Meeting 6.2: Reversible computing and quantum circuits

I. RSAT

II. Quantum circuits

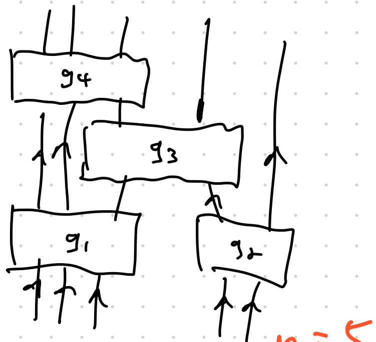
Next time: Solovay-Kitaev?

I. RSAT

Last time: a (Boolean) gate set $\mathcal{G}$ is a set of bijections

$$\{0,1\}^k \to \{0,1\}^k. \qquad (k \text{ variable})$$

A planar, reversible Boolean circuit R is a diagram like this:



$g_i \in \mathcal{G}$.

depth: 3

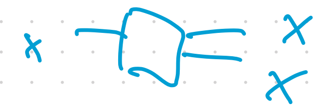
R encodes a function

$$R: \{0,1\}^n \to \{0,1\}^n$$

The circuit is a "planar $\mathcal{G}$-factorization" of this function.

Note: every Boolean Function (not necessarily reversible)

$$f: \{0,1\}^m \to \{0,1\}^n$$



can be built out of AND, OR, and NOT, FANOUT

$\{$AND, OR, NOT$\}$ is a universal set of logic gates.

If we want to find interesting computational problems for reversible circuits, $\mathcal{G}$ better be "sufficiently rich."

Lots of wiggle room!

Example:

$\mathcal{G} = \{F\}$, where F is the Fredkin gate:

Inputs			Outputs		
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	1	1

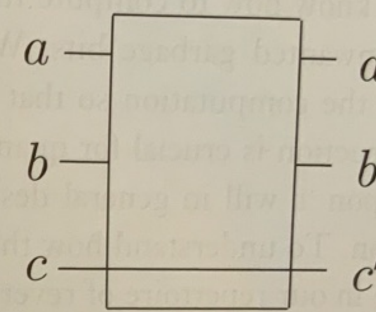


Figure 3.15. Fredkin gate truth table and circuit representation. The bits a and b are swapped if the control bit c is set, and otherwise are left alone.

Fredkin = "Controlled SWAP"

Given any $g: \{0,1\}^k \to \{0,1\}^k$

can always build "C-g" or "Controlled-g":

Just for Fun: Since F is "conservative"

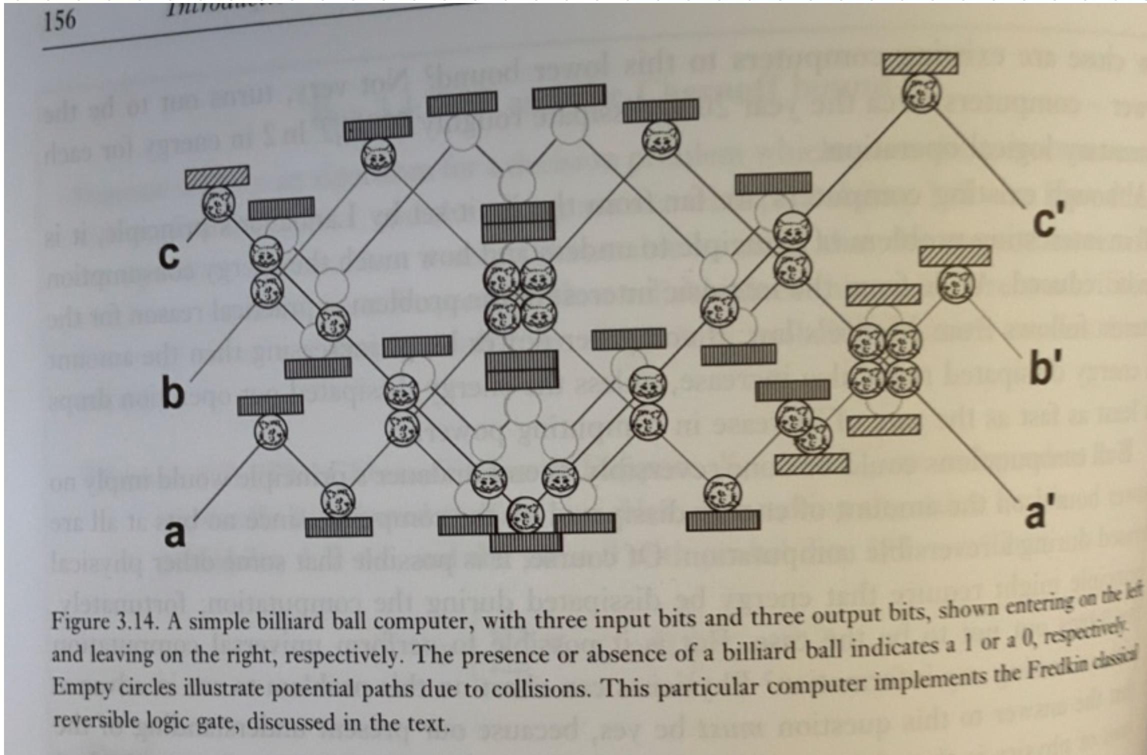


Figure 3.14. A simple billiard ball computer, with three input bits and three output bits, shown entering on the left and leaving on the right, respectively. The presence or absence of a billiard ball indicates a 1 or a 0, respectively. Empty circles illustrate potential paths due to collisions. This particular computer implements the Fredkin classical reversible logic gate, discussed in the text.

we can implement it with billiard balls!

If we allow extra "ancilla" bits, can encode AND, OR, NOT:

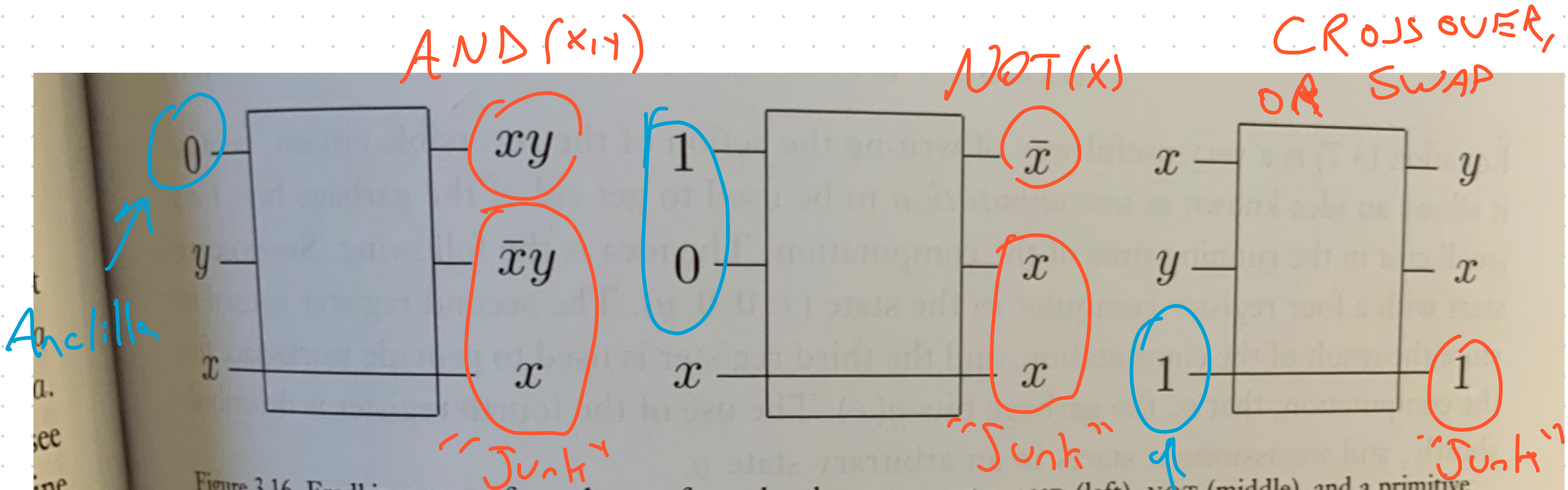
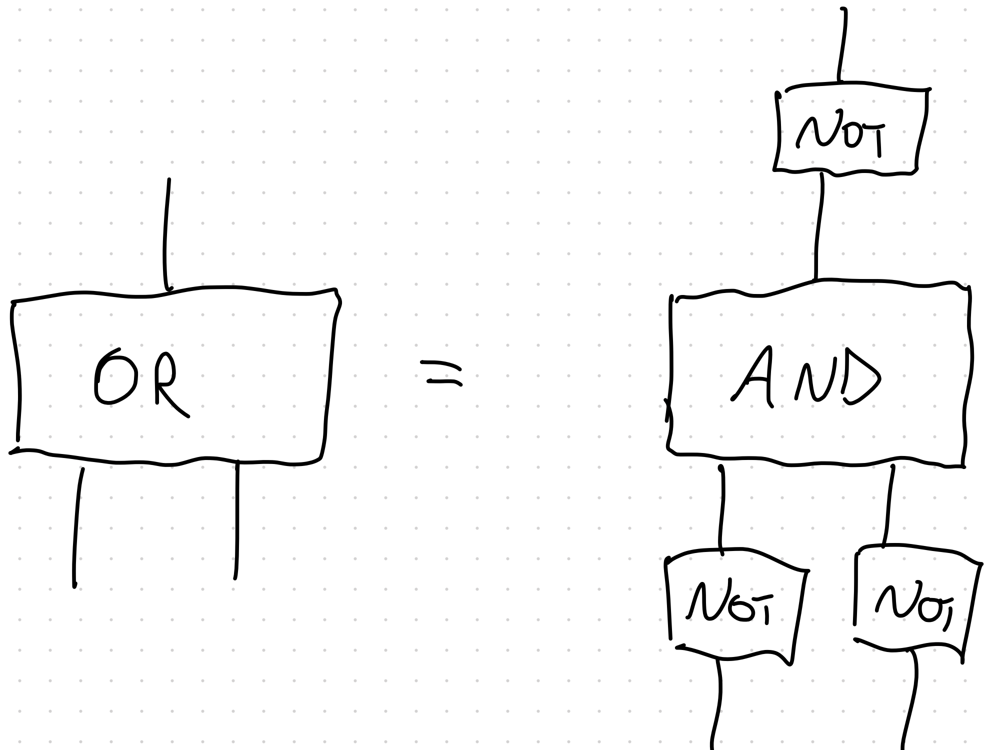


Figure 3.16. Fredkin gate configured to perform the elementary gates AND (left), NOT (middle), and a primitive routing function, the CROSSOVER (right). The middle gate also serves to perform the FANOUT operation, since it produces two copies of x at the output. Note that each of these configurations requires the use of extra 'ancilla' bits prepared in standard states – for example, the 0 input on the first line of the AND gate – and in general the output contains 'garbage' not needed for the remainder of the computation.

ANCILLA

Recall: De Morgan $x \vee y = \neg(\neg x \wedge \neg y)$



Since NOT and SWAP are reversible, might as well include them in $\mathcal{G}$ for now

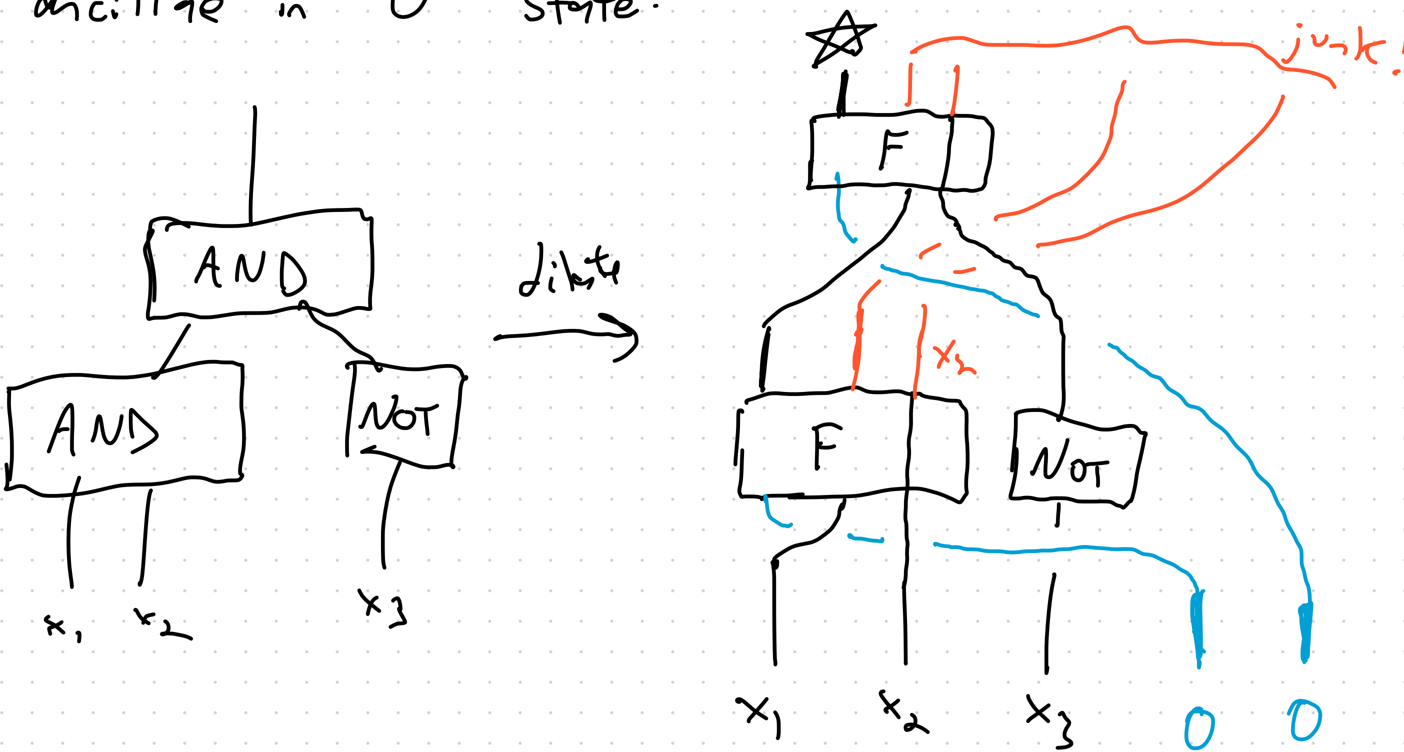
$$\mathcal{G} = \{F, NOT, SWAP\}$$

$$\mathbb{C}^2 \longrightarrow \mathbb{C}^2 \otimes \mathbb{C}^2$$

$$x \longmapsto x \otimes x$$

is not linear!

We can "dilate" every Boolean circuit to a reversible circuit, by replacing every AND with a Fredkin + ancillae in 0 state:



$\mathcal{G} = \{F, SWAP, NOT\}$.

RSAT($\mathcal{G}$), variant 1:

Instance: reversible (planar) $\mathcal{G}$-circuit R, with input divided into "data register" of width d and "ancilla register" of width $n-d$, where width$(R) = n$, and all ancillae set to 0.

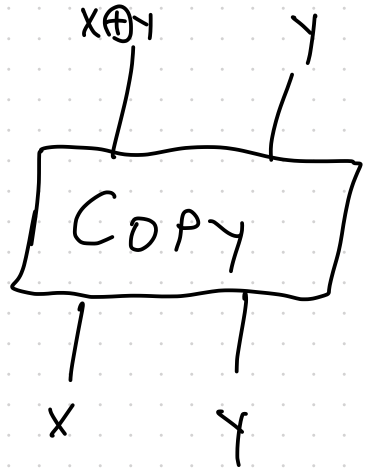
Problem: Does there exist $x \in \{0,1\}^d$ such that the first output bit of $R(x, \underbrace{0, \dots, 0}_{n-d})$ is 1?

Lemma: RSAT($\mathcal{G}$) is NP-complete.

Proof: Reduce from CSAT using dilation as on previous page. $\square$

If we include COPY in $\mathcal{G}$, we can build a somewhat less contrived variant of RSAT.

Here COPY is



COPY is "reversible copy" not "clone" or "fan out"

(If $x=0$, COPY copies y to x.)

$$\mathcal{G} = \{F, \text{SWAP}, \text{NOT}, \text{COPY}\}$$

RSAT($\mathcal{G}$), variant 2:

Instance: $\mathcal{G}$-circuit R with width$(R) = 2n$, with input divided into data and ancillae registers both of width n.

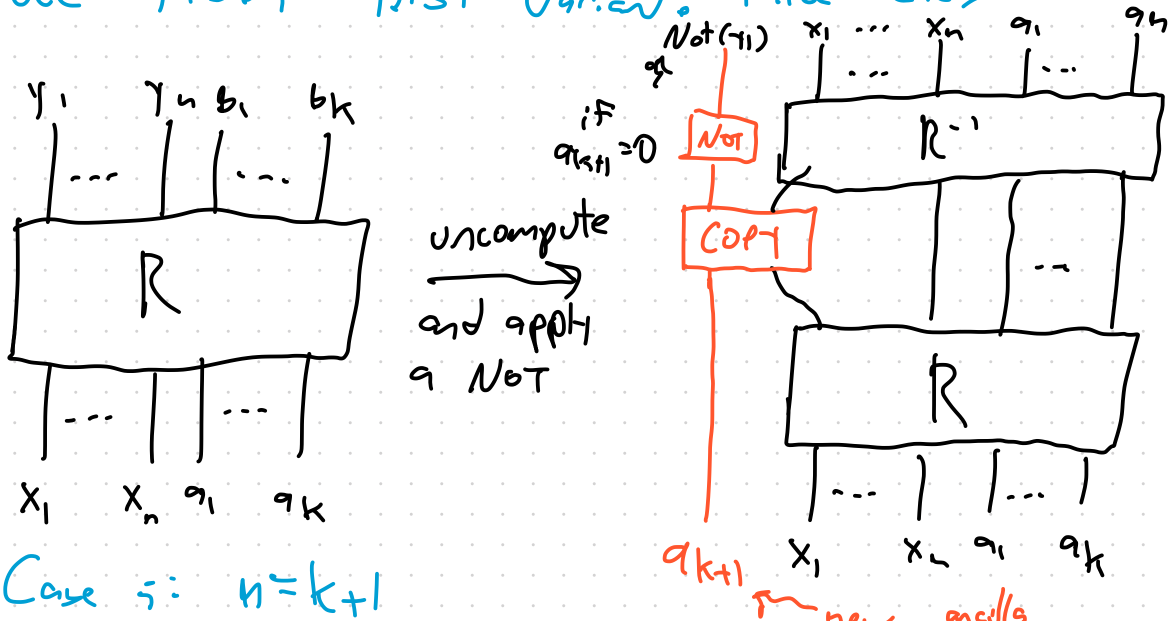
Problem: Do there exist $x, y \in \{0,1\}^n$ such that

$$R(x, \underbrace{0, \ldots, 0}_{\text{ancillae}}) = (y, \underbrace{0, \ldots, 0}_{\text{ancillae}})$$

Lemma: This problem is NP-complete.

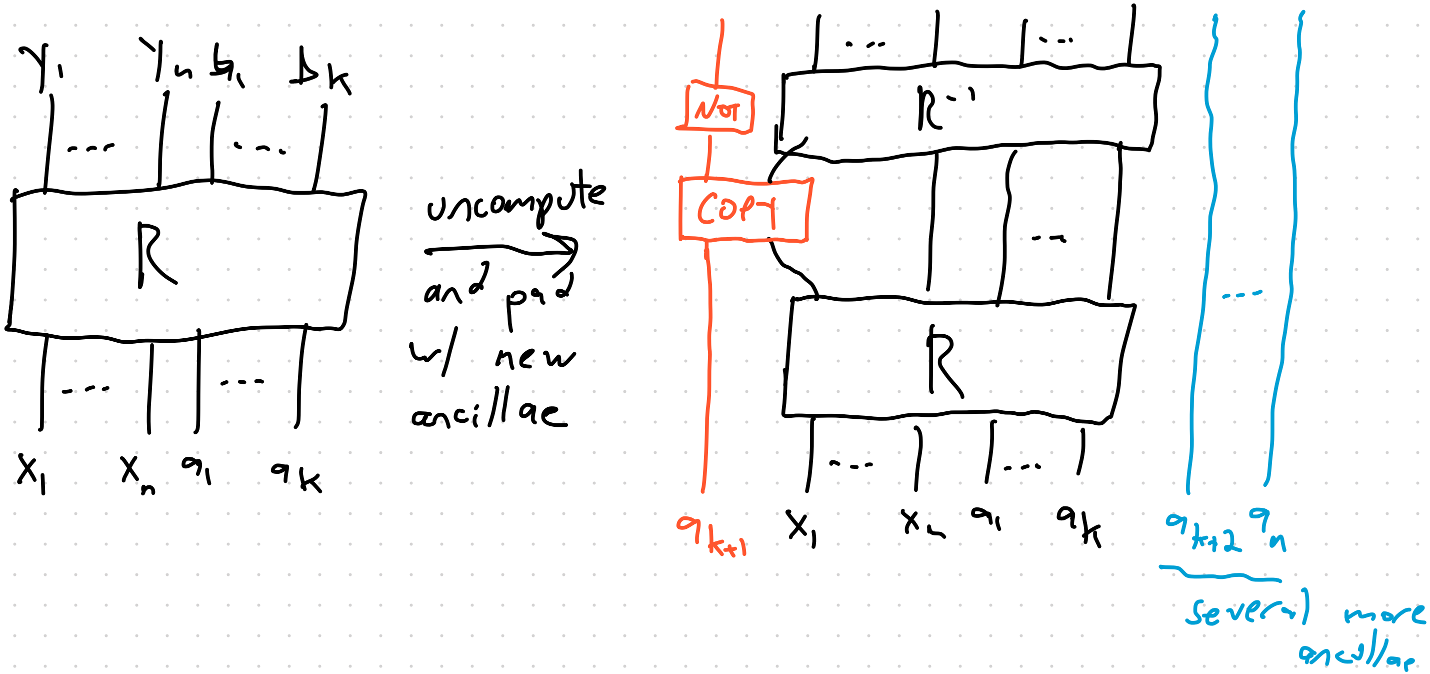
Proof: Key idea is "uncomputation," which is also useful in quantum computation and in complexity results in topology. (See "Computational complexity and 3-manifolds and zombies" by Kuperberg – S.)

Reduce from first variant. Three cases



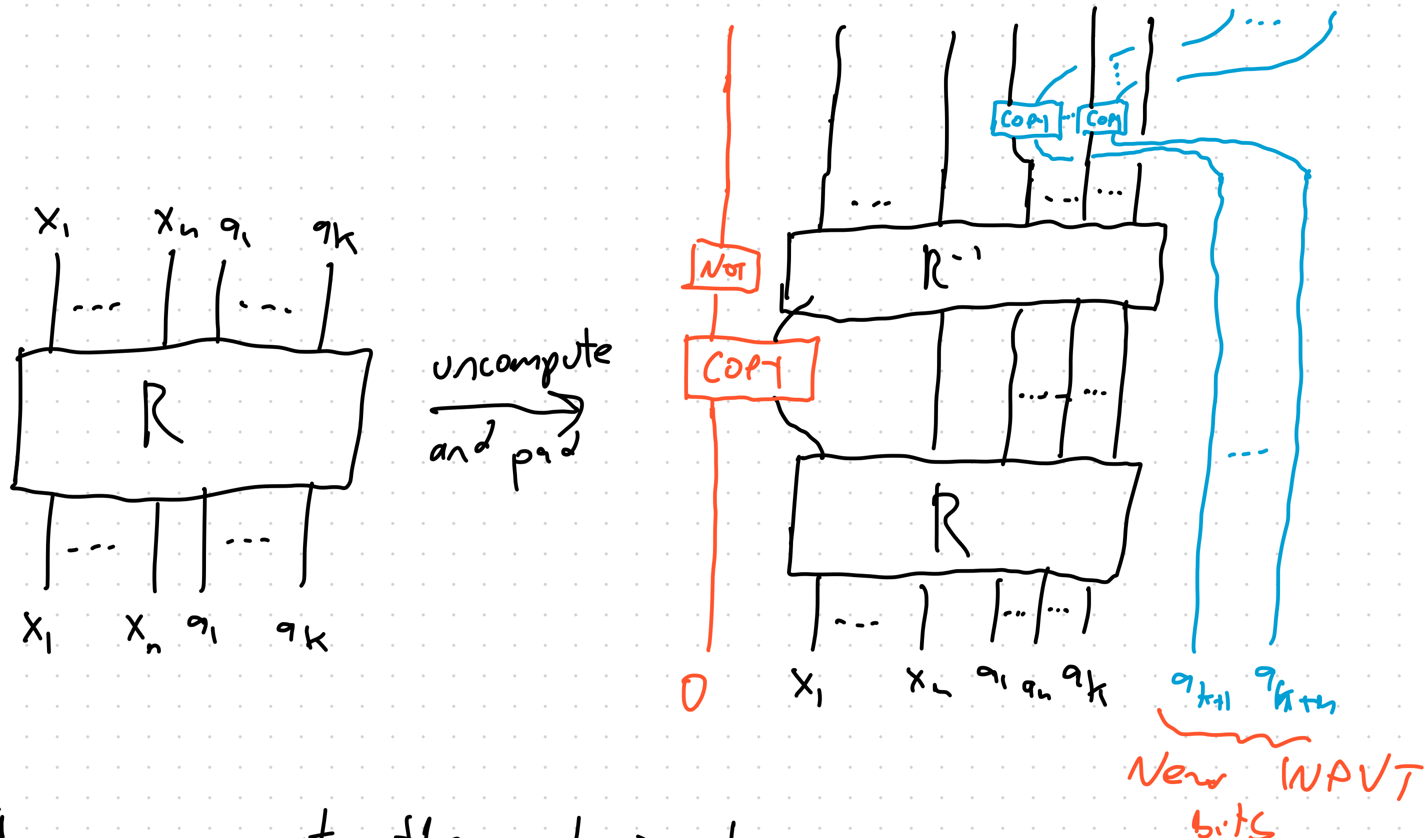
Case i: $n=k+1$

Case ii: $n > k+1$



"padding ancilla register"

Case iii: $n < k+1$

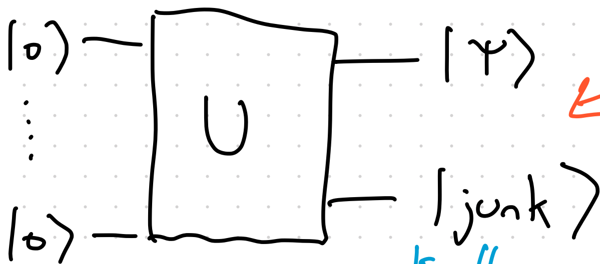


The copying at the end is to ensure a parsimonious reduction.

□

Why uncomputation is relevant to quantum computing:

We might work hard to prepare quantum state $|\psi\rangle$ so we can do useful things with it.



$$|\text{junk}\rangle = \sum_{i=0}^{2^k} c_i |i\rangle$$

Potentially imprecise notation! Output might be entangled w/ the junk!

Interesting question:

Given gate set $\mathcal{G}$, what's the complexity of $\text{RSAT}(\mathcal{G})$?

I.e.: "How powerful is $\mathcal{G}$?"

<u>Guess</u>: Either its in P

or its NP-complete?

(See Schaefer dichotomy theorem.)

II. Quantum Circuits

Call a unitary trasformation

$$U: \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{k} \longrightarrow \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$$

a **k-ary quantum gate.**

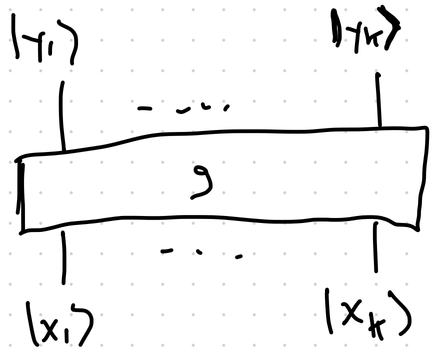
Any set $\mathcal{G}$ of quantum gates is called a **quantum** gate set.

Examples:

1. Any classical reversible gate g can be "linearized"

If $g: \{0,1\}^k \to \{0,1\}^k$
$(x_1,\dots,x_k) \mapsto (y_1,\dots y_k)$

Then



The quantum gate g permutes the computational basis of $(\mathbb{C}^2)^{\otimes k}$.

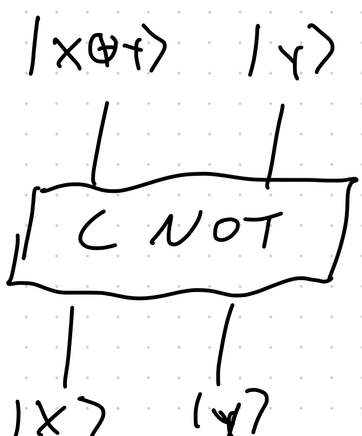
Take-away: quantum circuits include classical reversible circuits.

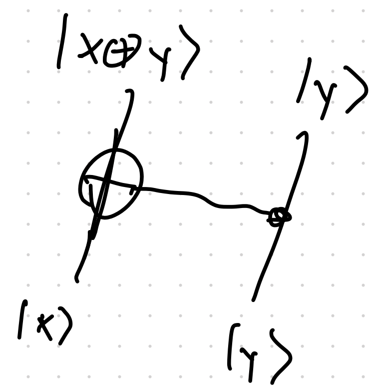
2. CNOT (aka COPY)

Linearization of CNOT

$$CNOT : \{0,1\}^2 \to \{0,1\}^2$$

$$\begin{array}{ccc} 00 & & 00 \\ 01 & \mapsto & 01 \\ 10 & & 11 \\ 11 & & 10 \end{array}$$



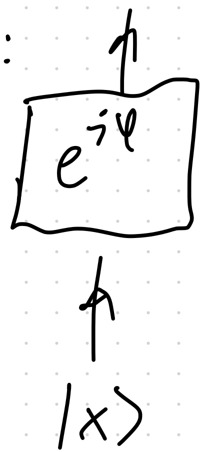


3. Single qubit gates, i.e. unitary operators on $\mathbb{C}^2$, i.e. Lie group $U(2)$.

Hadamard

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

Phase gates:



$$= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix} |x\rangle$$

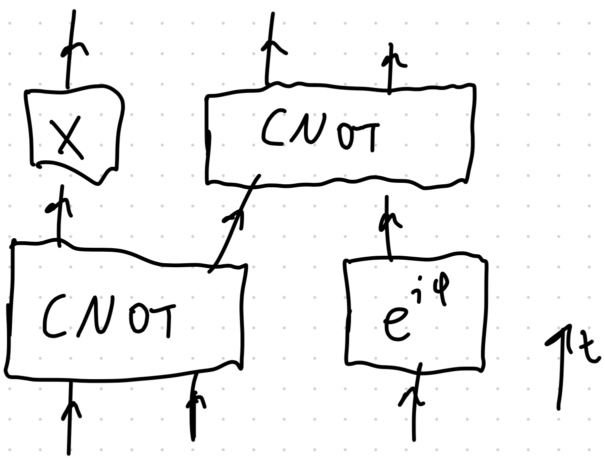
$$a|0\rangle + b|1\rangle \mapsto a|0\rangle + e^{i\varphi} b|1\rangle.$$

A quantum circuit over $\mathcal{G}$ is a circuit whose gates are elements of $\mathcal{G}$.

Just as for classical reversible circuits, quantum circuits have a width and a depth.

E.g.

C:



depth: 2 width: 3

C implements a unitary on $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$:

$$(X \otimes CNOT) \circ (CNOT \otimes e^{i\varphi})$$

Gate set $\mathcal{G}$ is precisely universal if (for n large enough...) every unitary $U: (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$ can be expressed as a $\mathcal{G}$-circuit. (Every $U \in U(2^n)$ can be factored as a product of elements of $\mathcal{G}$.)

$U(2)$ + CNOT is universal quantum gate set.

In fact: phase gates + H + CNOT is universal.

"Precisely universal" is overkill!

Why? Quantum computers are probabilistic and states that are too close can not ~~be~~ feasibly distinguished.

A better definition (but still arguably overkill...)

A gate $\mathcal{G}$ is (planar) quantum universal if for all n large enough, elements of $\mathcal{G}$ in

$$\underbrace{U(\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n}) \cong U(2^n)$$

(i.e., given $g \in \mathcal{G}$ that is binary, we get $n-1$ different unitaries of the form $\mathrm{Id}_{(\mathbb{C}^2)^{\otimes i}} \otimes g \otimes \mathrm{Id}_{(\mathbb{C}^2)^{n-2-i}}$)

generate (as a monoid) a dense subset

(For all $\varepsilon > 0$ for every $U \in U(2^n)$, we can find a $\mathcal{G}$-circuit U' s.t. $\|U - U'\| < \varepsilon$.)

Let

$$F: \{0,1\}^n \to \{0,1\}^m$$

be a function. A circuit U "computes" F to precision ε $(0 \le \varepsilon < 1/2)$ if for any $x \in \{0,1\}^n$

$$\sum_{z=0}^{2^m-1} \left| \langle F(x), \underbrace{z}_{0\ldots0} | U | x, 0^{N-n} \rangle \right| \ge 1 - \varepsilon.$$

(U has width N)