Termination Analysis of Quantum Programs

Yangjia Li, University of Tartu, Estonia

Joint work with Prof. Mingsheng Ying at
University of Technology, Sydney, Australia

Outline

- Introduction: program termination problems
- Formalization: quantum programs and their termination
- Result I: decidability for finite-dimensional programs
- Result II: LRSM-based approach for general programs
- Summary

Introduction

Termination problems

Termination analysis of classical programs

- Program termination is generally undecidable. (Halting problem)

- Incomplete approaches for positive results:
  -Linear Program, e.g. [Tiwari, CAV'04]
  -Multi-path Polynomial Program, e.g. [Bradley et al, VMCAI'05]
  -Predicate abstraction, e.g. [P. Cousot & R. Cousot, POPL'12]
  -Ranking functions, plenty of results, traced back to [Floyd, 1967]

- Boundary: hard even for some simple programs.

An open problem

- Termination problem of linear while-loop:

$$\textbf{while } x_1 > 0 \textbf{ do } \boldsymbol{x} := A\boldsymbol{x} \textbf{ od}$$

where $\boldsymbol{x} \in \mathbb{R}^6$ and A is a 6×6 real matrix.

An open problem

- Termination problem of linear while-loop:

$$\textbf{while } x_1 > 0 \textbf{ do } \boldsymbol{x} := A\boldsymbol{x} \textbf{ od}$$

  where $\boldsymbol{x} \in \mathbb{R}^6$ and A is a 6×6 real matrix.

- Computation of the Homogeneous Diophantine Approximation Type:

$$L(x) = \inf \left\{ c \in \mathbb{R} : \left| x - \frac{n}{m} \right| < \frac{c}{m^2} \text{ for some } n, m \in \mathbb{Z} \right\}$$

An open problem

- Termination problem of linear while-loop:

$$\textbf{while } x_1 > 0 \textbf{ do } \boldsymbol{x} := A\boldsymbol{x} \textbf{ od}$$

  where $\boldsymbol{x} \in \mathbb{R}^6$ and A is a 6×6 real matrix.

- Computation of the Homogeneous Diophantine Approximation Type:

$$L(x) = \inf \left\{ c \in \mathbb{R} : \left| x - \frac{n}{m} \right| < \frac{c}{m^2} \text{ for some } n, m \in \mathbb{Z} \right\}$$

- Reduction [Ouaknine and Worrell, SODA'14]: (1) $\Rightarrow$ (2).
  (1) Decidability of the termination problem;
  (2) Computability of $L(x)$ for a set of transcendental numbers x:

$$X = \left\{ \frac{\arg(p + \mathrm{i}q)}{2\pi} : p, q \in \mathbb{Q}, p, q \neq 0 \text{ and } p^2 + q^2 = 1 \right\}.$$

Termination analysis of quantum programs

- Motivation: verification of quantum programs, just like classical case.
  -In quantum Hoare logic [Ying, TOPLAS 33(2011),19]

  total correctness = partial correctness + termination analysis

- Method: quantum generalization of classical techniques

- Novelty: fundamental differences between classical and quantum.

Example: A simple quantum walk

- Consider a random walk on a square $ABCD$ starting at vertex A and terminating at vertex C. Then it terminates with probability 1.

$$
\begin{tikzcd}
A \arrow[r, leftrightarrow, "1/3"] \arrow[d, leftrightarrow, "1/3"'] & B \arrow[d, "1/3"] \\
D \arrow[r, "1/3"] & C
\end{tikzcd}
$$

Example: A simple quantum walk

- Consider a random walk on a square $ABCD$ starting at vertex A and terminating at vertex C. Then it terminates with probability 1.

$$\begin{tikzcd}
A \arrow[r, leftrightarrow, "1/3"] \arrow[d, leftrightarrow, "1/3"'] & B \arrow[d, "1/3"] \\
D \arrow[r, "1/3"] & C
\end{tikzcd}$$

- Consider a quantum version of the walk: unitary operations W_1 and W_2 are alternatingly taken during the process. Then it terminates with probability 0.

$$W_1 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 0 & -1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & -1 & 1 \end{pmatrix}, \; W_2 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 0 & 1 \\ -1 & 1 & -1 & 0 \\ 0 & 1 & 1 & -1 \\ 1 & 0 & -1 & -1 \end{pmatrix}$$

Formalisation

Quantum programs and termination problems

Syntax of quantum programs

Grammar of quantum while-programs (with nondeterminism)

$$
\begin{aligned}
P ::= \ & \mathbf{skip} \mid P_1; P_2 \mid q := |0\rangle \mid \overline{q} := U\overline{q} && (1)\\
& \mid \mathbf{if}\ \ (\square_m\ M[\overline{q}] = m \rightarrow P_m)\ \ \mathbf{fi} && (2)\\
& \mid \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ P\ \mathbf{od} && (3)\\
& \mid P_1 \sqcup P_2 \mid P_1 \sqcap P_2 \mid P_1 \parallel P_2 && (4)
\end{aligned}
$$

Syntax of quantum programs

Grammar of quantum while-programs (with nondeterminism)

$$
\begin{aligned}
P ::=\ & \mathbf{skip} \mid P_1; P_2 \mid q := |0\rangle \mid \overline{q} := U\overline{q} && (1)\\
& \mid \mathbf{if}\ \ (\Box_m\ M[\overline{q}] = m \to P_m)\ \ \mathbf{fi} && (2)\\
& \mid \mathbf{while}\ M[\overline{q}] = 1\ \mathbf{do}\ P\ \mathbf{od} && (3)\\
& \mid P_1 \sqcup P_2 \mid P_1 \sqcap P_2 \mid P_1 \parallel P_2 && (4)
\end{aligned}
$$

- Sequential quantum while-program with extension in (4)
- In (1), $\mathbf{skip}$ command and sequential composition are just as the same as in the classical case; quantum initialization and unitary transformation form quantum counterpart of the classical assignment command.
- (2) is the quantum case statement, and (3) is the quantum $\mathbf{while}$-loop, in both of which probabilistic choices are involved. (According to the Copenhagen interpretation.)
- (4) defines structures about nondeterminism: angelic choice, demonic choice and parallel composition, just like the classical programs.

Operational Semantics

Probabilistic transitions of while-loop:

$$\langle \textbf{while } M[\overline{q}] = 1 \textbf{ do } P \textbf{ od}, \rho \rangle \xrightarrow{\mathcal{M}_0} \langle \downarrow, M_0 \rho M_0^\dagger \rangle,$$

$$\langle \textbf{while } M[\overline{q}] = 1 \textbf{ do } P \textbf{ od}, \rho \rangle \xrightarrow{\mathcal{M}_1} \langle P; \textbf{while } M[\overline{q}] = 1 \textbf{ do } P \textbf{ od}, M_1 \rho M_1^\dagger \rangle$$

Operational Semantics

Probabilistic transitions of while-loop:

$$\langle \textbf{while } M[\overline{q}] = 1 \textbf{ do } P \textbf{ od}, \rho\rangle \xrightarrow{\mathcal{M}_0} \langle \downarrow, M_0\rho M_0^\dagger\rangle,$$

$$\langle \textbf{while } M[\overline{q}] = 1 \textbf{ do } P \textbf{ od}, \rho\rangle \xrightarrow{\mathcal{M}_1} \langle P; \textbf{while } M[\overline{q}] = 1 \textbf{ do } P \textbf{ od}, M_1\rho M_1^\dagger\rangle$$

Other transitions:

$$\langle \textbf{skip}, \rho\rangle \xrightarrow{\mathcal{I}} \langle \downarrow, \rho\rangle, \ \langle \overline{q} := U\overline{q}, \rho\rangle \xrightarrow{\mathcal{U}} \langle \downarrow, U\rho U^\dagger\rangle,$$

$$\langle q := 0, \rho\rangle \xrightarrow{|0\rangle_q\langle 0|\otimes tr_q} \langle \downarrow, |0\rangle_q\langle 0| \otimes tr_q(\rho)\rangle$$

$$\langle \textbf{if } \ (\square_m \ M[\overline{q}] = m \to P_m) \ \textbf{fi}, \rho\rangle \xrightarrow{\mathcal{M}_m} \langle P_m, M_m\rho M_m^\dagger\rangle \ \forall m.$$

$$\langle P_1 \sqcup P_2, \rho\rangle \xrightarrow{\mathcal{I}}_\sqcup \langle P_1, \rho\rangle, \ \langle P_1 \sqcup P_2, \rho\rangle \xrightarrow{\mathcal{I}}_\sqcup \langle P_2, \rho\rangle,$$

$$\langle P_1 \sqcap P_2, \rho\rangle \xrightarrow{\mathcal{I}}_\sqcap \langle P_1, \rho\rangle, \ \langle P_1 \sqcap P_2, \rho\rangle \xrightarrow{\mathcal{I}}_\sqcap \langle P_2, \rho\rangle$$

$$\frac{\langle P_1, \rho\rangle \xrightarrow{\mathcal{M}} \langle P_1', \rho'\rangle}{\langle P_1 \parallel P_2, \rho\rangle \xrightarrow{\mathcal{M}}_\sqcap \langle P_1' \parallel P_2, \rho'\rangle}, \ \frac{\langle P_2, \rho\rangle \xrightarrow{\mathcal{M}} \langle P_2', \rho'\rangle}{\langle P_1 \parallel P_2, \rho\rangle \xrightarrow{\mathcal{M}}_\sqcap \langle P_1 \parallel P_2', \rho'\rangle}$$

Denotational Semantics

State transformation: $\rho_{out} = [\![P]\!](\rho_{in})$

Denotational Semantics

State transformation: $\rho_{out} = [\![P]\!](\rho_{in})$

- Nondeterministic choices are made according to the history by an angelic scheduler σ and a demonic scheduler τ.
- Then the execution path p follows from the probabilistic choices:

  $$p = \langle P_0, \rho_0 \rangle \xrightarrow{\mathcal{E}_1} \cdots \xrightarrow{\mathcal{E}_n} \langle P_n, \rho_n \rangle.$$

  Define $[\![p]\!] = \mathcal{E}_n \circ \cdots \circ \mathcal{E}_1$.
- $$[\![P(\sigma, \tau)]\!] = \sum_{p \in Path(\sigma,\tau)} [\![p]\!],$$

  where $Path(\sigma, \tau)$ is the set of all paths p with $P_0 = P$ and $P_n = \downarrow$, under an angelic scheduler σ and a demonic scheduler τ.

Definitions

For a quantum prgram P and an input state ρ, define

- Termination probability:

$$TP_{\sigma,\tau}(\rho) = tr([\![P(\sigma,\tau)]\!](\rho_{in})).$$

- Expected running time:
  if $TP_{\sigma,\tau}(\rho) < 1$, $ET_{\sigma,\tau}(\rho) = \infty$; otherwise,

$$ET_{\sigma,\tau}(\rho) = \sum_{T=0}^{\infty} tr([\![P_T(\sigma,\tau)]\!](\rho)) \times T,$$

where $[\![P_T(\sigma,\tau)]\!] = \sum\{[\![p]\!] \mid p \in Path(\sigma,\tau), |p| = T\}$.

Termination problems

Definition I (Almost-sure termination)

A quantum program S is almost-surely terminating under input ρ, if

$$\exists\sigma\ \forall\tau.\ TP_{\sigma,\tau}(\rho) = 1. \tag{5}$$

Definition II (Finite termination)

A quantum program S is finitely terminating under input ρ_{in}, if

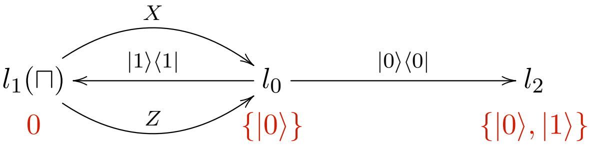
$$\exists\sigma\ \forall\tau.\ ET_{\sigma,\tau}(\rho) < \infty. \tag{6}$$

Result I

Decidability for finite-dimensional programs

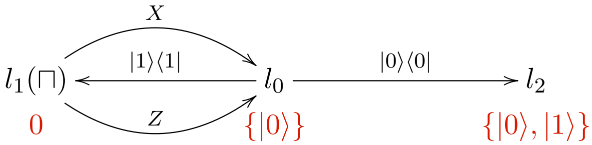
An illustrative example

- Program: $\textbf{while } M[q] = 1 \textbf{ do } q := X[q] \ \sqcap \ q := Z[q] \textbf{ od}$



An illustrative example

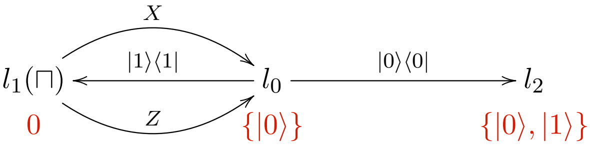
- Program: $\mathbf{while}\ M[q] = 1\ \mathbf{do}\ q := X[q]\ \sqcap\ q := Z[q]\ \mathbf{od}$



- Note: terminating subspaces form an invariant of the program.

An illustrative example

- Program: $\mathbf{while}\ M[q] = 1\ \mathbf{do}\ q := X[q]\ \sqcap\ q := Z[q]\ \mathbf{od}$



- Note: terminating subspaces form an invariant of the program.
- Difficulty in invariant generation: nontrivial invariant (neither I nor 0).

Result description

Theorem I (Computability of S_{in})

Given a finite-dimensional quantum program P, the set

$$S_{in} = \{|\psi\rangle : \exists\sigma.\forall\tau.\, TP_{\sigma,\tau}(|\psi\rangle\langle\psi|) = 1\}$$

of terminating initial pure states is computable.

Result description

Theorem I (Computability of S_{in})

Given a finite-dimensional quantum program P, the set

$$S_{in} = \{|\psi\rangle : \exists\sigma.\forall\tau.\, TP_{\sigma,\tau}(|\psi\rangle\langle\psi|) = 1\}$$

of terminating initial pure states is computable.

- A.S.T $\Leftrightarrow$ F.T for finite-dimensional programs.
- Termination is decidable by checking $\mathrm{supp}(\rho_{in}) \in S$, where
  1. Without angelic choice, S_{in} is the subspace S;
  2. With angelic choice, S_{in} would be a finite union of different subspaces due to different angelic strategy σ, then S can be any one of them.
- It is a generalization of our previous result for nondeterministic quantum loops. [Li et al, Acta Inform. 51(2015),1]

Generalized 0-1 Law

Lemma I (Generalized 0-1 Law)

Let X be an invariant subspace of a quantum Markov chain possibly with demonic choice, and $T(\rho)$ the termination probability starting from a state ρ, then

$$\inf_{|\psi\rangle \in X} T(|\psi\rangle\langle\psi|) = 0 \text{ or } 1.$$

Generalized 0-1 Law

Lemma I (Generalized 0-1 Law)

Let X be an invariant subspace of a quantum Markov chain possibly with demonic choice, and $T(\rho)$ the termination probability starting from a state ρ, then

$$\inf_{|\psi\rangle \in X} T(|\psi\rangle\langle\psi|) = 0 \text{ or } 1.$$

- In finite-dimensional case, infimum 0 is always reachable, but by a trickier proof than the classical case.
- The set of diverging states: $D_l = \{|\psi\rangle : T_l(|\psi\rangle\langle\psi|) = 0\}$.
- Condition for generation of termination subspaces: $S_l \cap D_l = 0$.

Generation of diverging states

- $\{D_l\}_l$ is the greatest fixed point under some transition relation.
- Algorithm: Generate $\{D_l\}_l$ firstly, and then $\{S_l\}_l$ accordingly.

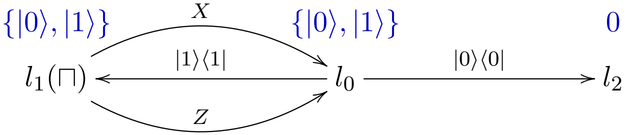
Generation of diverging states

- $\{D_l\}_l$ is the greatest fixed point under some transition relation.
- Algorithm: Generate $\{D_l\}_l$ firstly, and then $\{S_l\}_l$ accordingly.
- Example: $\mathbf{while}\ M[q] = 1\ \mathbf{do}\ q := X[q]\ \sqcap\ q := Z[q]\ \mathbf{od}$

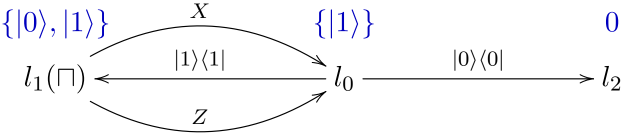
$$\begin{tikzcd}
l_1(\sqcap) \arrow[rr, bend left, "X"] \arrow[rr, bend right, "Z"'] & & l_0 \arrow[ll, "|1\rangle\langle 1|"'] \arrow[rr, "|0\rangle\langle 0|"] & & l_2
\end{tikzcd}$$

Generation of diverging states

- $\{D_l\}_l$ is the greatest fixed point under some transition relation.
- Algorithm: Generate $\{D_l\}_l$ firstly, and then $\{S_l\}_l$ accordingly.
- Example: $\textbf{while } M[q] = 1 \textbf{ do } q := X[q] \ \sqcap \ q := Z[q] \textbf{ od}$



- $\{D_l\}_l$ is the greatest fixed point under some transition relation.
- Algorithm: Generate $\{D_l\}_l$ firstly, and then $\{S_l\}_l$ accordingly.
- Example: $\textbf{while}\ M[q] = 1\ \textbf{do}\ q := X[q]\ \sqcap\ q := Z[q]\ \textbf{od}$



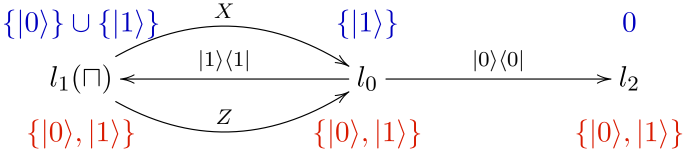
Generation of diverging states

- $\{D_l\}_l$ is the greatest fixed point under some transition relation.
- Algorithm: Generate $\{D_l\}_l$ firstly, and then $\{S_l\}_l$ accordingly.
- Example: $\textbf{while } M[q] = 1 \textbf{ do } q := X[q] \ \sqcap \ q := Z[q] \textbf{ od}$

$$
\begin{tikzcd}
\{|0\rangle\} \cup \{|1\rangle\} & \{|1\rangle\} & 0 \\
l_1(\sqcap) \arrow[r, bend left, "X"] \arrow[r, bend right, "Z"'] & l_0 \arrow[l, "|1\rangle\langle 1|"'] \arrow[r, "|0\rangle\langle 0|"] & l_2
\end{tikzcd}
$$

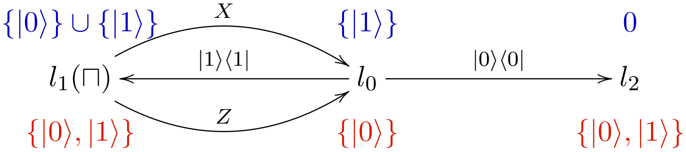
Generation of diverging states

- $\{D_l\}_l$ is the greatest fixed point under some transition relation.
- Algorithm: Generate $\{D_l\}_l$ firstly, and then $\{S_l\}_l$ accordingly.
- Example: $\mathbf{while}\ M[q] = 1\ \mathbf{do}\ q := X[q]\ \sqcap\ q := Z[q]\ \mathbf{od}$



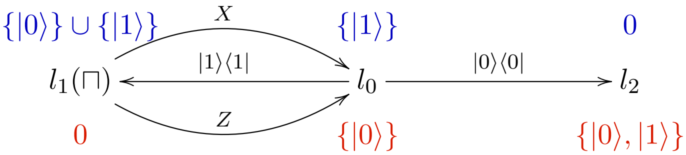
Generation of diverging states

- $\{D_l\}_l$ is the greatest fixed point under some transition relation.
- Algorithm: Generate $\{D_l\}_l$ firstly, and then $\{S_l\}_l$ accordingly.
- Example: $\textbf{while } M[q] = 1 \textbf{ do } q := X[q] \ \sqcap \ q := Z[q] \textbf{ od}$



Generation of diverging states

- $\{D_l\}_l$ is the greatest fixed point under some transition relation.
- Algorithm: Generate $\{D_l\}_l$ firstly, and then $\{S_l\}_l$ accordingly.
- Example: $\mathbf{while}\ M[q] = 1\ \mathbf{do}\ q := X[q]\ \sqcap\ q := Z[q]\ \mathbf{od}$



Complexity

Lemma II (Descending Chain Condition)

In a finite-dimensional Hilbert space, a descending chain

$$S_1 \supseteq S_2 \supseteq \cdots \supseteq S_k \supseteq \cdots$$

always terminates at some S_n, i.e., $S_m = S_n$ for all $m > n$, if each S_k is a finite union of subspaces.

Complexity

Lemma II (Descending Chain Condition)

In a finite-dimensional Hilbert space, a descending chain

$$S_1 \supseteq S_2 \supseteq \cdots \supseteq S_k \supseteq \cdots$$

always terminates at some S_n, i.e., $S_m = S_n$ for all $m > n$, if each S_k is a finite union of subspaces.

- A consequence of Hilbert's basis theorem.
- Particularly used here for generation of finite union D_l of subspaces.
- An Ackermannian function $A(d, n)$ w.r.t. the dimension d and the program size n can be found as a complexity upper bound.

Result II

An LRSM-based approach

Ranking function based approach

- The notion of Ranking Super-Martingale (RSM) has been introduced in the study of probabilistic programs, and successfully used for termination analysis of them.
  [Fioriti & Hermanns, POPL'15], [Chatterjee et al, POPL'16]

- We introduce the notion of Linear Ranking Super-Martingale (LRSM) as a quantum generalization of RSM, and apply it to termination analysis for quantum programs. [Li & Ying, POPL'18]

An illustrative example

- Program: $\mathbf{while}\ M[q] = 1\ \mathbf{do}\ q := X[q]\ \sqcap\ q := H[q]\ \mathbf{od}$

$$\begin{tikzcd}
l_1(\sqcap) \arrow[rr, bend left, "X"] \arrow[rr, bend right, "H"'] & & l_0 \arrow[ll, "|1\rangle\langle 1|"'] \arrow[rr, "|0\rangle\langle 0|"] & & l_2 \\
f_1(\rho) & & f_0(\rho) & & f_2(\rho)
\end{tikzcd}$$

An illustrative example

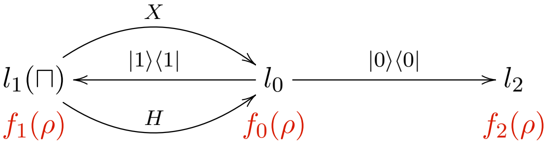
- Program: **while** $M[q] = 1$ **do** $q := X[q] \ \sqcap \ q := H[q]$ **od**

$$\begin{tikzcd}
l_1(\sqcap) \arrow[r, bend left, "X"] \arrow[r, bend right, "H"] & l_0 \arrow[l, "|1\rangle\langle 1|"'] \arrow[r, "|0\rangle\langle 0|"] & l_2 \\
f_1(\rho) & f_0(\rho) & f_2(\rho)
\end{tikzcd}$$

- Constraint: for all density operators ρ,
  (1) $f_0(\rho), f_1(\rho), f_2(\rho) \geq K$;
  (2) $f_1(\rho) \geq f_0(X\rho X) + \epsilon$, $f_1(\rho) \geq f_0(H\rho H) + \epsilon$, and
  $f_0(\rho) \geq f_2(\rho_{00} \cdot |0\rangle\langle 0|) + f_1(\rho_{11} \cdot |1\rangle\langle 1|) + \epsilon$.

An illustrative example

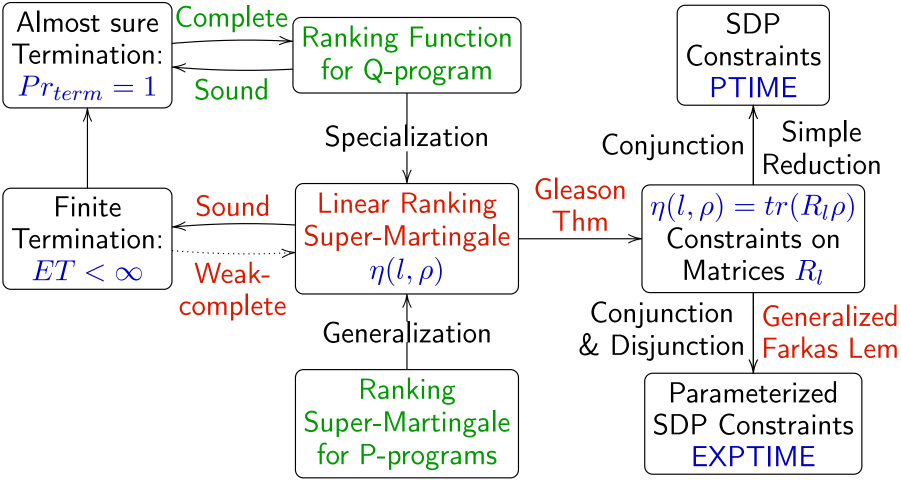
- Program: $\mathbf{while}\ M[q] = 1\ \mathbf{do}\ q := X[q]\ \sqcap\ q := H[q]\ \mathbf{od}$



- Constraint: for all density operators ρ,
  (1) $f_0(\rho), f_1(\rho), f_2(\rho) \geq K$;
  (2) $f_1(\rho) \geq f_0(X\rho X) + \epsilon$, $f_1(\rho) \geq f_0(H\rho H) + \epsilon$, and
  $f_0(\rho) \geq f_2(\rho_{00} \cdot |0\rangle\langle 0|) + f_1(\rho_{11} \cdot |1\rangle\langle 1|) + \epsilon$.
- Solution: choose $\epsilon = 0.5$, $K = f_2(\rho) = 0$, $f_0(\rho) = tr(A\rho)$, It suffices to find an operator A such that

$$0 \sqsubseteq A \ \wedge\ \max\{\langle 0|A|0\rangle, \langle -|A|-\rangle\} \cdot |1\rangle\langle 1| + I \sqsubseteq A.$$

Overview


Almost sure Termination: $Pr_{term} = 1$
Complete
Sound
Ranking Function for Q-program
SDP Constraints PTIME
Specialization
Conjunction
Simple Reduction
Finite Termination: $ET < \infty$
Sound
Weak-complete
Linear Ranking Super-Martingale $\eta(l, \rho)$
Gleason Thm
$\eta(l, \rho) = tr(R_l \rho)$ Constraints on Matrices R_l
Generalization
Conjunction & Disjunction
Generalized Farkas Lem
Ranking Super-Martingale for P-programs
Parameterized SDP Constraints EXPTIME

Linear Ranking Super-Martingale

A (K, ϵ)-Linear Ranking Super-Martingale for a quantum program P with respect to an invariant $\{O_l\}_{l \in L}$ is a function $\eta : L \times \mathcal{D}(\mathcal{H}) \to \mathbb{R}$ satisfying:

1. Linearity: $\eta(l, p\rho + q\sigma) = p\eta(l, \rho) + q\eta(l, \sigma)$
2. K-lower bounded: $\eta(l, \rho) \geq K + tr(O_l \rho) - 1$
3. ϵ-decreasing: $\eta(l, \rho) - pre_\eta(l, \rho) \geq \epsilon + tr(O_l \rho) - 1$

for all $l \in L$, density operators ρ and σ, and $p, q \geq 0$.

Linear Ranking Super-Martingale

A (K, ϵ)-Linear Ranking Super-Martingale for a quantum program P with respect to an invariant $\{O_l\}_{l \in L}$ is a function $\eta : L \times \mathcal{D}(\mathcal{H}) \to \mathbb{R}$ satisfying:

1. Linearity: $\eta(l, p\rho + q\sigma) = p\eta(l, \rho) + q\eta(l, \sigma)$
2. K-lower bounded: $\eta(l, \rho) \geq K + tr(O_l \rho) - 1$
3. ϵ-decreasing: $\eta(l, \rho) - pre_\eta(l, \rho) \geq \epsilon + tr(O_l \rho) - 1$

for all $l \in L$, density operators ρ and σ, and $p, q \geq 0$.

- L is the set of instructions and $\mathcal{H}$ is the state space of P;
- The pre-expectation of η is defined for a regular (resp. angelic, demonic) instruction l as

$$pre_\eta(l, \rho) = \Delta\{\eta(l', \rho') \mid (l, \rho) \to (l', \rho')\},$$

where $\Delta = \Sigma$ (resp. $\min$, $\max$).

Termination theorems

Theorem I (Termination under additive invariants)

If a quantum program P has a (K, ϵ)-LRSM, then it is finitely terminating with any input satisfying $tr(O_{in}\rho_{in}) = 1$, and

$$ET \leq \frac{\eta(l_{in}, \rho_{in}) - K}{\epsilon}.$$

- Proved in a similar but somehow different way to the probabilistic case.

Theorem II (Termination under multiplicative invariants)

If a quantum program P has a (K, ϵ)-LRSM w.r.t. a multiplicative invariant $\{O_l\}_l$, then it is finitely terminating with any input satisfying $tr(O_{in}\rho_{in}) > 1 - \epsilon$, and

$$ET \leq \frac{\eta(l_{in}, \rho_{in}) - K + 1 - tr(O_{in}\rho_{in})}{\epsilon + tr(O_{in}\rho_{in}) - 1}.$$

- Proved by reduction to the (classical) Foster Theorem

Weak completeness

Theorem III (Weak completeness)

A deterministic quantum program S is finite terminating for every input iff it has a $(0,1)$-LRSM with respect to the trivial invariant.

- Proof: LRSM can be constructed from the ET.
- Weakness: With nondeterministic choice ET may be non-linear.
- Special case: Quantum Markov Chain.
- A quantum generalization of the Foster Theorem on classical Markov chain.

LRSM synthesis by SDP

Gleason Theorem

If $\mathcal{H}$ is separable and $\dim \mathcal{H} > 2$, then for each measure m on $\mathcal{S}(\mathcal{H})$, there exists a unique positive Hermitian matrix R with $tr(R) = 1$ such that

$$m(X) = tr(RP_X)$$

for all $X \in \mathcal{S}(\mathcal{H})$, where P_X is the project onto X.

- Absence of angelic choice $\Rightarrow$ conjunction form $\Rightarrow$ SDP problem
  SDP constraints on R_l: $\sum_l \mathcal{A}_l(R_l) \sqsupseteq C$.
- Difficulty: Angelic choice $\Rightarrow$ disjunction form, e.g.,

$$\forall \rho \in \mathcal{D}(\mathcal{H}). \max\{\sum_l tr(\rho \mathcal{A}_l(R_l)), \sum_l tr(\rho \mathcal{B}_l(R_l))\} \geq tr(\rho C).$$

LRSM synthesis with angelic choice

Generalized Farkas Lemma for SDP

Let $H_1, \cdots, H_n$ be a finite number of Hermitian operators in a finitely dimensional Hilbert space $\mathcal{H}$. Then the following two statements are equivalent:

1. For any $\rho \in \mathcal{D}(\mathcal{H})$, $\bigvee_k (tr(\rho H_k) > 0)$;
2. There exist non-negative numbers $p_1 \geq 0, \cdots, p_n \geq 0$, such that $p_1 + \cdots + p_n > 0$ and $p_1 H_1 + p_2 H_2 + \cdots + p_n H_n \sqsupset 0$.

LRSM synthesis with angelic choice

Generalized Farkas Lemma for SDP

Let $H_1, \cdots, H_n$ be a finite number of Hermitian operators in a finitely dimensional Hilbert space $\mathcal{H}$. Then the following two statements are equivalent:

1. For any $\rho \in \mathcal{D}(\mathcal{H})$, $\bigvee_k (tr(\rho H_k) > 0)$;
2. There exist non-negative numbers $p_1 \geq 0, \cdots, p_n \geq 0$, such that $p_1 + \cdots + p_n > 0$ and $p_1 H_1 + p_2 H_2 + \cdots + p_n H_n \sqsupset 0$.

- Application: parameterized SDP form, e.g.,

$$\sum_l (p\mathcal{A}_l + q\mathcal{B}_l)(R_l) \sqsupseteq C \text{ for some } p, q \geq 0, p + q = 1.$$

- Parameterized SDP w.r.t. any error in EXPTIME.

Complexity

	Probabilistic	Quantum
The General Problem	PSPACE	2-EXPTIME by QE with CAD
Without Angelic Choice	PTIME	PTIME w.r.t. an error
With Angelic Choice	NP-hard	EXPTIME w.r.t. an error

Summary

Main contribution

- A nontrivial proof of decidability for finite-dimensional case.
- The LRSM-based approach for termination analysis.
- Some useful techniques: quantum generalizations of 0-1 Law and of Farkas lemma, and the application of Gleason theorem.

Future work

- Implementation in current quantum programming platforms.
- More efficient algorithms and better complexity upper bounds.
- Termination problems in expectation based QRHL.
- Relations to other problems in quantum theory: reachability analysis, quantum automata, measurement occurrence, etc.