Alessandro Scala

# Quantum Abstract Interpretation

Seminar for the **Introduction to Quantum Computing** course

Università di Pisa
Dipartimento di Informatica

Pisa, 24 Luglio 2023

# Roadmap

# Introduction

As quantum computing advances, we would like to have some means to prove correctness properties on quantum programs, *especially* since quantum programming is counterintuitive.

# Reasons

The naive way to check properties of a program is to run it and observe its behaviour.

# Reasons

The naive way to check properties of a program is to run it and **observe** its behaviour.

We **cannot observe** the state of a quantum program!

# Reasons

The naive way to check properties of a program is to run it and **observe** its behaviour.

We **cannot observe** the state of a quantum program!

Could **simulation** on a classical machine solve this issue?

# Reasons

The naive way to check properties of a program is to run it and **observe** its behaviour.

We **cannot observe** the state of a quantum program!

Could **simulation** on a classical machine solve this issue?

No: **exponential** space and time cost.

# Example

$$n_{qubits} = 1$$

$$|0\rangle \langle 0|$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$2^2 = 4$ complex numbers

# Example

$$n_{qubits} = 2$$

$$|00\rangle\langle 00|$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$2^4 = 16$ complex numbers

# Example

$$n_{qubits} = 3$$

$$|000\rangle \langle 000|$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$2^6 = 64$ complex numbers

# Example

$$n_{qubits} = 300$$

$$|0\rangle^{\otimes 300} \langle 0|^{\otimes 300}$$

?????

# Example

$$n_{qubits} = 300$$

$$|0\rangle^{\otimes 300} \langle 0|^{\otimes 300}$$

$$2^{600} = 41495155688809929585124078636911611510124462322424368\\999956573296906528114129081463997070489471037942881978866113\\007891823951510754117753078868748341139636870611818034015095\\23685376$$

Bigger than the number of atoms in the universe.

# Reasons

The naive way to check properties of a program is to run it and **observe** its behaviour.

We **cannot observe** the state of a quantum program!

Could **simulation** on a classical machine solve this issue?

No: **exponential** space and time cost.

# Reasons

The naive way to check properties of a program is to run it and **observe** its behaviour.

We **cannot observe** the state of a quantum program!

Could **simulation** on a classical machine solve this issue?

No: **exponential** space and time cost.

Static checking of programs by hand?

# Reasons

The naive way to check properties of a program is to run it and **observe** its behaviour.

We **cannot observe** the state of a quantum program!

Could **simulation** on a classical machine solve this issue?

No: **exponential** space and time cost.

Static checking of programs by hand?

Yes, but time consuming. Needs to be adapted to the specific program.

# Reasons

The naive way to check properties of a program is to run it and **observe** its behaviour.

We **cannot observe** the state of a quantum program!

Could **simulation** on a classical machine solve this issue?

No: **exponential** space and time cost.

Static checking of programs by hand?

Yes, but time consuming. Needs to be adapted to the specific program.

Solution: abstract interpretation

# What is abstract interpretation?

Instead of considering the **concrete domain**, we restrict our analysis to a **more coarse domain**, an **abstract domain**.

# What is abstract interpretation?

Instead of considering the **concrete domain**, we restrict our analysis to a **more coarse domain**, an **abstract domain**.

The abstract domain we consider should be:

# What is abstract interpretation?

Instead of considering the **concrete domain**, we restrict our analysis to a **more coarse domain**, an **abstract domain**.

The abstract domain we consider should be:

fine enough to capture the information we are interested in (e.g. the state of the quantum system belongs to the span of two vectors)

# What is abstract interpretation?

Instead of considering the **concrete domain**, we restrict our analysis to a **more coarse domain**, an **abstract domain**.

The abstract domain we consider should be:

fine enough to capture the information we are interested in (e.g. the state of the quantum system belongs to the span of two vectors)

as coarse as possible to discard information we are not interested in, and thus be substantially more efficient to be analyzed

# What is abstract interpretation?

Instead of considering the **concrete domain**, we restrict our analysis to a **more coarse domain**, an **abstract domain**.

The abstract domain we consider should be:

fine enough
: to capture the information we are interested in (e.g. the state of the quantum system belongs to the span of two vectors)

as coarse as possible
: to discard information we are not interested in, and thus be substantially more efficient to be analyzed

Abstract interpretation is usually sound, but not complete:

- AI returns **true** $\Rightarrow$ Property is **true**
- AI returns **false** $\Rightarrow$ Property can be either **true** or **false**

# Ingredients

- Abstract domain

# Ingredients

- Abstract domain
  - Abstraction function: from more concrete to more abstract domain

# Ingredients

- Abstract domain
  - Abstraction function: from more concrete to more abstract domain
  - Concretization function: from more abstract to more concrete domain

# Ingredients

- Abstract domain
  - Abstraction function: from more concrete to more abstract domain
  - Concretization function: from more abstract to more concrete domain
  - Abstract operations: to represent concrete operations in the abstract domain

# Ingredients

- Abstract domain
  - Abstraction function: from more concrete to more abstract domain
  - Concretization function: from more abstract to more concrete domain
  - Abstract operations: to represent concrete operations in the abstract domain
- Assertions: properties we can prove with abstract interpretation

# Roadmap

# Density Matrix

Instead of dealing with a state $|\phi\rangle$ in vector form, we use its *density matrix*:

$\rho_\phi = |\phi\rangle\langle\phi|$ (For a pure state)

- positive semi-definite
- $Tr(\rho) = 1$
- projection ($P = P^\dagger = P^2$)

# Density Matrix

Instead of dealing with a state $|\phi\rangle$ in vector form, we use its *density matrix*:

$\rho_\phi = |\phi\rangle\langle\phi|$ (For a pure state)

- positive semi-definite
- $Tr(\rho) = 1$
- projection ($P = P^\dagger = P^2$)

Example:

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$\rho_{\beta_{00}} = |\beta_{00}\rangle\langle\beta_{00}|$$

# Density Matrix

Instead of dealing with a state $|\phi\rangle$ in vector form, we use its *density matrix*:

$$\rho_\phi = |\phi\rangle\langle\phi| \text{ (For a pure state)}$$

- positive semi-definite
- $Tr(\rho) = 1$
- projection ($P = P^\dagger = P^2$)

Example:

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$\rho_{\beta_{00}} = |\beta_{00}\rangle\langle\beta_{00}| = \frac{1}{2}(|00\rangle + |11\rangle)(\langle 00| + \langle 11|)$$

# Density Matrix

Instead of dealing with a state $|\phi\rangle$ in vector form, we use its *density matrix*:

$$\rho_\phi = |\phi\rangle\langle\phi| \text{ (For a pure state)}$$

- positive semi-definite
- $Tr(\rho) = 1$
- projection ($P = P^\dagger = P^2$)

Example:

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$\rho_{\beta_{00}} = |\beta_{00}\rangle\langle\beta_{00}| = \frac{1}{2}(|00\rangle + |11\rangle)(\langle 00| + \langle 11|)$$

$$= \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)$$

# Density Matrix

Instead of dealing with a state $|\phi\rangle$ in vector form, we use its *density matrix*:

$$\rho_\phi = |\phi\rangle\langle\phi| \text{ (For a pure state)}$$

- positive semi-definite
- $Tr(\rho) = 1$
- projection ($P = P^\dagger = P^2$)

Example:

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$\begin{aligned}
\rho_{\beta_{00}} &= |\beta_{00}\rangle\langle\beta_{00}| = \frac{1}{2}(|00\rangle + |11\rangle)(\langle 00| + \langle 11|) \\
&= \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|) \\
&= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}$$

# Reduced Density Matrix

Suppose we have a composite quantum system $AB = A \otimes B$, and we want to focus our attention on a state $|\phi\rangle \in AB$ with respect to subsystem $A$.

# Reduced Density Matrix

Suppose we have a composite quantum system $AB = A \otimes B$, and we want to focus our attention on a state $|\phi\rangle \in AB$ with respect to subsystem $A$.

$$A = \mathbb{C}^{2^n} \times \mathbb{C}^{2^n} \quad B = \mathbb{C}^{2^m} \times \mathbb{C}^{2^m}$$

$$AB = (\mathbb{C}^{2^n} \times \mathbb{C}^{2^n}) \otimes (\mathbb{C}^{2^m} \times \mathbb{C}^{2^m})$$

# Reduced Density Matrix

Suppose we have a composite quantum system $AB = A \otimes B$, and we want to focus our attention on a state $|\phi\rangle \in AB$ with respect to subsystem $A$.

$$A = \mathbb{C}^{2^n} \times \mathbb{C}^{2^n} \quad B = \mathbb{C}^{2^m} \times \mathbb{C}^{2^m}$$

$$AB = (\mathbb{C}^{2^n} \times \mathbb{C}^{2^n}) \otimes (\mathbb{C}^{2^m} \times \mathbb{C}^{2^m})$$

$$Tr_B[\rho] : AB \to A \qquad Tr_A[\rho] : AB \to B$$

$$Tr_B[\alpha \otimes \beta] = \alpha \cdot Tr(\beta) \qquad Tr_A[\alpha \otimes \beta] = Tr(\alpha) \cdot \beta$$

$$\left.\begin{array}{l} Tr_S[\rho + \sigma] = Tr_S[\rho] + Tr_S[\sigma] \\ Tr_S[a \cdot \rho] = a \cdot Tr_S[\rho] \end{array}\right\} \text{(Linearity)}$$

# Reduced Density Matrix

Suppose we have a composite quantum system $AB = A \otimes B$, and we want to focus our attention on a state $|\phi\rangle \in AB$ with respect to subsystem $A$.

$$A = \mathbb{C}^{2^n} \times \mathbb{C}^{2^n} \quad B = \mathbb{C}^{2^m} \times \mathbb{C}^{2^m}$$
$$AB = (\mathbb{C}^{2^n} \times \mathbb{C}^{2^n}) \otimes (\mathbb{C}^{2^m} \times \mathbb{C}^{2^m})$$

$$Tr_B[\rho] : AB \to A \qquad Tr_A[\rho] : AB \to B$$
$$Tr_B[\alpha \otimes \beta] = \alpha \cdot Tr(\beta) \qquad Tr_A[\alpha \otimes \beta] = Tr(\alpha) \cdot \beta$$

$$\left.\begin{array}{l} Tr_S[\rho + \sigma] = Tr_S[\rho] + Tr_S[\sigma] \\ Tr_S[a \cdot \rho] = a \cdot Tr_S[\rho] \end{array}\right\} \text{(Linearity)}$$

Partial trace $Tr_B[\rho]$ **traces out** subsystem $B$.

# Example

$$A = \mathbb{C}^2 \times \mathbb{C}^2 \quad B = \mathbb{C}^2 \times \mathbb{C}^2 \quad AB = A \otimes B$$

$$\rho_{\beta_{00}} = |\beta_{00}\rangle \langle \beta_{00}| = \frac{|00\rangle \langle 00| + |00\rangle \langle 11| + |11\rangle \langle 00| + |11\rangle \langle 11|}{2}$$

$$Tr_B[\rho_{\beta_{00}}] =$$

# Example

$$A = \mathbb{C}^2 \times \mathbb{C}^2 \quad B = \mathbb{C}^2 \times \mathbb{C}^2 \quad AB = A \otimes B$$

$$\rho_{\beta_{00}} = |\beta_{00}\rangle \langle\beta_{00}| = \frac{|00\rangle \langle 00| + |00\rangle \langle 11| + |11\rangle \langle 00| + |11\rangle \langle 11|}{2}$$

$$Tr_B[\rho_{\beta_{00}}] = \frac{(Tr_B[|00\rangle \langle 00|] + Tr_b[|00\rangle \langle 11|] + Tr_b[|11\rangle \langle 00|] + Tr_b[|11\rangle \langle 11|])}{2}$$

# Example

$$A = \mathbb{C}^2 \times \mathbb{C}^2 \quad B = \mathbb{C}^2 \times \mathbb{C}^2 \quad AB = A \otimes B$$

$$\rho_{\beta_{00}} = |\beta_{00}\rangle \langle\beta_{00}| = \frac{|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|}{2}$$

$$Tr_B[\rho_{\beta_{00}}] = \frac{(Tr_B[|00\rangle\langle 00|] + Tr_b[|00\rangle\langle 11|] + Tr_b[|11\rangle\langle 00|] + Tr_b[|11\rangle\langle 11|])}{2}$$
$$= \frac{(|0\rangle\langle 0| \cdot \langle 0|0\rangle) + (|0\rangle\langle 1| \cdot \langle 0|1\rangle) + (|1\rangle\langle 0| \cdot \langle 1|0\rangle) + (|1\rangle\langle 1| \cdot \langle 1|1\rangle)}{2}$$

# Example

$$A = \mathbb{C}^2 \times \mathbb{C}^2 \quad B = \mathbb{C}^2 \times \mathbb{C}^2 \quad AB = A \otimes B$$

$$\rho_{\beta_{00}} = |\beta_{00}\rangle \langle \beta_{00}| = \frac{|00\rangle \langle 00| + |00\rangle \langle 11| + |11\rangle \langle 00| + |11\rangle \langle 11|}{2}$$

$$Tr_B[\rho_{\beta_{00}}] = \frac{(Tr_B[|00\rangle \langle 00|] + Tr_b[|00\rangle \langle 11|] + Tr_b[|11\rangle \langle 00|] + Tr_b[|11\rangle \langle 11|])}{2}$$

$$= \frac{(|0\rangle \langle 0| \cdot \langle 0|0\rangle) + (|0\rangle \langle 1| \cdot \langle 0|1\rangle) + (|1\rangle \langle 0| \cdot \langle 1|0\rangle) + (|1\rangle \langle 1| \cdot \langle 1|1\rangle)}{2}$$

# Example

$$A = \mathbb{C}^2 \times \mathbb{C}^2 \quad B = \mathbb{C}^2 \times \mathbb{C}^2 \quad AB = A \otimes B$$

$$\rho_{\beta_{00}} = |\beta_{00}\rangle \langle \beta_{00}| = \frac{|00\rangle \langle 00| + |00\rangle \langle 11| + |11\rangle \langle 00| + |11\rangle \langle 11|}{2}$$

$$\begin{aligned} Tr_B[\rho_{\beta_{00}}] &= \frac{(Tr_B[|00\rangle \langle 00|] + Tr_b[|00\rangle \langle 11|] + Tr_b[|11\rangle \langle 00|] + Tr_b[|11\rangle \langle 11|])}{2} \\ &= \frac{(|0\rangle \langle 0| \cdot \langle 0|0\rangle) + (|0\rangle \langle 1| \cdot \langle 0|1\rangle) + (|1\rangle \langle 0| \cdot \langle 1|0\rangle) + (|1\rangle \langle 1| \cdot \langle 1|1\rangle)}{2} \\ &= \frac{|0\rangle \langle 0| + |1\rangle \langle 1|}{2} \end{aligned}$$

# Loss of precision

Computing a reduced density matrix **discards information**!

$$\rho_{\beta_{00}} = \frac{|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|}{2} \qquad \text{(Pure state)}$$

$$\rho_2 = \frac{|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| + |11\rangle\langle 11|}{4} \qquad \text{(Mixed state)}$$

# Loss of precision

Computing a reduced density matrix **discards information**!

$$\rho_{\beta_{00}} = \frac{|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|}{2} \qquad \text{(Pure state)}$$
$$\rho_2 = \frac{|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| + |11\rangle\langle 11|}{4} \qquad \text{(Mixed state)}$$

$$Tr_B[\rho_{\beta_{00}}] = \frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2} = Tr_B[\rho_2]$$

The partial traces of two different initial states can be equal.

For a state $\rho \in A \otimes B$, even if we know $Tr_B[\rho]$ and $Tr_A[\rho]$, we cannot uniquely determine $\rho$.

# Projection Subspaces

Each projection $P$ corresponds to a linear subspace $S_P = \{v \mid Pv = v\}$.

# Projection Subspaces

Each projection $P$ corresponds to a linear subspace $S_P = \{v \mid Pv = v\}$.

The support of a matrix $P$ is the subspace orthogonal to its kernel, i.e., the set $supp(P) = \{v \mid Pv \neq 0\}$.

# Projection Subspaces

Each projection $P$ corresponds to a linear subspace $S_P = \{v \mid Pv = v\}$.

The support of a matrix $P$ is the subspace orthogonal to its kernel, i.e., the set $supp(P) = \{v \mid Pv \neq 0\}$.

This duality between projections and subspaces will be employed multiple times and will often be implied.

# Roadmap

# Abstract Domain

$$\mathcal{D} = \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}, \quad S = (s_1, ..., s_m), \quad 1 \leq m \leq 2^n, \quad s_i \subseteq [n]$$

# Abstract Domain

$$\mathcal{D} = \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}, \quad S = (s_1, \dots, s_m), \quad 1 \leq m \leq 2^n, \quad s_i \subseteq [n]$$

$$AbsDom(S) = \left\{ (P_{s_1}, \dots, P_{s_m}) \mid P_{s_i} \text{ is a projection in } \mathbb{C}^{2^{|s_i|}} \otimes \mathbb{C}^{2^{|s_i|}} \right\}$$

# Abstract Domain

$$\mathcal{D} = \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}, \quad S = (s_1, \ldots, s_m), \quad 1 \leq m \leq 2^n, \quad s_i \subseteq [n]$$

$$AbsDom(S) = \left\{ (P_{s_1}, \ldots, P_{s_m}) \mid P_{s_i} \text{ is a projection in } \mathbb{C}^{2^{|s_i|}} \otimes \mathbb{C}^{2^{|s_i|}} \right\}$$

Intuitively, given a tuple $S$ of sets of qubits, an abstract state $\overline{\sigma} \in AbsDom(S)$ is a tuple of projections over those qubits.

# Abstract Domain

$$\mathcal{D} = \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}, \quad S = (s_1, \dots, s_m), \quad 1 \leq m \leq 2^n, \quad s_i \subseteq [n]$$

$$AbsDom(S) = \left\{ (P_{s_1}, \dots, P_{s_m}) \mid P_{s_i} \text{ is a projection in } \mathbb{C}^{2^{|s_i|}} \otimes \mathbb{C}^{2^{|s_i|}} \right\}$$

Intuitively, given a tuple $S$ of sets of qubits, an abstract state $\overline{\sigma} \in AbsDom(S)$ is a tuple of projections over those qubits.

Special case:

$$T = ([n]) \Rightarrow AbsDom(T) \simeq \mathcal{D}$$

# Fineness Relation

Let $S = (s_1, ..., s_m)$ and $T = (t_1, ..., t_m)$ (with $1 \leq m \leq 2^n$), then:

$$\underbrace{S \trianglelefteq T}_{\text{“T is finer than S”}} \quad \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$T$ is “more concrete” than S.

# Fineness Relation

Let $S = (s_1, ..., s_m)$ and $T = (t_1, ..., t_m)$ (with $1 \leq m \leq 2^n$), then:

$$\underbrace{S \trianglelefteq T}_{\text{"T is finer than S"}} \quad \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$T$ is "more concrete" than S.

Least element: $\perp = (\emptyset, ..., \emptyset)$.
Greatest element: $\top = ([n], ...[n]) = [n]^m$.

# Fineness Relation

Let $S = (s_1, ..., s_m)$ and $T = (t_1, ..., t_m)$ (with $1 \leq m \leq 2^n$), then:

$$\underbrace{S \trianglelefteq T}_{\text{“T is finer than S”}} \quad \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$T$ is “more concrete” than S.

Least element: $\bot = (\emptyset, ..., \emptyset)$.
Greatest element: $\top = ([n], ...[n]) = [n]^m$.

$AbsDom(\bot)$ corresponds to a state so abstract that it holds no information at all.
$AbsDom(\top)$ corresponds to tuples where every projection is a concrete state.

# Abstraction Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\alpha_{T \to S} : AbsDom(T) \to AbsDom(S)$$

# Abstraction Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\alpha_{T \to S} : AbsDom(T) \to AbsDom(S)$$

$$\alpha_{T \to S}(Q_{t_1}, ..., Q_{t_m}) = (P_{s_1}, ..., P_{s_m})$$

$$P_{s_i} = \bigcap_{t_j.\ s_i \subseteq t_J} supp(Tr_{t_j \setminus s_i}[Q_{t_j}])$$

# Abstraction Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\alpha_{T \to S} : AbsDom(T) \to AbsDom(S)$$

$$\alpha_{T \to S}(Q_{t_1}, ..., Q_{t_m}) = (P_{s_1}, ..., P_{s_m})$$

$$P_{s_i} = \bigcap_{t_j.\ s_i \subseteq t_J} supp(Tr_{t_j \setminus s_i}[Q_{t_j}])$$

Given an abstract state $\overline{\tau} \in AbsDom(T) = (Q_{t_1}, ..., Q_{t_m})$, we want to compute $\overline{\sigma} \in AbsDom(S) = (P_{s_1}, ..., P_{s_m})$. For each $i \in [m]$:

# Abstraction Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\alpha_{T \to S} : AbsDom(T) \to AbsDom(S)$$

$$\alpha_{T \to S}(Q_{t_1}, \dots, Q_{t_m}) = (P_{s_1}, \dots, P_{s_m})$$

$$P_{s_i} = \bigcap_{t_j.\ s_i \subseteq t_J} supp(Tr_{t_j \setminus s_i}[Q_{t_j}])$$

Given an abstract state $\overline{\tau} \in AbsDom(T) = (Q_{t_1}, \dots, Q_{t_m})$, we want to compute $\overline{\sigma} \in AbsDom(S) = (P_{s_1}, \dots, P_{s_m})$. For each $i \in [m]$:

1. Find all $Q_{t_j}$s such that $s_i \subseteq t_j$. We know that at least one exists (for $j = i$), since $S \trianglelefteq T$.

# Abstraction Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\alpha_{T \to S} : AbsDom(T) \to AbsDom(S)$$

$$\alpha_{T \to S}(Q_{t_1}, \dots, Q_{t_m}) = (P_{s_1}, \dots, P_{s_m})$$

$$P_{s_i} = \bigcap_{t_j.\ s_i \subseteq t_J} supp(Tr_{t_j \setminus s_i}[Q_{t_j}])$$

Given an abstract state $\overline{\tau} \in AbsDom(T) = (Q_{t_1}, \dots, Q_{t_m})$, we want to compute $\overline{\sigma} \in AbsDom(S) = (P_{s_1}, \dots, P_{s_m})$. For each $i \in [m]$:

1. Find all $Q_{t_j}$s such that $s_i \subseteq t_j$. We know that at least one exists (for $j = i$), since $S \trianglelefteq T$.
2. For each $Q_{t_j}$ found, trace out the qubits in $t_j$ that are not in $s_i$.

# Abstraction Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\alpha_{T \to S} : AbsDom(T) \to AbsDom(S)$$

$$\alpha_{T \to S}(Q_{t_1}, ..., Q_{t_m}) = (P_{s_1}, ..., P_{s_m})$$

$$P_{s_i} = \bigcap_{t_j.\ s_i \subseteq t_J} supp(Tr_{t_j \setminus s_i}[Q_{t_j}])$$

Given an abstract state $\overline{\tau} \in AbsDom(T) = (Q_{t_1}, ..., Q_{t_m})$, we want to compute $\overline{\sigma} \in AbsDom(S) = (P_{s_1}, ..., P_{s_m})$. For each $i \in [m]$:

1. Find all $Q_{t_j}$s such that $s_i \subseteq t_j$. We know that at least one exists (for $j = i$), since $S \trianglelefteq T$.
2. For each $Q_{t_j}$ found, trace out the qubits in $t_j$ that are not in $s_i$.
3. Compute the support of the traced matrices (to preserve the structure of projections).

# Abstraction Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\alpha_{T \to S} : AbsDom(T) \to AbsDom(S)$$

$$\alpha_{T \to S}(Q_{t_1}, \ldots, Q_{t_m}) = (P_{s_1}, \ldots, P_{s_m})$$

$$P_{s_i} = \bigcap_{t_j.\ s_i \subseteq t_J} supp(Tr_{t_j \setminus s_i}[Q_{t_j}])$$

Given an abstract state $\overline{\tau} \in AbsDom(T) = (Q_{t_1}, \ldots, Q_{t_m})$, we want to compute $\overline{\sigma} \in AbsDom(S) = (P_{s_1}, \ldots, P_{s_m})$. For each $i \in [m]$:

1. Find all $Q_{t_j}$s such that $s_i \subseteq t_j$. We know that at least one exists (for $j = i$), since $S \trianglelefteq T$.
2. For each $Q_{t_j}$ found, trace out the qubits in $t_j$ that are not in $s_i$.
3. Compute the support of the traced matrices (to preserve the structure of projections).
4. Compute the intersection of the supports.

# Concretization Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\gamma_{S \to T} : AbsDom(S) \to AbsDom(T)$$

# Concretization Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\gamma_{S \to T} : AbsDom(S) \to AbsDom(T)$$

$$\gamma_{S \to T}(P_{s_1}, ..., P_{s_m}) = (Q_{t_1}, ..., Q_{t_m})$$

$$Q_{t_j} = \bigcap_{s_i.\ s_i \subseteq t_J} P_{s_i} \otimes I_{t_j \setminus s_i}$$

# Concretization Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$

$$\gamma_{S \to T} : AbsDom(S) \to AbsDom(T)$$

$$\gamma_{S \to T}(P_{s_1}, ..., P_{s_m}) = (Q_{t_1}, ..., Q_{t_m})$$

$$Q_{t_j} = \bigcap_{s_i.\ s_i \subseteq t_J} P_{s_i} \otimes I_{t_j \setminus s_i}$$

Given an abstract state $\overline{\sigma} \in AbsDom(S) = (P_{s_1}, ..., P_{s_m})$, we want to compute $\overline{\tau} \in AbsDom(T) = (Q_{t_1}, ..., Q_{t_m})$. For each $j \in [m]$:

# Concretization Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$
$$\gamma_{S \to T} : AbsDom(S) \to AbsDom(T)$$

$$\gamma_{S \to T}(P_{s_1}, \ldots, P_{s_m}) = (Q_{t_1}, \ldots, Q_{t_m})$$
$$Q_{t_j} = \bigcap_{s_i .\ s_i \subseteq t_J} P_{s_i} \otimes I_{t_j \setminus s_i}$$

Given an abstract state $\overline{\sigma} \in AbsDom(S) = (P_{s_1}, \ldots, P_{s_m})$, we want to compute $\overline{\tau} \in AbsDom(T) = (Q_{t_1}, \ldots, Q_{t_m})$. For each $j \in [m]$:

1. Find all $P_{s_i}$s such that $s_i \subseteq t_j$. We know at least one exists (for $i = j$), since $S \trianglelefteq T$.

# Concretization Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$
$$\gamma_{S \to T} : AbsDom(S) \to AbsDom(T)$$

$$\gamma_{S \to T}(P_{s_1}, \ldots, P_{s_m}) = (Q_{t_1}, \ldots, Q_{t_m})$$
$$Q_{t_j} = \bigcap_{s_i .\ s_i \subseteq t_J} P_{s_i} \otimes I_{t_j \setminus s_i}$$

Given an abstract state $\overline{\sigma} \in AbsDom(S) = (P_{s_1}, \ldots, P_{s_m})$, we want to compute $\overline{\tau} \in AbsDom(T) = (Q_{t_1}, \ldots, Q_{t_m})$. For each $j \in [m]$:

1. Find all $P_{s_i}$s such that $s_i \subseteq t_j$. We know at least one exists (for $i = j$), since $S \trianglelefteq T$.
2. Extend the projection to the space of all qubits in $t_j$, by computing the tensor product with the identity matrix.

# Concretization Function

$$S \trianglelefteq T \triangleq \forall i \in [m].\ s_i \subseteq t_i$$
$$\gamma_{S \to T} : AbsDom(S) \to AbsDom(T)$$

$$\gamma_{S \to T}(P_{s_1}, ..., P_{s_m}) = (Q_{t_1}, ..., Q_{t_m})$$
$$Q_{t_j} = \bigcap_{s_i.\ s_i \subseteq t_J} P_{s_i} \otimes I_{t_j \setminus s_i}$$

Given an abstract state $\overline{\sigma} \in AbsDom(S) = (P_{s_1}, ..., P_{s_m})$, we want to compute $\overline{\tau} \in AbsDom(T) = (Q_{t_1}, ..., Q_{t_m})$. For each $j \in [m]$:

1. Find all $P_{s_i}$s such that $s_i \subseteq t_j$. We know at least one exists (for $i = j$), since $S \trianglelefteq T$.
2. Extend the projection to the space of all qubits in $t_j$, by computing the tensor product with the identity matrix.
3. Compute the intersection of the extended projections.

# Abstract Operations

Operations on the concrete domain are unitary operators $U$.
To apply them to a concrete state $\rho$ we compute $U\rho U^\dagger$.
We want to define $U^\sharp : AbsDom(S) \rightarrow AbsDom(S)$.

# Abstract Operations

Operations on the concrete domain are unitary operators $U$.
To apply them to a concrete state $\rho$ we compute $U\rho U^\dagger$.
We want to define $U^\sharp : AbsDom(S) \rightarrow AbsDom(S)$.

Let $T = (t_1, ..., t_m)$ s.t. $t_i = s_i \cup s_U$, where $s_U$ denotes the set of qubits that matrix $U$ acts on.

# Abstract Operations

Operations on the concrete domain are unitary operators $U$.
To apply them to a concrete state $\rho$ we compute $U\rho U^\dagger$.
We want to define $U^\sharp : AbsDom(S) \to AbsDom(S)$.

Let $T = (t_1, ..., t_m)$ s.t. $t_i = s_i \cup s_U$, where $s_U$ denotes the set of qubits that matrix $U$ acts on.
Let $U^{cg}(\overline{\sigma}) = (UT_{t_1}U^\dagger, ..., UT_{t_m}U^\dagger)$

# Abstract Operations

Operations on the concrete domain are unitary operators $U$.
To apply them to a concrete state $\rho$ we compute $U\rho U^\dagger$.
We want to define $U^\sharp : AbsDom(S) \to AbsDom(S)$.

Let $T = (t_1, ..., t_m)$ s.t. $t_i = s_i \cup s_U$, where $s_U$ denotes the set of qubits that matrix $U$ acts on.
Let $U^{cg}(\overline{\sigma}) = (UT_{t_1}U^\dagger, ..., UT_{t_m}U^\dagger)$, then

$$U^\sharp = \alpha_{T\to S} \circ U^{cg} \circ \gamma_{S\to T}$$

# Focus

$$U^\sharp = \alpha_{T \to S} \circ U^{cg} \circ \gamma_{S \to T}$$

# Focus

$$U^\sharp = \alpha_{T \to S} \circ U^{cg} \circ \gamma_{S \to T}$$

$$AbsDom(S) \xrightarrow[\text{Focus}]{\gamma_{S \to T}} AbsDom(T) \xrightarrow[\text{Apply}]{U^{cg}} AbsDom(T) \xrightarrow[\text{Unfocus}]{\alpha_{T \to S}} AbsDom(S)$$

# Focus

$$U^\sharp = \alpha_{T\to S} \circ U^{cg} \circ \gamma_{S\to T}$$

$$AbsDom(S) \xrightarrow[\text{Focus}]{\gamma_{S\to T}} AbsDom(T) \xrightarrow[\text{Apply}]{U^{cg}} AbsDom(T) \xrightarrow[\text{Unfocus}]{\alpha_{T\to S}} AbsDom(S)$$

Focus - concretize to a new, finer domain with sufficient precision to accurately represent the operation;

# Focus

$$U^\sharp = \alpha_{T\to S} \circ U^{cg} \circ \gamma_{S\to T}$$

$$AbsDom(S) \xrightarrow[\text{Focus}]{\gamma_{S\to T}} AbsDom(T) \xrightarrow[\text{Apply}]{U^{cg}} AbsDom(T) \xrightarrow[\text{Unfocus}]{\alpha_{T\to S}} AbsDom(S)$$

Focus - concretize to a new, finer domain with sufficient precision to accurately represent the operation;

Apply - apply the unitary operator to all the elements in the tuple

# Focus

$$U^\sharp = \alpha_{T \to S} \circ U^{cg} \circ \gamma_{S \to T}$$

$$AbsDom(S) \xrightarrow[\text{Focus}]{\gamma_{S\to T}} AbsDom(T) \xrightarrow[\text{Apply}]{U^{cg}} AbsDom(T) \xrightarrow[\text{Unfocus}]{\alpha_{T\to S}} AbsDom(S)$$

Focus - concretize to a new, finer domain with sufficient precision to accurately represent the operation;

Apply - apply the unitary operator to all the elements in the tuple

Unfocus - abstract back to the original abstract domain to keep the representation more compact

# Assertions

We want to check that a state of a program lies in the span of two vectors.

# Assertions

We want to check that a state of a program lies in the span of two vectors.
We define an assertion as:

$$A = span\{v_1 = |a_1\rangle \dots |a_n\rangle\ ,\quad v_2 = |b_1\rangle \dots |b_n\rangle\}$$

# Assertions

We want to check that a state of a program lies in the span of two vectors.
We define an assertion as:

$$A = span\{v_1 = |a_1\rangle \dots |a_n\rangle\,,\ \ v_2 = |b_1\rangle \dots |b_n\rangle\}$$

And a projection $proj(A)$ onto this subspace, such that:

$$proj(A)v_1 = v_1 \qquad proj(A)v_2 = v_2$$

# Order Relation on Abstract States

$$1 \leq m \leq 2^n, \quad S = (s_1, ...s_m), \quad \forall i \in [m].\ s_i \subseteq [n]$$
$$\overline{\sigma} \in AbsDom(S) = (P_{s_1}, ..., P_{s_m}), \quad \overline{\tau} \in AbsDom(S) = (Q_{s_1}, ..., Q_{s_m})$$

$$\overline{\sigma} \sqsubseteq \overline{\tau} \triangleq \forall i \in [m].\ \underbrace{P_{s_i} \subseteq Q_{s_i}}_{\text{Subspace interpretation of projections}}$$

# Monotonicity and Galois Connection

Monotonicity of *abstraction*, *concretization*, and *abstract operations*:

$$S \trianglelefteq T$$

$$\forall \overline{\sigma}, \overline{\tau} \in AbsDom(T).$$

$$\begin{pmatrix} \overline{\sigma} \sqsubseteq \overline{\tau} \Rightarrow \alpha_{T \to S}(\overline{\sigma}) \sqsubseteq \alpha_{T \to S}(\overline{\tau}) \\ \wedge \\ \overline{\sigma} \sqsubseteq \overline{\tau} \Rightarrow \gamma_{T \to S}(\overline{\sigma}) \sqsubseteq \gamma_{T \to S}(\overline{\tau}) \\ \wedge \\ \overline{\sigma} \sqsubseteq \overline{\tau} \Rightarrow U^{\sharp}(\overline{\sigma}) \sqsubseteq U^{\sharp}(\overline{\tau}) \end{pmatrix}$$

# Monotonicity and Galois Connection

Monotonicity of *abstraction*, *concretization*, and *abstract operations*:

$$S \trianglelefteq T$$
$$\forall \overline{\sigma}, \overline{\tau} \in AbsDom(T).$$
$$\begin{pmatrix} \overline{\sigma} \sqsubseteq \overline{\tau} \Rightarrow \alpha_{T\to S}(\overline{\sigma}) \sqsubseteq \alpha_{T\to S}(\overline{\tau}) \\ \wedge \\ \overline{\sigma} \sqsubseteq \overline{\tau} \Rightarrow \gamma_{T\to S}(\overline{\sigma}) \sqsubseteq \gamma_{T\to S}(\overline{\tau}) \\ \wedge \\ \overline{\sigma} \sqsubseteq \overline{\tau} \Rightarrow U^\sharp(\overline{\sigma}) \sqsubseteq U^\sharp(\overline{\tau}) \end{pmatrix}$$

Galois connection:

$$\forall \overline{\sigma} \in AbsDom(S).\ \forall \overline{\tau} \in AbsDom([n]^m).$$
$$\overline{\tau} \sqsubseteq \gamma_{S\to[n]^m}(\overline{\sigma}) \quad \Leftrightarrow \quad \alpha_{[n]^m\to S}(\overline{\tau}) \sqsubseteq \overline{\sigma}$$

# Connectivity

$$S = (s_1, \ldots, s_m)$$

$$S \text{ is connected} \triangleq \forall k \in [n-1].\ \exists r \in [m].\ k \in s_r \wedge k+1 \in s_r$$

# Connectivity

$$S = (s_1, ..., s_m)$$

$$S \text{ is connected} \triangleq \forall k \in [n-1].\ \exists r \in [m].\ k \in s_r \wedge k+1 \in s_r$$

Example:

$$n = 5, \quad m = 3$$
$$S = (\{0, 2, 3\}, \{0, 1, 2\}, \{3, 4, 5\})$$

# Connectivity

$$S = (s_1, ..., s_m)$$

$$S \text{ is connected} \triangleq \forall k \in [n-1].\ \exists r \in [m].\ k \in s_r \wedge k+1 \in s_r$$

Example:

$$n = 5, \quad m = 3$$

$$S = (\{0,2,3\}, \{0,1,2\}, \{3,4,5\})$$

$$0, 1 \in \{0,1,2\}$$

# Connectivity

$$S = (s_1, ..., s_m)$$

$$S \text{ is connected} \triangleq \forall k \in [n-1].\ \exists r \in [m].\ k \in s_r \wedge k+1 \in s_r$$

Example:

$$n = 5, \quad m = 3$$

$$S = (\{0,2,3\}, \{0,1,2\}, \{3,4,5\})$$

$$1, 2 \in \{0,1,2\}$$

# Connectivity

$$S = (s_1, \ldots, s_m)$$

$$S \text{ is connected} \triangleq \forall k \in [n-1].\ \exists r \in [m].\ k \in s_r \wedge k+1 \in s_r$$

Example:

$$n = 5, \quad m = 3$$
$$S = (\{0, 2, 3\}, \{0, 1, 2\}, \{3, 4, 5\})$$

$$2, 3 \in \{0, 2, 3\}$$

# Connectivity

$$S = (s_1, ..., s_m)$$

$$S \text{ is connected} \triangleq \forall k \in [n-1].\ \exists r \in [m].\ k \in s_r \land k+1 \in s_r$$

Example:

$$n = 5, \quad m = 3$$

$$S = (\{0, 2, 3\}, \{0, 1, 2\}, \{3, 4, 5\})$$

$$3, 4 \in \{3, 4, 5\}$$

# Connectivity

$$S = (s_1, ..., s_m)$$

$$S \text{ is connected} \triangleq \forall k \in [n-1].\ \exists r \in [m].\ k \in s_r \wedge k+1 \in s_r$$

Example:

$$n = 5, \quad m = 3$$

$$S = (\{0, 2, 3\}, \{0, 1, 2\}, \{3, 4, 5\})$$

$$4, 5 \in \{3, 4, 5\}$$

# Connectivity

$$S = (s_1, ..., s_m)$$

$$S \text{ is connected} \triangleq \forall k \in [n-1].\ \exists r \in [m].\ k \in s_r \land k+1 \in s_r$$

Example:

$$n = 5, \quad m = 3$$
$$S = (\{0,2,3\}, \{0,1,2\}, \{3,4,5\})$$

For an assertion $A$, if $S$ is connected, then:

$$proj(A) = \gamma_{S \to [n]^m}(\alpha_{[n]^m \to S}(proj(A)))$$

# Assertion Checking

Given an assertion $A$, if the final state of a computation is $v$ and the final abstract state of the abstract interpretation is $\overline{v} \in AbsDom(S)$, with $S$ connected, then:

$$\overline{v} \sqsubseteq \alpha_{[n]^m \to S}(proj(A)) \quad \Rightarrow \quad v \in A$$

# Assertion Checking

Given an assertion $A$, if the final state of a computation is $v$ and the final abstract state of the abstract interpretation is $\overline{v} \in AbsDom(S)$, with $S$ connected, then:

$$\overline{v} \sqsubseteq \alpha_{[n]^m \to S}(proj(A)) \quad \Rightarrow \quad v \in A$$

Essentially, if the abstract state satisfies the assertion, then the concrete one does as well.

# Choice of Tuple S

Fundamental choice for the shape of the abstract domain and, in turn, the success of the analysis.

# Choice of Tuple S

Fundamental choice for the shape of the abstract domain and, in turn, the success of the analysis.

| Type of $S$ | Pros | Cons |
|---|---|---|
| Less, smaller sets | Less computational cost and memory footprint | Less precision |
| More, bigger sets | More computational cost and memory footprint | More precision |

# Choice of Tuple S

Fundamental choice for the shape of the abstract domain and, in turn, the success of the analysis.

| Type of $S$ | Pros | Cons |
| --- | --- | --- |
| Less, smaller sets | Less computational cost and memory footprint | Less precision |
| More, bigger sets | More computational cost and memory footprint | More precision |

Examples:

- $S_0 =$ all $2^n$ combinations of up to $n$ qubits
- $S_1 =$ all $\binom{n}{k}$ combinations of exactly $k$ qubits
- $S_2 =$ sets that contain qubits used by at least two 3-qubit gates
- ...

# Choice of Tuple S (ctd.)

Another important factor in the choice for the abstract state is which qubits are put together in projections.
A good choice would make the abstract interpretation *flow-sensitive*.

# Choice of Tuple S (ctd.)

Another important factor in the choice for the abstract state is which qubits are put together in projections.
A good choice would make the abstract interpretation *flow-sensitive*.

Example:

- $U_1$ acts on qubits $\{1, 2, 3\}$ with $\{1, 2\}$ as input and $\{3\}$ as output.
- $U_2$ acts on qubits $\{3, 4, 5\}$ with $\{3, 4\}$ as input and $\{5\}$ as output.

# Choice of Tuple S (ctd.)

Another important factor in the choice for the abstract state is which qubits are put together in projections.
A good choice would make the abstract interpretation *flow-sensitive*.

Example:

- $U_1$ acts on qubits $\{1, 2, 3\}$ with $\{1, 2\}$ as input and $\{3\}$ as output.
- $U_2$ acts on qubits $\{3, 4, 5\}$ with $\{3, 4\}$ as input and $\{5\}$ as output.

A good choice which would improve the precision of the abstract interpretation is to have the set $\{1, 2, 3, 4, 5\}$ in the abstract state.

# Assertion Checking Example

We want to check (partial) correctness of Grover's algorithm.

# Assertion Checking Example

We want to check (partial) correctness of Grover's algorithm.

## Grover's algorithm steps:

1. Prepare the initial state $|\phi\rangle^{(0)}$

# Assertion Checking Example

We want to check (partial) correctness of Grover's algorithm.

## Grover's algorithm steps:

1. Prepare the initial state $|\phi\rangle^{(0)}$
2. Repeat approx. $\frac{\pi}{4}\sqrt{N}$ times $|\phi\rangle^{(t+1)} = G\,|\phi\rangle^{(t)}$.

# Assertion Checking Example

We want to check (partial) correctness of Grover's algorithm.

## Grover's algorithm steps:

1. Prepare the initial state $|\phi\rangle^{(0)}$
2. Repeat approx. $\frac{\pi}{4}\sqrt{N}$ times $|\phi\rangle^{(t+1)} = G\,|\phi\rangle^{(t)}$.
3. Measure in the computational basis.

# Assertion Checking Example

We want to check (partial) correctness of Grover's algorithm.

## Grover's algorithm steps:

1. Prepare the initial state $|\phi\rangle^{(0)}$
2. Repeat approx. $\frac{\pi}{4}\sqrt{N}$ times $|\phi\rangle^{(t+1)} = G\,|\phi\rangle^{(t)}$.
3. Measure in the computational basis.

With this abstract interpretation framework, we can prove the loop invariant:

$$|\phi\rangle^{(t)} \in A = \mathit{span}\{|\beta\rangle, |\phi\rangle\}$$

# Assertion Checking Example (ctd.)

$$P(v) \triangleq v \in A = span\{|\beta\rangle, |\phi\rangle\}$$

# Assertion Checking Example (ctd.)

$$P(v) \triangleq v \in A = span\{|\beta\rangle, |\phi\rangle\}$$

We need to check:

1. $P\left(|\phi\rangle^{(0)}\right)$

# Assertion Checking Example (ctd.)

$$P(v) \triangleq v \in A = span\{|\beta\rangle, |\phi\rangle\}$$

We need to check:

1. $P\left(|\phi\rangle^{(0)}\right) \triangleq |\phi\rangle \in span\{|\beta\rangle, |\phi\rangle\}$

# Assertion Checking Example (ctd.)

$$P(v) \triangleq v \in A = span\{|\beta\rangle, |\phi\rangle\}$$

We need to check:

1. $P\left(|\phi\rangle^{(0)}\right) \triangleq |\phi\rangle \in span\{|\beta\rangle, |\phi\rangle\}$
2. $P\left(|\phi\rangle^{(t)}\right) \Rightarrow P\left(|\phi\rangle^{(t+1)}\right)$

# Assertion Checking Example (ctd.)

$$P(v) \triangleq v \in A = span\{|\beta\rangle, |\phi\rangle\}$$

We need to check:

1. $P\left(|\phi\rangle^{(0)}\right) \triangleq |\phi\rangle \in span\{|\beta\rangle, |\phi\rangle\}$
2. $P\left(|\phi\rangle^{(t)}\right) \Rightarrow P\left(|\phi\rangle^{(t+1)}\right) \triangleq |\phi\rangle^{(t)} \in A \Rightarrow G|\phi\rangle^{(t)} \in A$

# Assertion Checking Example (ctd.)

$$P(v) \triangleq v \in A = span\{|\beta\rangle, |\phi\rangle\}$$

We need to check:

1. $P\left(|\phi\rangle^{(0)}\right) \triangleq |\phi\rangle \in span\{|\beta\rangle, |\phi\rangle\}$
2. $P\left(|\phi\rangle^{(t)}\right) \Rightarrow P\left(|\phi\rangle^{(t+1)}\right) \triangleq |\phi\rangle^{(t)} \in A \Rightarrow G|\phi\rangle^{(t)} \in A$

$G|\phi\rangle^{(t)} \in A = G|\beta\rangle \in A \wedge G|\phi\rangle \in A$ By linearity of G and $|\phi\rangle^{(t)} \in A$

# Roadmap

# Future developments

- Programs with measurements
- Conditionals
- Loops
- Mix of classical and quantum computation
- Choosing the optimal abstract space
- Other kinds of assertions

# Computing the Projection of a Support

[noframenumbering] To compute a projection corresponding to $supp(A)$, we:

1. take the rows $\{r_1, ..., r_n\}$ of A;
2. extract an orthonormal set of vectors $\{b_1, ..., b_n\}$ that span the same subspace as the rows;
3. create the matrix $B = \begin{pmatrix} b_1 \\ ... \\ b_n \end{pmatrix}$;
4. return $BB^\dagger$.

# Computing the Projection of an Intersection

[noframenumbering]

$$\{P_1, ..., P_k\}, \quad \forall i \in [k].\, P_i = C^n \times C^n$$

$$\bigcap_{i \in [k]} P_i \triangleq I_n - supp(kI_n - \sum_{i \in [k]} P_i)$$

# Computing the Assertion Projection

[noframenumbering] Given two vectors

$$v_1 = |a_1\rangle \dots |a_n\rangle$$
$$v_2 = |b_1\rangle \dots |b_n\rangle$$

1. Create a matrix $P = \begin{pmatrix} v_1^T \\ v_2^T \\ \mathbf{0} \\ \dots \\ \mathbf{0} \end{pmatrix}$
2. Return $supp(P)$

# Partial trace

$$Tr_B[\rho] = \sum_{v=0}^{2^m} (I_A \otimes \langle v|)\rho(I_A \otimes |v\rangle) \quad Tr_A[\rho] = \sum_{v=0}^{2^n} (\langle v| \otimes I_B)\rho(|v\rangle \otimes I_B)$$

Where $v$ labels vectors of an orthonormal basis of the subspace we are tracing out.

# Weak Galois Connection

$$S \trianglelefteq T$$

$$\forall \overline{\sigma} \in AbsDom(S).\ \forall \overline{\tau} \in AbsDom(T).$$

$$\left( \begin{array}{c} \overline{\tau} \sqsubseteq \gamma_{S \to T}(\overline{\sigma}) \Rightarrow \alpha_{T \to S}(\overline{\tau}) \sqsubseteq \overline{\sigma} \\ \wedge \\ \left(\exists \overline{\rho} \in AbsDom([n]^m).\ \overline{\tau} = \alpha_{[n]^m \to T}(\overline{\rho})\right) \Rightarrow \overline{\tau} \sqsubseteq \gamma_{S \to T}(\overline{\sigma}) \Leftrightarrow \alpha_{T \to S}(\overline{\tau}) \sqsubseteq \overline{\sigma} \end{array} \right)$$