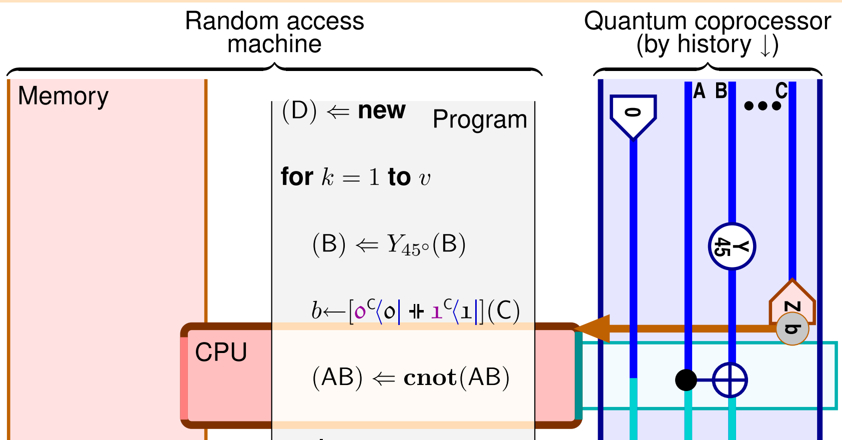
Models of Quantum Computation:
Monoidal Coprocessors

- Computation with coprocessors.
- Monoids for oracle computation.
- Linear models of computation and readouts.
- Coprocessor extensions.
- Lie groups and their power.

E. "Manny" Knill: knill@lanl.gov, http://www.lanl.gov/~knill

Computation with Coprocessors



- Quantum computation = deterministic computation + $\left\{\begin{array}{l}\text{quantum network coprocessor}\\ \text{or}\\ \ldots g_k(\underset{0}{P_0} \dotplus \underset{1}{P_1}) g_{k-1}\ldots\end{array}\right.$

Why Study Quantum Information?

- QI is a tool.
  - A language for quantum control.
  - Generalization of information theory.
  - Linearization of the theory of computation.
- QI's potential.
  - Efficient quantum physics simulation.
  - Communication security.
  - Efficient factoring and other number theory algorithms.
  - Insights into complexity theory?
- QI may be realizable.
  - Quantum noise is in principle controllable.
  - Numerous experimental proposals.
- . . . when?

Monoidal Gates

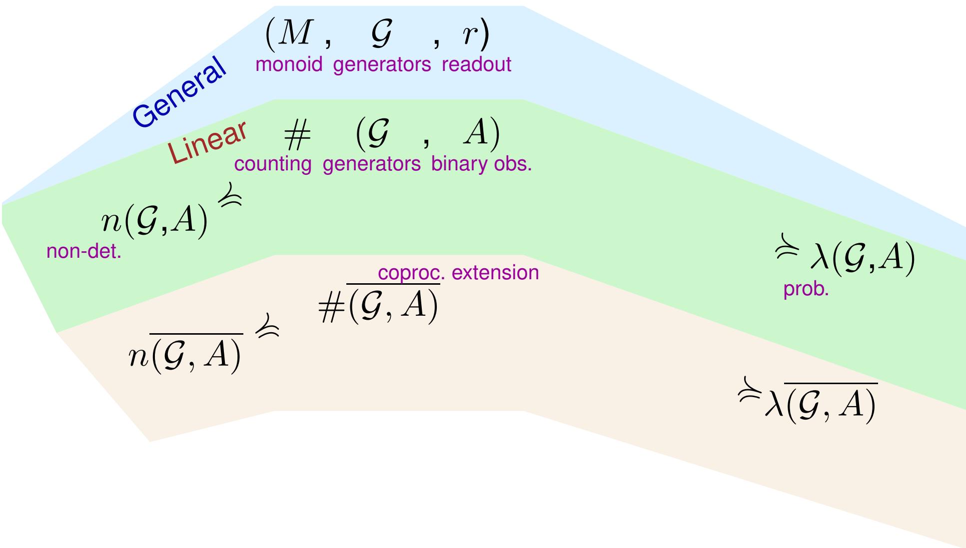
- Set with associative product and an identity: *Monoid*.

  Ex.: The free monoid with generators $a, b, \ldots$ is the set of all words over the alphabet $\{a, b, \ldots\}$. The product is concatenation. The empty word is the identity.
- A *one-line* model of computation is defined by:
  1. Monoid M.
  2. Generators $g_0, g_1, \ldots \in M$ (countable).
  3. A readout function $r : M \to R$.

  E.g. $R = \{p{:}o \dotplus (1-p){:}1\}_{p\in[0,1]}$ for probabilistic readout.

  Ex. Deterministic circuits.
  1. $M = \{f : \{o, 1\}^n \to \{o, 1\}^n\}$ with composition.
  2. $g_{k,l,m} : \mathbf{b} \mapsto \mathbf{c}$, with $c_k = \underbrace{\neg(b_l \wedge b_m)}$ and $c_j = b_j$ for $j \neq k$.
  3. $r(f) = f(\mathbf{0})_1$. $\quad \mathbf{nand}(x, y) = \neg(x \wedge y)$: 

Overview of Concepts



Linear Representations

- A representation of a monoid M is a homomorphism $\pi : M \to \mathrm{Mat}_n(\mathbb{C})$. I.e. $\pi(gh) = \pi(g)\pi(h)$.
- Readout functions induced by a representation and $A \in \mathrm{Mat}_n(\mathbb{C})$ with $0 < A \leq \mathbb{I}$. Let $A_\perp = \mathbb{I} - A$.
  Def.: $p(g) = \mathrm{tr}\big(A\,\pi(g)A\pi(g)^\dagger\big)$, $q(g) = \mathrm{tr}\big(A_\perp\,\pi(g)A\pi(g)^\dagger\big)$.
  - Probabilistic. ∼"bounded error"
    $$r(g) = \tfrac{q(g)}{p(g)+q(g)}{:}\mathtt{0} + \tfrac{p(g)}{p(g)+q(g)}{:}\mathtt{1}$$
  - Non-deterministic.
    $$r(g) = \begin{cases} \mathtt{0} & \text{if } p(g) = 0 \\ \mathtt{1} & \text{otherwise} \end{cases}$$
  - Counting. $r(g) = \left(\tfrac{p(g)}{p(g)+q(g)}, \tfrac{q(g)}{p(g)+q(g)}\right)$ ∼ #P

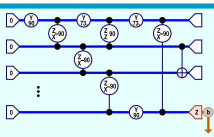
Extending Deterministic Computation

- Let $\mathcal{M} = (M, \mathcal{G}, r)$ be a one-line model of computation.
  RAM$^{\mathcal{M}}$ is a random access machine with an $\mathcal{M}$-oracle:
  $$b \leftarrow r(g_{k_N} \ldots g_{k_1})$$
- Defining common models using RAM$^{\mathcal{M}}$.
  - Probabilistic computation (PC):
    $M = \{1\} \subseteq \mathbb{C}$, $\mathcal{G} = (1)$, r is prob. readout with $A = 1/2$.
  - Quantum computation (QC):
    $M = \oplus_n U(2^n)$, $\mathcal{G} =$ "quantum gates",
    r is prob. readout with $A = |\mathbf{0}\rangle\langle\mathbf{0}|$.
  - QC with one bit ($\mathrm{Q_1C}$): Like QC but using $A = |\mathtt{0}\rangle_1\langle\mathtt{0}|$.
  - Non-deterministic PC|QC|$\mathrm{Q_1C}$:
    Like PC|QC|$\mathrm{Q_1C}$, but using non-deterministic readout.

Extending Deterministic Computation

- Let $\mathcal{M} = (M, \mathcal{G}, r)$ be a one-line model of computation.
  RAM$^{\mathcal{M}}$ is a random access machine with an $\mathcal{M}$-oracle:
  $$b \leftarrow r(g_{k_N} \ldots g_{k_1})$$
- Defining common models using RAM$^{\mathcal{M}}$.
  - Probabilistic computation (PC):
    $M = \{1\} \subseteq \mathbb{C}$, $\mathcal{G} = (1)$, r is prob. readout with $A = 1/2$.
  - Quantum computation (QC):
    $M = \oplus_n U(2^n)$, $\mathcal{G} =$ "quantum gates",
    r is prob. readout with $A = |\mathbf{0}\rangle\langle\mathbf{0}|$.

Goto: Quantum network primer.



The Pauli Group

$$\sigma_{00} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \sigma_{01} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \sigma_{11} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ \sigma_{10} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Def. (by ex.): A Pauli product: $\sigma_{001011} = \sigma_{00} \otimes \sigma_{10} \otimes \sigma_{11}$

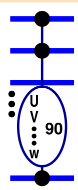
- The set $\mathcal{P}_n$ of Pauli products on $\mathbb{C}^{2^n} \cong (\mathbb{C}^2)^{\otimes n}$ is a projective unitary representation of $(\mathbb{Z}_2)^{2n}$:

$$\sigma_{\mathbf{b}}\sigma_{\mathbf{c}} \propto \sigma_{\mathbf{b+c}}$$

- $\mathcal{N}_n \doteq$ normalizer of $\overline{\mathcal{P}} = (\{1, -1, i, -i\}\mathcal{P}_n)$ in $SU(\mathbb{C}^{2^n})$.
  Define a $2n \times 2n$ matrix over $\mathbb{Z}_2$ by $S = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \otimes \mathbb{I}_n$.
- **Theorem 1:** $\mathcal{N}_n$ is an extension of $\overline{\mathcal{P}}$ by the $2n \times 2n$ invertible matrices X over $\mathbb{Z}_2$ that satisfy $X^T S X = S$.

See: Calderbank&*al.* 1997 [1]

An $\mathcal{N}$-based Model of Computation



- **Theorem 2:**
  $\mathcal{N}$ is generated by *rotations* of the form $e^{-i\sigma_{\mathbf{b}}\pi/4}$ with *weight* $w(\mathbf{b}) \leq 2$.
- Counting $\mathcal{N}$-computation ($\#\mathcal{N}$C).
  Monoid: $\mathcal{N} \subseteq SU(\mathbb{C}^{2^n})$.
  Generators: $e^{-i\sigma_{\mathbf{b}}\pi/4}$ with $w(\mathbf{b}) \leq 2$.
  Readout: Counting, induced by $A = |\mathfrak{o}\rangle\langle\mathfrak{o}| \otimes \mathbb{I}_2 \otimes \mathbb{I}_2 \otimes \ldots$.

  Rem.: $r(g_N \ldots g_1) \propto (\mathrm{tr}(Ag_N \ldots g_1 A g_1^\dagger \ldots g_N^\dagger), \ldots)$
- **Theorem 3:** $\#\mathcal{N}$C is efficiently determ. simulatable.

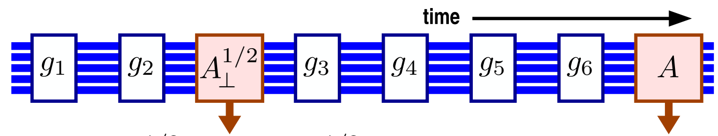
  Note: Efficiently in N.

  Proof idea: Compute $g_N \ldots g_1$ in the representation as $2n \times 2n$ matrices over $\mathbb{Z}_2$. Calculate traces using $A = (\mathbb{I} + \sigma_{10\,00\,\ldots})/2$.

Gottesman 1998 [2]

Coprocessor Extensions

- Let $\mathcal{L} = (\mathcal{G}, A)$ be a linear counting one-line model of computation.
  - $\mathcal{G}$ consists of bounded linear operators.
  - $\mathrm{tr}(A\,\cdot\,)$ well defined on $\langle\mathcal{G}\rangle$.
  - Readout is counting, induced by A.
- The coprocessor extension of $\mathcal{L}$ is $(\mathcal{G} \cup \{A^{1/2}, A_\perp^{1/2}\}, \mathbb{I})$.



$\mathrm{tr}(A_\perp^{1/2} g_2 g_1 \mathbb{I} g_1^\dagger g_2^\dagger A_\perp^{1/2})$ $\quad$ $\mathrm{tr}(A^{1/2} g_6 \ldots \mathbb{I} \ldots g_6^\dagger A^{1/2})$

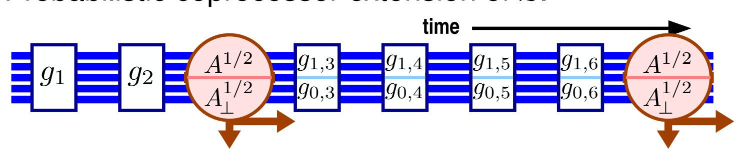
- **Theorem 4:** The coprocessor extension of $\#\mathcal{N}$C is efficiently deterministically simulatable.

Gottesman 1998 [2]

Rem.: $\#\mathcal{N}$C is counting $\mathcal{N}$-computation.

Probabilistic Coprocessor Extensions

- Let $\mathcal{L} = (\mathcal{G}, A)$ be a linear probabilistic one-line model of computation.
  - $\mathcal{G}$ consists of bounded linear operators.
  - $\mathrm{tr}(A\,\cdot\,)$ well defined on $\langle\mathcal{G}\rangle$.
  - Readout is probabilistic, induced by A.
- Probabilistic coprocessor extension of $\mathcal{L}$.



As a linear one-line model:

Monoid:	$M' = \oplus_{\mathbf{b} \in \{\mathfrak{o}, \mathfrak{1}\}^*} \mathbf{b} \cdot M$.
Generators:	$g_\tau(\mathbf{b} \cdot x) = \mathbf{b} \cdot g_{\tau(\mathbf{b})}(x)$,
	$g_A(\mathbf{b} \cdot x) = \mathfrak{1}\mathbf{b} \cdot A^{1/2} x + \mathfrak{o}\mathbf{b} \cdot A_\perp^{1/2} x$.
Readout:	Probabilistic, induced by $\mathbb{I}$.

Models of Standard Quantum Computation

- Let $\mathcal{G}_n$ consist of the 45° rotations $e^{-i\sigma_b\pi/8}$ on n qubits and $A_n = |\mathfrak{o}\rangle\langle\mathfrak{o}| \otimes \mathbb{I}_{2^{n-1}}/2^{n-1}$.
  $\mathsf{Q}_1\mathsf{C}_n$ is the prob. linear one-line model induced by $(\mathcal{G}_n, A_n)$.
  $\mathsf{Q}_1\mathsf{C}_n$: "one qubit quantum computation with $n-1$ random bits"
  Knill&Laflamme 1998 [3]
- $\mathsf{Q}_1\mathsf{C} = \oplus_{n=1}^{\infty}\mathsf{Q}_1\mathsf{C}_n$,
  induced by $\left(\bigcup_{n=1}^{\infty} (\ldots \mathbf{0} \oplus \mathcal{G}_n \oplus \mathbf{0} \oplus \ldots), \oplus_{n=1}^{\infty} A_n \right)$.
- Let $B_n = |\mathfrak{o}\rangle\langle\mathfrak{o}| \otimes |\mathfrak{o}\rangle\langle\mathfrak{o}| \otimes \ldots$.
  QC_n is the model induced by $(\mathcal{G}_n, B_n)$. $\mathsf{QC} = \oplus_{n=1}^{\infty}\mathsf{QC}_n$.
- **Theorem 5:** QC $\cong$ coprocessor extension of $\mathsf{Q}_1\mathsf{C}$.
  Proof idea ($\Leftarrow$): Use the prob. projectors of the coprocessor extension and an outcome-dependent flip to "prepare" B_n.

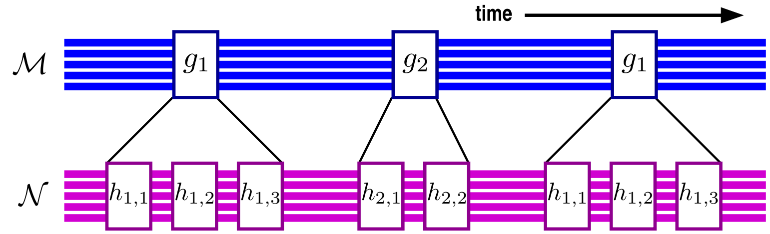
Forward to: Faithful Sim. | Counting QC

Turing Simulation

- Let $\mathcal{M} = (M, \mathcal{G}, r)$, $\mathcal{N} = (N, \mathcal{H}, s)$ be one-line models.
  $\mathcal{M} \preccurlyeq_{\text{PRAM}} \mathcal{N}$: $\mathcal{M}$ is (efficiently) *simulatable* by $\mathcal{N}$ if $r(g_k \ldots g_1)$ can be computed in polynomial time by $\text{PRAM}^{\mathcal{N}}$.
  I.e.: Assume prob. RAM as the basic machine.
  Require: Binary representations for generators.
  If $\mathcal{M}, \mathcal{N}$ are linear, finite dimensional: Use non-uniform circuit model to compare complexity non-asymptotically.
- $\mathcal{M}$ linear: $\#\mathcal{M}$: $\mathcal{M}$ with counting readout,
  $n\mathcal{M}$: $\mathcal{M}$ with non-deterministic readout,
  $\lambda\mathcal{M}$: $\mathcal{M}$ with probabilistic readout (default).
- **Observation 6:** $\#\mathcal{M} \preccurlyeq \lambda\mathcal{M}$
  $n\mathcal{M} \preccurlyeq \#\mathcal{M}$

Faithful Simulation

- Let $\mathcal{M} = (\mathcal{G}, A, r)$, $\mathcal{N} = (\mathcal{H}, B, s)$ be linear models with probabilistic readouts.
  $\mathcal{M} \preccurlyeq_f \mathcal{N}$: $\mathcal{M}$ is *faithfully simulatable* by $\mathcal{N}$.



$\mathcal{M} \preccurlyeq_f \mathcal{N}$ if there exists $\pi : \mathcal{G} \to \langle\mathcal{H}\rangle$ with $\pi(g) = h_{g,K} \ldots h_{g,1}$, the complexity of $h_{g,K} \ldots$ is polynomial in that of g_k, $r(g_L \ldots g_1) = s(\pi(g_L) \ldots \pi(g_1))$.

Strengthening: *Homomorphically* simulatable.

Counting QC

- #QC is like QC, but with counting readout.
  #P is (an oracle for) "permanent of integer matrices".
- **Theorem 7:** #QC is polynomially equivalent to $\text{RAM}^{\#\text{P}}$.
  Proof (outline). Prob(rand. sol. satisfies ϕ) $\propto$ (# of sat. sols.).
- **Theorem 8:** RAM^{QC} can factor in PolyTime. Shor 1994 [5]
  Belief: $\mathbb{O} \prec$ QC. Def.: $\mathbb{O}$ is the trivial linear model $(\{\mathbb{I}\}, \mathbb{I})$
- $\mathcal{M}$ linear: $\#\overline{\mathcal{M}}$: Counting coproc. extension of $\mathcal{M}$
  $\lambda\overline{\mathcal{M}}$: Probabilistic coproc. extension of $\mathcal{M}$.
- **Observation**: For the linear models $\mathcal{M}$ studied so far:
  QC $\preccurlyeq_f \lambda\overline{\mathcal{M}}$ or $\#\overline{\mathcal{M}} \preccurlyeq \mathbb{O}$.

Note: QC $\preccurlyeq_f \lambda\overline{\mathcal{M}}$ and $\#\overline{\mathcal{M}} \preccurlyeq \mathbb{O}$ implies QC $\preccurlyeq \mathbb{O}$ ... contrary to belief.

Forward to: Conclusion

Fermionic Linear Optics

- Consider $\mathfrak{so}_{2n}\mathbb{R}$ with generators $s_{ij} = e_{ij} - e_{ji}$.

  s_{ij}: "standard" basis for $2n \times 2n$ antisymmetric real matrices.

  Representation in $\mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2$ using generators:

$$\mathfrak{g} = \left\{ i \ldots \otimes \mathbb{I} \otimes \left\{ \begin{matrix} \sigma_x \\ \sigma_y \end{matrix} \right\} \otimes \sigma_z \otimes \ldots \otimes \sigma_z \otimes \left\{ \begin{matrix} \sigma_x \\ \sigma_y \end{matrix} \right\} \otimes \mathbb{I} \otimes \ldots \right\}$$

- Define a linear model $\mathcal{M}$ with $\mathcal{G} = \{e^{tg} \mid g \in \mathfrak{g}, t \in \mathbb{R}\}$, $A = (\sigma_z + \mathbb{I}) \otimes \mathbb{I} \otimes \ldots$.
- **Theorem 9:** $\#\overline{\mathcal{M}} \preccurlyeq \mathbb{O}$.

  Valiant 2001 [6], Terhal&DiVincenzo 2001 [7]
  Knill 2001 [8]

  Require: Proper treatment of computation over $\mathbb{R}$, or restriction to rational e^{tg}.
- Implication: Implementing QC with fermionic linear optics devices, particle detectors and feedback is probably not possible.

Complexification

- $\mathfrak{g} = \left\{ i \ldots \otimes \mathbb{I} \otimes \left\{ \begin{matrix} \sigma_x \\ \sigma_y \end{matrix} \right\} \otimes \sigma_z \otimes \ldots \otimes \sigma_z \otimes \left\{ \begin{matrix} \sigma_x \\ \sigma_y \end{matrix} \right\} \otimes \mathbb{I} \otimes \ldots \right\}$

  $\mathcal{M}$ induced by $(\mathcal{G}, A)$ with $\mathcal{G} = \{e^{tg} \mid g \in \mathfrak{g}, t \in \mathbb{R}\}$, $A = (\sigma_z + \mathbb{I}) \otimes \mathbb{I} \otimes \ldots$.
- Let $\mathbb{C}\mathcal{M}$ be the linear model of computation induced by $\mathcal{H} = \{e^{\alpha g} \mid g \in \mathfrak{g}, \alpha \in \mathbb{C}\}$ and $\mathbb{I}$.
- **Theorem 10:** $\mathbb{C}\mathcal{M}$ is faithfully approx. equi-reducible to $\#\overline{\mathcal{M}}$.

  Proof idea:

  $A = \lim_{t\to\infty} e^{-t} e^{-it\sigma_z \otimes \mathbb{I} \otimes \ldots}$, $A_\perp = \lim_{t\to\infty} e^{-t} e^{it\sigma_z \otimes \mathbb{I} \otimes \ldots}$.

  The *approximation* concept is required for the limit.
- Second example: Bosonic linear optics model, $\mathcal{B}$. Hamiltonians of degree ≤ 2 in creation/annihilation ops. A is "position".
- **Theorem 11:** $\#\overline{\mathcal{B}} \preccurlyeq \mathbb{O}$ and $\mathbb{C}\mathcal{B} \equiv_{\tilde{f}} \#\overline{\mathcal{B}}$.

  Bartlett&*al.* 2001 [9]

Forward to: Conclusion

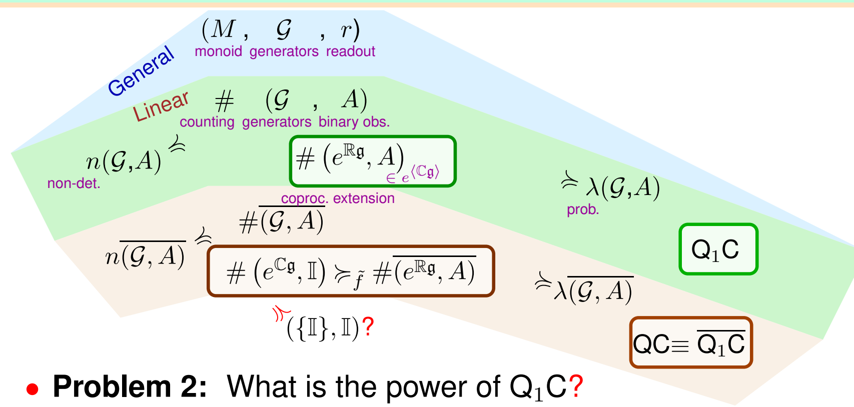
Lie Group Models

- Let $h_1 \ldots h_n$ be a basis of skew-hermitian operators forming a real Lie algebra $\mathfrak{h}$. Let $A, A_\perp$ be in the closure of $e^{\mathbb{C}\mathfrak{h}}$.

  $\#\mathcal{H}, n\mathcal{H}, \lambda\mathcal{H}$: Linear models induced by $\left(\{e^{th_i}\}_{t\in\mathbb{R}, 1\leq i\leq n}, A\right)$.

  $\#\mathbb{C}\mathcal{H}, \ldots$: $\left(\{e^{\alpha h_i}\}_{\alpha\in\mathbb{C}, 1\leq i\leq n}, \mathbb{I}\right)$
- **Theorem 12:** $\#\overline{\mathcal{H}} \preccurlyeq_{\tilde{f}} \#\mathbb{C}\mathcal{H}$.
- **Problem 1:** Is the complexity of $\#\mathbb{C}\mathcal{H}$ polynomial in n independent of the dimension of the operators?
  - This is the case for "fermionic linear optics", with operator dimension $O(2^n)$), "bosonic linear optics", with operator dimension ∞.
  - Use "number" instead of "position" for bosonic l.o.: Then the coprocessor extension is equivalent to QC.

Conclusion



- **Problem 2:** What is the power of Q$_1$C?
- **Problem area 3:** Investigate the power of $\#\mathbb{C}\mathcal{H}$ for different classes of Lie algebras and their representations.
- **Problem area 4:** Investigate the emergence of "information units" and subsystems through generator relationships.

Contents

References

[1] A.R. Calderbank, E.M. Rains, P.W. Shor, and N.J.A. Sloane. Quantum error correction and orthogonal geometry. *Phys. Rev. A*, 78:405–408, 1997.

[2] D. Gottesman. The heisenberg representation of quantum computers. quant-ph/9807006, 1998.

[3] E. Knill and R. Laflamme. On the power of one bit of quantum information. *Phys. Rev. Lett.*, 81:5672–5675, 1998.

[4] M. H. Freedman. P/np and the quantum field computer. *Proc. Nat. Ac. of Sci. USA*, 95:98–101, 1998.

[5] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26:1484–1509, 1997.

[6] L. G. Valiant. Quantum computers that can be simulated classically in polynomial time. In ?, editor, *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computation (STOC'01)*, page ?, El Paso, Texas, 2001. ACM Press.

[7] B. M. Terhal and D. P. DiVincenzo. Classical simulation of quantum circuits using non-interacting fermions. quant-ph/0108010, 2001.

[8] R. Laflamme, E. Knill, C. Negrevergne, R. Martinez, and D. Cory. Introduction to NMR quantum information processing. Technical Report LAUR-02-6132, Los Alamos National Laboratory, 2001. To appear in LA Science.

[9] S. D. Bartlett, B. C. Sanders, S. L. Braunstein, and K. Nemoto. Efficient classical simulation of continuous variable quantum information processes. quant-ph/0109047, 2001.