

Equipo 6: Qutips



Problema de GATO

Challenge: Mentorship Program Screening Task

Thank you for your interest in the QC Mentorship program!

We decided to select participants based on how they will manage to do some simple “screening tasks”

These tasks have been designed to:

- find out if you have the skills necessary to succeed in our program.
- be doable with basic QC knowledge - nothing should be too hard for you to quickly learn.
- allow you to learn some interesting concepts of QC.
- give you some choices depending on your interests.

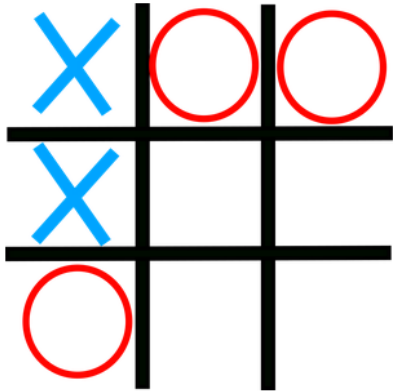
What we mean by skills is not knowledge and expertise in QC. It's the ability to code, learn new concepts and to meet deadlines.

What are we looking for in these applications? Coding skills – clear, readable, well-structured code Communication – well-described results, easy to understand, tidy. Reliability – submitted on time, all the points from the task description are met Research skills – asking good questions and answering them methodically

Also, feel free to be creative – once you finish the basic version of the task, you can expand it. Bonus questions provide just some ideas on how to expand a given topic.

Task 3 Find best results

For this problem you have the following situation: you are playing a game of tic-tac-toe, and you find the situation in the figure below, next is your turn, develop a quantum algorithm to be able to find the best decisions with higher probability.



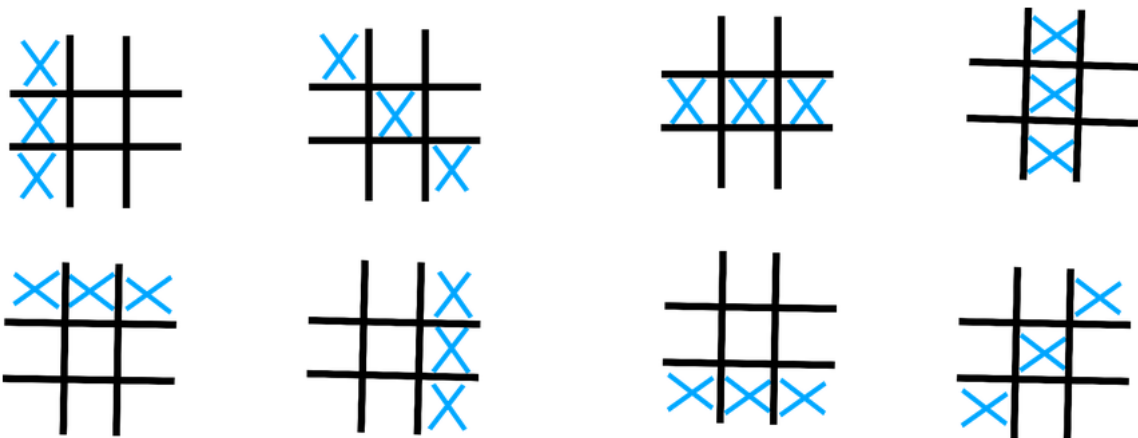
The following considerations apply:

- You are the X's.
- The matrix as a qubit and the state of the X's is $|1\rangle$ and of the

O's is $|0\rangle$, of the empty cells an unknown state.

- What are the valid combinations to win?
- You have at most 2 turns

A hint for this exercise should consider all the possible ways to win that (there's 8 of them), for this exercise must obtain with probability the state with the highest probability.



for the output only give the status of the empty boxes, you consider this example:

If you think the solution is put the values

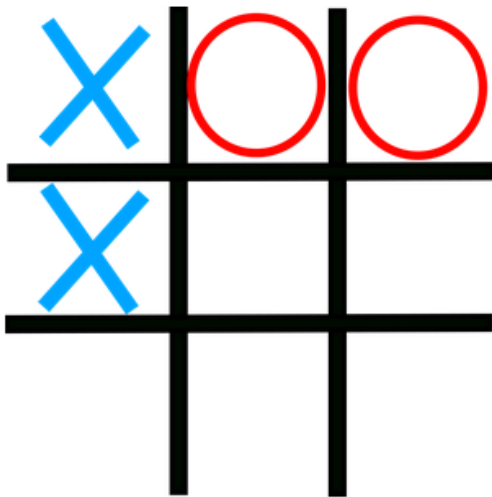
X | O | O

X | X | X

O | O | O

The state output must be $|1100\rangle$

Bonus : what if we start one step earlier and your opponent has not chosen yet, as shown in the following image, it shows with higher probability the chances of you winning. Please refer to the above considerations.

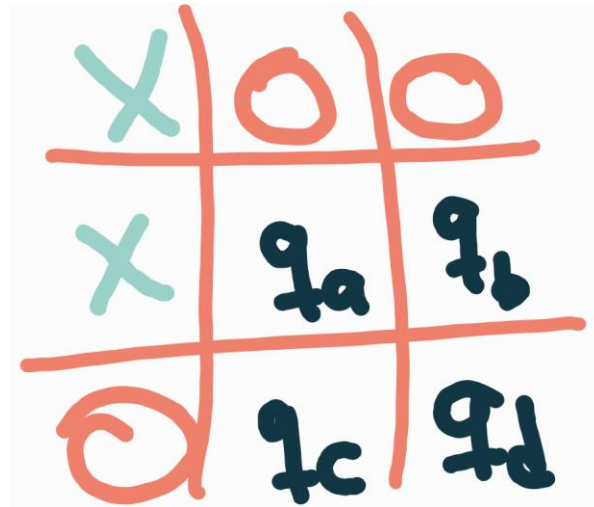


El Gato

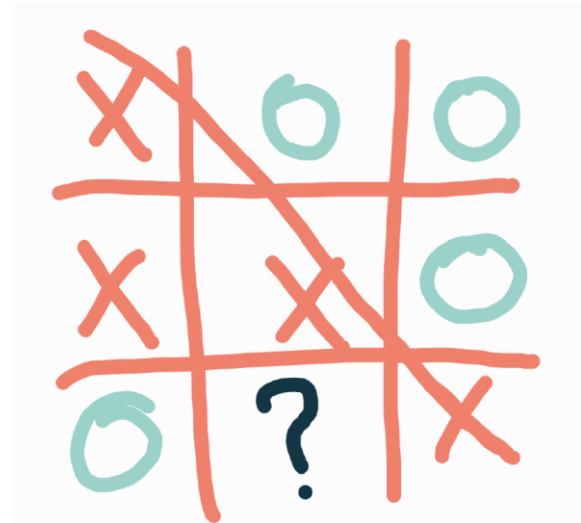
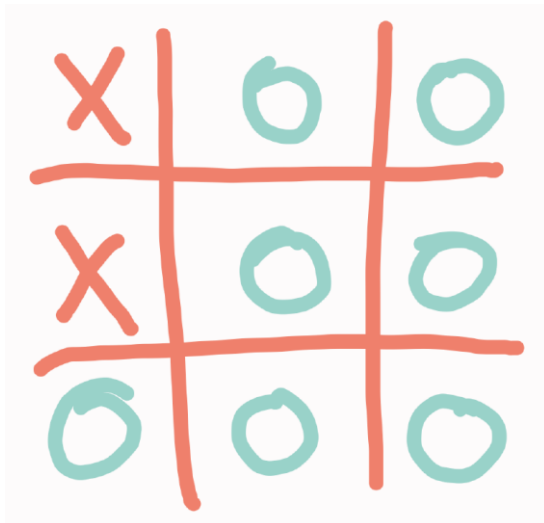
El problema consta de encontrar las probabilidades de ganar de entre todas las posibles combinaciones, que se puedan formar siguiendo una restricción ya aplicada.

Dichas restricciones constan de tener un juego de gato ya iniciado y derivado de ello la tabla de posibilidades de jugadas se ve limitada.

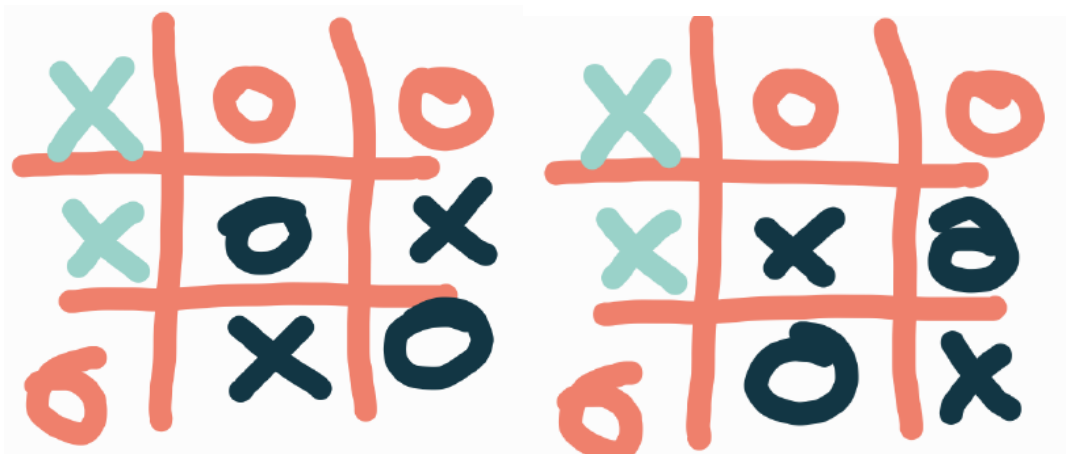
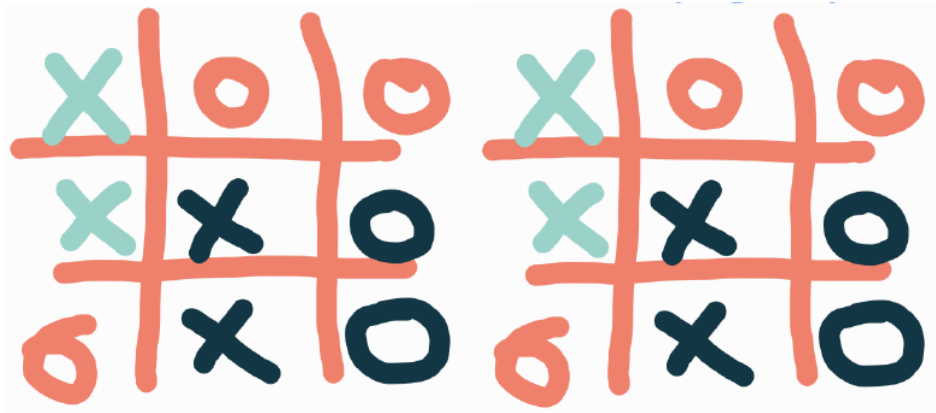
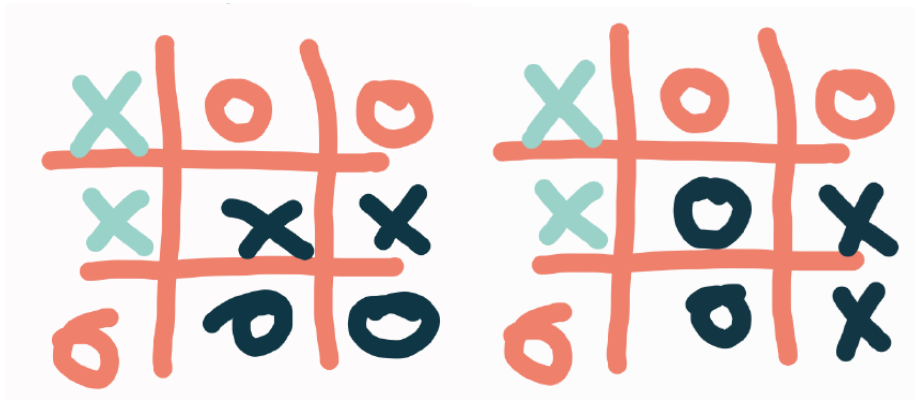
Al contar con un gato ya definido solo nos encontramos con 6 casos que puede ser posible encontrar el tablero. Dichas posiciones no están definidas por tanto el tablero se encuentra en superposición de estado definidos como $\{Q_A, Q_B, Q_C, Q_D\}$.



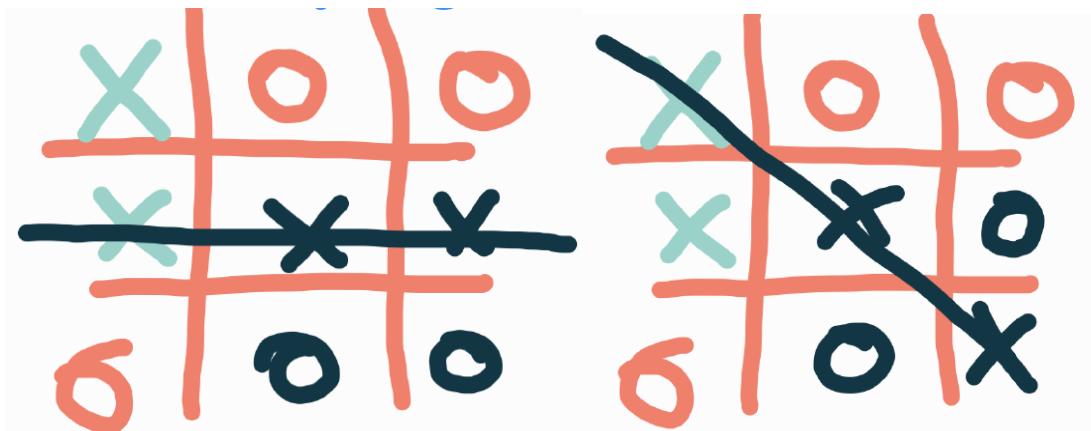
El tablero que definimos solo puede construirse de 6 maneras diferentes, aunque realmente existen muchas mas posibilidades las restricciones del juego de gato, descartan algunas de ellas. Debido a que los turnos de los jugadores se irán alternando a medida que pasa la partida, y también que alguien gana antes que el otro jugador, ya que se construye primero una fila consecutiva de patrones. No existen dichos casos.



Los posibles resultados solo se determinarían de la siguiente forma.



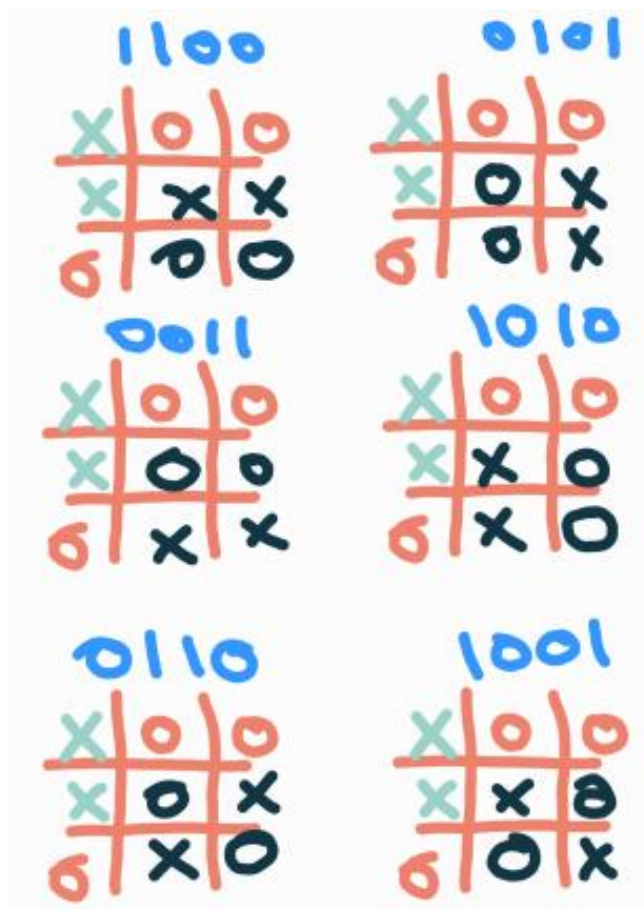
Pero como estados ganadores a:



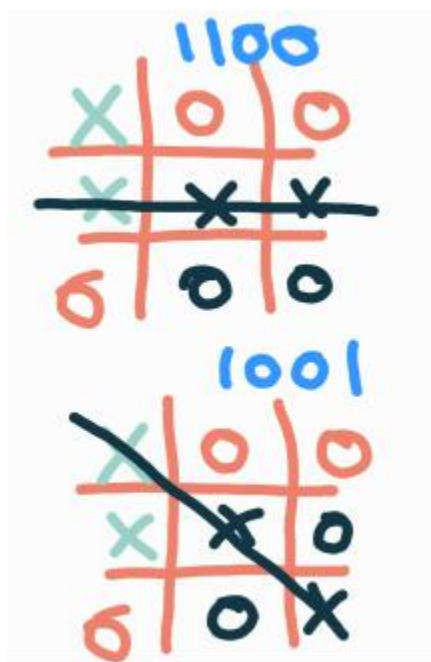
Recordando que el tablero se representa como

- 1 para el valor de X
- 0 para el valor de 0

La configuración nos permite identificar a los estados posibles, pero ahora con una representación diferente que nos ayude a identificar algún patrón.



Y aquellos estados ganadores como:



Se observa que ya se cuenta con una configuración inicial de 1001 para describir al tablero iniciando a contar los espacios de izquierda a derecha, de arriba hacia abajo. Entonces nuestra secuencia de numero se describe de la siguiente manera. (1001ab0de) tomando como variables $\{a, b, c, d\}$.

	Fijo	Regla					
	1001	a	b	c	d	e	
0		0	0		0	0	- 0
1		0	0	.	0	1	- 1
2		0	0		1	0	- 1
3		0	0	.	1	1	- 2
4		0	1		0	0	- 1
5		0	1	.	0	1	. 2
6		0	1		1	0	. 2
7		0	1	0	1	1	- 3
8		1	0	.	0	0	- 1
9		1	0		0	1	. 2
10		1	0	.	1	0	- 2
11		1	0		1	1	- 3
12		1	1	.	0	0	- 2
13		1	1		0	1	- 3
14		1	1		1	0	- 3
15		1	1	0	1	1	- 4

Con ello nos damos cuenta que ocurre algún curioso, parece que los 1 sumados dentro de la misma configuración de $\{a, b, c, d\}$. Dan un total de 2 y da la casualidad que estos valores son los posibles casos que encontraríamos al tablero.

	Fijo	Regla					
	1001	a	b	c	d	e	
0		0	0		0	0	-0
1		0	0	.	0	1	-1
2		0	0		1	0	-1
3		0	0	.	1	1	-2
4		0	1		0	0	-1
5		0	1	.	0	1	-2
6		0	1		1	0	-2
7		0	1	.	1	1	-3
8		1	0	.	0	0	-1
9		1	0		0	1	-2
10		1	0	.	1	0	-2
11		1	0		1	1	-3
12		1	1	.	0	0	-2
13		1	1		0	1	-3
14		1	1		1	0	-3
15		1	1	.	1	1	-4

Y dentro de los casos posibles solo 2 son los ganadores para X, como lo son el caso:

- 1100
- 1001

Siendo ambos con un mismo patrón de 1 sobre el primer valor a y 0 en su tercer valor c.

A	B	C	D
1	1	0	0
1	0	0	1

Una vez descrito su comportamiento procedemos a codificar.