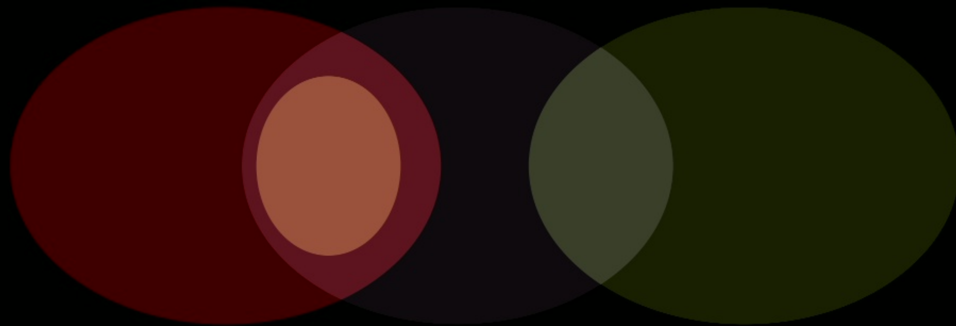
Title: New Quantum Algorithm for NP-Complete problems: Efficiency to be determined

Speakers: Cole Coughlin
Date: February 06, 2023 - 2:00 PM

URL: https://pirsa.org/23020034

Abstract: Quantum algorithms have been found which are able to solve important problems exponentially faster than any known classical algorithm. The most well known example is Shor's algorithm which would be able to break all RSA encryption if fault tolerant quantum computers existed for it to be run on. It is currently not believed that quantum computers will be able to efficiently solve NP-Complete problems, but the answer is still unknown. I present a novel quantum algorithm which is able to solve 3-SAT along with numerical simulations to see how it performs on small instances, but new methods of analyzing the complexity of quantum algorithms will need to be developed before we can say exactly how it performs. This will be the goal of my PSI essay.

Zoom Link: https://pitp.zoom.us/j/95795655548?pwd=aU5jNFhFZHEzUWpLK1FvWjd2Q1c2Zz09



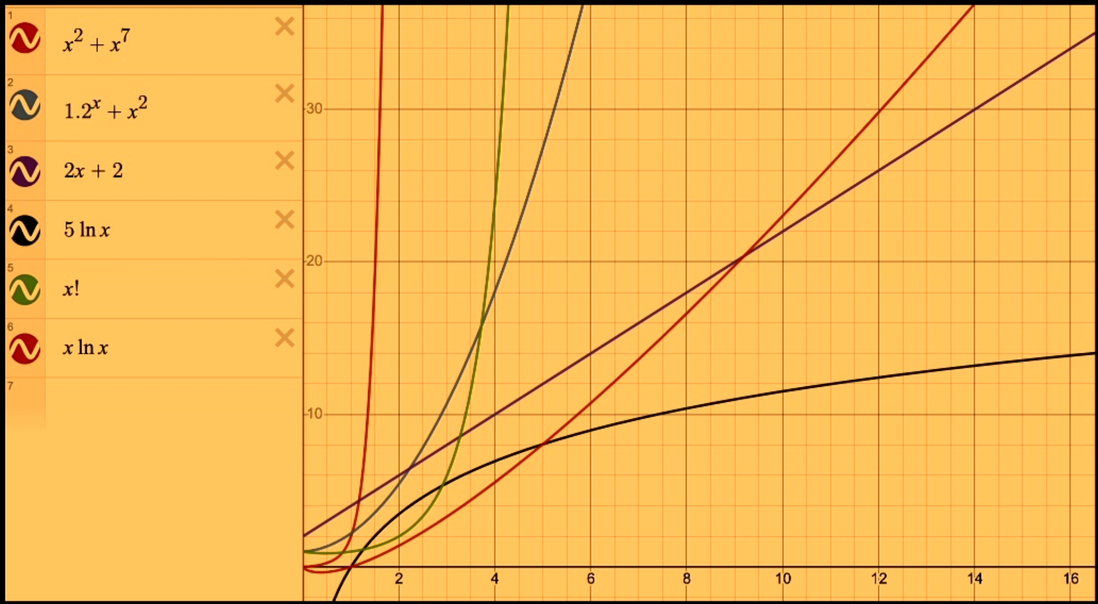
NEW QUANTUM NP-COMPLETE ALGORITHM

EFFICIENCY TO BE DETERMINED

Cole Coughlin - Feb 2023

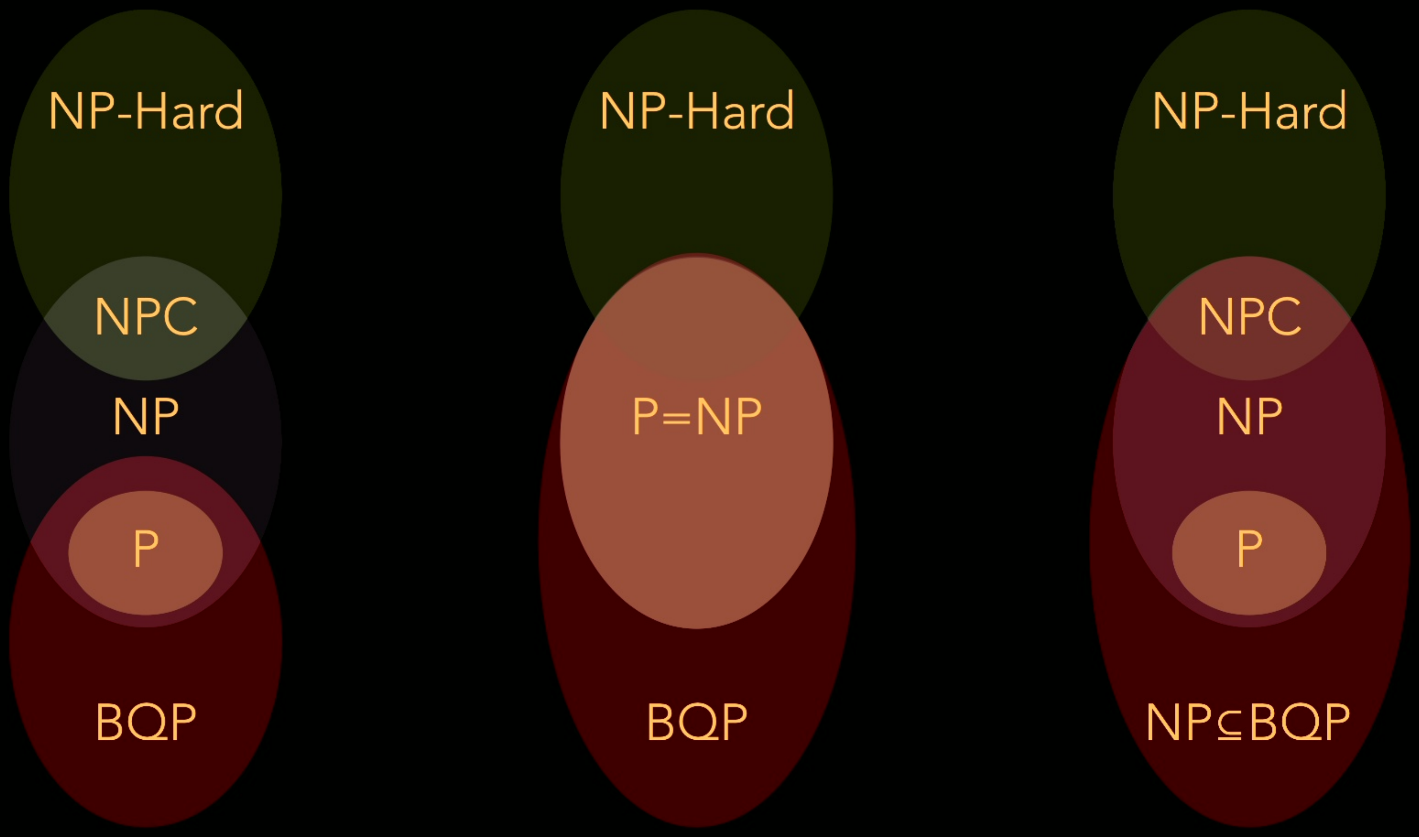
BIG O NOTATION

- N - Size of the input
- Worst case running time
- Fastest growing term in complexity



Linear	Polynomial	Exponential
$2N + 2 \rightarrow O(N)$	$N^2 + N^7 \rightarrow O(N^7)$	$1.2^N + N^2 \rightarrow O(1.2^N)$

INTO THE COMPLEXITY ZOO



WHY DO WE CARE ABOUT NP-COMPLETE PROBLEMS?

- They are Complete!
- Deals with the fundamental limits of what is computable
- All problems which are verifiable in polynomial time are in NP - Including bounded length Mathematical proofs!

SATISFIABLITY

- The first NP-Complete problem
- Given a propositional logic formula consisting of Boolean variables and the operations AND, OR and NOT.
- If there is an assignment of the variables such that the logical statement evaluates TRUE, then it is said to be satisfiable.

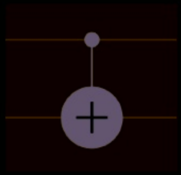
$$\Phi = (a \vee \neg b \vee \neg c) \wedge (\neg a \vee b \vee d)$$

$$\Phi = \text{TRUE for } a = 1,\ d = 1$$

MULTI QUBIT QUANTUM GATES

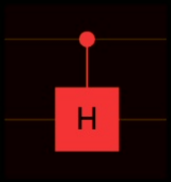
C-NOT

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



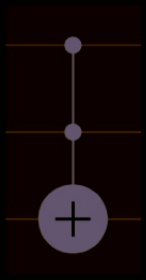
C-Hadamard

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



Toffoli - CC-NOT

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

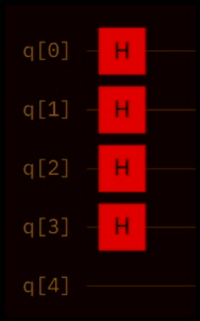


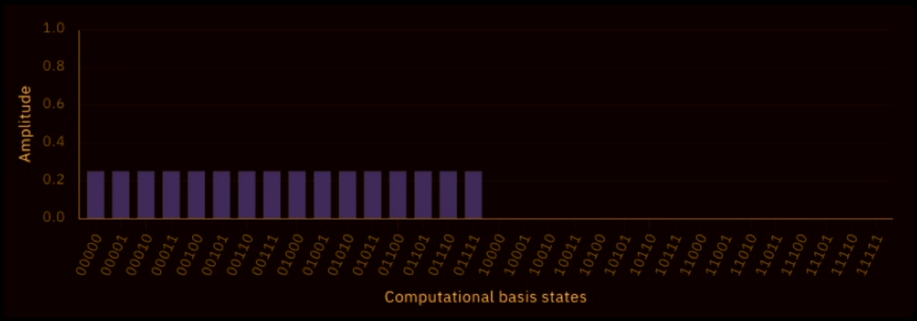
HOW TO BUILD A QUANTUM CIRCUIT TO SOLVE 3SAT

We will have one qubit for every variable in the 3SAT instance and one ancillary qubit

$$\Phi = (a \vee \neg b \vee \neg c) \wedge (\neg a \vee b \vee d) \rightarrow 5 \text{ qubits}$$

The qubits corresponding to the variables will start out in equal superposition





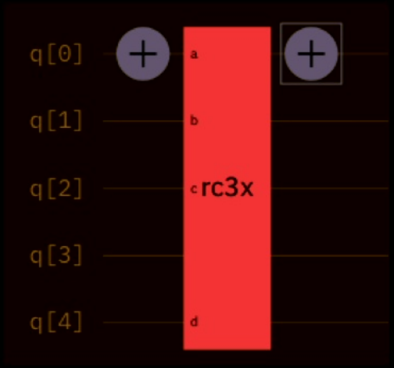
HOW TO BUILD A QUANTUM CIRCUIT TO SOLVE 3SAT

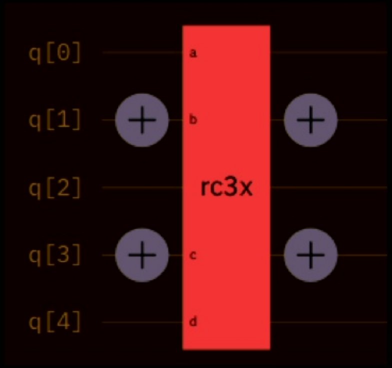
Instead of working with the 3SAT instance

$$\Phi = (a \vee \neg b \vee \neg c) \wedge (\neg a \vee b \vee d)$$

We will be working with the logical NOT of the instance

$$\neg\Phi = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg d)$$

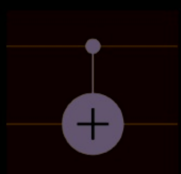




MULTI QUBIT QUANTUM GATES

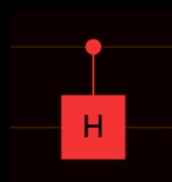
C-NOT

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



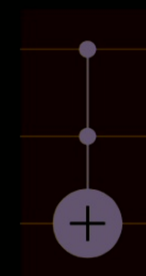
C-Hadamard

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



Toffoli - CC-NOT

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



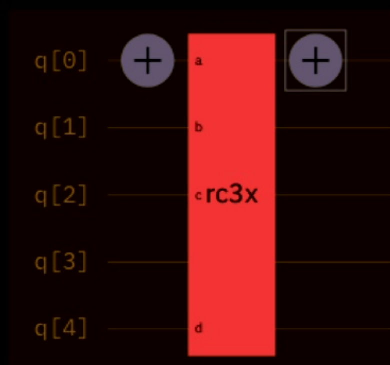
HOW TO BUILD A QUANTUM CIRCUIT TO SOLVE 3SAT

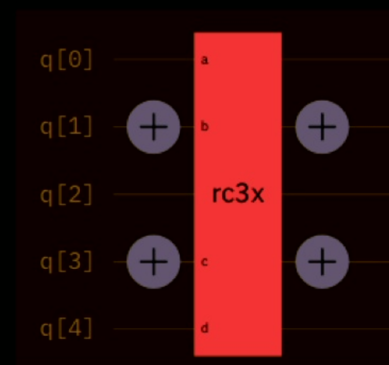
Instead of working with the 3SAT instance

$$\Phi = (a \vee \neg b \vee \neg c) \wedge (\neg a \vee b \vee d)$$

We will be working with the logical NOT of the instance

$$\neg\Phi = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg d)$$



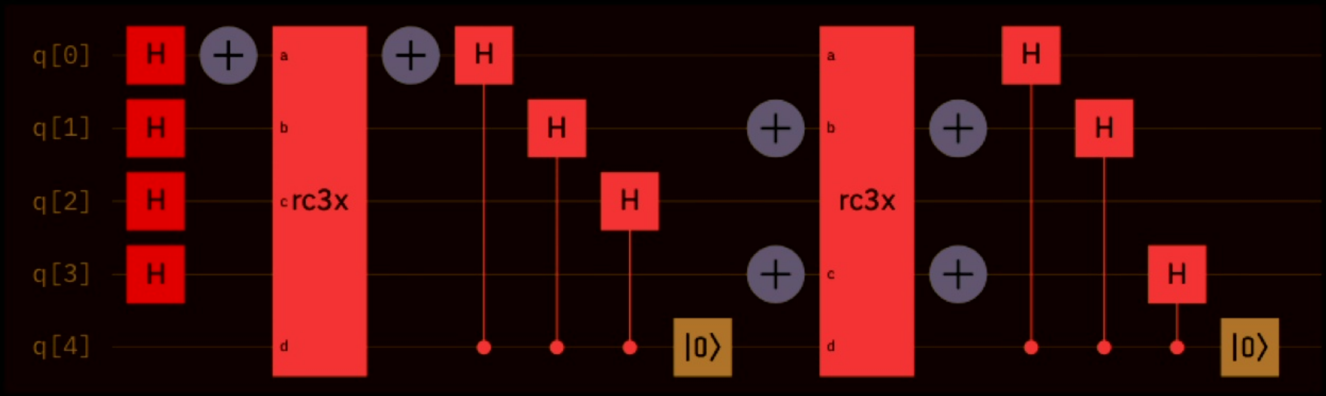


HOW TO BUILD A QUANTUM CIRCUIT TO SOLVE 3SAT

Perform a controlled Hadamard on the variables in the clause

Then reset the ancillary qubit

$$\neg\Phi = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg d)$$

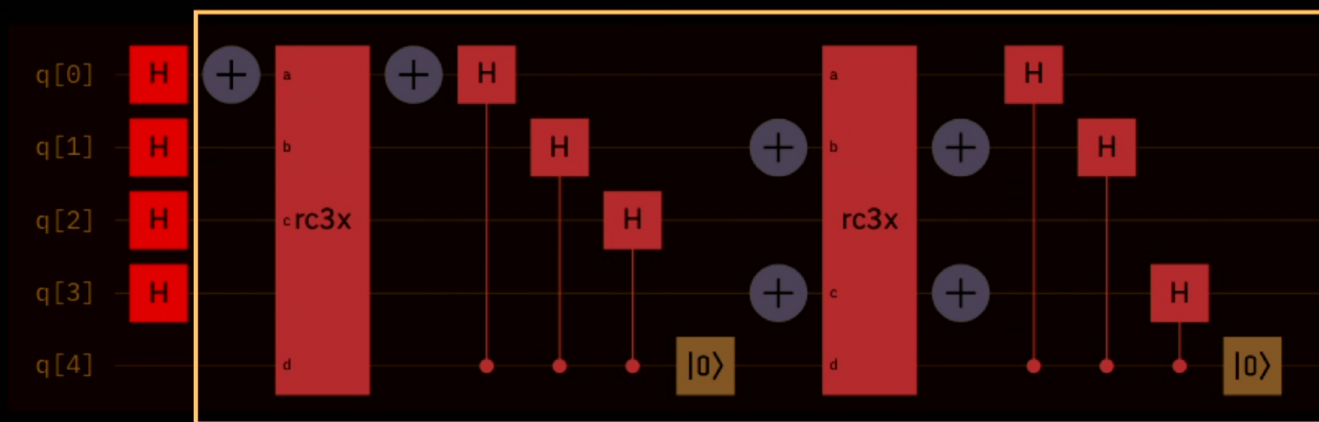


HOW TO BUILD A QUANTUM CIRCUIT TO SOLVE 3SAT

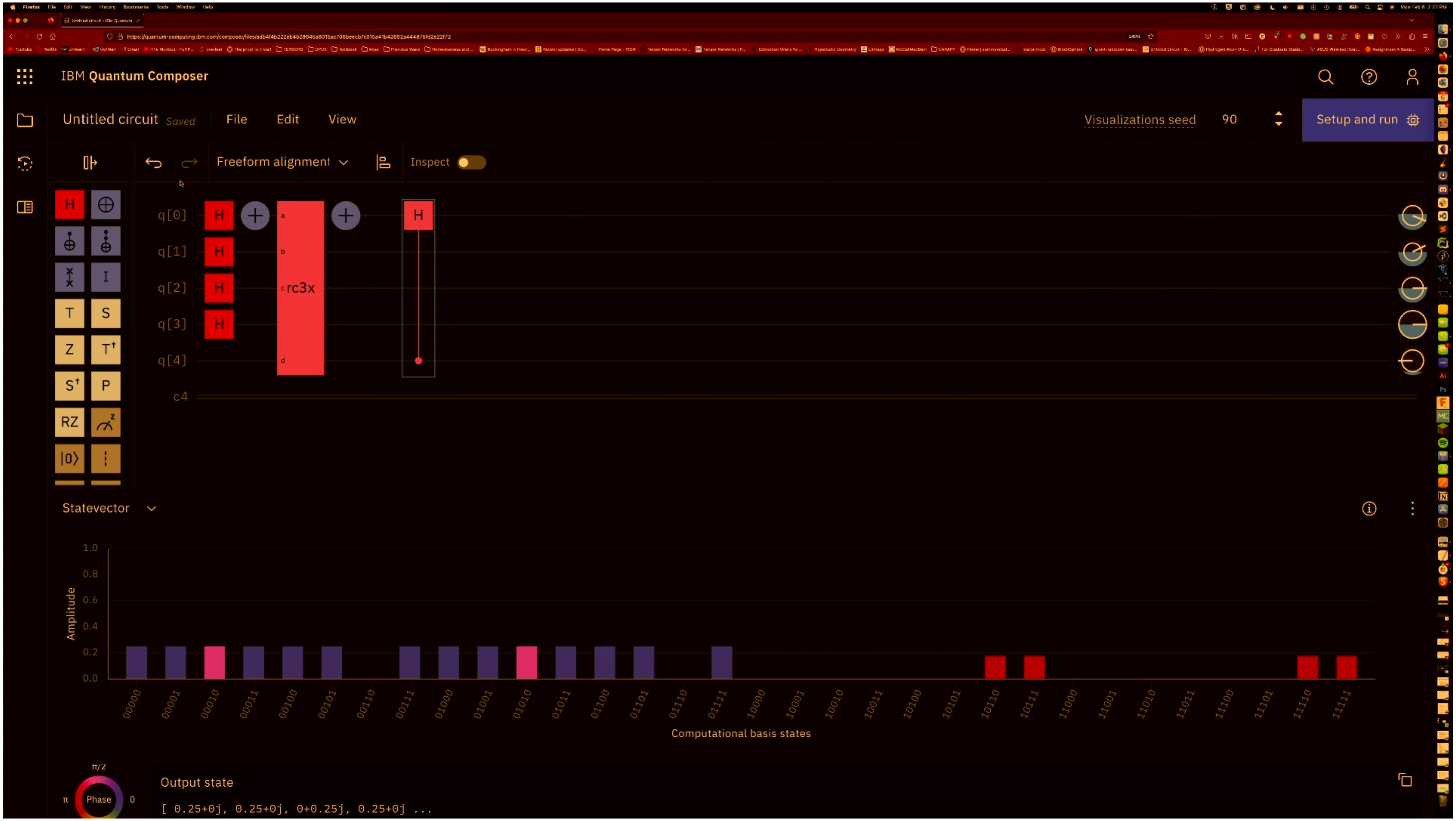
Perform a controlled Hadamard on the variables in the clause

Then reset the ancillary qubit

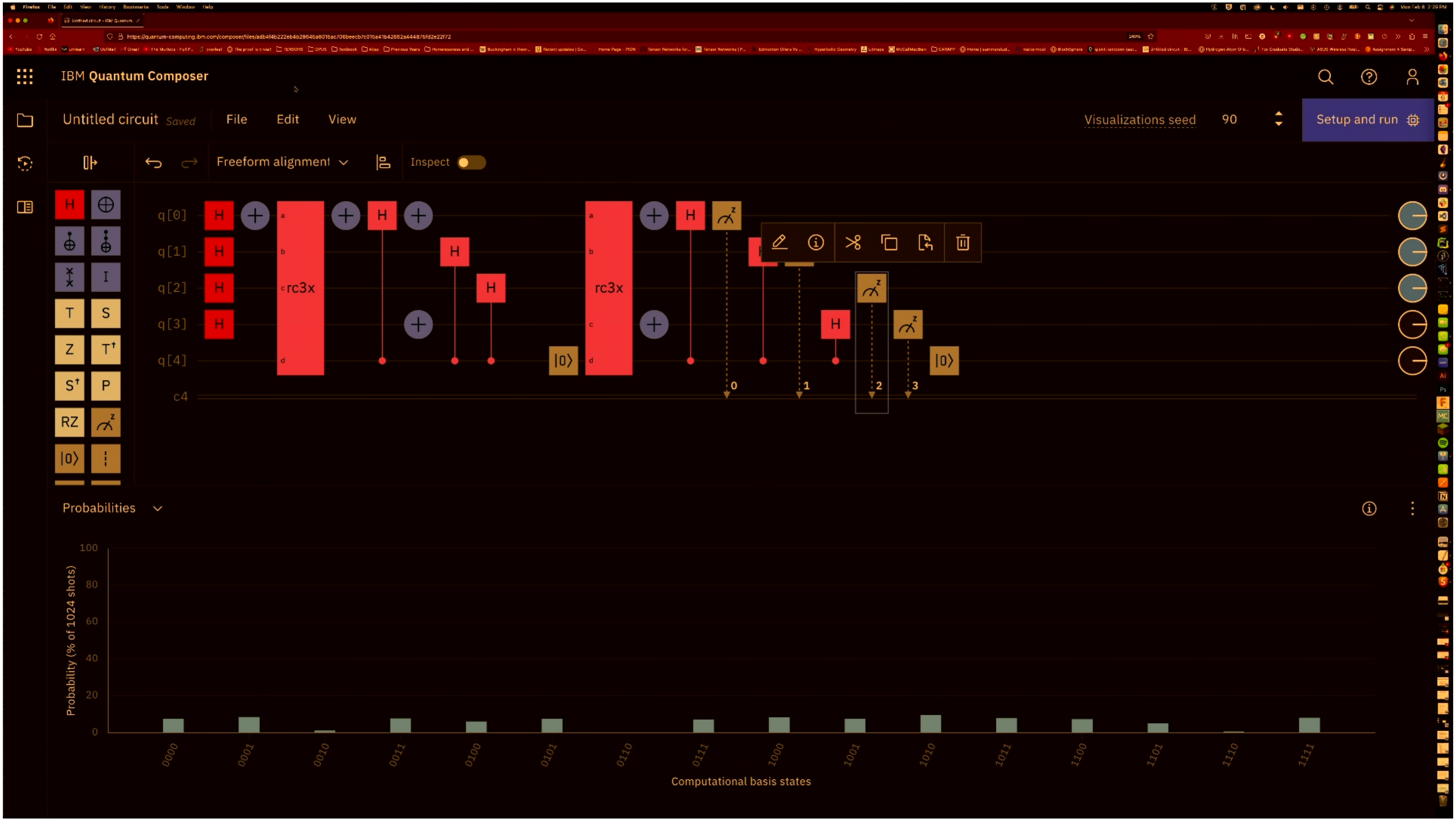
$$\neg\Phi = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg d)$$



U_{SAT}



IBM Quantum Composer
Untitled circuit Saved
File
Edit
View
Visualizations seed
90
Setup and run
Freeform alignment
Inspect
q[0]
q[1]
q[2]
q[3]
q[4]
c4
rc3x
Statevector
Amplitude
Computational basis states
Output state
[0.25+0j, 0.25+0j, 0+0.25j, 0.25+0j ...
Phase

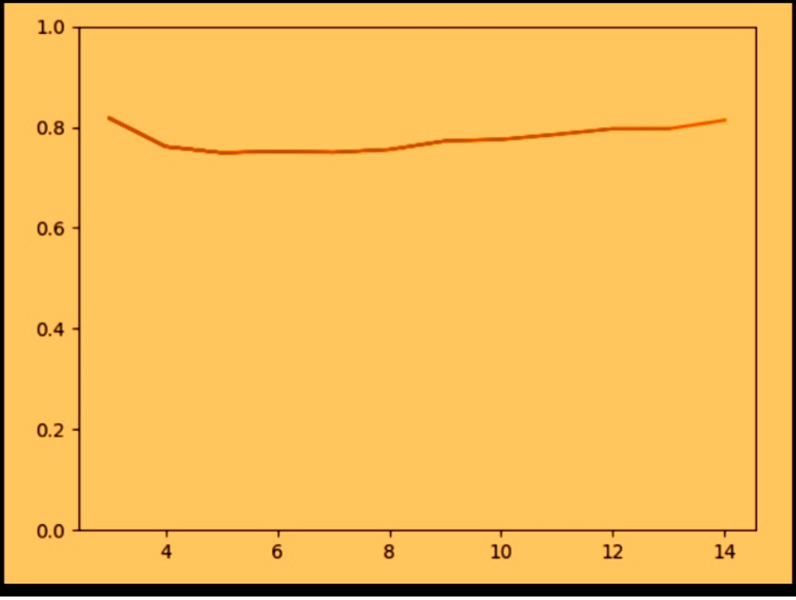

IBM Quantum Composer
Untitled circuit Saved
File
Edit
View
Visualizations seed
90
Setup and run
Freeform alignment
Inspect
q[0]
q[1]
q[2]
q[3]
q[4]
c4
rc3x
Probabilities
Probability (% of 1024 shots)
Computational basis states
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

PRELIMINARY RESULTS

- Build 3SAT instances with 1 solution and the minimum number of clauses $c = 3n - 2$
- Apply U_{SAT} c times
- See how the algorithm scales as we increase the number of variables

PRELIMINARY RESULTS

- Build 3SAT instances with 1 solution and the minimum number of clauses $c = 3n - 2$
- Apply U_{SAT} c times
- See how the algorithm scales as we increase the number of variables



NEXT STEPS

- Prove algorithm correctness - it will solve any problem in an infinite number of iterations
- Test how it performs in small instances against classical random solvers
- Prove how many iterations are necessary to give solution with 1/3 probability

THANK YOU

- Craig McRae
- David Gosset and Anirban Ch Narayan Chowdhury