Algebraic Turing Machines with Applications to Quantum Computation

Jonathan A. Poritz

jonathan.poritz@gmail.com
www.poritz.net/jonathan

Department of Mathematics & Physics
Colorado State University, Pueblo
2200 Bonforte Blvd.
Pueblo, CO 81001-4901

University of Colorado, Colorado Springs
Rings & Wings Seminar
15 November 2017

The *Entscheidungsproblem* *[="decision problem"]*

David Hilbert, 1928:
Does there exist an algorithm which on input of a statement in first order logic outputs *Yes* if the statement is universally valid and *No* otherwise?

But, what is this "**algorithm**?"

Alonzo Church, 1936:
Algorithms are defined using "the λ-calculus."
And, Herr Doktor Professor Hilbert: **Nope**.

Alan Turing [Church's Ph.D. student!], 1936:
Algorithms are defined using "Turing machines."
And, Herr Doktor Professor Hilbert: **Nope**.

Turing Machines: Formally, Set-up

A **Turing machine** [**TM**] consists of two sets and a map:

- a finite set Σ (the **alphabet** of symbols to be written on the tape);
- a finite set $\mathcal{S}$ (of **states** of the control device), which has two special elements $\boxed{START}, \boxed{HALT} \in \mathcal{S}$ (in which the state machine starts, and which indicates the algorithm is finished); and
- a map $T : \mathcal{S} \times \Sigma \to \mathcal{S} \times \Sigma \times \{-1, +1\}$ (the **state machine transition function**).

[It suffices to work with the alphabet $\Sigma = \{0, 1\}$, and we shall always do so.]
A **tape** is a map $\tau : \mathbb{Z} \to \Sigma$ (think of it as an infinite tape written with symbols from Σ); denote the set of such tapes as $\mathcal{T}_\Sigma$.
Given a TM $\{\Sigma, \mathcal{S}, T\}$ and some tape τ (the **input tape**), we set

- the **current state** $s \in \mathcal{S}$ to be $s = \boxed{START}$, and
- the **read/write head location** $j \in \mathbb{Z}$ to be $j = 0$.

Turing Machines: Formally, Processing

Repeat the following steps until $s = \boxed{HALT}$:

- say $T(s, \tau(j)) = (s', \sigma, m)$:
- then change the **current state** to the new value s' and
- change the **tape** to a new one τ' given by

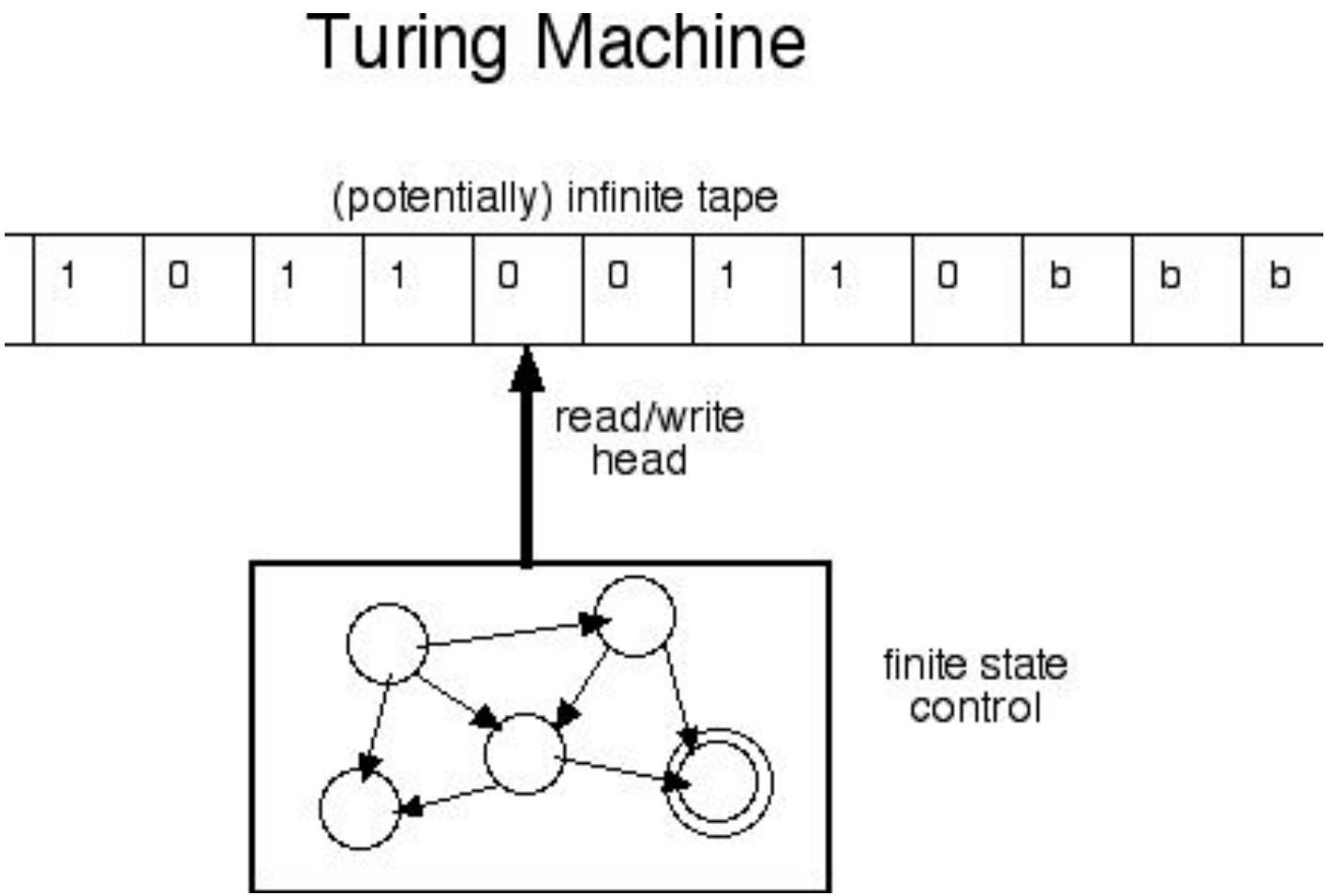
$$\tau'(i) = \begin{cases} \tau(i) & \text{if } i \neq j \\ \sigma & \text{if } i = j \end{cases}$$

  and

- change the **head location** to the new value $j + m$.

Whatever is the value of the tape when [actually, *if*] the TM halts is the **output tape**.

Turing Machines: The Picture



Turing Machines: Universality

A **Universal Turing Machine** [**UTM**] is a TM U and two maps

- an **encoding map** e which takes as input any TM M and any tape τ and produces a new tape $e(M, \tau) \in \mathcal{T}_\Sigma$ and
- a **decoding map** $d : \mathcal{T}_\Sigma \to \mathcal{T}_\Sigma$

such that

- for any TM M and input tape τ
- M halts on input τ with output τ'

  if and only if
- U halts on input $e(M, \tau)$ with an output tape τ'' satisfying $\tau' = d(\tau'')$.

Turing: UTMs exist!

Turing Machines: Use in Complexity Theory

TMs are very good at describing the *cost* of computation:

- How much of the tape is used during a computation – *i.e.*, the largest value of the current head location j – is a measure of the *space complexity* of the computation.
- How many steps the read/write had to take – *i.e.*, how many times through the main loop of the "repeat the following..." above – is a measure of the *time complexity* of the computation.

If we have a computational problem which can have instances of different sizes and there is some bound on the time complexity of a TM which solves that problem then we would say the problem is solvable with that bound.

We often talk about *polynomial-time* problems, and the set $\mathcal{P}$ of such.

E.g., Multiplying integers is in $\mathcal{P}$, but it is not know if factoring is. The gap between these two is what makes most of public-key crypto work!

Turing Machines: Now With Randomness

We want to talk about TMs with access to *randomness*, formalized by

- replacing the above deterministic description with one where the steps in the main loop are thought of as drawing according to specified probability distributions; or
- giving a deterministic TM access to a second tape full of random bits, and drawing conclusions about distributions of results based on the distributions of the input randomness.

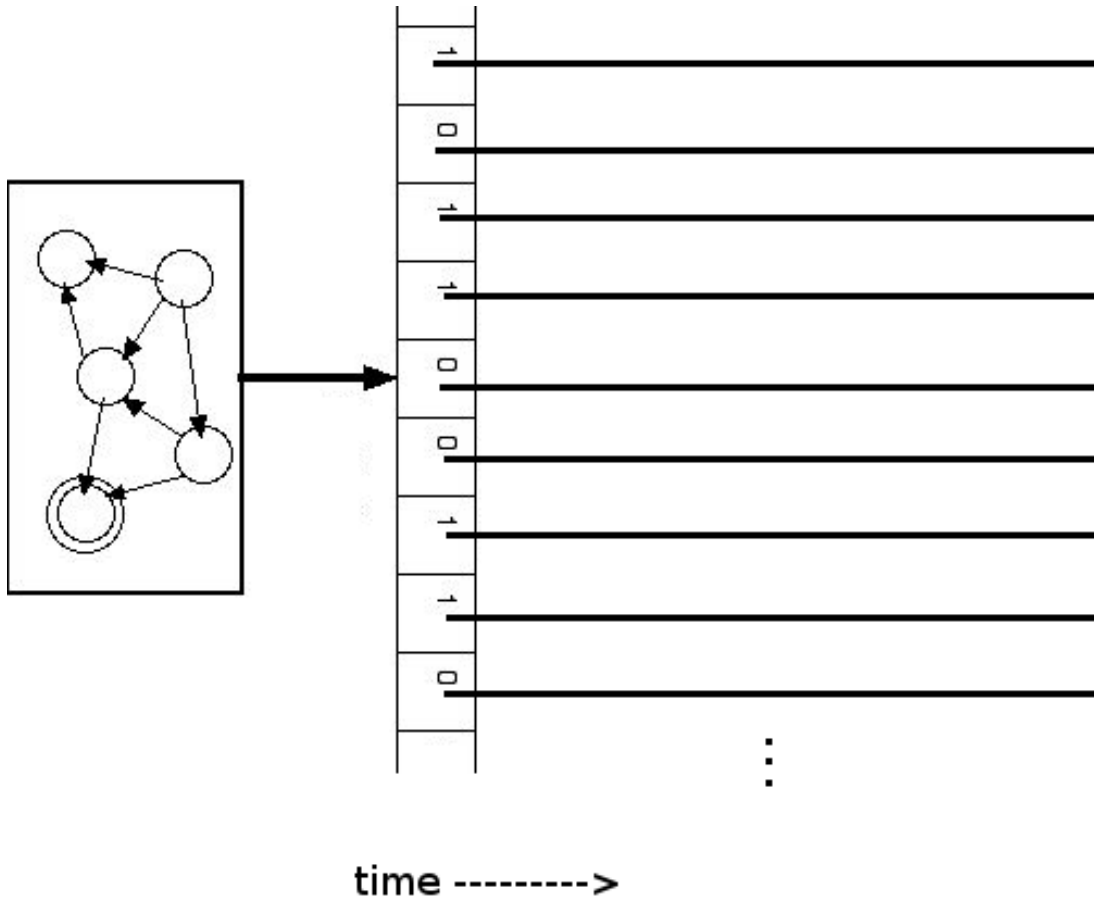
We shall not go into the details for these randomized TMs.

Strangely, randomness seems to make TMs **more** powerful! [Below, we shall see a hint of why that might be the case.]

For this reason, one usually considers that all actors in a cryptographic protocol have access to *probabilistic, polynomial-time TMs*, [PPTMs]. Most public-key crypto today is based on the fact that there are PPTMs for multiplication but none is known for factoring. Hence multiplication is a good candidate for a *cryptographic one-way function*.

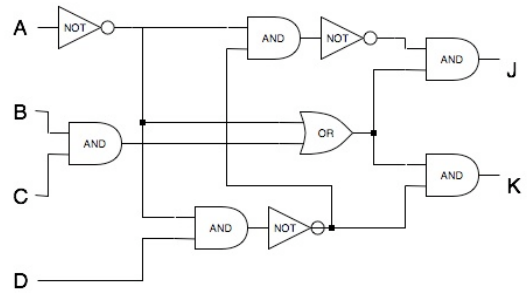
Circuits: Starting with Turing Machines

TMs, while good for the *Entscheidungsproblem* and complexity theory, are not so practical to build in the real world. We would rather make circuits. So imagine:



Circuits: Boolean Basics

These are similar to *[OK, "vaguely reminiscent of"]* Boolean circuits, *i.e.*, things that look like:



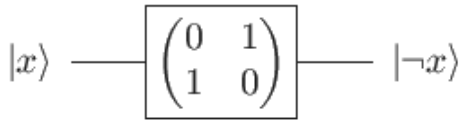
Boolean circuits are built out of wires which carry either a 0 or 1 *[like the cells on a TM tape, stretched out in time!]*, the *logic gates* NOT, AND, and OR and, implicitly, the *utility gates* FANOUT and SWAP.

There is a kind of universality for Boolean circuits: while we thought, from simple logic, that we needed all of the logic gates, it turns out instead that **NAND** [=NOT∘AND] **is universal** in the sense that, given any Boolean circuit, you can build an equivalent circuit which only uses the logic gate NAND.

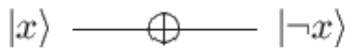
Sometimes it is convenient to add extra wires which do not carry parts of the input data but are instead pre-loaded always with 0s, and whose values at the end are not counted as part of the output. These are called **ancilla**.

Circuits: We Want More Algebra!

Let's make our circuit diagrams look more like the streched-out-TM picture from before. And with More Algebra™.
All wires will go straight across our entire diagram and never simply appear or disappear. The wires will be viewed as carrying a vector in some fixed vector space over a field k – we will usually use $k = \mathbb{C}$, but leave open the possibility that could change – with basis $|0\rangle$ and $|1\rangle$. *[yes, weird notation: blame Dirac!]*
Vectors in $\mathbb{C}|0\rangle \oplus \mathbb{C}|1\rangle$ are called **qubits**. *[for reasons which shall emerge eventually]*
Here is a "one-qubit gate" to do the basic logic operation NOT:

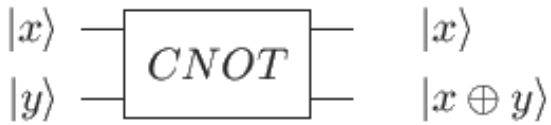


Actually, this NOT usualy gets the special notation

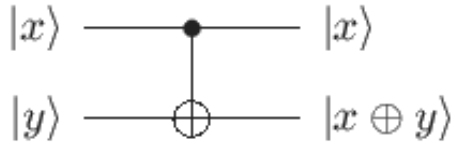


Circuits: Controlled Gates, CNOT

Often we use the idea of a **controlled gate**, which passes through some of the qubites without change and modifies others if the control lines are all 1s. For example, the **controlled-not** or **CNOT** is the gate



[that "$\oplus$" is XOR to computer scientists and addition mod 2 to mathmeaticians] with special notation



,

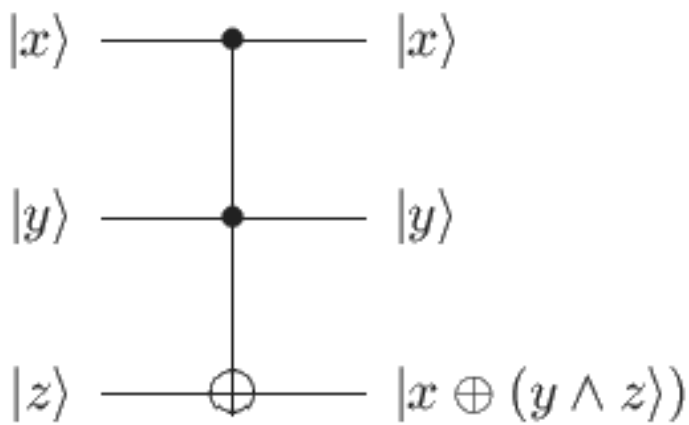
where

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} .$$

Circuits: Controlled Gates, TOFFOLI

There is a famous *doubly controlled gate*, called **TOFFOLI** with matrix, corresponding special notation, and logical action:

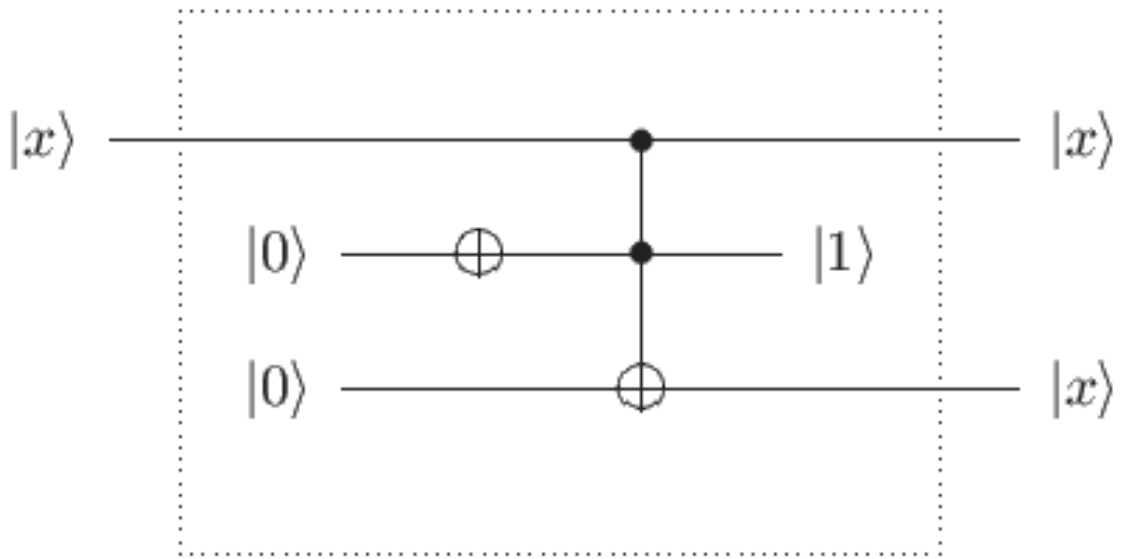
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$



Note that TOFFOLI is *invertible*, so by using it (and NOT or CNOT, which are also invertible), we are doing **reversible computation**. There are reasons, the physicists tell us, that that is a very good thing to do. TOFFOLI can be used with ancilla to do *all of the other logical gates*, so it is, in yet another sense, **universal for reversible computation**.

Circuits: Digression 1, Universality of TOFFOLI

This is fun to work out. Here is one of the more surprising ones, FANOUT:



Exercise: Figure out how to use TOFFOLI to do AND, OR, *etc.*

Circuits: Where Do Those Vectors Live? and Input

A vertical (time) slice of a circuit diagram should have a particular qubit vector on each "wire," but one may ask: where does the whole thing live?

Probably, it seems natural to put the separate vectors together in the direct sum (Cartesian product). But, instead, we shall use the **tensor prodcut**. The **state vector** for any time slice of our diagram will thus live in $\mathbb{C}^{2^n}$ if we are working with n wires, the n-**qubit space**.

We shall call the basis

$$\begin{aligned} |\mathbf{0}\rangle = |0\ldots0\rangle &= |0\rangle \otimes \cdots \otimes |0\rangle \\ |\mathbf{1}\rangle = |0\ldots1\rangle &= |0\rangle \otimes \cdots \otimes |1\rangle \\ &\vdots \\ |\mathbf{2^n-1}\rangle = |1\ldots1\rangle &= |1\rangle \otimes \cdots \otimes |1\rangle \end{aligned}$$

the **computational basis**, and we use it to set up our input of n binary digits.

Circuits: Where Do Those Vectors Live? and Output

Now, *assume* that the n-qubit vector at the output side of a vector is a unit vector in $\mathbb{C}^{2^n}$, so it can be written as

$$\sum_{j=0}^{2^n-1} a_j |\mathbf{j}\rangle,$$

where each $a_j \in \mathbb{C}$ and

$$\sum_{j=0}^{2^n-1} |a_j| = 1 \ .$$

Then we shall say that our circuit yields the n-bit output string which is the binary expression of the number k with probability $|a_k|$, and the above constraint on coefficients implies this is a well-defined probability distribution on the set of 2^n possible n-bit strings. This whole process is called **measurement with respect to the computational basis**.

Circuits: No Probability [Yet]

Note that the gates we have seen are all permutation matrices. Inputs, as we have done them, always make us start with a basis vector. Therefore, computation built out of the these gates will always yield a basis vector at the end of the computation.

This corresponds to an output distribution which gives one bit string with probability 1 – so it is deterministic.

In a certain sense, we are saying that the *structure group* of classical (reversible ... but, with ancilla and fan-out that doesn't change things much) computation is the permutation group S_{2^n}. The circuits we have described are ways of building all of the elements of S_{2^n} by products of matrices which are tensors of some size identity on the left and the right and one of a small set of convenient gates/permutation matrices.

Circuits: Unitary Universalist

The structure group of the real world is the unitary group – because of **quantum mechanics**. So it would make sense to imagine special gates that we like to build (analoguously to the special permutation matrices we looked at above) which are now *unitary matrices*. Then since products and tensor products of unitary matrices are unitary, the entire computation done by a circuit will be a matrix in $U(2^n)$.
The way we built inputs, our circuits always start with unit vectors.
Therefore, our final output vector will also be a unit vector, and we can do measurement with respect to the computational basis.
The computation will be *probabilistic*, however.
We also must imagine that our laboratories can only produce a finite number of specific basic gates. Therefore there is no way the products and tensor products with identities will yield all of $U(2^n)$. Instead, in this context, we say that a set of basic gates of **universal for quantum computation** if it generates a dense subset of $U(2^n)$.

Circuits: BvN Machines

Actually, we haven't used inverses anywhere. So we could explore in a different direction which doesn't go from permutation matrices to unitary matrices, but instead goes to *doubly stochastic matrices*. Recall that, by the **Birkhoff-von Neumann Theorem**, the convext hull in $\mathbb{R}^{n^2}$ of the set of permutation matrices is the set of doubly stochastic matrices.

Circuits: Digression 2, Physics – Landauer's Principle

Rolf Landauer (IBM) 1961 says, version by Charlie Bennett (IBM):

> *Any logically irreversible manipulation of information, such as the erasure of a bit or the merging of two computation paths, must be accompanied by a corresponding entropy increase in non-information bearing degrees of freedom of the information processing apparatus or its environment.*

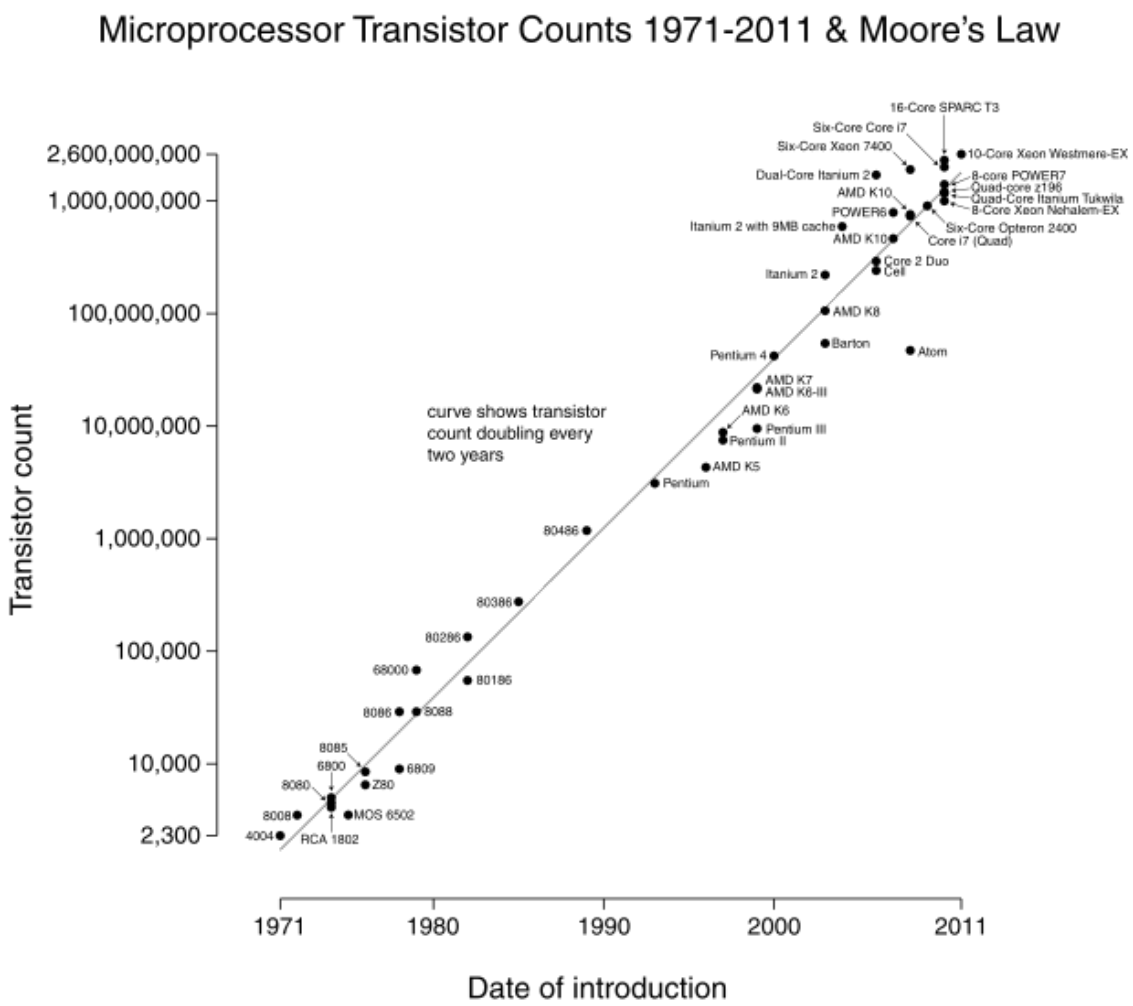
Hence the minimum amount of enegergy to change (erase, *etc.*) one bit is

$$kT \ln 2 \ ,$$

where k is Boltzmann's constant and T is the temperature.
Moving around bits *without erasure* can (in theory) be done **with zero energy cost**.

Circuits: Digression 2, Physics – Moore's Law 1

Described by Gordon Moore, on of Intel's co-founders, in 1965.



Circuits: Digression 2, Physics – Nanoscale Problems

$\rightsquigarrow$ A Nanometer is a billionth of a meter, $10^{-9}m$.

$\rightsquigarrow$ How small is that?

- $\rightharpoonup$ A human red blood cell is about $7000nm$.
- $\rightharpoonup$ An Ebola virus is about $1500nm$ long and $50nm$ wide.
- $\rightharpoonup$ In a silicon wafer, silicon atoms line up about one every half nm.
- $\rightharpoonup$ Hydrogen atoms are usually said to have a diameter of about an angstrom. [$1\text{Å} = 1/10nm = 10^{-10}m$.]

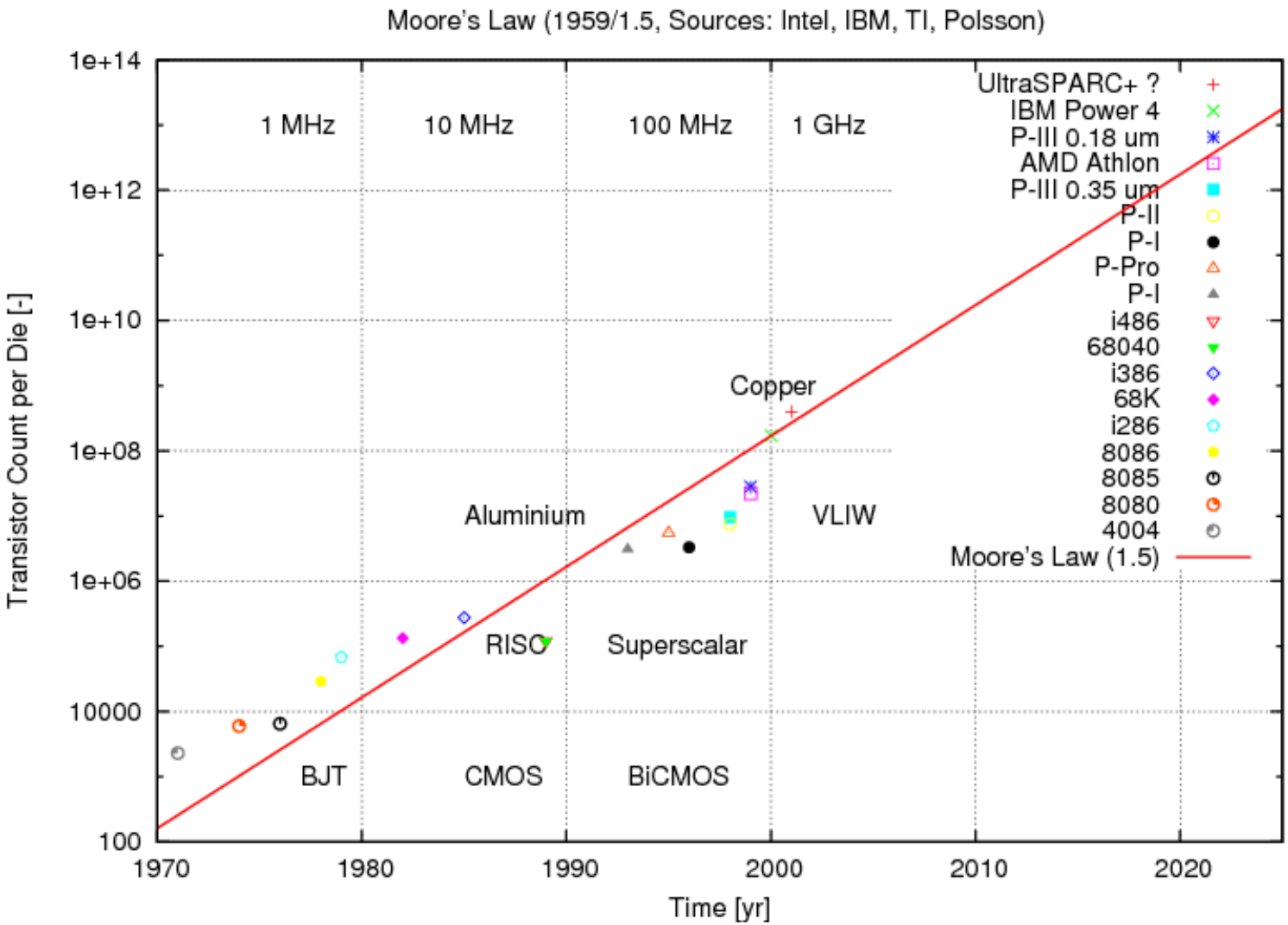
$\rightsquigarrow$ *Feature sizes* of integrated circuits this millennium are described as $130nm$, $90nm$, $65nm$, $45nm$, $32nm$, and $22nm$, the current best.

$\rightsquigarrow$ Current best means the gate length [basically, how wide is the wire] is around $20nm$, *e.g.,* for a NAND gate.

But, at around those sizes, dopants don't dope, insulators don't insulate, conductors have significant resistance and inductance....

[Numbers on this page may be off by a factor of two or so.]

Circuits: Digression 2, Physics – Moore's Law 2



Quantum Mechanics: Postulate I

Postulate I: the State Space

The state of an isolated quantum system is described by a unit vector (these are written $|\phi\rangle$) in a Hilbert space, the *state space* of that system.

$\rightsquigarrow$ Hilbert spaces are complete, complex vector spaces.

$\rightsquigarrow$ The linear structure [adding vectors] is odd: in classical mechanics, state spaces are curved spaces (usually the cotangent bundles of some space of configurations).

$\rightsquigarrow$ The complex structure – that vectors can be multiplied by complex numbers $a + bi$ – is odd: how should it make sense to multiply by $\sqrt{-1}$ in a real physical system.

$\rightsquigarrow$ In applications, the Hilbert spaces are often infinite-dimensional (and required to be separable), such as the L^2-completions of spaces of (twice-differentiable) functions, representing the "probability density of a particle."

A Nice State Space

Take a *two-state* quantum system, such as

- $\rightsquigarrow$ a single photon, which can be *vertically polarized* or *horizontally polarized*
- $\rightsquigarrow$ a single electron, which can be *spin up* or *spin down*
- $\rightsquigarrow$ an ion suspended in an electromagnatic field, cooled to near absolute zero, which we try to keep in one of its two lowest energy states
- $\rightsquigarrow$ a cat in a box, about which some maniac (Erwin Schrödinger) only cares whether it is *alive* or *dead*

and call the states $|0\rangle$ and $|1\rangle$.
The state space is then the *one-qubit space* $\mathcal{H} = \mathbb{C}^2 = \mathbb{C}|0\rangle \oplus \mathbb{C}|1\rangle$ and state vectors will be $\alpha|0\rangle + \beta|1\rangle$ where $\alpha, \beta \in \mathbb{C}$ satisfy $|\alpha|^2 + |\beta|^2 = 1$.

A gruesome example is $\frac{1}{\sqrt{2}}|\mathit{alive}\rangle + \frac{1}{\sqrt{2}}|\mathit{dead}\rangle$ *for Schrödinger's cat.*

Quantum Mechanics: Postulate II

Postulate II: Unitary Time Evolution

The time evolution of an isolated system is given by a *unitary operator* acting on the state vectors, $|\phi\rangle \mapsto U(|\phi\rangle)$

The unitary operator is determined by the specific physics of the situation. *E.g,* sometimes one figures out the Hamiltonian H of the system (a Hermitian operator) and time evolution is the unitary operator $U = e^{itH}$. **Or**, in the case of a 1-qubit system, we might look for physical processes which can act as some desired 2×2 unitary matrix, such as

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix};$$

X, Y, and Z are the *Pauli matrices* and H is the *Hadamard gate*

Quantum Mechanics: Postulate III

Postulate III: Measurement

Measurements correspond to a family of operators $\{M_m\}$ where if the system is in state $|\phi\rangle$ before the measurement, the probability of seeing the value m is $p(m) = \left\langle \phi \mid M_m^\dagger M_m \mid \phi \right\rangle$, and if m is the observed value, then the system is left in state $M_m(|\phi\rangle)/\sqrt{p(m)}$.

For us, it suffices to make measurements *in the computational basis*, which means we are in the 1-qubit space $\mathcal{H}$ with operators $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$; measurement of the state $\alpha|0\rangle + \beta|1\rangle$ yields 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$, leaving the system in state $\frac{\alpha}{|\alpha|}|0\rangle$ or $\frac{\beta}{|\beta|}|1\rangle$, respectively.
This is the most mysterious of all: states transform in a **non-unitary** way when observed, so observation is apparently not time evolution of a physical system!

Quantum Mechanics: Postulate IV

Postulate IV: Combining Systems

When two systems are allowed to interact, the combined state space is the *tensor product* of the separate state spaces.

Let n 1-qubit systems interact gives the n-qubit state space $\mathcal{H}^{\otimes n}$, which is a complex vector space of dimension 2^n with basis

$$\begin{array}{rcccl}
|0\rangle & = & |00\cdots00\rangle & = & |0\rangle \otimes \cdots \otimes |0\rangle \otimes |0\rangle \\
|1\rangle & = & |00\cdots01\rangle & = & |0\rangle \otimes \cdots \otimes |0\rangle \otimes |1\rangle \\
|2\rangle & = & |00\cdots10\rangle & = & |0\rangle \otimes \cdots \otimes |1\rangle \otimes |0\rangle \\
\vdots & & \vdots & & \vdots \\
|2^n-1\rangle & = & |1\ldots1\rangle & = & |1\rangle \otimes \cdots \otimes |1\rangle \otimes |1\rangle
\end{array}$$

For example, there are *entangled* states, such as the Bell states

Quantum Mechanics: All Together Now

Postulate I: the State Space

The state of an isolated quantum system is described by a unit vector (these are written $|\phi\rangle$) in a Hilbert space, the *state space* of that system.

Postulate II: Unitary Time Evolution

The time evolution of an isolated system is given by a *unitary operator* acting on the state vectors, $|\phi\rangle \mapsto U(|\phi\rangle)$

Postulate III: Measurement

Measurements correspond to a family of operators $\{M_m\}$ where if the system is in state $|\phi\rangle$ before the measurement, the probability of seeing the value m is $p(m) = \left\langle \phi \mid M_m^\dagger M_m \mid \phi \right\rangle$, and if m is the observed value, then the system is left in state $M_m(|\phi\rangle)/\sqrt{p(m)}$.

Postulate IV: Combining Systems

When two systems are allowed to interact, the combined state space is the *tensor product* of the separate state spaces.

Quantum Circuits

Just like regular circuits, only more linear algebra:
put $2^n \times 2^n$ **unitary** matrices in the boxes of a reversible circuit diagram.
The ends of some wires can be measured in the computational basis.

Universality here means building (*exactly or approximately*) any gate on n qubits out of combinations of a fixed set of gates on a fixed (small) number of qubits.
Several collections are universal, such as $U(2) \cup \{CNOT\}$.

There are also **qudits**, based on d-state quantum systems.

Quantum algorithms exists which, *e.g.:*

search unsorted databases in $O(\sqrt{n})$ time [Grover], and
factor integers in polynomial time [Shor].

Resistance Is Futile: Quantum Computers Are Coming.

[So we should now be training the next generation of quantum computer scientists.]

Example Quantum Algorithms: Deutsch-Jozsa

Problem: given a function $f : \{0,1\}^n \to \{0,1\}$ which is either

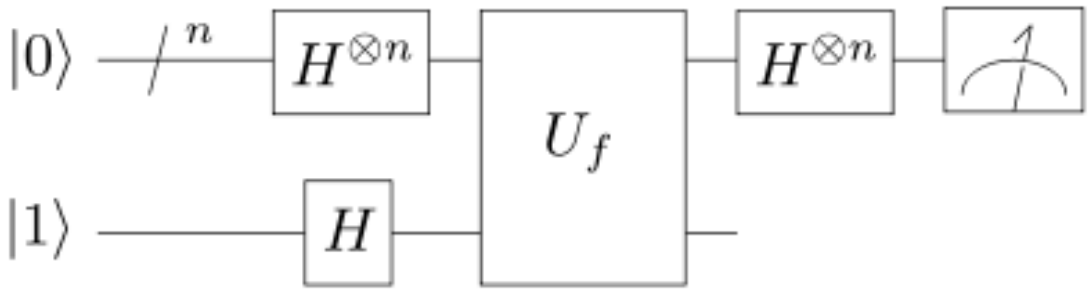
$$\textit{constant} \text{ or } \textit{balanced } [\#(f^{-1}(0)) = \#(f^{-1}(1))],$$

determine which it is.

Use a *quantum oracle* for U, the quantum gate

$$U_f(|x\rangle \otimes |y\rangle) = |x\rangle \otimes |y \oplus f(x)\rangle$$

Then do:



Vary the Structure Group!

Because it's cheaper than inter-dimensional travel

The *structure group* of our universe seems to be $U(n)$.

The *structure group* of the universe of classical computation is S_n.

What about other groups [universes]?

Circuits built on a family of groups and representations work with diagrams like quantum (or classical reversible) circuits, but the wires have values in a fixed vector space V and the boxes are in some chosen groups, which act on appropriate tensor powers of V via some family of representations

So What? Part 1

Well, how about *BvN Machines?*
Why only a group?
How about a semigroup?
Use the **Birkhoff Polytope** B_n!
The Birkhoff-von Neumann Theorem: B_n is the convex hull in $\mathbb{R}^{n^2}$ of the permutation matrices.
Includes things like $RAND_{1/2} = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$. Use the output of such a gate to control another operation ...

BvN machines compute the same things as classical probabilistic Turing machines

So What? Part B

My conjecture, which is mine: only groups and representations which have some negative matrix coefficients admit quantum-type exponential speed-up over classical algorithms.

Other types of groups:

- what about infinite dimensional groups?
- issues of universality to do with generating interesting groups by a finite number of generators
- find some intermediate group between S^n and $U(n)$ which allows quantum-fast algorithms; in fact, $O(3)$ does fine. ... so find a physical system which can implement $O(3)$ computing.

References

A great book on qunatum computation:

Quantum Computation and Quantum Information, Michael A. Nielsen and Isaac L. Chuang, MIT Press, 10^{th} Anv ed. (2011)

Details of varying the structure group:

Universal Gates in Other Universes, Jonathan A. Poritz, appeared in G.W. Dueck and D.M. Miller (Eds.): RC 2013, LNCS 7948, pp. 155-167; Springer-Verlag Berlin Heidelberg 2013 (also on `poritz.net/jonathan`)