Quantum Computing Research Directions

Kenneth W. Regan[1]
University at Buffalo (SUNY)

19 March, 2019

[1] Joint work with Amlan Chakrabarti, U. Calcutta, and Chaowen Guan, UB.

Algebraic and Logical QC Simulations

Algebraic and Logical QC Simulations

- Idea is to postpone exponential blowup until the end. . .

Algebraic and Logical QC Simulations

- Idea is to postpone exponential blowup until the end…
- …when a full spec can be fed to equation solvers or SAT solvers.

Algebraic and Logical QC Simulations

- Idea is to postpone exponential blowup until the end. . .
- . . . when a full spec can be fed to equation solvers or SAT solvers.
- Algebraic side is joint work with Amlan Chakrabarti (U. Calcutta) since 2007. [show GLL page]

Algebraic and Logical QC Simulations

- Idea is to postpone exponential blowup until the end...
- ...when a full spec can be fed to equation solvers or SAT solvers.
- Algebraic side is joint work with Amlan Chakrabarti (U. Calcutta) since 2007. [show GLL page]
- Logical side with Chaowen Guan, UB.

Algebraic and Logical QC Simulations

- Idea is to postpone exponential blowup until the end...
- ...when a full spec can be fed to equation solvers or SAT solvers.
- Algebraic side is joint work with Amlan Chakrabarti (U. Calcutta) since 2007. [show GLL page]
- Logical side with Chaowen Guan, UB. Jointly became paper [RCG18] in *Transactions on Computational Science*, 2018.

Algebraic and Logical QC Simulations

- Idea is to postpone exponential blowup until the end...
- ...when a full spec can be fed to equation solvers or SAT solvers.
- Algebraic side is joint work with Amlan Chakrabarti (U. Calcutta) since 2007. [show GLL page]
- Logical side with Chaowen Guan, UB. Jointly became paper [RCG18] in *Transactions on Computational Science*, 2018.
- Logic-based full QC simulator, 8,000+ lines of C++.

Algebraic and Logical QC Simulations

- Idea is to postpone exponential blowup until the end...
- ...when a full spec can be fed to equation solvers or SAT solvers.
- Algebraic side is joint work with Amlan Chakrabarti (U. Calcutta) since 2007. [show GLL page]
- Logical side with Chaowen Guan, UB. Jointly became paper [RCG18] in *Transactions on Computational Science*, 2018.
- Logic-based full QC simulator, 8,000+ lines of C++.
- Partly included in textbook with Richard Lipton, *Quantum Algorithms Via Linear Algebra* (MIT Press, 2014; 2nd. ed. to come this year).

Notable Theoretical Advance (just this past month)

Notable Theoretical Advance (just this past month)

- *Stabilizer circuits* ($\equiv$ *Clifford circuits*) are the most salient classically solvable case.

Notable Theoretical Advance (just this past month)

- *Stabilizer circuits* ($\equiv$ *Clifford circuits*) are the most salient classically solvable case.
- Vital in quantum error-correcting codes for fault-tolerant QC.

Notable Theoretical Advance (just this past month)

- *Stabilizer circuits* ($\equiv$ *Clifford circuits*) are the most salient classically solvable case.
- Vital in quantum error-correcting codes for fault-tolerant QC.
- Classical simulation of n qubits takes $O(n^2)$ time per single-qubit measurement [Aaronson-Gottesman, 2004], $O(n^3)$ time to measure all n qubits.

Notable Theoretical Advance (just this past month)

- *Stabilizer circuits* ($\equiv$ *Clifford circuits*) are the most salient classically solvable case.
- Vital in quantum error-correcting codes for fault-tolerant QC.
- Classical simulation of n qubits takes $O(n^2)$ time per single-qubit measurement [Aaronson-Gottesman, 2004], $O(n^3)$ time to measure all n qubits.
- We improve to time $O(n^\omega)$ where $\omega < 2.3729$ is the known exponent for $n \times n$ matrix multiplication.

Notable Theoretical Advance (just this past month)

- *Stabilizer circuits* ($\equiv$ *Clifford circuits*) are the most salient classically solvable case.
- Vital in quantum error-correcting codes for fault-tolerant QC.
- Classical simulation of n qubits takes $O(n^2)$ time per single-qubit measurement [Aaronson-Gottesman, 2004], $O(n^3)$ time to measure all n qubits.
- We improve to time $O(n^\omega)$ where $\omega < 2.3729$ is the known exponent for $n \times n$ matrix multiplication.
- Also give $O(N)$-time reduction $(N = n^2)$ from computing $n \times n$ matrix rank over $\mathbb{F}_2$ to the QC simulation.

Notable Theoretical Advance (just this past month)

- *Stabilizer circuits* ($\equiv$ *Clifford circuits*) are the most salient classically solvable case.
- Vital in quantum error-correcting codes for fault-tolerant QC.
- Classical simulation of n qubits takes $O(n^2)$ time per single-qubit measurement [Aaronson-Gottesman, 2004], $O(n^3)$ time to measure all n qubits.
- We improve to time $O(n^\omega)$ where $\omega < 2.3729$ is the known exponent for $n \times n$ matrix multiplication.
- Also give $O(N)$-time reduction $(N = n^2)$ from computing $n \times n$ matrix rank over $\mathbb{F}_2$ to the QC simulation.
- Means that the n^2-vs.-n^ω weak/strong simulation gap canot be closed unless matrix rank is in $O(n^2)$ time over $\mathbb{F}_2$.

How Achieved

How Achieved

- Stabilizer circuits C yield *classical* quadratic forms q_C over $\mathbb{Z}_4$.

How Achieved

- Stabilizer circuits C yield *classical* quadratic forms q_C over $\mathbb{Z}_4$.
- Exploit normal form q' for q_C by Schmidt [2009].

How Achieved

- Stabilizer circuits C yield *classical* quadratic forms q_C over $\mathbb{Z}_4$.
- Exploit normal form q' for q_C by Schmidt [2009].
- Apply new algorithm for LDU decompositions over $\mathbb{F}_2$ by Dumas-Pernet [2018].

How Achieved

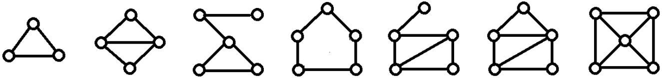
- Stabilizer circuits C yield *classical* quadratic forms q_C over $\mathbb{Z}_4$.
- Exploit normal form q' for q_C by Schmidt [2009].
- Apply new algorithm for LDU decompositions over $\mathbb{F}_2$ by Dumas-Pernet [2018].
- Invert the LDU process but calculating in $\mathbb{Z}_4$.

How Achieved

- Stabilizer circuits C yield *classical* quadratic forms q_C over $\mathbb{Z}_4$.
- Exploit normal form q' for q_C by Schmidt [2009].
- Apply new algorithm for LDU decompositions over $\mathbb{F}_2$ by Dumas-Pernet [2018].
- Invert the LDU process but calculating in $\mathbb{Z}_4$.
- Painstaking analysis of how distributions of values were mapped yields a simple recursion then gives the result from the final "spectrum."

How Achieved

- Stabilizer circuits C yield *classical* quadratic forms q_C over $\mathbb{Z}_4$.
- Exploit normal form q' for q_C by Schmidt [2009].
- Apply new algorithm for LDU decompositions over $\mathbb{F}_2$ by Dumas-Pernet [2018].
- Invert the LDU process but calculating in $\mathbb{Z}_4$.
- Painstaking analysis of how distributions of values were mapped yields a simple recursion then gives the result from the final "spectrum."
- Also yields an apparently new class of undirected graphs:



Boolean Logic Simulation

Boolean Logic Simulation

- Allocate free variables x_i for every input (qu)bit, z_i for corresponding outputs, and y_j for every nondeterministic gate (wlog. Hadamard gate).

Boolean Logic Simulation

- Allocate free variables x_i for every input (qu)bit, z_i for corresponding outputs, and y_j for every nondeterministic gate (wlog. Hadamard gate).
- Also maintain "forced" variables giving the current *phase* and *location* of every Feynman path.

Boolean Logic Simulation

- Allocate free variables x_i for every input (qu)bit, z_i for corresponding outputs, and y_j for every nondeterministic gate (wlog. Hadamard gate).
- Also maintain "forced" variables giving the current *phase* and *location* of every Feynman path.
- Translation from circuit C to Boolean formula ϕ_C is again real-time.

Boolean Logic Simulation

- Allocate free variables x_i for every input (qu)bit, z_i for corresponding outputs, and y_j for every nondeterministic gate (wlog. Hadamard gate).
- Also maintain "forced" variables giving the current *phase* and *location* of every Feynman path.
- Translation from circuit C to Boolean formula ϕ_C is again real-time.
- Solution counts over each phase for a target location yield its amplitude.

Boolean Logic Simulation

- Allocate free variables x_i for every input (qu)bit, z_i for corresponding outputs, and y_j for every nondeterministic gate (wlog. Hadamard gate).
- Also maintain "forced" variables giving the current *phase* and *location* of every Feynman path.
- Translation from circuit C to Boolean formula ϕ_C is again real-time.
- Solution counts over each phase for a target location yield its amplitude.
- *#SAT solvers* such as *sharpSAT* and *Cachet* give hope of heuristic simulations of harder classes of circults.

Boolean Logic Simulation

- Allocate free variables x_i for every input (qu)bit, z_i for corresponding outputs, and y_j for every nondeterministic gate (wlog. Hadamard gate).
- Also maintain "forced" variables giving the current *phase* and *location* of every Feynman path.
- Translation from circuit C to Boolean formula ϕ_C is again real-time.
- Solution counts over each phase for a target location yield its amplitude.
- *#SAT solvers* such as *sharpSAT* and *Cachet* give hope of heuristic simulations of harder classes of circults.
- Our C++ simulator outputs DIMACS-compliant files for these solvers. SAT solvers have seen great success in many areas

Boolean Logic Simulation

- Allocate free variables x_i for every input (qu)bit, z_i for corresponding outputs, and y_j for every nondeterministic gate (wlog. Hadamard gate).
- Also maintain "forced" variables giving the current *phase* and *location* of every Feynman path.
- Translation from circuit C to Boolean formula ϕ_C is again real-time.
- Solution counts over each phase for a target location yield its amplitude.
- *#SAT solvers* such as *sharpSAT* and *Cachet* give hope of heuristic simulations of harder classes of circults.
- Our C++ simulator outputs DIMACS-compliant files for these solvers. SAT solvers have seen great success in many areas, but maybe not QC...

Boolean Logic Simulation

- Allocate free variables x_i for every input (qu)bit, z_i for corresponding outputs, and y_j for every nondeterministic gate (wlog. Hadamard gate).
- Also maintain "forced" variables giving the current *phase* and *location* of every Feynman path.
- Translation from circuit C to Boolean formula ϕ_C is again real-time.
- Solution counts over each phase for a target location yield its amplitude.
- *#SAT solvers* such as *sharpSAT* and *Cachet* give hope of heuristic simulations of harder classes of circults.
- Our C++ simulator outputs DIMACS-compliant files for these solvers. SAT solvers have seen great success in many areas, but maybe not QC...
- Second main purpose of simulator [show] is to enable tinkering with approximative methods.

Higher Algebra and Applications

Higher Algebra and Applications

- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and effort to keep coherence.

Higher Algebra and Applications

- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and effort to keep coherence.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates.

Higher Algebra and Applications

- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and effort to keep coherence.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates. Apply similar to quantum circuits?

Higher Algebra and Applications

- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and effort to keep coherence.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates. Apply similar to quantum circuits?
- Already hard to formulate n-partite entanglement of (pure or mixed) *states*.

Higher Algebra and Applications

- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and effort to keep coherence.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates. Apply similar to quantum circuits?
- Already hard to formulate n-partite entanglement of (pure or mixed) *states*. How to define for *circuits*?

Higher Algebra and Applications

- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and effort to keep coherence.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates. Apply similar to quantum circuits?
- Already hard to formulate n-partite entanglement of (pure or mixed) *states*. How to define for *circuits*? Plausible axioms:

$$\begin{aligned} e(C^*) &= e(C), \\ e(C_1 \otimes C_2) &= e(C_1) + e(C_2), \\ e(C; \mathit{measure}) &\leq e(C), \\ e(C + \text{LOCC}) &= e(C) \end{aligned}$$

Higher Algebra and Applications

- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and effort to keep coherence.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates. Apply similar to quantum circuits?
- Already hard to formulate n-partite entanglement of (pure or mixed) *states*. How to define for *circuits*? Plausible axioms:

$$\begin{aligned} e(C^*) &= e(C), \\ e(C_1 \otimes C_2) &= e(C_1) + e(C_2), \\ e(C; \textit{measure}) &\leq e(C), \\ e(C + \mathrm{LOCC}) &= e(C) \end{aligned}$$

- Also apply to study T-*gate count*, singular points. . .

Summary

Summary

- Novel research ideas.

Summary

- Novel research ideas.
- Development of program infrastructure to experiment with them.

Summary

- Novel research ideas.
- Development of program infrastructure to experiment with them.
- Indo-US collaboration.

Summary

- Novel research ideas.
- Development of program infrastructure to experiment with them.
- Indo-US collaboration.
- Avenues for dissemination: *Gödel's Lost Letter* blog, textbook with MIT Press going to second edition this summer.

Summary

- Novel research ideas.
- Development of program infrastructure to experiment with them.
- Indo-US collaboration.
- Avenues for dissemination: *Gödel's Lost Letter* blog, textbook with MIT Press going to second edition this summer.
- Involvement in the general debate over *Quantum Advantage*.