Analysis of Groups Generated by Quantum Gates

David C. Gajewski

University of Toledo

April 24, 2009

This is a Math Talk

- Studying computation as computation = Computer Science
- Studying computation as a form of abstract algebra = Math

Bits

- A **bit** is an element of $\mathbb{F}_2 = \{0, 1\}$
- A **binary string** is a string $b_{n-1} \cdots b_2 b_1 b_0$ where each b_i is a bit.

Logic can be encoded with 1 for True and 0 for False.
Numbers encoded as binary strings via their base 2 representation.

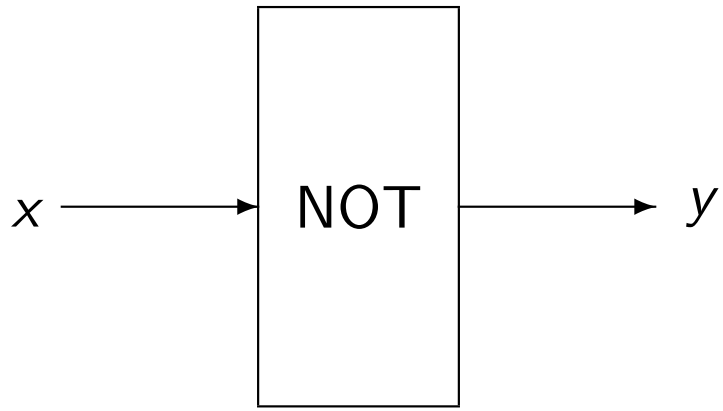
- $1 = 001$
- $2 = 010$
- $5 = 101$
- $7 = 111$

Binary Computers

A Binary Computer is a machine which acts on binary strings of a given length.

- These operations are called gates.
- Gates take some number of input bits and create a number of output bits.

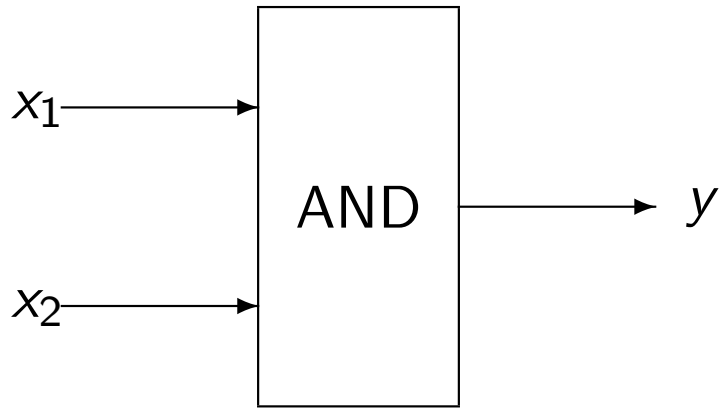
1-Bit Gate Example - Logical NOT



$$y = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x = 1 \end{cases}$$

$$y = 1 + x \pmod 2$$

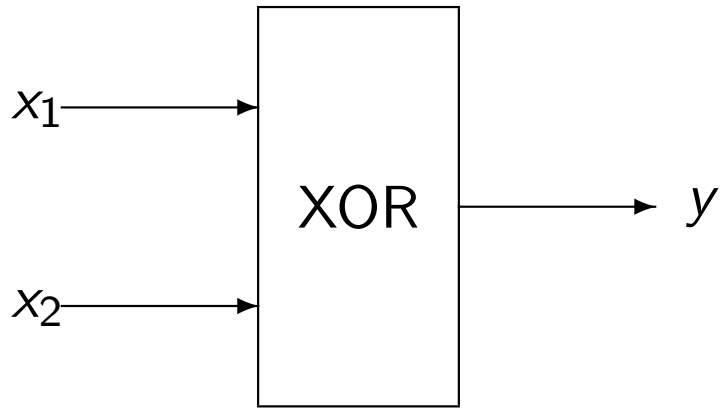
2-Bit Gate Example - Logical AND



$$y = \begin{cases} 1 & \text{if } x_1 = x_2 = 1, \\ 0 & \text{otherwise} \end{cases}$$

$$y = x_1 \cdot x_2 \pmod 2$$

2-Bit Gate Example - Logical XOR



$$y = \begin{cases} 0 & \text{if } x_1 = x_2, \\ 1 & \text{if } x_1 \neq x_2 \end{cases}$$

$$y = x_1 + x_2 \pmod 2$$

Reversible Computing

- Gates with less outputs than inputs destroy bits, requiring energy and creating heat.
- Gates with more outputs than inputs creates bits, requiring energy.

So # of inputs = # of outputs is best.

Reversible Computing Continued

- Additional output bits can be made by copying over some inputs.
- Additional input bits can be added which are fixed as 0 or 1

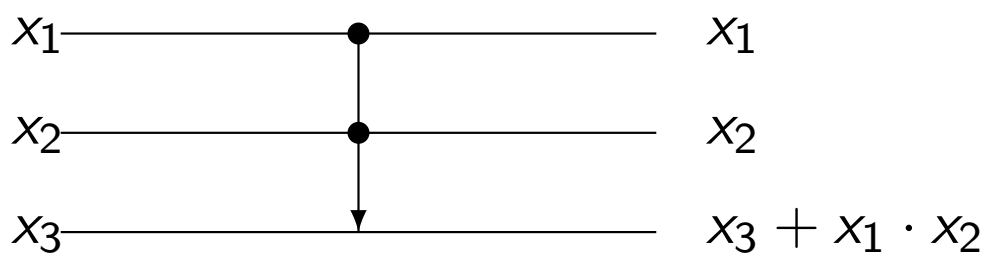
Such a gate can be made to be a permutation on binary strings of a certain length, and thus is reversible/invertible.

Universal Gate

A Universal Gate is a gate which can simulate any other gate by way of multiple applications across different wires.

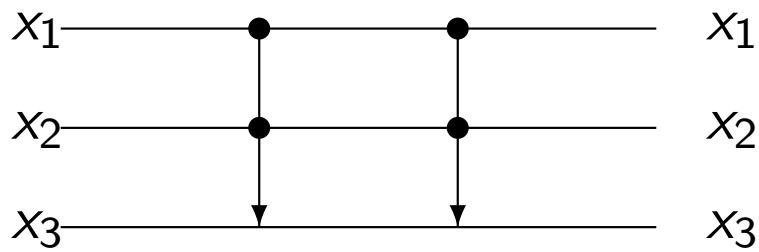
There are no universal 1-bit or 2-bit gates.

The 3-bit Universal Gate: Toffoli



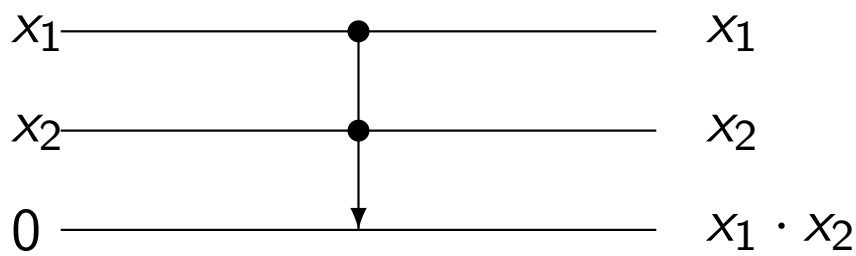
- x_1 and x_2 are called the control bits and x_3 the target bit.
- x_1 and x_2 are left alone by the operation.
- x_3 becomes $x_3 + x_1 \cdot x_2 \pmod 2$.
- Also called Controlled-Controlled NOT as x_3 only changes if both x_1 and x_2 have value 1.

The Toffoli Gate is Reversible



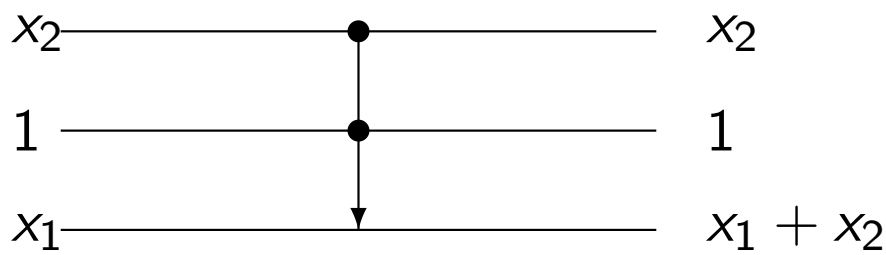
$$
\begin{aligned}
x_3 &\longrightarrow x_3 + x_1 \cdot x_2 \pmod 2 \\
&\longrightarrow (x_3 + x_1 \cdot x_2) + x_1 \cdot x_2 \pmod 2 \\
&= x_3 \pmod 2
\end{aligned}
$$

Examples of Universality
Simulating Logical AND



Last bit after operation: $0 + x_1 \cdot x_2$
Same result as AND.

Examples of Universality
Simulating Logical XOR



Last bit after operation: $x_1 + 1 \cdot x_2$
Same result as XOR.

The Toffoli Group

The Toffoli gate acting on all binary strings of length n by all possible combinations of 3 wires generates the Toffoli group T_n.

Call the operation of the Toffoli gate using bits i and j as control with bit k as target $t_{i,j}^k$.

So $T_n = \langle t_{i,j}^k \mid 0 \leq i, j, k < n, \text{ and distinct} \rangle$.

Toffoli Group Continued

Definition: The **bit weight** of a binary string is the number of 1 bits it contains.

Theorem $T_3 = \mathrm{Sym}(\{3, 5, 6, 7\}) = \mathrm{Sym}(\{011, 101, 110, 111\})$. For $n > 3$, T_n is the alternating group on all binary strings of bit weight ≥ 2.

Toffoli Group Continued

Proof for $n = 3$

- $t^2_{0,1}$ takes $011 \longrightarrow 111$ and $111 \longrightarrow 011$ only.
- $t^1_{0,2}$ takes $101 \longrightarrow 111$ and $111 \longrightarrow 101$ only.
- $t^0_{1,2}$ takes $110 \longrightarrow 111$ and $111 \longrightarrow 110$ only.

Giving

$t^2_{0,1} = t^2_{1,0} = (3\ 7), \qquad t^1_{0,2} = t^1_{2,0} = (5\ 7), \qquad t^0_{1,2} = t^0_{2,1} = (6\ 7)$

$$\begin{aligned} T_3 &= \langle t^2_{0,1}, t^1_{0,2}, t^0_{1,2} \rangle \\ &= \langle (3\ 7), (5\ 7), (6\ 7) \rangle \\ &= \mathrm{Sym}(\{3, 5, 6, 7\}) \end{aligned}$$

Toffoli Group Continued

For $n > 3$

- Use induction on n, the number of wires.
- On $n+1$ wires apply T_n in different ways, ignore one wire.
- Any extension of a permutation gives two copies - one for the ignored bit being 0, another for it being 1.

Hence why T_n is alternating for $n \geq 4$.

Toffoli Group Continued

Examples of extending $011 \longleftrightarrow 111$

- $\_011 \longleftrightarrow \_111$
- Giving $0011 \longleftrightarrow 0111$ and $1011 \longleftrightarrow 1111$
- This is the permutation (3 7)(11 15)

- $01\_1 \longleftrightarrow 11\_1$
- Giving $0101 \longleftrightarrow 1101$ and $0111 \longleftrightarrow 1111$
- This is the permutation (5 13)(7 15)

Computation in General

Examples of computation alongside Binary Computers

- Turing Machines
- λ-Calculus
- Automata

Church-Turing Thesis

> *Any 'reasonable' model of computation can be efficiently simulated by any other model.*

Quantum Computing

A Quantum Computer is an computational object which makes use of systems at atomic scales and smaller.

History of Quantum Computing

- 1982 - Richard Feynman - Introduced Quantum Computing
- 1992 - David Deutsch - First non-classical quantum algorithm
- 1994 - Peter Schor - Polynomial time factoring algorithm
- 1996 - L. K. Grover - Sub-linear time search algorithm

A possible counter-example to the Church-Turing Thesis?

Quantum Computing Continued

Instead of bits, we have quantum bits, or **q-bits**.

- 010 as a binary string is written $|010\rangle$ in Dirac notation.
- Also can be written e_{010}

Quantum Gates

- On n q-bits they act on $\mathbb{C} \otimes \cdots \otimes \mathbb{C}$ (n times)
- Contains the Toffoli gate.
- Contains more interesting gates!

Quantum Gate: Hadamard

The Hadamard Gate (h) is a 1 q-bit quantum gate.

$$h(|0\rangle) = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$h(|1\rangle) = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

This does not give a permutation of q-bit strings, but a linear combination!

Quantum States

A quantum state is a linear combination of q-bit strings.

$$\sum \alpha_i e_i, \qquad \sum |\alpha_i|^2 = 1$$

- Quantum Gates act as linear operators.
- This is quantum entanglement!

Quantum States Continued

Observing a quantum state destroys it.

- $\sum \alpha_i e_i$ is observed as e_i with probability $|\alpha_i|^2$.

Example: $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ is observed as

- $|0\rangle$ 50% of the time
- $|1\rangle$ 50% of the time

The result of an algorithm may have all q-bit strings entangled.

The goal is that the correct answer is observed with a high degree of probability.

Hadamard Continued

Linear representation of the Hadamard gate for 1 q-bit over the basis $\{e_0, e_1\}$

$$h = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Hadamard Continued

2 q-bits over the basis $\{e_{00}, e_{01}, e_{10}, e_{11}\}$, acting on the first q-bit and second q-bit.

$$h \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes h = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$

Observations

Observation 1:

- If σ is a permutation matrix, then $(h \otimes I_{2^{n-1} \times 2^{n-1}})^{\sigma}$ is a change of basis.
- This gives a permutation of the basis into pairs $\{f_{i0}, f_{i1}\}_i$ on which the standard Hadamard acts like $h \otimes I_{2^{n-1} \times 2^{n-1}}$.

So

$$h \otimes I\left(f_{i0}\right) = \frac{1}{\sqrt{2}} f_{i0} + \frac{1}{\sqrt{2}} f_{i1}$$
$$h \otimes I\left(f_{i1}\right) = \frac{1}{\sqrt{2}} f_{i0} - \frac{1}{\sqrt{2}} f_{i1}$$

Call $\{f_{i0}, f_{i1}\}_i$ the pairing under σ.

Column Vectors

Consider column vectors with entries α_i in $\mathbb{Z}[1/\sqrt{2}]$. With $\sum \alpha_i^2 = 1$ (like a quantum state).

There exists $k \geq 0$ such that $b_i = \alpha_i \sqrt{2}^k$ are integers for all i and with some b_i odd.

This k is called the **column weight**.

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n} \end{pmatrix} = \frac{1}{\sqrt{2}^k} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{2^n} \end{pmatrix}$$

More Observations

Observation 2

- $\sum \alpha_i^2 = 1$ implies $\sum b_i^2 = 2^k$.
- Taken modulo 2 this shows that
  - an even number of the b_i's are odd
  - an even number of the b_i's are even

Observation 3

- even + even = even, even - even = even
- odd + odd = even, odd - odd = even

More Observations

Observation 4

- We can find a permutation σ which makes pairs of
  - the even number of even b_i's
  - the even number of odd b_i's

Operating on a column of weight k

$$\begin{aligned}(h \otimes I)^{\sigma}(f_{i0}) &= \frac{1}{\sqrt{2}} f_{i0} + \frac{1}{\sqrt{2}} f_{i1} \\ &= \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}^{k}} \cdot (\text{even number}) \\ &= \frac{1}{\sqrt{2}^{k+1}} \cdot 2 \cdot (\text{a number}) \\ &= \frac{1}{\sqrt{2}^{k-1}} \cdot (\text{a number})\end{aligned}$$

This reduces the column weight!

Column Reducing Example

$$\frac{1}{\sqrt{2}^2}\begin{pmatrix} -1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Pair 1^{st} with 2^{nd}, 3^{rd} with 4^{th}

$$(h \otimes I)^{\sigma} \cdot \frac{1}{\sqrt{2}^2}\begin{pmatrix} -1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}^3}\begin{pmatrix} 0 \\ -2 \\ 2 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix}$$

Pair 1^{st} with 4^{th}, 2^{nd} with 3^{rd}

$$(h \otimes I)^{\sigma} \cdot \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}^2}\begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Main Theorem

Group of Quantum Computing is generated by Hadamard and Toffoli gates.

Let P_{2n} be all permutation matrices of alternating type (think Toffoli T_{2n})

Theorem $\mathrm{SO}_{2n}(\mathbb{Z}[1/2]) \cdot \langle h \rangle = \langle (h \otimes I)^{\sigma} \,|\sigma \in P_{2n} \rangle$

- Induction on $2n$.
- Reduce first column.
- Reduce second column, respecting the first column.
- A 2×2 identity matrix appears.
- Keep the pairing giving the $I_{2\times 2}$ and reduce the remaining $(2n-2) \times (2n-2)$ matrix

Example

$$\frac{1}{\sqrt{2}^4}\begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 3 & -1 & 1 & -1 \\ 0 & 0 & 1 & 3 & -1 & 0 & -1 & 2 \\ 0 & 0 & 2 & 0 & 1 & 3 & -1 & -1 \\ 0 & 0 & 1 & 1 & -2 & -1 & 0 & -3 \\ 0 & 0 & 1 & -1 & 1 & -2 & -3 & 0 \\ 0 & 0 & 3 & -1 & 0 & -1 & 2 & 1 \end{pmatrix}$$

Example

$$\frac{1}{\sqrt{2}^3}\begin{pmatrix} -2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -2 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 & 0 & 1 & 2 & -1 \\ 0 & 0 & 1 & -1 & -1 & 2 & -1 & 0 \\ 0 & 0 & -2 & 0 & 1 & 1 & -1 & 1 \\ 0 & 0 & 0 & -2 & 1 & -1 & -1 & -1 \\ 0 & 0 & 1 & -1 & 1 & 0 & 1 & 2 \end{pmatrix}$$

Example

$$\frac{1}{\sqrt{2}^2}\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 \\ 0 & 0 & -1 & 1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 \end{pmatrix}$$

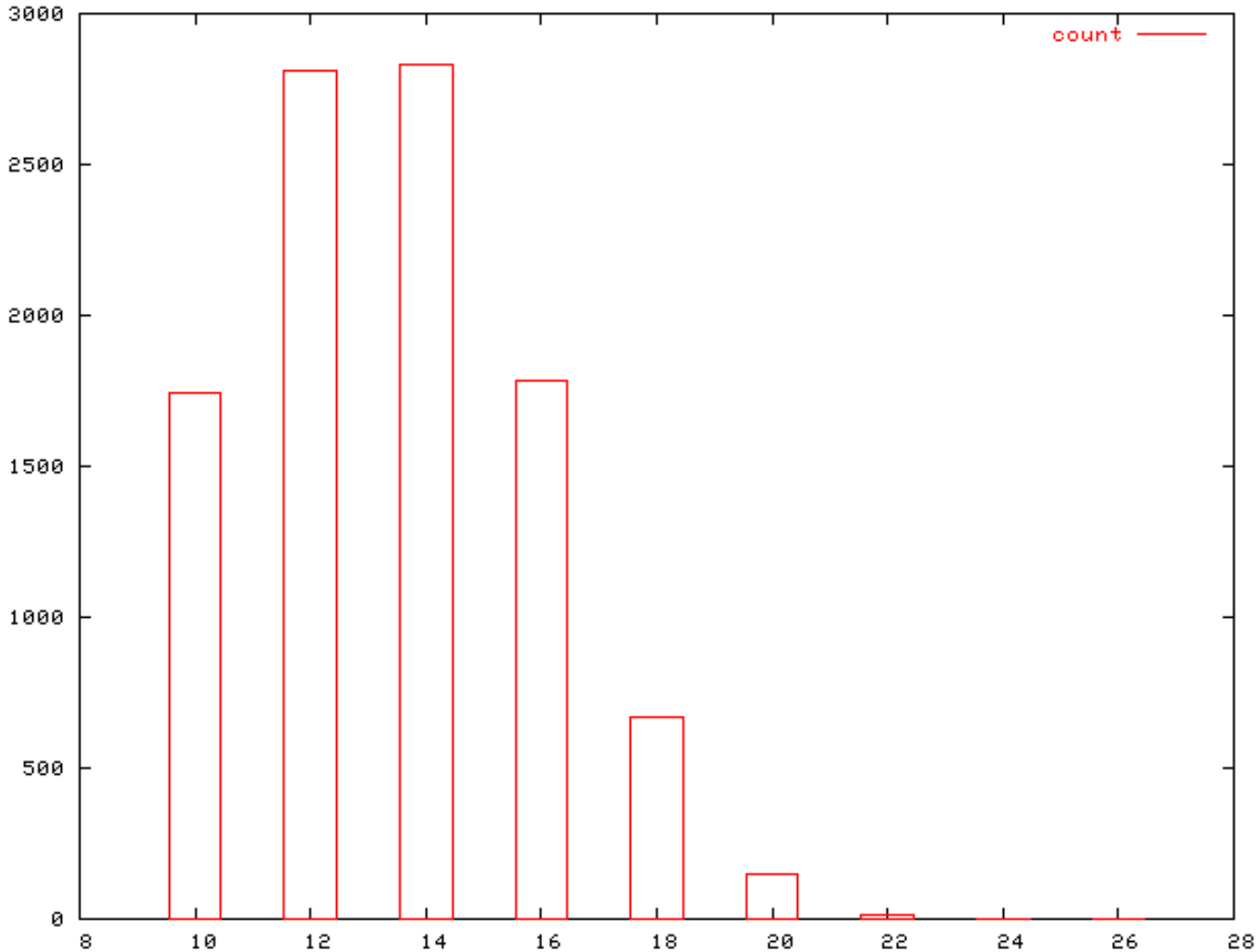
Example

$$\frac{1}{\sqrt{2}}\begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

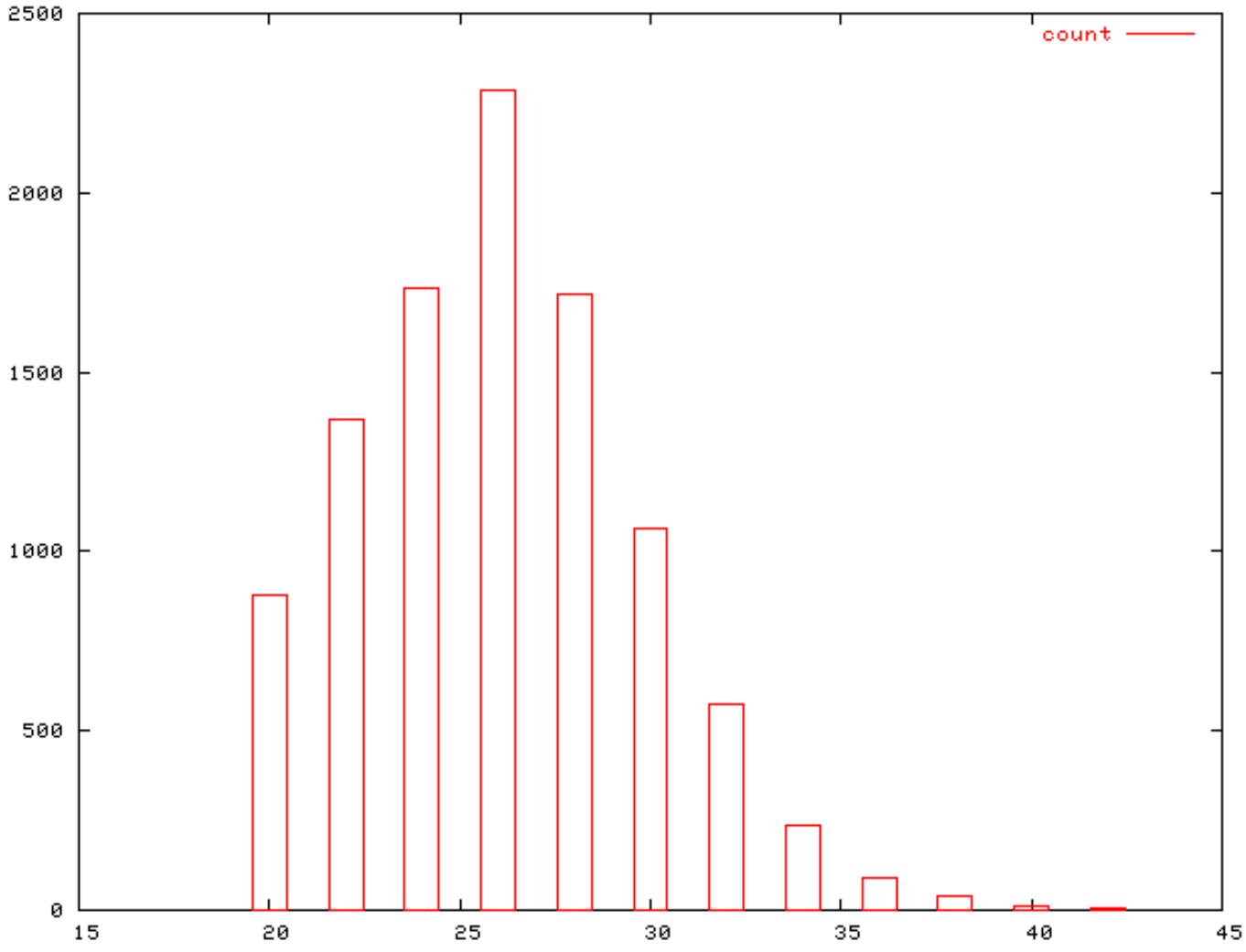
Example

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

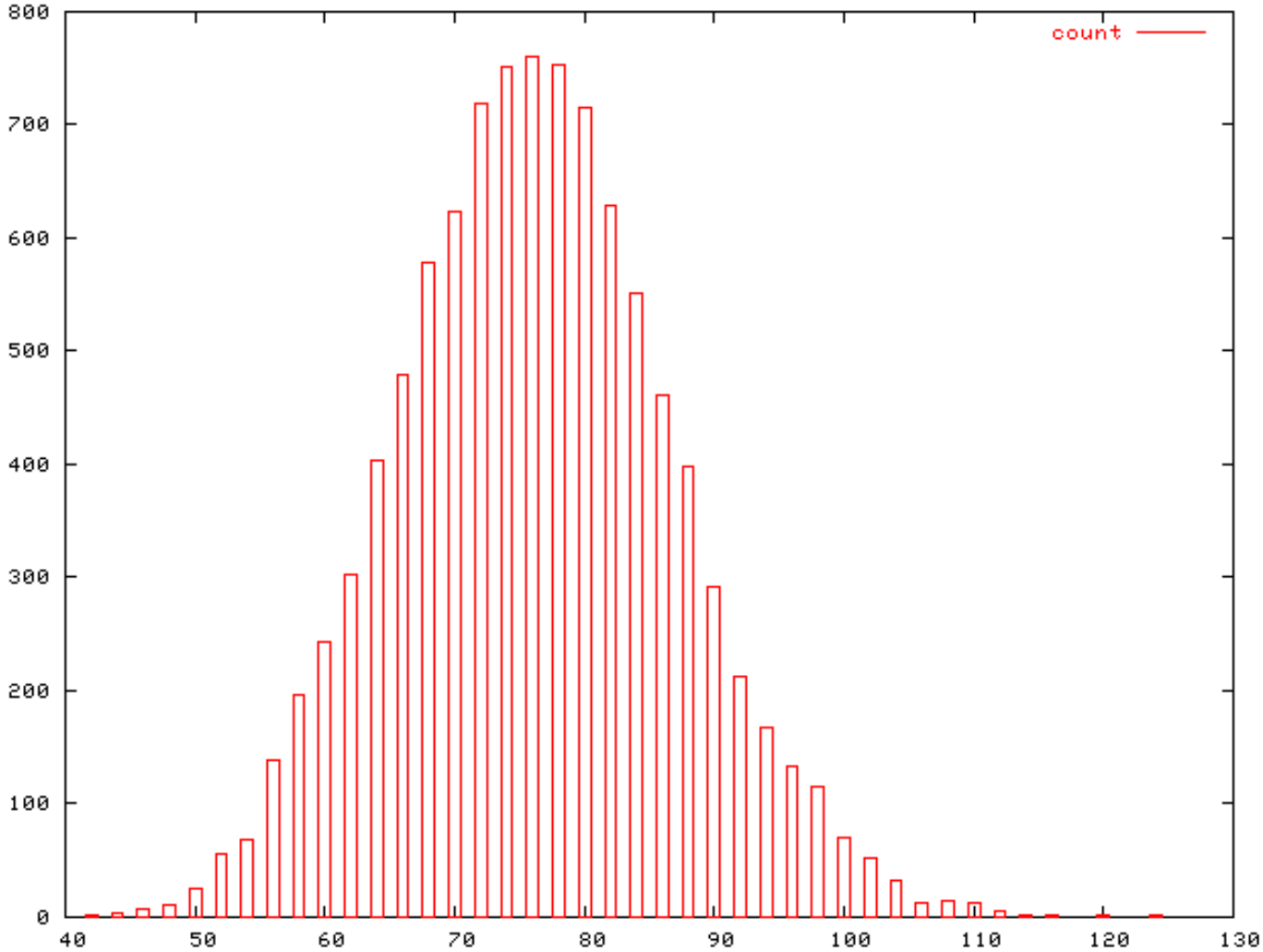
Decomposition Length of 10,000 8×8 Matrices of Weight 10



Decomposition Length of 10,000 8×8 Matrices of Weight 20



Decomposition Length of 10,000 16 × 16 Matrices of Weight 10



Fin

This concludes the talk.

Thank you for your time.

Any questions?