symQV: Automated Symbolic Verification of Quantum Programs

Fabian Bauer-Marquart, Stefan Leue, **Christian Schilling**
University of Konstanz Aalborg University

FM 2023



Motivation

- Quantum computers on the rise but face the same problems as classical computers
- Verification of classical programs well studied
- Verification of quantum programs under-explored
  - Interactive proof assistants
  - Automated program equivalence checking
  - Programs with fixed input
- This work
  - Automatic verification against FOL specifications
  - Reduction to SMT solving
  - Efficient encoding and overapproximation

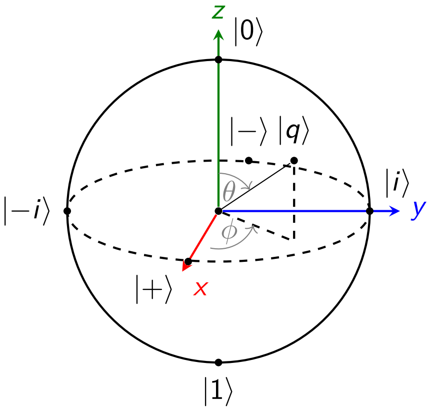
Overview

Overview

Qubit

- Ground state $|0\rangle$
- Excited state $|1\rangle$
- Superposition $|q\rangle = \alpha |0\rangle + \beta |1\rangle, \quad \alpha, \beta \in \mathbb{C}$
- Constraint $|\alpha|^2 + |\beta|^2 = 1$
- Written as 2D vector: $|q\rangle \equiv \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$

Bloch sphere



- Polar coordinates: $|q\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$

Multiple qubits

$$
\begin{aligned}
|q_0 q_1\rangle = |q_0\rangle \otimes |q_1\rangle \quad &\equiv \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} \otimes \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} \\
&= \alpha_0\alpha_1 |00\rangle + \alpha_0\beta_1 |01\rangle + \beta_0\alpha_1 |10\rangle + \beta_0\beta_1 |11\rangle \\
&\equiv \begin{bmatrix} \alpha_0\alpha_1 \\ \alpha_0\beta_1 \\ \beta_0\alpha_1 \\ \beta_0\beta_1 \end{bmatrix}
\end{aligned}
$$

Quantum gates

- Invertible matrix operations
- Example: swapping of two qubits

$$SWAP(|q_0\rangle \otimes |q_1\rangle) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} \otimes \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0\alpha_1 \\ \alpha_0\beta_1 \\ \beta_0\alpha_1 \\ \beta_0\beta_1 \end{bmatrix} = \begin{bmatrix} \alpha_0\alpha_1 \\ \beta_0\alpha_1 \\ \alpha_0\beta_1 \\ \beta_0\beta_1 \end{bmatrix} = |q_1\rangle \otimes |q_0\rangle$$

Measurement and challenges

- Measurement: converts to classical bit
- Challenges with quantum programs
  - Measurement destroys state (not invertible)
  - Simulation is probabilistic and requires many runs
  - Exponential state space
  - Entanglement: dependency between qubits (ignored in this presentation)

Overview

Qubit encoding

- Encode a qubit $|q\rangle$ as a 5-tuple $(\alpha, \beta_R, \beta_I, \phi, \theta) \in \mathbb{R}^5$
- Add constraints for values
  - $\alpha = \cos\frac{\theta}{2} \quad \wedge \quad \beta_R = \cos\phi \cdot \sin\frac{\theta}{2} \quad \wedge \quad \beta_I = \sin\phi \cdot \sin\frac{\theta}{2}$
  - $0 \leq \theta \leq \pi \quad \wedge \quad 0 \leq \phi < 2\pi$
  - $\theta = 0 \implies \phi = 0 \quad \wedge \quad \theta = \pi \implies \phi = 0$

Gates, measurements etc.

- Common gates can be encoded efficiently in a symbolic way
  Example: $SWAP(|q_0\rangle, |q_1\rangle) \rightsquigarrow (|q_1\rangle, |q_0\rangle)$
- In general we need the (exponential) matrix representation (see the paper)
- Measurement is just a projection

Soundness and completeness

Theorem

The quantum program model (= our encoding) preserves the semantics of the quantum circuit model (= standard model)

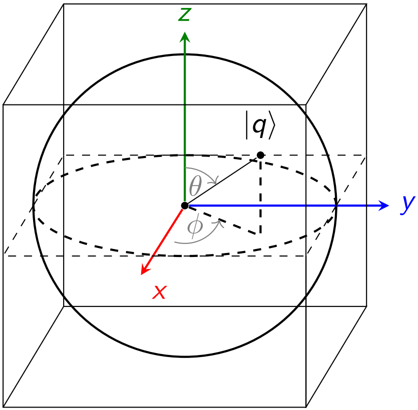
Corollary

Given a quantum program model with encoding $M_{\mathcal{Q}}$ and a specification φ, the program satisfies φ if and only if $M_{\mathcal{Q}} \wedge \neg\varphi$ is unsatisfiable

- $M_{\mathcal{Q}}$ is a formula containing nonlinear real arithmetic with trigonometric expressions
- Undecidable but δ-decidable[1]

[1] S. Gao, J. Avigad, and E. M. Clarke. *IJCAR*. 2012.

Overapproximation



$$-1 \leq \alpha \leq 1 \quad \wedge \quad -1 \leq \beta_R \leq 1 \quad \wedge \quad -1 \leq \beta_I \leq 1$$

Overview

Benchmark problems

Program	**Description**	**Depth**	**Input**
Toffoli	Toffoli gate	5	bit vector
TP	Quantum teleportation	6	infinite
ADD-8	8-qubit quantum adder	48	bit vector
QFT-n	n-qubit quantum Fourier transform	$\mathcal{O}(n^2)$	bit vector
QPE-n	n-qubit quantum phase estimation	$\mathcal{O}(n^2)$	singleton[1]
GDO-n	n-qubit Grover's diffusion operator	$\mathcal{O}(n)$	infinite

[1] Parameterized gates

Algorithms

- **Simulation**
- **Matrix**: SMT encoding with (exponential) matrix/vector representation
- **Exact**: SMT encoding with gate mapping but without overapproximation
- **symQV**: SMT encoding with gate mapping and overapproximation

Benchmark results

Benchmark	**Simulation**	**Matrix**	**Exact**	**symQV**
Toffoli	**0.02 sec**	11.1 sec	1.3 sec	0.4 sec
TP	N/A	44.8 sec	**21.6 sec**	31.0 sec
ADD-8	6.1 h	`OOM`	**7.6 sec**	7.8 sec
QFT-3	**0.005 sec**	12.8 sec	5.8 sec	1.0 sec
QFT-5	**0.03 sec**	17.6 min	2.6 min	26.4 sec
QFT-10	**1.5 sec**	1.2 h	10.9 h	1.6 h
QFT-12	**14.0 sec**	4.0 h	`timeout`	7.4 h

Benchmark results

Benchmark	Simulation	Matrix	Exact	symQV
QPE-3	N/A	19.2 sec	34.0 sec	**8.7 sec**
QPE-5	N/A	18.2 min	42.3 min	**3.9 min**
GDO-5	N/A	`timeout`	9.2 sec	**1.3 sec**
GDO-10	N/A	`timeout`	3.2 min	**17.0 sec**
GDO-12	N/A	`timeout`	14.2 min	**20.2 sec**
GDO-15	N/A	`timeout`	2.9 h	**1.0 min**
GDO-18	N/A	`timeout`	`timeout`	**4.9 min**
GDO-20	N/A	`timeout`	`timeout`	**17.1 min**
GDO-22	N/A	`timeout`	`timeout`	**1.1 h**
GDO-24	N/A	`timeout`	`timeout`	**4.2 h**

Varying the δ parameter

δ	GDO-12	GDO-15	GDO-18
10^{-4}	20.2 sec	1.0 min	4.9 min
10^{-6}	20.5 sec	28.0 min	33.1 min
10^{-8}	20.8 sec	49.4 min	58.7 min
10^{-10}	21.1 sec	52.3 min	1.2 h

Overview

Conclusion and future work

- SMT encoding of quantum programs
- Fully automatic verification via δ-satisfiability
- Symbolic encoding can sometimes avoid exponential blow-up
- Simple overapproximation sometimes useful in practice
- Future directions:
  - Other approximation techniques
  - Falsification and CEGAR