Foundations of Quantum Programming (Lecture 3)

Mingsheng Ying

Centre for Quantum Software and Information
University of Technology Sydney

Outline

Outline

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...
- Quantum operating systems: concurrency

Theoretical Issues

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...
- Quantum operating systems: concurrency
  - Multithreading

Theoretical Issues

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...
- Quantum operating systems: concurrency
  - Multithreading
  - Multiprocessing

Theoretical Issues

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...
- Quantum operating systems: concurrency
  - Multithreading
  - Multiprocessing
  - Mutual exclusion

Theoretical Issues

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...
- Quantum operating systems: concurrency
  - Multithreading
  - Multiprocessing
  - Mutual exclusion
  - Deadlock

Theoretical Issues

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...
- Quantum operating systems: concurrency
  - Multithreading
  - Multiprocessing
  - Mutual exclusion
  - Deadlock
  - Fairness

Theoretical Issues

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...
- Quantum operating systems: concurrency
  - Multithreading
  - Multiprocessing
  - Mutual exclusion
  - Deadlock
  - Fairness
  - .....

Theoretical Issues

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...
- Quantum operating systems: concurrency
  - Multithreading
  - Multiprocessing
  - Mutual exclusion
  - Deadlock
  - Fairness
  - .....

Theoretical Issues

- Execution model of parallel quantum programs?

Parallel Quantum Programming?

- Parallel architecture of quantum computers: IBM, Google, ...
- Quantum operating systems: concurrency
  - Multithreading
  - Multiprocessing
  - Mutual exclusion
  - Deadlock
  - Fairness
  - .....

Theoretical Issues

- Execution model of parallel quantum programs?
- How to reason about parallel quantum programs?

Outline

Owicki-Gries and Lamport Method

1. S. Owicki and D. Gries, An axiomatic proof techniques for parallel programs, *Acta Informatica* 1976.

Owicki-Gries and Lamport Method

1. S. Owicki and D. Gries, An axiomatic proof techniques for parallel programs, *Acta Informatica* 1976.
2. L. Lamport, Proving the correctness of multiprocess programs, *IEEE Transactions on Software Engineering* 1977.

Disjoint Parallel Programs

$$S ::= S_1 \| \cdots \| S_n$$

where $var(S_i) \cap var(S_j) = \emptyset \ (i \neq j)$.

Interleaving Semantics

$$\frac{\langle S_i, \sigma \rangle \rightarrow \langle S_i', \sigma' \rangle}{\langle S_1 \| \cdots \| S_i \| \cdots \| S_n, \sigma \rangle \rightarrow \langle S_1 \| \cdots \| S_i' \| \cdots \| S_n, \sigma' \rangle}$$

Interleaving Semantics

$$\frac{\langle S_i, \sigma\rangle \to \langle S_i', \sigma'\rangle}{\langle S_1 \| \cdots \| S_i \| \cdots \| S_n, \sigma\rangle \to \langle S_1 \| \cdots \| S_i' \| \cdots \| S_n, \sigma'\rangle}$$

Proof Rule for Disjoint Parallelism

$$(\mathit{Rule\ PC}) \quad \frac{\{P_i\} S_i \{Q_i\}\ (i = 1, ..., n)}{\{\bigwedge_{i=1}^{n} P_i\}\ S_1 \| \cdots \| S_n\ \{\bigwedge_{i=1}^{n} Q_i\}}$$

where $\mathit{free}(P_i, Q_i) \cap \mathit{change}(S_j) = \emptyset\ (i \neq j)$.

Question: *Is rule (Rule PC) complete?*

Example

$$\{x = y\}x := x + 1 \| y := y + 1\{x = y\}$$

cannot be derived by (Rule PC)!

Example

$$\{x = y\}x := x + 1 \| y := y + 1\{x = y\}$$

cannot be derived by (Rule PC)!

Auxiliary Variables

$A \subseteq var(S)$ is a set of auxiliary variables of S if each variable $x \in A$ occurs only in assignments of the form $z := t$ with $z \in A$.

Proof Rule for Auxiliary Variables

$$(\mathit{Rule\ AV}) \qquad \frac{\{P\}S\{Q\}}{\{P\}S_0\{Q\}}$$

where S_0 results from S by deleting all assignments to variables in A for some set A of auxiliary variables with $\mathit{free}(Q) \cap A = \emptyset$.

- Exercise: Use (Rule PC) + (Rule AV) to prove $\{x = y\}x := x + 1 \| y := y + 1\{x = y\}$.

Proof Rule for Auxiliary Variables

$$(\textit{Rule AV}) \qquad \frac{\{P\}S\{Q\}}{\{P\}S_0\{Q\}}$$

where S_0 results from S by deleting all assignments to variables in A for some set A of auxiliary variables with $\textit{free}(Q) \cap A = \emptyset$.

- Exercise: Use (Rule PC) + (Rule AV) to prove $\{x = y\}x := x + 1 \| y := y + 1\{x = y\}$.

Completeness Theorem

Hoare logic + (Rule PC) + (Rule AV) is (relative) complete for disjoint parallel programs.

Parallel Programs with Shared Variables

$$\frac{\{x = 0\}x := 0\{x = 0 \vee x = 2\} \quad \{x = 0\}x := x + 2\{x = 0 \vee x = 2\}}{\{x = 0\}x = 0 \| x := x + 2\{x = 0 \vee x = 2\}}\,!$$

$$\frac{\{x = 0\}x := 0\{x = 0 \vee x = 2\} \quad \{x = 0\}x := x + 1; x := x + 1\{x = 0 \vee x = 2\}}{\{x = 0\}x = 0 \| x := x + 1; x := x + 1\{x = 0 \vee x = 2\}}\,?$$

Interference Freedom

- A statement T with precondition $pre(T)$ does not interfere with a proof of $\{P\}S\{Q\}$ if for all assertions R in the proof:

$$\vdash_{\text{par}} \{R \wedge pre(T)\}T\{R\}$$

Interference Freedom

- A statement T with precondition $pre(T)$ does not interfere with a proof of $\{P\}S\{Q\}$ if for all assertions R in the proof:

$$\vdash_{\text{par}} \{R \wedge pre(T)\}T\{R\}$$

- Proofs of $\{P_i\}S_i\{Q_i\}$ $(i = 1, ..., n)$ are interference free if no atomic statement in S_i interferes with proof $\{P_j\}S_j\{Q_j\}$ for $i \neq j$.

Proof Rule for Shared-Memory Concurrency

$$(\textit{Rule } PC') \quad \frac{\text{Proofs of } \{P_i\}S_i\{Q_i\}\ (i = 1, ...n) \text{ are interference free}}{\{\bigwedge_{i=1}^{n} P_i\}S_1 \| \cdots \| S_n\{\bigwedge_{i=1}^{n} Q_i\}}$$

Proof Rule for Shared-Memory Concurrency

$$(\textit{Rule PC}') \quad \frac{\text{Proofs of } \{P_i\}S_i\{Q_i\}\ (i = 1, ...n) \text{ are interference free}}{\{\bigwedge_{i=1}^{n} P_i\}S_1 \| \cdots \| S_n\{\bigwedge_{i=1}^{n} Q_i\}}$$

Completeness Theorem

Hoare logic + (Rule PC') + (Rule AV) is (relative) complete for parallel programs with shared variables.

Outline

Syntax

Generated by the syntax of **qWhile** together with:

$$P ::= P_1 \| \cdots \| P_n$$

where $n > 1$, $P_1, ..., P_n$ are quantum **while**-programs

Syntax

Generated by the syntax of **qWhile** together with:

$$P ::= P_1 \| \cdots \| P_n$$

where $n > 1$, $P_1, ..., P_n$ are quantum **while**-programs

Operational semantics

- A configuration ensemble as a multi-set $\mathcal{A} = \{|\langle P_i, \rho_i \rangle|\}$ of configurations with $\sum_i tr(\rho_i) \leq 1$.
- Interleaving concurrency:

$$\frac{\langle P_i, \rho \rangle \to \left\{ |\langle P'_{ij}, \rho'_j \rangle| \right\}}{\langle P_1 \| ... \| P_{i-1} \| P_i \| P_{i+1} \| ... \| P_n, \rho \rangle \to \left\{ |\langle P_1 \| ... \| P_{i-1} \| P'_{ij} \| P_{i+1} \| ... \| P_n, \rho'_j \rangle| \right\}}$$

Denotational semantics

- For a configuration ensemble $\mathcal{A}$,

$$val(\mathcal{A}) = \sum \{ |\rho' : \langle \downarrow, \rho' \rangle \in \mathcal{A}| \} .$$

Denotational semantics

- For a configuration ensemble $\mathcal{A}$,

$$val(\mathcal{A}) = \sum \left\{ |\rho' : \langle \downarrow, \rho' \rangle \in \mathcal{A}| \right\}.$$

- A computation of a program P starting in a state $\rho \in \mathcal{D}(\mathcal{H}_P)$ is a maximal finite sequence

$$\pi = \langle P, \rho \rangle \to \mathcal{A}_1 \to \cdots \to \mathcal{A}_n \nrightarrow$$

or an infinite sequence

$$\pi = \langle P, \rho \rangle \to \mathcal{A}_1 \to \cdots \to \mathcal{A}_n \to \cdots.$$

Denotational semantics

- For a configuration ensemble $\mathcal{A}$,

$$val(\mathcal{A}) = \sum \left\{ |\rho' : \langle \downarrow, \rho' \rangle \in \mathcal{A}| \right\}.$$

- A computation of a program P starting in a state $\rho \in \mathcal{D}(\mathcal{H}_P)$ is a maximal finite sequence

$$\pi = \langle P, \rho \rangle \to \mathcal{A}_1 \to \cdots \to \mathcal{A}_n \nrightarrow$$

or an infinite sequence

$$\pi = \langle P, \rho \rangle \to \mathcal{A}_1 \to \cdots \to \mathcal{A}_n \to \cdots.$$

- Value of computation π:

$$val(\pi) = \begin{cases} val(\mathcal{A}_n) & \text{if } \pi \text{ is finite and } \mathcal{A}_n \text{ is the last ensemble,} \\ \lim_{n\to\infty} val(\mathcal{A}_n) & \text{if } \pi \text{ is infinite.} \end{cases}$$

Denotational semantics

- For a parallel program P and an input ρ,

$$\mathcal{V}(P,\rho) = \{val(\pi) : \pi \text{ is a computation of } P \text{ starting in } \rho\}$$

$$\overline{\mathcal{V}(P,\rho)} = \left\{ \bigsqcup_k \rho_k : \{\rho_k\} \text{ is an increasing chain in } (\mathcal{V}(P,\rho), \sqsubseteq) \right\}$$

where $\sqsubseteq$ is the Löwner order, $\bigsqcup$ stands for the least upper bound in CPO $(\mathcal{D}(\mathcal{H}_P), \sqsubseteq)$

Denotational semantics

- For a parallel program P and an input ρ,

$$\mathcal{V}(P,\rho) = \{val(\pi) : \pi \text{ is a computation of } P \text{ starting in } \rho\}$$

$$\overline{\mathcal{V}(P,\rho)} = \left\{ \bigsqcup_k \rho_k : \{\rho_k\} \text{ is an increasing chain in } (\mathcal{V}(P,\rho), \sqsubseteq) \right\}$$

  where $\sqsubseteq$ is the Löwner order, $\bigsqcup$ stands for the least upper bound in CPO $(\mathcal{D}(\mathcal{H}_P), \sqsubseteq)$

- Semantic function of P is the mapping $[\![P]\!] : \mathcal{D}(\mathcal{H}_P) \to 2^{\mathcal{D}(\mathcal{H}_P)}$:

$$[\![P]\!](\rho) = \left\{ \text{maximal elements of } \left(\overline{\mathcal{V}(P,\rho)}, \sqsubseteq \right) \right\}$$

Denotational semantics

- For a parallel program P and an input ρ,

$$\mathcal{V}(P,\rho) = \{val(\pi) : \pi \text{ is a computation of } P \text{ starting in } \rho\}$$

$$\overline{\mathcal{V}(P,\rho)} = \left\{ \bigsqcup_k \rho_k : \{\rho_k\} \text{ is an increasing chain in } (\mathcal{V}(P,\rho), \sqsubseteq) \right\}$$

  where $\sqsubseteq$ is the Löwner order, $\bigsqcup$ stands for the least upper bound in CPO $(\mathcal{D}(\mathcal{H}_P), \sqsubseteq)$

- Semantic function of P is the mapping $[\![P]\!] : \mathcal{D}(\mathcal{H}_P) \to 2^{\mathcal{D}(\mathcal{H}_P)}$:

$$[\![P]\!](\rho) = \left\{ \text{maximal elements of } \left(\overline{\mathcal{V}(P,\rho)}, \sqsubseteq \right) \right\}$$

- Zorn's lemma: $[\![P]\!](\rho)$ is nonempty.

Correctness of parallel quantum programs

Correctness formula (Hoare triple) $\{A\}P\{B\}$ is true in the sense of total correctness (respectively, partial correctness), written

$$\models_{tot} \{A\}P\{B\} \quad (resp. \ \models_{par} \{A\}P\{B\}),$$

if for each input ρ, we have:

$$\mathrm{tr}(A\rho) \leq \mathrm{tr}(B\rho')$$
$$resp.\ \mathrm{tr}(A\rho) \leq \mathrm{tr}(B\rho') + [\mathrm{tr}(\rho) - \mathrm{tr}(\rho')] \text{ for all } \rho' \in [\![P]\!](\rho)$$

Outline

Proof Rule for (Classical) Disjoint Parallelism

$$(PC) \quad \frac{\{P_i\} S_i \{Q_i\} \ (i = 1, ..., n)}{\{\bigwedge_{i=1}^{n} P_i\} \ S_1 \| \cdots \| S_n \ \{\bigwedge_{i=1}^{n} Q_i\}}$$

where $free(P_i, Q_i) \cap change(S_j) = \emptyset \ (i \neq j)$.

Proof Rule for (Classical) Disjoint Parallelism

$$(PC) \quad \frac{\{P_i\}S_i\{Q_i\}\ (i=1,...,n)}{\{\bigwedge_{i=1}^n P_i\}\, S_1 \| \cdots \| S_n\, \{\bigwedge_{i=1}^n Q_i\}}$$

where $free(P_i, Q_i) \cap change(S_j) = \emptyset\ (i \neq j)$.

Problem

Find a (relative) complete (!) axiomatic system for parallel quantum programs.

- Quantum counterparts of conjunctives $\bigwedge_{i=1}^n P_i$ and $\bigwedge_{i=1}^n Q_i$?

$$(QPC) \quad \frac{\{A_i\}\, P_i\, \{B_i\}\ \ (i=1,...,n)}{\{\bigotimes_{i=1}^n A_i\}\, P_1 \| \cdots \| P_n\, \{\bigotimes_{i=1}^n B_i\}}$$

Entanglement?

Entanglement?

Theorem [Gurvits and Barmun 2003]

Let $\mathcal{H}_1, \cdots, \mathcal{H}_n$ be finite-dimensional Hilbert spaces, and let A be a positive operator in $\bigotimes_{i=1}^{n} \mathcal{H}_n$. If

$$\|A - I\|_2 \leq \frac{1}{2^{n/2-1}}$$

then A is separable, where $\|\cdot\|_2$ is the Hilbert-Schmidt (Frobenius) norm:

$$\|X\|_2 := \sqrt{\mathrm{Tr}(X^\dagger X)}.$$

Entanglement?

Theorem [Gurvits and Barmun 2003]

Let $\mathcal{H}_1, \cdots, \mathcal{H}_n$ be finite-dimensional Hilbert spaces, and let A be a positive operator in $\bigotimes_{i=1}^{n} \mathcal{H}_n$. If

$$\|A - I\|_2 \leq \frac{1}{2^{n/2-1}}$$

then A is separable, where $\|\cdot\|_2$ is the Hilbert-Schmidt (Frobenius) norm:

$$\|X\|_2 := \sqrt{\mathrm{Tr}(X^\dagger X)}.$$

Transfer separable predicates to entangled

$$\text{(S2E)} \quad \frac{\{(1-\epsilon)I + \epsilon A\}\, P\, \{(1-\epsilon)I + \epsilon B\}}{\{A\}\, P\, \{B\}}$$

Completeness Theorem

(QPC) + (S2E) + (SP) + (AP) + (TP) is (relative) complete.

$$\text{(SP)}\quad \frac{P_i : \text{Abort}(C_i)\ \ P_i : \text{Term}(D_i)\ \ \{D_i + A_i\}\, P_i\, \{B_i\} \qquad (i = 1, ..., n)}{\{I - \bigotimes_{i=1}^{n}(I_i - C_i) + \bigotimes_{i=1}^{n} A_i\}\, P_1 \| \cdots \| P_n\, \{\bigotimes_{i=1}^{n} B_i\}}$$

$$\text{(AP)}\quad \frac{P_i : \text{Abort}(A_i)\ (i = 1, \cdots, n)}{P_1 \| \cdots \| P_m : \text{Abort}\,(I - \bigotimes_{i=1}^{n}(I_i - A_i))}$$

$$\text{(TP)}\quad \frac{P_i : \text{Term}(A_i)\ (i = 1, \cdots, n)}{P_1 \| \cdots \| P_m : \text{Term}\,(I - \bigotimes_{i=1}^{m}(I_i - A_i))}$$

[1] M. S. Ying, L. Zhou, Y. J. Li and Y. Feng, A proof system for disjoint parallel quantum programs, *TCS* 2022.

Completeness Theorem

(QPC) + (S2E) + (SP) + (AP) + (TP) is (relative) complete.

$$\text{(SP)}\quad \frac{P_i : \text{Abort}(C_i)\ \ P_i : \text{Term}(D_i)\ \ \{D_i + A_i\}\, P_i\, \{B_i\} \qquad (i = 1, ..., n)}{\{I - \bigotimes_{i=1}^{n}(I_i - C_i) + \bigotimes_{i=1}^{n} A_i\}\, P_1 \| \cdots \| P_n\, \{\bigotimes_{i=1}^{n} B_i\}}$$

$$\text{(AP)}\quad \frac{P_i : \text{Abort}(A_i)\ (i = 1, \cdots, n)}{P_1 \| \cdots \| P_m : \text{Abort}\,(I - \bigotimes_{i=1}^{n}(I_i - A_i))}$$

$$\text{(TP)}\quad \frac{P_i : \text{Term}(A_i)\ (i = 1, \cdots, n)}{P_1 \| \cdots \| P_m : \text{Term}\,(I - \bigotimes_{i=1}^{m}(I_i - A_i))}$$

[1] M. S. Ying, L. Zhou, Y. J. Li and Y. Feng, A proof system for disjoint parallel quantum programs, *TCS* 2022.

Open problem

Infinite-dimensional?

Outline

Proof Rule for Shared-Memory Concurrency

$$(\textit{Rule PC}') \qquad \frac{\text{Proofs of } \{P_i\}S_i\{Q_i\}\ (i = 1, ...n) \text{ are interference free}}{\{\bigwedge_{i=1}^{n} P_i\}S_1 \| \cdots \| S_n\{\bigwedge_{i=1}^{n} Q_i\}}$$

- Quantum counterparts of conjunctives $\bigwedge_{i=1}^{n} P_i$ and $\bigwedge_{i=1}^{n} Q_i$?

$$\bigotimes_{i=1}^{n} A_i \qquad \bigotimes_{i=1}^{n} B_i?$$

Proof Rule for Shared-Memory Concurrency

$$(\textit{Rule PC}') \qquad \frac{\text{Proofs of } \{P_i\}S_i\{Q_i\} \ (i = 1, ...n) \text{ are interference free}}{\{\bigwedge_{i=1}^{n} P_i\}S_1 \| \cdots \| S_n\{\bigwedge_{i=1}^{n} Q_i\}}$$

- Quantum counterparts of conjunctives $\bigwedge_{i=1}^{n} P_i$ and $\bigwedge_{i=1}^{n} Q_i$?

$$\bigotimes_{i=1}^{n} A_i \qquad \bigotimes_{i=1}^{n} B_i?$$

- $\mathcal{H}_{S_1\|\cdots\|S_n} \neq \bigotimes_{i=1}^{n} \mathcal{H}_{S_i}$!

Reasoning about Quantum Shared-Memory Concurrency

This problem is closely related to:

- **Quantum Marginal Problem**: Given a family $\mathcal{J}$ of subsets of $\{1, ..., n\}$, and for each $J \in \mathcal{J}$, given a mixed state (density operator) ρ_J in $\mathcal{H}_J$. Is there a join (global state) of $\{\rho_j\}_{J \in \mathcal{J}}$ in $\mathcal{H} = \bigotimes_{i=1}^{n} \mathcal{H}_i$?

[2] A.A. Klyachko, Quantum marginal problem and N-representability, *Journal of Physics* 2006.
[3] M. S. Ying, Chapter 9 of *Foundations of Quantum Programming* (2024).

Reasoning about Quantum Shared-Memory Concurrency

This problem is closely related to:

- **Quantum Marginal Problem**: Given a family $\mathcal{J}$ of subsets of $\{1, ..., n\}$, and for each $J \in \mathcal{J}$, given a mixed state (density operator) ρ_J in $\mathcal{H}_J$. Is there a join (global state) of $\{\rho_j\}_{J \in \mathcal{J}}$ in $\mathcal{H} = \bigotimes_{i=1}^{n} \mathcal{H}_i$?

[2] A.A. Klyachko, Quantum marginal problem and N-representability, *Journal of Physics* 2006.
[3] M. S. Ying, Chapter 9 of *Foundations of Quantum Programming* (2024).

THANKS!