

Title: Classical and quantum circuit obfuscation with braids

Date: Apr 08, 2013 04:00 PM

URL: <http://pirsa.org/13040111>

Abstract: A circuit obfuscator is an algorithm that translates logic circuits into functionally-equivalent similarly-sized logic circuits that are hard to understand. While ad hoc obfuscators have been implemented, theoretical progress has mainly been limited to no-go results. In this work, we propose a new notion of circuit obfuscation, which we call partial indistinguishability. We then prove that, in contrast to previous definitions of obfuscation, partial indistinguishability obfuscation can be achieved by a polynomial-time algorithm. Specifically, our algorithm re-compiles the given circuit using a gate that satisfies the

relations of the braid group, and then reduces to a braid normal form. Variants of our obfuscation algorithm can be applied to both classical and quantum circuits.

Circuit Obfuscation with Braids

Stephen Jordan

In collaboration with:

Gorjan Alagic

Stacey Jeffery

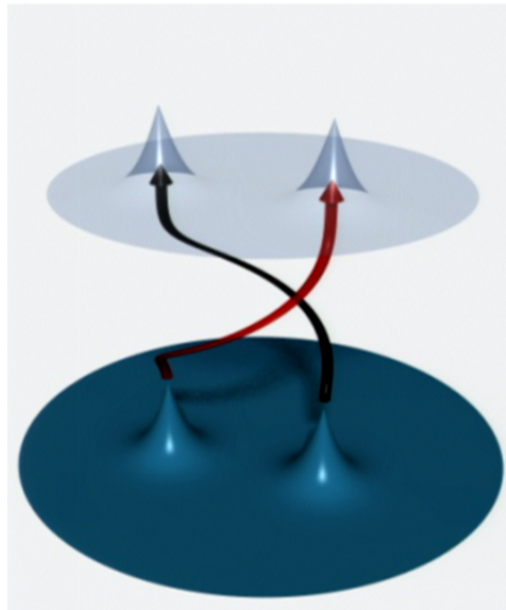
[ArXiv: 1212.6458]

NIST

April 8, 2013

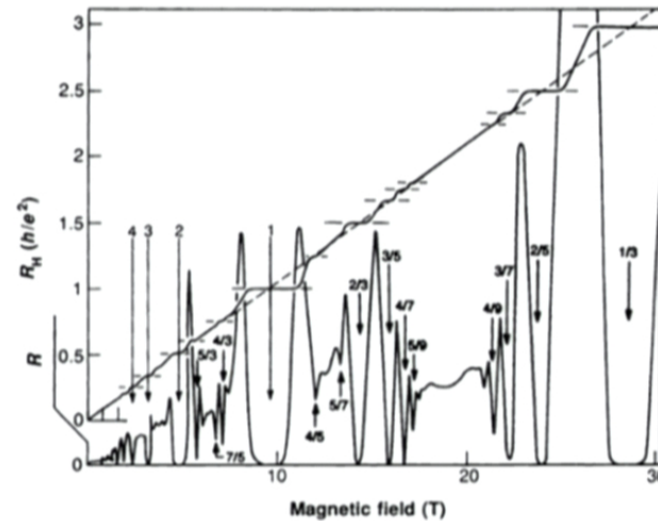
Anyons

- In (2+1)-D winding number is well-defined
- Particle exchange can induce phase $\neq \pm 1$



Non-Abelian Anyons

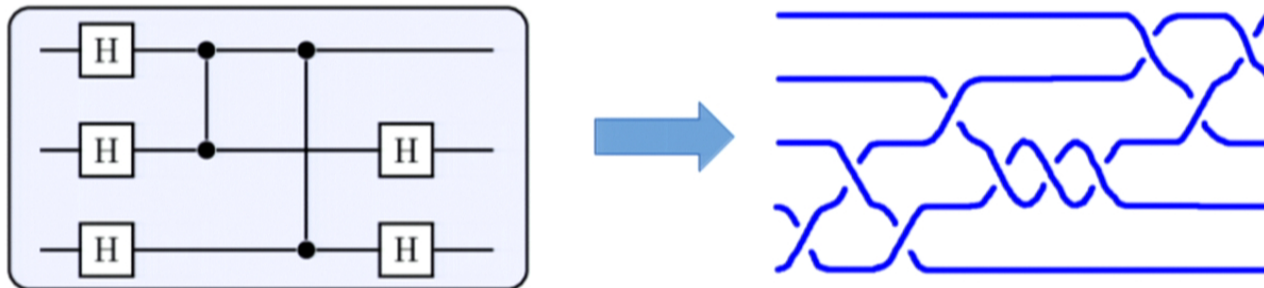
- Two-dimensional condensed-matter systems may have anyonic quasiparticle excitations.



- Braiding can induce unitary transformations within degenerate ground space.

Non-Abelian Anyons

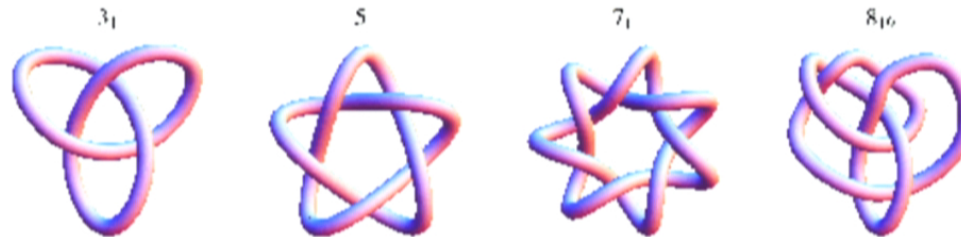
- Non-abelian anyons give us a unitary representation of the braid group.
- In some cases the set of unitary transformations induced by elementary crossings is a universal set of quantum gates.



“topological quantum computation”

Immediate Applications?

- Topological quantum computers are a long way off.
- However, the theory of anyons has yielded many new results in mathematics, such as invariants of links and 3-manifolds.



- How about applications in computer science?

What is Obfuscation?



An obfuscator is a compiler that transforms any program into an obfuscated program that has the same input-output functionality as the original program, but is “unintelligible.”



Shafi Goldwasser and Guy Rothblum
“On best possible obfuscation” (2007)

Why Obfuscate?

- **In practice:**
 - Prevent reverse engineering - “protect IP”
 - Prevent tampering
- **In theory:**
 - Convert private-key encryption to public-key
 - Do homomorphic encryption

Obfuscation in Practice



Red Gate's .NET obfuscator

Protect your .NET code and Intellectual Property

Download free trial



Protect your .NET code and IP with SmartAssembly

.NET applications are easy to disassemble if they haven't been obfuscated. SmartAssembly is an obfuscator that helps protect your application against reverse-engineering, cracking, and modification.

If you don't want your C# or VB.NET code exposed internationally, or if your entire business rests on the IP embodied in your software, then obfuscating your code becomes a necessity, not a luxury.



Pricing (exc. support & upgrades) **from \$795**

SmartAssembly is licensed per production build machine (any machine you use to produce release builds). A Support & Upgrade package is available for 25% of the product's purchase price.

Pricing

Got a question?

Obfuscation in Practice

The screenshot shows the website for PreEmptive Solutions, specifically the page for Dotfuscator for .NET. The page features a dark navigation bar with the company logo and menu items like 'PRODUCTS', 'SUPPORT', 'KNOW MORE', and 'COMPANY'. There are also buttons for 'SALES CHAT', 'MY ACCOUNT', and 'CONTACT SUPPORT'. The main content area has a large banner with the product name and a description: 'Protect & defend your .NET apps from reverse engineering & tampering'. Below the banner, there are three columns of text: '50 reasons to choose Dotfuscator', 'Obfuscation is only the beginning', and 'Application Instrumentation'. A red circular icon with an exclamation mark is positioned above a section titled 'Professional Edition 4.9.8500 - Our latest release.' To the right, there is a box titled 'Need help finding a solution?' with a 'Buying Guide' link. At the bottom right, there is a currency selector set to 'US Dollar'.

Please log in. Home News & Events Blog Contact

PreEmptive Solutions PRODUCTS SUPPORT KNOW MORE COMPANY SALES CHAT MY ACCOUNT CONTACT SUPPORT

PreEmptive Protection dotfuscator for .NET

Protect & defend your .NET apps from reverse engineering & tampering

.Net Obfuscation | Dotfuscator [Overview](#) [Compare Editions](#) [See It Work](#) [Licensing](#)

50 reasons to choose Dotfuscator

There are more than 50, but we had to draw a line somewhere!
[Take Me There](#)

Obfuscation is only the beginning

Managing risk is much more than munging MSIL - it's about efficient integration of development controls.
[Learn More](#)

Application Instrumentation

With injection, Dotfuscator can easily add application monitoring to existing apps and new development.
[See It Work](#)

Professional Edition 4.9.8500 - Our latest release.

Our continued investment in Dotfuscator is reflected in the stream of product upgrades and enhancements. To learn more about our latest, visit our [change log](#).

Need help finding a solution?
You can reference our buying guide, support forum, or just ask us!
[+ Buying Guide](#)

Currency:

An Example

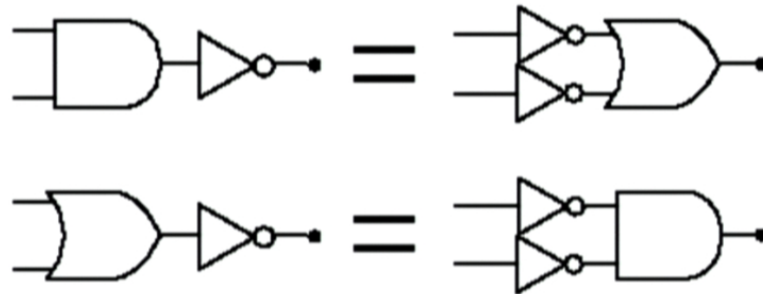
```
#include<stdio.h> #include<string.h> main(){char*0,l[999]=
''‘acgo\177~|xp .-\OR^8)NJ6%K40+A2M(*0ID57$3G1FBL";while(0=
fgets(l+45,954,stdin)){*l=0[strlen(0)[0-1]=0,strupn(0,l+11)];
while(*0)switch((*l&&isalnum(*0))-!*l){case-1:{char*I=(0+=
strupn(0,l+12)+1)-2,0=34;while(*I&&(0=(0-16<<1)+*I---’-’)<80);
putchar(0&93?*I&8||!( I=memchr( l , 0 , 44 ) ) ??’:I-l+47:32);
break;case 1: ;}*l=(*0&31)[l-15+(*0>61)*32];while(putchar(45+*l%2),
(*l=*l+32>>1)>35);case 0:putchar((++0,32));}putchar(10);}}
```

Fig. 1. The winning entry of the 1998 *International Obfuscated C Code Contest*, an ASCII/Morse code translator by Frans van Dorsselaer [vD98] (adapted for this paper).

B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich,
A. Sahai, S. Vadhan, and K. Yang
“On the (Im)possibility of obfuscating programs” (2001)

State of the Art

- **Practice:** Apply local circuit identities [**ad hoc**]



- **Theory:**
 - Several formal definitions proposed
 - Positive results for very weak models of computation (e.g. point functions)
 - **Primarily no-go theorems**


Definition 1: Black Box

Black box obfuscation: Anything one can efficiently compute from the obfuscated circuit, one should be able to efficiently compute given just oracle access to the function that the circuit computes.

B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich,
A. Sahai, S. Vadhan, and K. Yang
“On the (Im)possibility of obfuscating programs” (2001)

Definition 1: Black Box

Black box obfuscation: Anything one can efficiently compute from the obfuscated circuit, one should be able to efficiently compute given just oracle access to the function that the circuit computes.

 Impossible!

B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich,
A. Sahai, S. Vadhan, and K. Yang
“On the (Im)possibility of obfuscating programs” (2001)

Definition 2: Indistinguishability

Indistinguishability obfuscation: Let O be an obfuscator. If circuits A and B compute the same function then $O(A)$ is indistinguishable from $O(B)$.

Variants:

$O(A) = O(B)$

$O(A)$ and $O(B)$ have small total variation distance

$O(A)$ and $O(B)$ are computationally indistinguishable

B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich,
A. Sahai, S. Vadhan, and K. Yang

“On the (Im)possibility of obfuscating programs” (2001)

Definition 2: Indistinguishability

Indistinguishability obfuscation: Let O be an obfuscator. If circuits A and B compute the same function then $O(A)$ is indistinguishable from $O(B)$.

Possible – output the lexicographically first circuit implementing the given function... but this takes exponential time.

Efficient Indistinguishability?

Variants:

$O(A)=O(B)$

➔ **Impossible**: $coNP$ would equal P

$O(A)$ and $O(B)$ have small total variation distance

➔ **Impossible**: PH would collapse to 2nd level*

$O(A)$ and $O(B)$ are computationally indistinguishable

➔ **Open**: but impossible in random oracle model*

*[Goldwasser & Rothblum, 2007]

Intuition

Best-possible obfuscation: $O(A)$ leaks no more information than any other similarly-sized circuit computing the same function.

Black box \geq Best-possible \geq Indistinguishability

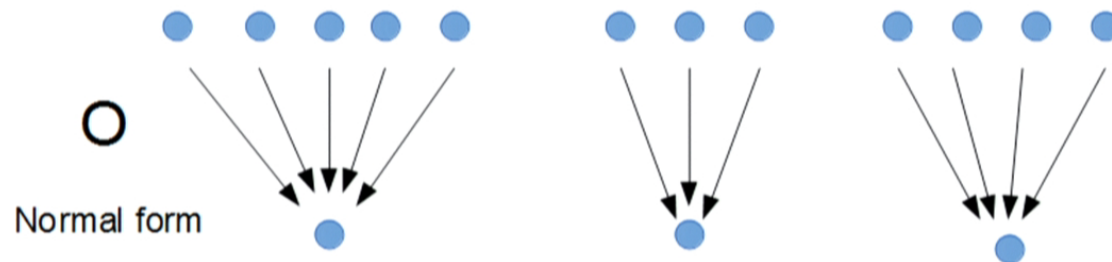
For polynomial-time obfuscators:

Best possible = Indistinguishability

[Goldwasser & Rothblum, 2007]

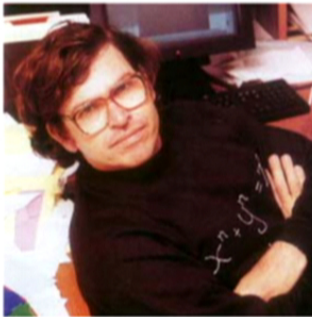
More Intuition

- A perfect indistinguishability obfuscator maps circuits to a normal form, which depends only on their blackbox behavior.



- Many-to-one map erases all information about implementation.

Thurston's Algorithm



A braid word w in the Artin generators can be reduced to normal form in $O(|w|^2)$ time.

[Thurston, 1992]

$$B_n = \left\langle \sigma_1, \dots, \sigma_{n-1} \mid \begin{array}{l} \sigma_i \sigma_j = \sigma_j \sigma_i \quad |i - j| \geq 2 \\ \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \quad \forall i \end{array} \right\rangle$$

If two braids are equivalent, they will be reduced to the same normal form.

➡ “Perfect indistinguishability obfuscation of braids.”

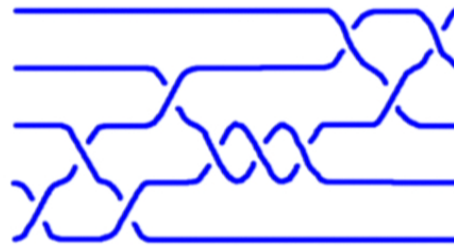
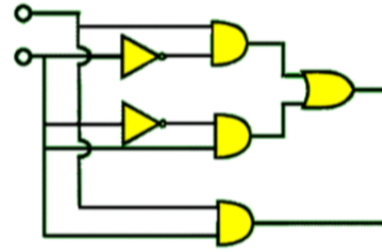
Kitaev's Gate



There exists a universal reversible gate obeying the relations of the braid group.
[Kitaev, 1997]

- Kitaev considered an anyonic model called the quantum double of S_5
- Two kinds of particles: “charges” and “fluxes”
- Braiding of fluxes is represented by permutation matrices.
- The permutations do universal classical computation.

Our Obfuscation Algorithm



Normal form

Definition 4: Partial-indistinguishability

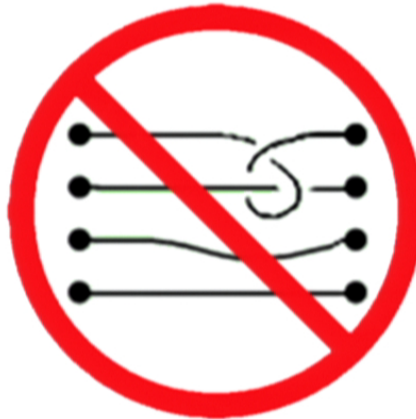
Partial-indistinguishability obfuscation: Let A and B be circuits. Let Γ be a set of identities obeyed by the gates. If A is obtainable from B via Γ then $O(A)=O(B)$.

If Γ is a complete set of identities this is (deterministic) indistinguishability obfuscation.

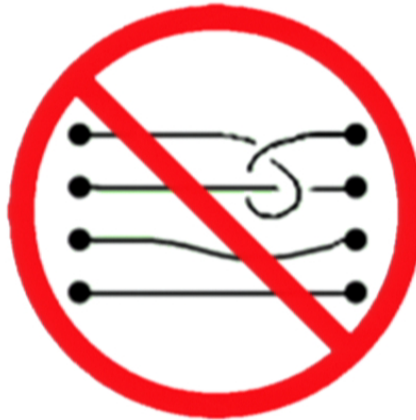
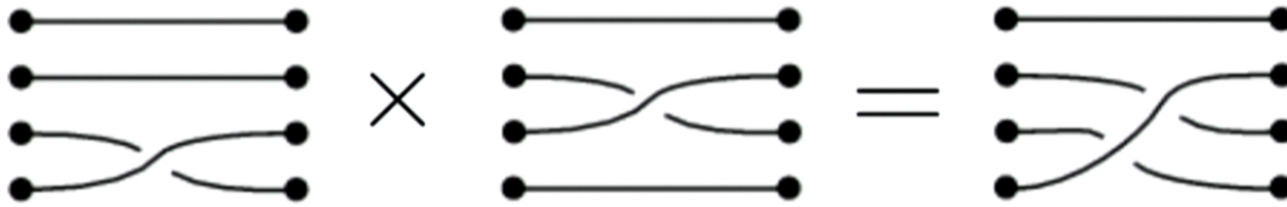
Main result: We have constructed a nontrivial polynomial-time partial-indistinguishability obfuscator.

[Alagic, Jeffery, Jordan, 2013]

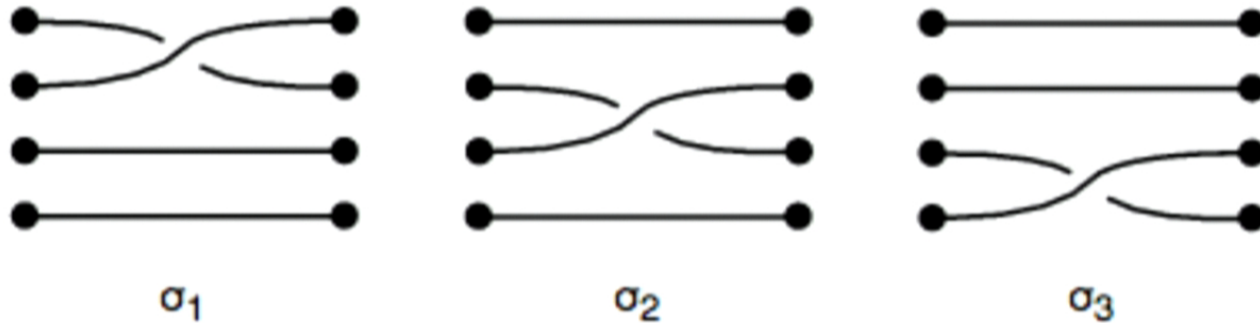
The Braid Group



The Braid Group



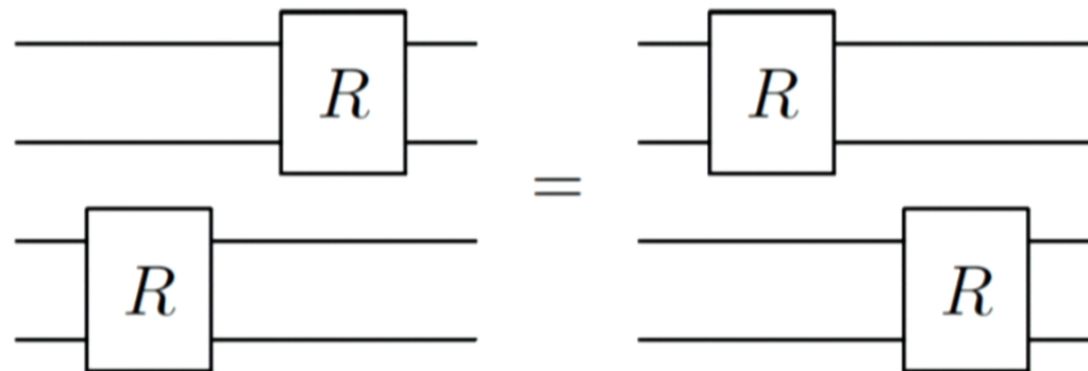
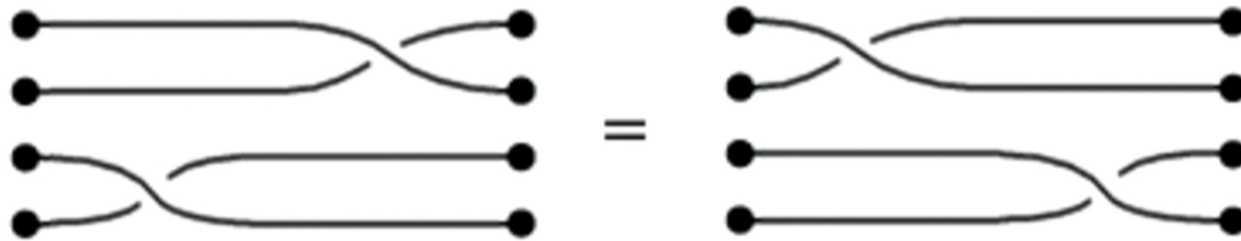
The Artin Generators



$$B_n = \left\langle \sigma_1, \dots, \sigma_{n-1} \mid \begin{array}{l} \sigma_i \sigma_j = \sigma_j \sigma_i \quad |i - j| \geq 2 \\ \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \quad \forall i \end{array} \right\rangle$$

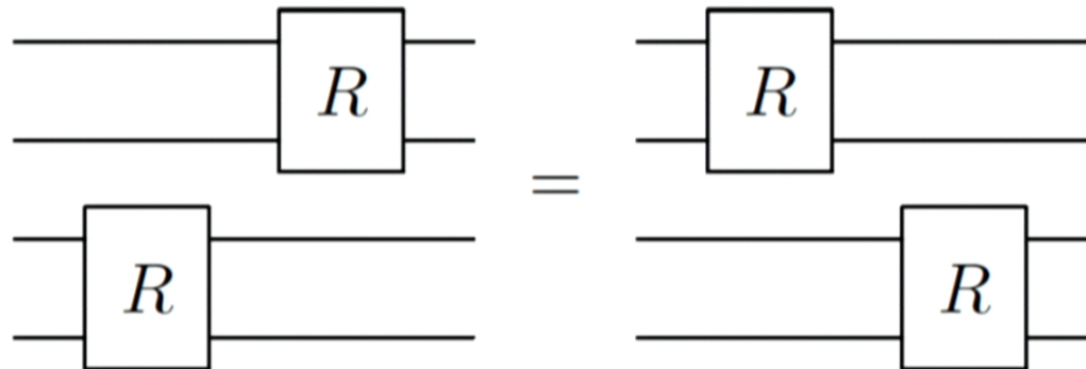
Commutation

$$\sigma_i \sigma_j = \sigma_j \sigma_i \quad |i - j| \geq 2$$



Commutation

$$\sigma_i \sigma_j = \sigma_j \sigma_i \quad |i - j| \geq 2$$



Yang-Baxter Computing

- Any matrix R satisfying the Yang-Baxter equation yields a representation of the braid group

$$\sigma_1 \mapsto R \otimes \mathbb{1} \otimes \mathbb{1} \otimes \dots$$

$$\sigma_2 \mapsto \mathbb{1} \otimes R \otimes \mathbb{1} \otimes \dots$$

$$\vdots$$

- Unitary R : quantum computation by braiding
- Permutation R : classical computation by braiding
- Universal?
- How to find R ?

Yang-Baxter Computing

- Any matrix R satisfying the Yang-Baxter equation yields a representation of the braid group

$$\sigma_1 \mapsto R \otimes \mathbb{1} \otimes \mathbb{1} \otimes \dots$$

$$\sigma_2 \mapsto \mathbb{1} \otimes R \otimes \mathbb{1} \otimes \dots$$

⋮

- Unitary R : quantum computation by braiding
- Permutation R : classical computation by braiding
- Universal?
- How to find R ?

Man vs. Machine

There are 133,081 permutation matrices satisfying the 25x25 Yang-Baxter equation, none of which is a universal gate.



Just use the quantum double of S_5 !



Man vs. Machine

There are 133,081 permutation matrices satisfying the 25x25 Yang-Baxter equation, none of which is a universal gate.



A_5 will do!



Quantum Double

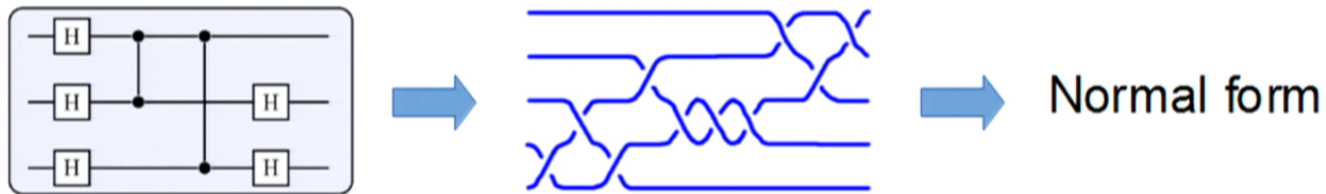
- Fluxes:



- Satisfies Yang-Baxter for any group
- Is classically universal for non-solvable groups

How about Quantum Circuits?

- **Use the Fibonacci representation:** [Freedman, Larsen, Wang]
 - $\rho : B_n \rightarrow U(F_{n-1})$
 - $\rho(\sigma_1), \dots, \rho(\sigma_{n-1})$ modulo phase generate dense subgroup of $SU(F_{n-1})$
 - Arbitrary quantum circuits can be built from them.
 - Efficiently? Yes, due to local structure (not exactly tensor product structure).
- **Obfuscate just like a classical circuit:**



Application: Testing Quantum Computers



This is a quantum computer.
You can buy it for \$10,000,000.



I don't trust you.
Let me look inside.



No way! It's proprietary.



Factor: 127849991827001928378165598299019311
298365026502637469299265655022363038110474
298734569230984261656483999938171717182221



I can't.
I only have 100 qubits.

Application: Testing Quantum Computers

- Bob needs a quantum computation that:
 - Can be done with 100 qubits
 - Can't be done classically
 - He can check the answer
- Proposal:
 - Express a simple classical computation using quantum gates
 - Obfuscate the quantum circuit and hand it to Alice

Can Alice Recognize Classicality?

- **Not generically.**
- **CLASS:** Given a quantum circuit implementing a unitary U that is ϵ -close to a polynomial-size reversible circuit, find the reversible circuit.
- **Theorem:** If $\text{CLASS} \in \text{FP}$ then $\text{QCMA} \in \text{P}^{\text{NP}}$

Proof: [Alagic, Jeffery, Jordan. ArXiv: 1212.6458]

Application: Testing Quantum Computers

- Bob needs a quantum computation that:
 - Can be done with 100 qubits
 - Can't be done classically
 - He can check the answer
- Proposal:
 - Express a simple classical computation using quantum gates
 - Obfuscate the quantum circuit and hand it to Alice

Can Alice Recognize Classicality?

- **Not generically.**
- **CLASS:** Given a quantum circuit implementing a unitary U that is ϵ -close to a polynomial-size reversible circuit, find the reversible circuit.
- **Theorem:** If $\text{CLASS} \in \text{FP}$ then $\text{QCMA} \in \text{P}^{\text{NP}}$

Proof: [[Alagic, Jeffery, Jordan. ArXiv: 1212.6458](#)]

Other Relations?

- *Partial* indistinguishability: R obeys additional identities besides the braid relations.
- For example, $R^{60} = \mathbb{1}$
- More relations give stronger obfuscation.
- If we add more relations can we still compute a normal form in polynomial time?



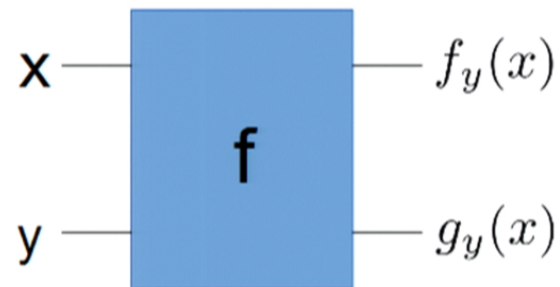
All the Relations?

Could we use all the relations of a reversible gate set?

- First glance: **no**, because deciding equivalence of logic circuits is coNP-complete.
- Second glance: **maybe**, because this would decide *strong* equivalence, including ancillas.
- Third glance: **no**, deciding strong equivalence is coNP-complete too.

Strong vs. Weak Equivalence

- Reversible circuits cannot erase information.



- Weak equivalence: f_0
- Strong equivalence: f_y, g_y

Strong equivalence is coNP-complete

- **Containment in coNP**: easy, if two circuits are inequivalent, an input on which they differ is a witness.
- **coNP-hardness**: might depend on gate set.
 - If gate-set G can be constructed from gate set F then hardness for G implies hardness for F .
 - Let's consider the weakest standard reversible gate set: {Fredkin}.

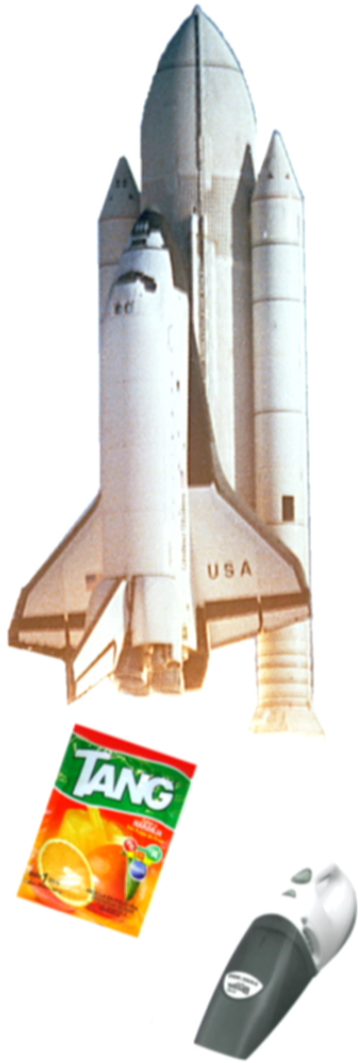


Proof Sketch

- Reduce from UNSAT to identity-check.
- Width-5 branching programs:
 - We have a 5-state dit.
 - We apply controlled-permutations on it, controlled by input bits.
 - Accept if resulting permutation is non-identity.
- **Barrington's theorem:** Using $O(4^d)$ controlled-permutations we can evaluate any depth-d Boolean formula.

Review

- New obfuscation definition: partial-indistinguishability obfuscation.
- Similar to local-replacement methods used in practice today, but the existence of a normal form adds qualitatively new feature.
- Constructed nontrivial polynomial-time example: express (quantum or classical) circuits as braids and reduce to normal form.
- Early stages: many open questions and side-avenues



Spinoffs!

- coNP-completeness of strong equivalence of reversible circuits
- Impossibility of efficient classicality testing
- All permutation matrices up to 25×25 satisfying Yang-Baxter
- No-go results for Yang-Baxter computation*
- Random reversible circuits are non-malleable one-time pads.

*[Alagic, Bapat, Jordan, 2013]

Open Problems

- Fuller sets of relations?
- What is the minimum dimension of a universal Yang-Baxter gate? (Quantum universality?)
- Can our obfuscation scheme be broken in practice?
- Does the no-go theorem for black-box obfuscation carry over to quantum circuits?
- Are those Yang-Baxter permutations good for something?
- Quantum obfuscation of quantum circuits?

Thanks to my collaborators



and thank you for your attention.