Preparazione dello spazio di ricerca

Tutti i possibili valori che abbiamo bisogno di trovare.

Nel caso del sudoku si parla di una griglia 2x2 → 

quindi per rappresentarla servono 2 qubit: $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

di conseguenza $N = 2^2 = 4 \rightarrow \{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}$

Per risolvere il problema del sudoku, possiamo creare una funzione f che prende una possibile soluzione x e ritorna $f(x) = 0$ se non è una soluzione ($x \neq w$) e $f(x) = 1$ se è una valida soluzione ($x = w$)

L'oracolo può essere descritto come: $U_w|x\rangle = (-1)^{f(x)}|x\rangle$, dove la matrice viene descritta come:

$$U_w = \begin{bmatrix} (-1)^{f(0)} & 0 & \dots & 0 \\ 0 & (-1)^{f(1)} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & (-1)^{f(2^n-1)} \end{bmatrix}$$

Risolvere il problema del SUDOKU

Sudoku 2x2

V_0	V_1
V_2	V_3

Regole:

$V_0 \neq V_1$ riga
$V_0 \neq V_2$ colonna
$V_1 \neq V_3$ colonna
$V_2 \neq V_3$ riga

} in qiskit può diventare una lista di clausole → clause_list = [[0,1], [0,2], [1,3], [2,3]]

La funzione che ci serve in aiuto è lo XOR ($\oplus$) che:

$$XOR(x,y) = \begin{cases} 0 & \text{se } x = y \\ 1 & \text{se } x \neq y \end{cases}$$

In qiskit per implementare lo xor abbiamo bisogno di due operazioni CNOT:

$$CNOT(c,t) = \begin{cases} t & \text{se } c = 0 \\ \neg t & \text{se } c = 1 \end{cases}$$

Per implementare XOR(x, y) e salvare il risultato in un qubit di output devo:

- x e y sono i qubit di controllo (c)
- output è il qubit target (t)

Faccio prima $CNOT(x, output) = \begin{cases} 0 & \text{se } x = 0 \\ 1 & \text{se } x = 1 \end{cases}$ con output inizializzato a 0

(stiamo copiando x in output). Stessa cosa per y:

$$CNOT(y, output) = \begin{cases} output & \text{se } y = 0 \\ \neg output & \text{se } y = 1 \end{cases}$$

⚠ l'output che viene restituito dipende in input dalla prima operazione di CNOT su x.

Questi due CNOT ci restituiscono il calcolo che fa XOR:

- se $x = y$ l'output torna a 0;
- se $x \neq y$ l'output torna a 1;

esempio di esecuzione

1) $x = y = 0$ e output = 0
   - CNOT(x, output) → output = 0
   - CNOT(y, output) → output = 0 ⇒ questo ci dice che $x = y$ ✓

2) $x = 0$ $y = 1$ e output = 0

- CNOT(x, output) → output = 0
- CNOT(y, output) → output = 1 => questo ci dice che $x \neq y$ ✓

3) $x = 1$ $y = 0$ e output = 0
- CNOT(x, output) → output = 1
- CNOT(y, output) → output = 1 => questo ci dice che $x \neq y$ ✓

4) $x = 1$ $y = 1$ output = 0
- CNOT(x, output) → output = 1
- CNOT(y, output) → output = 0 => questo ci dice che $x = y$ ✓