Techniques algébriques en calcul quantique

E. Jeandel

Laboratoire de l'Informatique du Parallélisme
LIP, ENS Lyon, CNRS, INRIA, UCB Lyon

8 Avril 2005

Algebraic Techniques in Quantum Computing

E. Jeandel

Laboratoire de l'Informatique du Parallélisme
LIP, ENS Lyon, CNRS, INRIA, UCB Lyon

April 8th, 2005

Outline

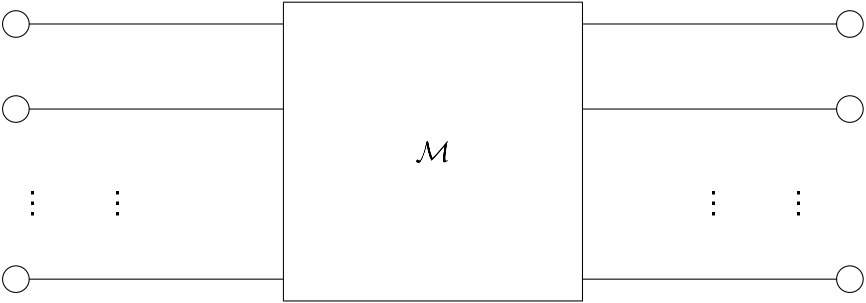
1. Combinatorial setting: Quantum gates
   - Definitions
   - Completeness and Universality
2. Algebraic setting
   - Quantum gates are unitary matrices
   - Computing the group
   - Density
3. Conclusion
   - Automata
   - Conclusion

Introduction

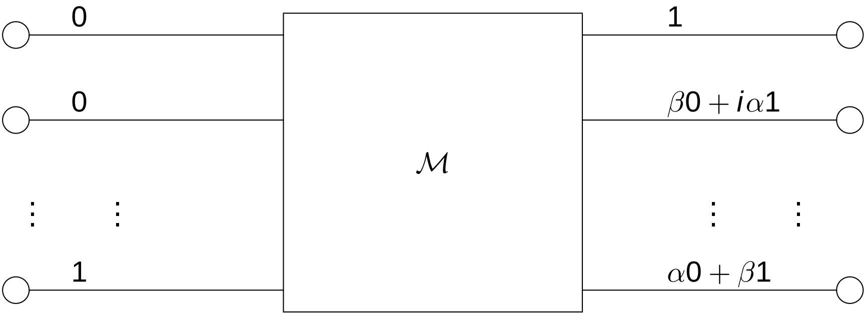
	Classical	**Quantum**
State	q	$\sum \alpha_i q_i$ The system may be in all states simultaneously
Operators	Maps	Unitary (hence reversible) maps

Outline

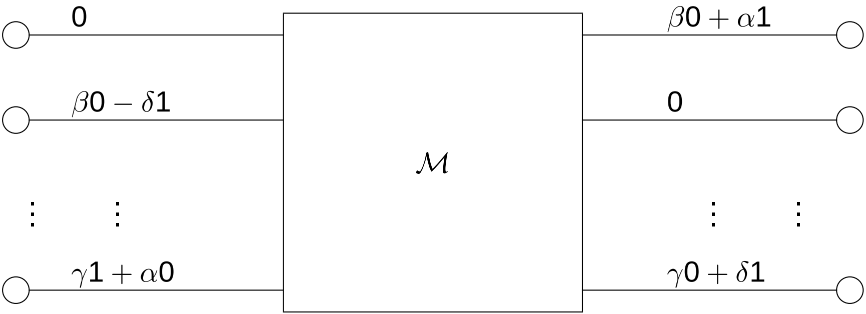
What is a quantum gate ?



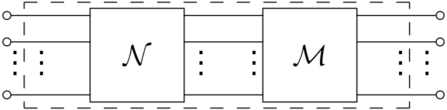
What is a quantum gate ?



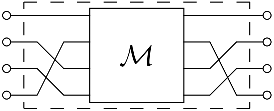
What is a quantum gate ?



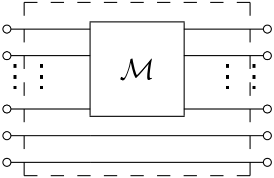
What can we do with quantum gates ?



(a) The multiplication $\mathcal{M}\mathcal{N}$



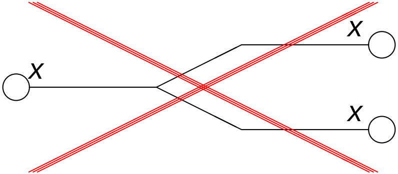
(b) $\mathcal{M}[\sigma]$



(c) The operation $\mathcal{M} \otimes \mathcal{I}$

A quantum circuit is everything we can obtain by applying these constructions.

What we cannot do



Quantum mechanics implies no-cloning.

Outline

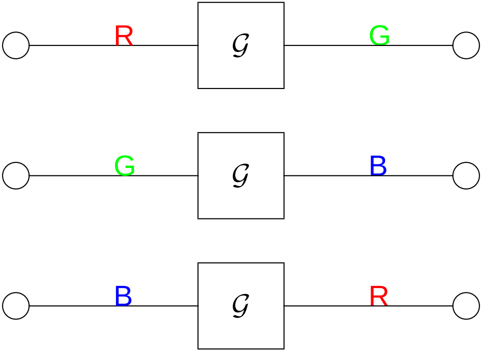
Completeness

- A (finite) set of gates is **complete** if every quantum gate can be obtained by a quantum circuit built on these gates.
- How to show that some set of gates is complete ?

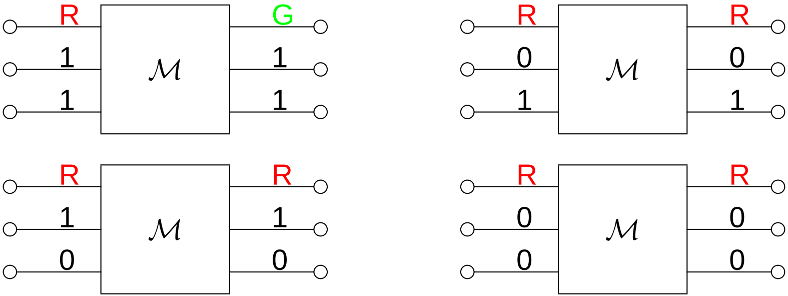
Completeness

- A (finite) set of gates is **complete** if every quantum gate can be obtained by a quantum circuit built on these gates.
- How to show that some set of gates is complete ?

Game: Design this gate



Toolkit 1

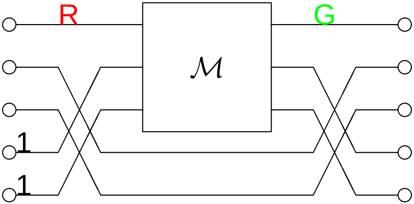


Toolkit 1: Universality

Fact

If there are two wires set to 1, we can make the gate G.

This is called **universality with ancillas**.

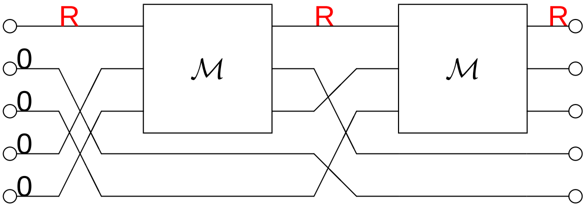


Toolkit 1: Non-completeness

Fact

If among the additional wires, strictly less than 2 are set to 1, the gate G cannot be made.

Any circuit, even the most intricate, cannot produce any 1 using only the gate $\mathcal{M}$.

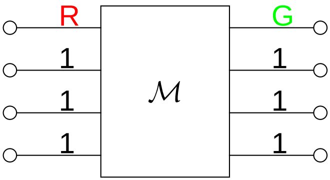


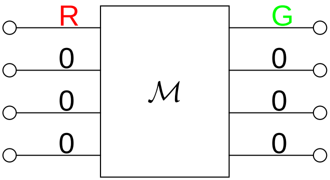
Toolkit 1: Summary

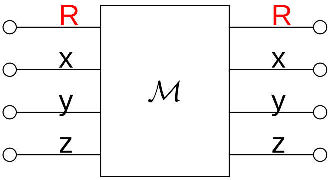
Theorem (8.7)

There exists a set of gates $\mathcal{B}_i$ such that $\mathcal{B}_i$ is 2-universal but neither 1-universal nor k-complete.

Toolkit 2







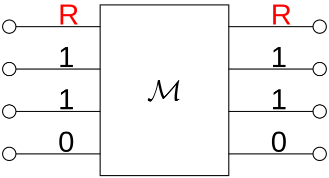
otherwise

Toolkit 2: Non-completeness

Fact

Without any additional wire, we cannot realise the gate G.

If the three given wires are set to $1, 1$ and 0 there is no mean to have three 1 or three 0.



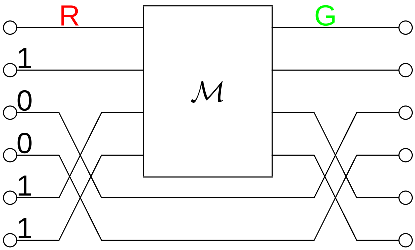
Toolkit 2: 2 additional wires

- We are given two additional 0/1-wires.
- We have now five 0/1-wires. 3 of them must be equal !

Problem: The wiring depends on the 3 equal wires.

Toolkit 2: 2 additional wires

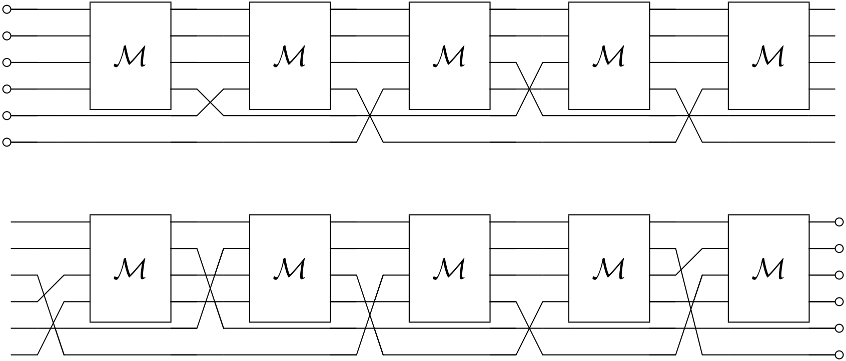
- We are given two additional 0/1-wires.
- We have now five 0/1-wires. 3 of them must be equal !



Problem: The wiring depends on the 3 equal wires.

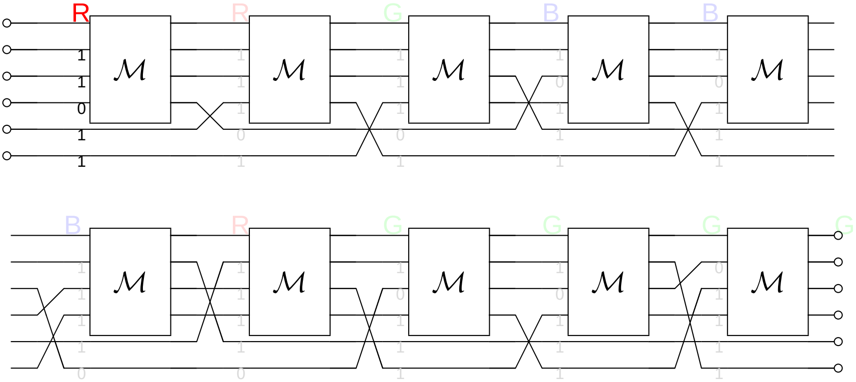
Toolkit 2: Solution

Consider the following circuit:



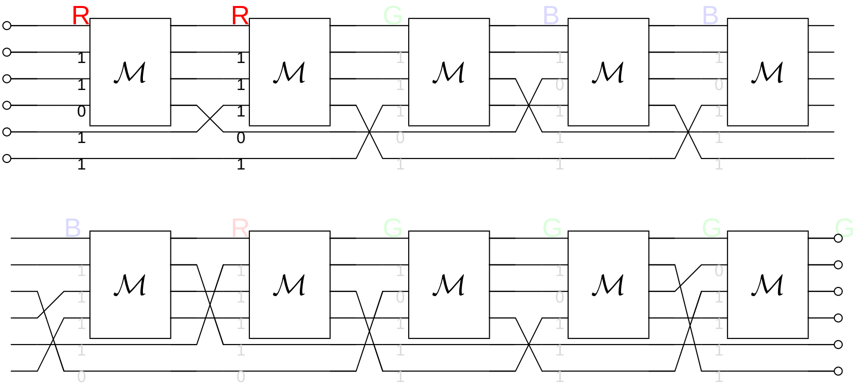
Toolkit 2: Solution

If 4 bits are equal:



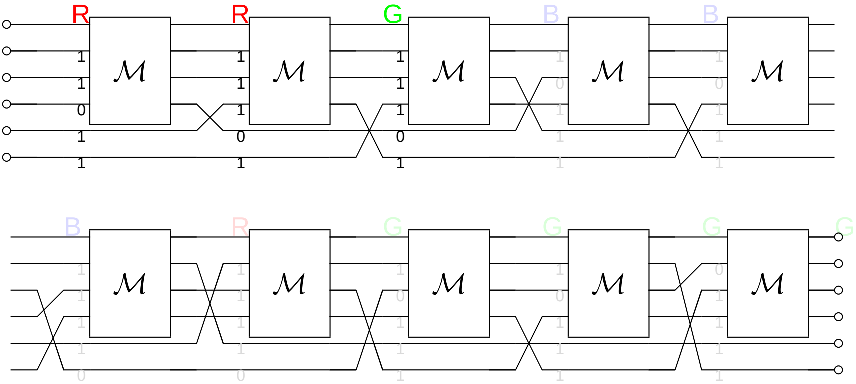
Toolkit 2: Solution

If 4 bits are equal:



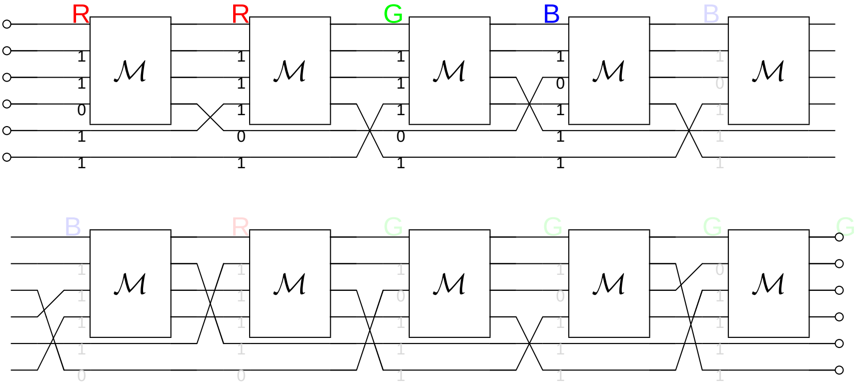
Toolkit 2: Solution

If 4 bits are equal:



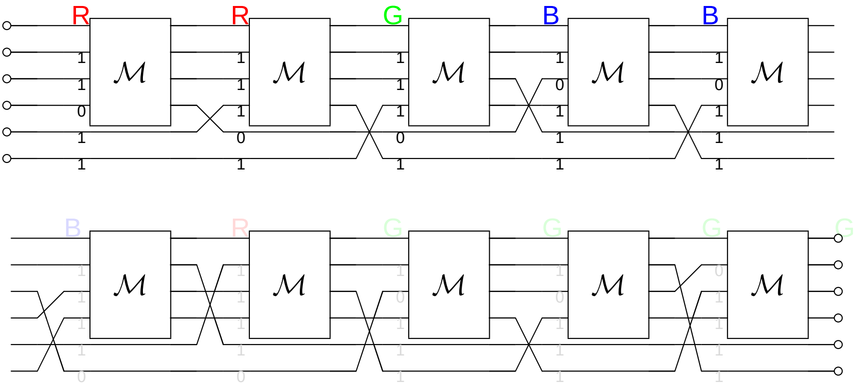
Toolkit 2: Solution

If 4 bits are equal:



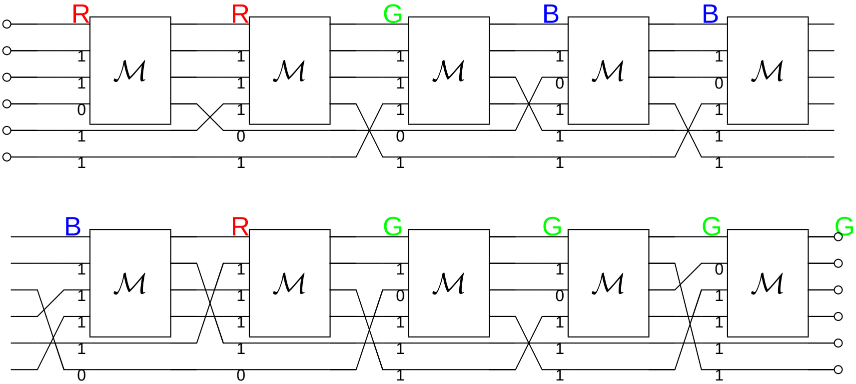
Toolkit 2: Solution

If 4 bits are equal:



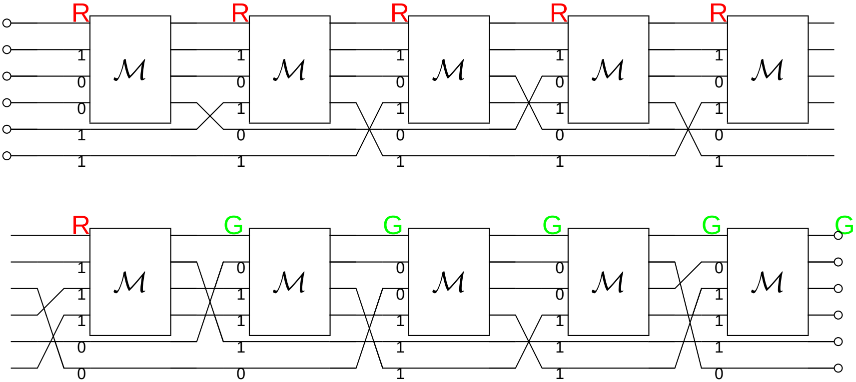
Toolkit 2: Solution

If 4 bits are equal:



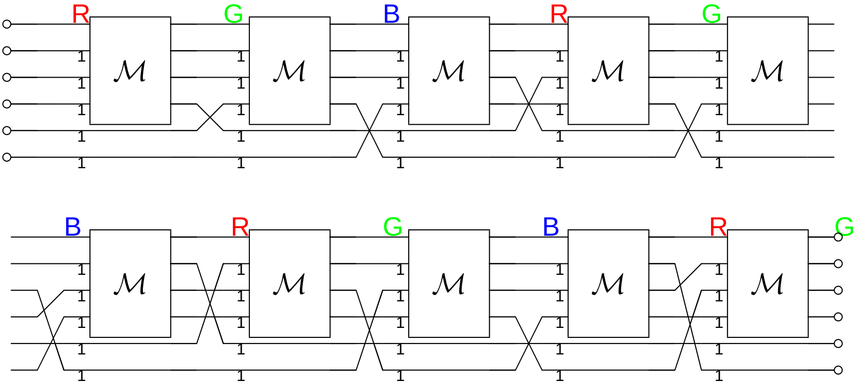
Toolkit 2: Solution

If 3 bits are equal:



Toolkit 2: Solution

If all 5 bits are equal:



Toolkit 2: Summary

Fact

The previous circuit simulates the gate G whatever the bits on the wires are.

This is called 2-**completeness** (since we use 2 additional wires).
Up to some technical details, we obtain:

Theorem (8.8)

There exists a set of gates $\mathcal{B}_i$ such that $\mathcal{B}_i$ is 3-complete but not complete.

Toolkit 2: Summary

Fact

The previous circuit simulates the gate G whatever the bits on the wires are.

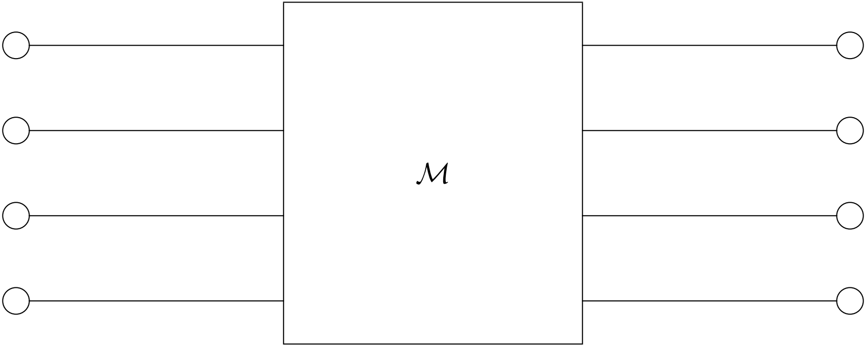
This is called 2-**completeness** (since we use 2 additional wires).
Up to some technical details, we obtain:

Theorem (8.8)

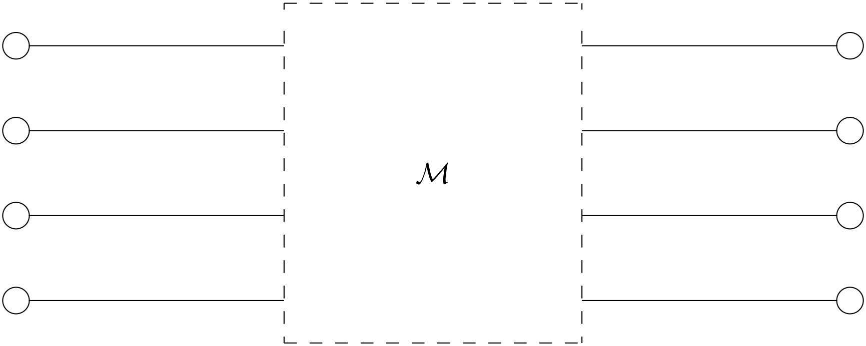
There exists a set of gates $\mathcal{B}_i$ such that $\mathcal{B}_i$ is 3-complete but not complete.

Outline

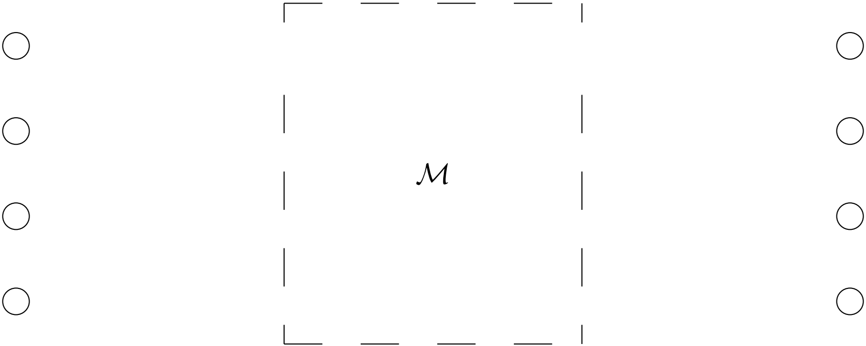
What is a quantum gate ?



What is a quantum gate ?



What is a quantum gate ?



What is a quantum gate ?

$$\mathcal{M}$$

What is a quantum gate ?

A quantum gate over n qubits

$$\mathcal{M}$$

is a $2^n \times 2^n$ unitary matrix

Approximating Quantum Circuits

Problem

Given unitary matrices $\mathcal{X}_1 \dots \mathcal{X}_n$ and a unitary matrix $\mathcal{M}$, is $\mathcal{M}$ in the group generated by the $\mathcal{X}_i$?

In the real life, we do not try to obtain quantum gates, but rather to approximate them.

Problem

Given unitary matrices $\mathcal{X}_1 \dots \mathcal{X}_n$ and a unitary matrix $\mathcal{M}$, is $\mathcal{M}$ in the euclidean closure of the group generated by the $\mathcal{X}_i$?
(More generally, investigate finitely generated compact groups)

Approximating Quantum Circuits

Problem

Given unitary matrices $\mathcal{X}_1 \ldots \mathcal{X}_n$ and a unitary matrix $\mathcal{M}$, is $\mathcal{M}$ in the group generated by the $\mathcal{X}_i$?

In the real life, we do not try to obtain quantum gates, but rather to approximate them.

Problem

Given unitary matrices $\mathcal{X}_1 \ldots \mathcal{X}_n$ and a unitary matrix $\mathcal{M}$, is $\mathcal{M}$ in the euclidean closure of the group generated by the $\mathcal{X}_i$?
(More generally, investigate finitely generated compact groups)

Why compact groups ?

Property

A compact group G of $M_n(\mathbb{R})$ is algebraic. That is there exists polynomials $p_1 \dots p_k$ such that $\mathcal{X} \in G \iff \forall i, p_i(\mathcal{X}) = 0$

For instance, if $G = O_2(\mathbb{R})$, then

$$G = \left\{ X = \begin{pmatrix} a & b \\ c & d \end{pmatrix} : XX^T = \mathcal{I} \right\} = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} : \left\{ \begin{array}{rcl} a^2 + b^2 - 1 & = & 0 \\ c^2 + d^2 - 1 & = & 0 \\ ac + bd & = & 0 \end{array} \right. \right\}$$

We can compute things !
Now we focus on algebraic groups.

Outline

Question

Problem

Given matrices $\mathcal{X}_1 \dots \mathcal{X}_n$, compute the algebraic group generated by the matrices $\mathcal{X}_i$.

Computing the group means finding polynomials p_i such that

$$\mathcal{X} \in G \iff \forall i, p_i(\mathcal{X}) = 0$$

Algebraic sets (defined by polynomials) are the closed sets of a topology called the Zariski topology.

Irreducible groups

Theorem

If G_1 and G_2 are irreducible algebraic groups given by polynomials, one may obtain polynomials for $\langle G_1, G_2 \rangle$ by the following algorithm:

1. $H := \overline{G_1 \cdot G_2}$
2. *While* $\overline{H \cdot H} \neq H$ *do*
   $H := \overline{H \cdot H}$

($\overline{A}$ is the *Zariski-closure* of A, the smallest algebraic set containing A. $\overline{A \cdot B}$ may be obtained by using Groebner basis techniques)

Irreducible groups

Theorem

If G_1 and G_2 are irreducible algebraic groups given by polynomials, one may obtain polynomials for $\langle G_1, G_2 \rangle$ by the following algorithm:

1. $H := \overline{G_1 \cdot G_2}$
2. *While* $\overline{H \cdot H} \neq H$ *do*
   $H := \overline{H \cdot H}$

Sketch of proof: At each step H is an irreducible algebraic variety. If $\overline{H \cdot H} \neq H$, $\overline{H \cdot H}$ is of a greater dimension, which proves that the algorithm terminates.

General groups

Fact

Let G be an algebraic group generated by $X_1 \dots X_k$. Then $G = S \cdot H$ with

1. $\forall i, X_i \in S \cdot H$
2. *H is an irreducible algebraic group*
3. $S \cdot H \cdot S \cdot H = S \cdot H$
4. *H is normal in G : $S \cdot H \cdot S^{-1} = H$*
5. *S is finite*

Furthermore, if the conditions are satisfied by some S and H, then $G = S \cdot H$ is the algebraic group generated by the X_i.

General groups

Fact

Let G be an algebraic group generated by $X_1 \dots X_k$. Then $G = S \cdot H$ with

1. $\forall i, X_i \in S \cdot H$
2. *H is an irreducible algebraic group*
3. $S \cdot S \subseteq S \cdot H$
4. *H is normal in G : $S \cdot H \cdot S^{-1} = H$*
5. *S is finite*

Furthermore, if the conditions are satisfied by some S and H, then $G = S \cdot H$ is the algebraic group generated by the X_i.

Sketch of an algorithm

Define by induction

1. $S_0 = \{X_i\}$, $H_0 = \{\mathcal{I}\}$
2. $H_{n+1} := \overline{H_n \cdot H_n}$
3. $S_{n+1} := S_n$.
   For X, Y in S_n, if $X \cdot Y \not\in S_n H_n$ then $S_{n+1} := S_{n+1} \cup \{X \cdot Y\}$
4. For X in S_n do $H_{n+1} := \overline{X \cdot H_{n+1} \cdot X^{-1} \cdot H_{n+1}}$

Then the limit $S = \bigcup S_n, H = \bigcup H_n$ satisfies all conditions of the previous fact. . . except perhaps the last one.

Sketch of an algorithm

Define by induction

1. $S_0 = \{X_i\}$, $H_0 = \{\mathcal{I}\}$
2. $H_{n+1} := \overline{H_n \cdot H_n}$
3. $S_{n+1} := S_n$.
   For X, Y in S_n, if $X \cdot Y \notin S_n H_n$ then $S_{n+1} := S_{n+1} \cup \{X \cdot Y\}$
4. For X in S_n do $H_{n+1} := \overline{X \cdot H_{n+1} \cdot X^{-1} \cdot H_{n+1}}$

Then the limit $S = \bigcup S_n, H = \bigcup H_n$ satisfies all conditions of the previous fact…except perhaps the last one.

General groups revisited

Fact

Let G be an algebraic group generated by $X_1 \dots X_k$. Then $G = S \cdot H$ with

1. $\forall i, X_i \in S \cdot H$
2. *H is an irreducible algebraic group*
3. $S \cdot S \subseteq S \cdot H$
4. *H is normal in G : $S \cdot H \cdot S^{-1} = H$*
5. *S is finite*

Furthermore, if the conditions are satisfied by some S and H, then S is finite and $G = S \cdot H$ is the algebraic group generated by the X_i.

General groups revisited

Fact

Let G be an algebraic group generated by $X_1 \dots X_k$. Then $G = S \cdot H$ with

1. $\forall i, X_i \in S \cdot H$
2. *H is an irreducible algebraic group*
3. $S \cdot S \subseteq S \cdot H$
4. *H is normal in G : $S \cdot H \cdot S^{-1} = H$*
5. *$\forall X \in S$ there exists $n > 0$ such that $X^n \in H$.*

Furthermore, if the conditions are satisfied by some S and H, then S is finite and $G = S \cdot H$ is the algebraic group generated by the X_i.

Sketch of an algorithm, revisited

Define by induction

1. $S_0 = \{X_i\}$, $H = \{\mathcal{I}\}$
2. $H_{n+1} := \overline{H_n \cdot H_n}$
3. $S_{n+1} := S_n$.
   For X, Y in S_n, if $X \cdot Y \notin S_n H_n$ then $S_{n+1} := S_{n+1} \cup \{X \cdot Y\}$
4. For X in S_n do $H_{n+1} := \overline{X \cdot H_{n+1} \cdot X^{-1} \cdot H_{n+1}}$
5. For X in S_n, compute the group $G_X = S_X H_X$ generated by X and add H_X to H_{n+1} : $H_{n+1} := \overline{H_X \cdot H_{n+1}}$

Then the limit $S = \bigcup S_n, H = \bigcup H_n$ satisfies all conditions of the previous fact. In particular, S is finite.

The new algorithm works

Theorem

The previous algorithm terminates and gives sets S, H *such that* $G = S \cdot H$ *is the algebraic group generated by the* X_i*.*

We need only to know how to compute the group generated by one matrix.

Group generated by one matrix : example

$$X = \begin{pmatrix} \beta^2 & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \beta\gamma^{-3} & 0 \\ 0 & 0 & 0 & \gamma \end{pmatrix}$$

The group generated by X is

$$\langle X \rangle = \left\{ \begin{pmatrix} \beta^{2k} & 0 & 0 & 0 \\ 0 & \beta^k & 0 & 0 \\ 0 & 0 & \beta^k\gamma^{-3k} & 0 \\ 0 & 0 & 0 & \gamma^k \end{pmatrix}, k \in \mathbb{Z} \right\}$$

The algebraic group generated by X is

$$\left\{ \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{pmatrix}, ab^{-2} = 1, b^{-1}d^3c = 1 \right\}$$

Group generated by one matrix

A unitary matrix, up to a change of basis is of the form

$$\begin{pmatrix} \alpha_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \alpha_n \end{pmatrix}$$

(Multiplicative) relationships between the α_i is the key point:

$$(m_1, \dots, m_n) \in \Gamma \iff \prod_i \alpha_i^{m_i} = 1$$

The algebraic group generated by X is then

$$\left\{ \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{pmatrix} : \prod_i \lambda_i^{m_i} = 1 \ \forall (m_1, \dots, m_n) \in \Gamma \right\}$$

To find Γ, we must find bounds for the m_i.

Group generated by one matrix

Theorem (Ge)

There exists a polynomial-time algorithm which given the α_i computes the multiplicative relations between the α_i.

Corollary

There exists an algorithm which computes the compact group generated by a unitary matrix X.

Theorem

There exists an algorithm which computes the algebraic group generated by a matrix X.

Group generated by one matrix

Theorem (Ge)

There exists a polynomial-time algorithm which given the α_i computes the multiplicative relations between the α_i.

Corollary

There exists an algorithm which computes the compact group generated by a unitary matrix X.

Theorem

There exists an algorithm which computes the algebraic group generated by a matrix X.

Group generated by one matrix

Theorem (Ge)

There exists a polynomial-time algorithm which given the α_i computes the multiplicative relations between the α_i.

Corollary

There exists an algorithm which computes the compact group generated by a unitary matrix X.

Theorem

There exists an algorithm which computes the algebraic group generated by a matrix X.

Summary

Theorem (3.3)

There exists an algorithm which given matrices X_i computes the algebraic group generated by the X_i.

Due to the method (keep going until it stabilises), there is absolutely no bound of complexity for the algorithm.

Theorem

There exists an algorithm which given unitary matrices X_i computes the compact group generated by the X_i.

Summary

Theorem (3.3)

There exists an algorithm which given matrices X_i computes the algebraic group generated by the X_i.

Due to the method (keep going until it stabilises), there is absolutely no bound of complexity for the algorithm.

Theorem

There exists an algorithm which given unitary matrices X_i computes the compact group generated by the X_i.

Outline

Question

Problem

Given matrices $\mathcal{X}_1 \dots \mathcal{X}_k$, decide if the group generated by the matrices $\mathcal{X}_i$ is dense in the algebraic group G.

The good notion of "density" for an algebraic group is the Zariski-density.

Problem

Given unitary matrices $\mathcal{X}_1 \dots \mathcal{X}_k$ of dimension n, decide if the group generated by the matrices $\mathcal{X}_i$ is dense in U_n

Question

Problem

Given matrices $\mathcal{X}_1 \dots \mathcal{X}_k$, decide if the group generated by the matrices $\mathcal{X}_i$ is dense in the algebraic group G.

The good notion of "density" for an algebraic group is the Zariski-density.

Problem

Given unitary matrices $\mathcal{X}_1 \dots \mathcal{X}_k$ of dimension n, decide if the group generated by the matrices $\mathcal{X}_i$ is dense in U_n

Simple groups

A simple group has no non-trivial normal irreducible subgroups.
This gives an algorithm for a simple group:

Theorem

H is dense in a simple group G iff H is infinite and H is normal in G.

There exists an algorithm from Babai, Beals and Rockmore to test if a finitely generated group is finite.
We only have to find a way to show that H is normal in G.

Normal groups

$$H \text{ is normal in } G \iff \forall X \in G, XHX^{-1} = H$$

Denote by K_G the set $\{M \mapsto XMX^{-1}, X \in G\}$. K_G is a set (in fact a group) of endomorphisms of M_n.

$$H \text{ is normal in } G \iff \forall \phi \in K_G, \phi(H) = H$$

Fact

$\forall \phi \in K_H, \phi(H) = H.$

Corollary

If $K_H = K_G$ then H is normal in G.

Testing $K_H = K_G$ is not that easy..

Normal groups

$$H \text{ is normal in } G \iff \forall X \in G, XHX^{-1} = H$$

Denote by K_G the set $\{M \mapsto XMX^{-1}, X \in G\}$. K_G is a set (in fact a group) of endomorphisms of M_n.

$$H \text{ is normal in } G \iff \forall \phi \in K_G, \phi(H) = H$$

Fact

$\forall \phi \in K_H, \phi(H) = H.$

Corollary

If $K_H = K_G$ then H is normal in G.

Testing $K_H = K_G$ is not that easy..

Normal groups

$$H \text{ is normal in } G \iff \forall X \in G, XHX^{-1} = H$$

Denote by K_G the set $\{M \mapsto XMX^{-1}, X \in G\}$. K_G is a set (in fact a group) of endomorphisms of M_n.

$$H \text{ is normal in } G \iff \forall \phi \in K_G, \phi(H) = H$$

Fact

$\forall \phi \in K_H, \phi(H) = H$.

Corollary

If $K_H = K_G$ then H is normal in G.

Testing $K_H = K_G$ is not that easy..

Normal groups

$$H \text{ is normal in } G \iff \forall X \in G, XHX^{-1} = H$$

Denote by K_G the set $\{M \mapsto XMX^{-1}, X \in G\}$. K_G is a set (in fact a group) of endomorphisms of M_n.

$$H \text{ is normal in } G \iff \forall \phi \in K_G, \phi(H) = H$$

Fact

$\forall \phi \in K_H, \phi(H) = H$.

Corollary

If $K_H = K_G$ then H is normal in G.

Testing $K_H = K_G$ is not that easy..

Normal groups

Denote by $\mathrm{Span}(S)$ the vector space generated by S.

Theorem (2.5)

If $\mathrm{Span}(K_H) = \mathrm{Span}(K_G)$, then H is normal in G.

Proof.

We use Lie algebras techniques. The condition implies that the Lie algebra of H is an ideal of the Lie algebra of G.

Fact

Testing whether $\mathrm{Span}(K_H) = \mathrm{Span}(K_G)$ is easy.

Normal groups

Denote by $\mathrm{Span}(S)$ the vector space generated by S.

Theorem (2.5)

If $\mathrm{Span}(K_H) = \mathrm{Span}(K_G)$, then H is normal in G.

Proof.

We use Lie algebras techniques. The condition implies that the Lie algebra of H is an ideal of the Lie algebra of G. □

Fact

Testing whether $\mathrm{Span}(K_H) = \mathrm{Span}(K_G)$ is easy.

Normal groups

Denote by $\mathrm{Span}(S)$ the vector space generated by S.

Theorem (2.5)

If $\mathrm{Span}(K_H) = \mathrm{Span}(K_G)$, then H is normal in G.

Proof.

We use Lie algebras techniques. The condition implies that the Lie algebra of H is an ideal of the Lie algebra of G.

□

Fact

Testing whether $\mathrm{Span}(K_H) = \mathrm{Span}(K_G)$ is easy.

Computing $\mathrm{Span}(K_H)$

Let E be the vector space generated by the morphisms $M \mapsto X_i M X_i^{-1}$
While E is not stable by multiplication (composition), let
$E := EE = \{\phi \circ \psi : \phi \in E, \psi \in E\}$

Theorem

For every simple group G, there exists a polynomial time algorithm which decides if a finitely generated subgroup H is dense in G.

Computing $\mathrm{Span}(K_H)$

Let E be the vector space generated by the morphisms $M \mapsto X_i M X_i^{-1}$
While E is not stable by multiplication (composition), let
$E := EE = \{\phi \circ \psi : \phi \in E, \psi \in E\}$

Theorem

For every simple group G, there exists a polynomial time algorithm which decides if a finitely generated subgroup H is dense in G.

Generalisation

Theorem (2.26)

For every reductive group G, there exists a polynomial time algorithm which decides if a finitely generated subgroup H is Zariski-dense in G.

Theorem (2.27)

For every compact group G, there exists a polynomial time algorithm which decides if a finitely generated subgroup H is dense in G.

Generalisation

Theorem (2.26)

For every reductive group G, there exists a polynomial time algorithm which decides if a finitely generated subgroup H is Zariski-dense in G.

Theorem (2.27)

For every compact group G, there exists a polynomial time algorithm which decides if a finitely generated subgroup H is dense in G.

Back to circuits

Theorem (8.5)

There exists a polynomial time algorithm which decides if a set of gates is complete.

Theorem (8.4)

There exists an algorithm which decides if a set of gates is universal.

Back to circuits

Theorem (8.5)

There exists a polynomial time algorithm which decides if a set of gates is complete.

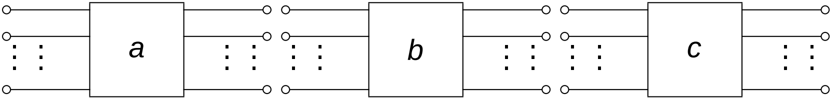
Theorem (8.4)

There exists an algorithm which decides if a set of gates is universal.

Outline

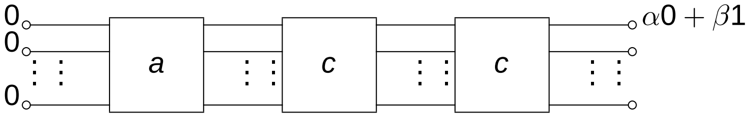
Automata (Sketch)

We are given a gate for each letter $a, b, c \ldots$.

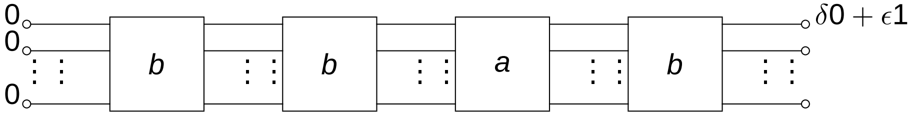


The value (or probability) of a word ω is function of the result of the circuit corresponding to ω.

Automata (Sketch)



acc is accepted with probability $|\alpha|^2$.



bbab is accepted with probability $|\delta|^2$.

Theorems

Some theorems about quantum automata :

Theorem (5.4)

We can decide given an automaton A and a threshold λ if there exists a word accepted with a probability strictly greater than λ.

We use the algorithm which computes the group generated by some matrices.

Theorem (7.1)

Non-deterministic quantum automata with an isolated threshold recognise only regular languages.

The proof introduces a new model of automata, called topological automata.

Theorems

Some theorems about quantum automata :

Theorem (5.4)

We can decide given an automaton A and a threshold λ if there exists a word accepted with a probability strictly greater than λ.

We use the algorithm which computes the group generated by some matrices.

Theorem (7.1)

Non-deterministic quantum automata with an isolated threshold recognise only regular languages.

The proof introduces a new model of automata, called topological automata.

Outline

Conclusion

- Study of quantum objects using algebraic groups techniques.
- New algorithms about algebraic groups.
- Many other potentially interesting things.

Conclusion

- Study of quantum objects using algebraic groups techniques.
- New algorithms about algebraic groups.
- Many other potentially interesting things.

Conclusion

- Study of quantum objects using algebraic groups techniques.
- New algorithms about algebraic groups.
- Many other potentially interesting things.

Perspectives and open problems

Problem

What if the number of auxiliary wires depends on the gate to realise (∞-universality) ?
Is it equivalent to m-universality for some m ?

Problem

Find an efficient algorithm to decide whether some matrix $\mathcal{X}$ is in the algebraic group generated by the matrices $\mathcal{X}_i$.

More generally, use the structure of the algebraic groups more efficiently.

Perspectives and open problems

Problem

What if the number of auxiliary wires depends on the gate to realise (∞-universality) ?
Is it equivalent to m-universality for some m ?

Problem

Find an efficient algorithm to decide whether some matrix $\mathcal{X}$ is in the algebraic group generated by the matrices $\mathcal{X}_i$.

More generally, use the structure of the algebraic groups more efficiently.