

Approches Quantiques
pour une nouvelle
recherche opérationnelle ?!

Les TPs du mercredi (Grover)

Eric Bourreau

Université de Montpellier



Philippe Lacomme
(LIMOS)

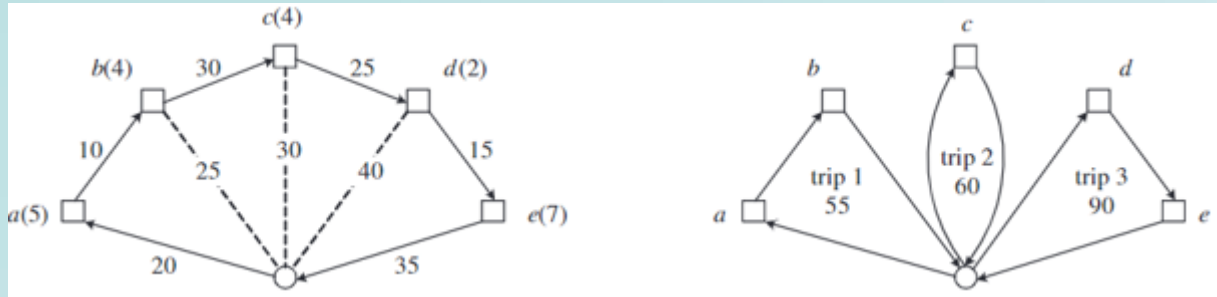
Julien Rodriguez
(Master Université de Montpellier)



Imran Meghazi



Grover et les problèmes combinatoires



$$\left\{ \begin{array}{l} \text{Max} \sum_{i=1}^n c_i x_i \\ \text{s.t.} \sum_{i=1}^n a_{ij} x_i \leq b_j \quad j = 1, \dots, m \\ \text{et } x_i \in \{0,1\} \end{array} \right.$$

- Ingrédients requis pour résoudre un problème combinatoire
 - Booléens ou Entiers (représentation binaire sur des Qubits)
 - Graphe (structure de données, modélisation)
 - Somme (sous forme d'opérateur)
 - Contraintes (sous la forme d'un Oracle)
 - Recherche de solution (opérateur de Grover) optimale

Exercices

- Exercice 1 - Savoir coder ses opérateurs
 - Application : Somme Quantique, comparateur
- Exercice 2 – Savoir encoder un Oracle
 - Application : 3-SAT (avec des booléens)
- Exercice 3 - Encoder un problème de minimisation sur un graphe
 - Application : Problème de Coloration (avec des entiers)

Exercice 1 – encoder ses opérateurs

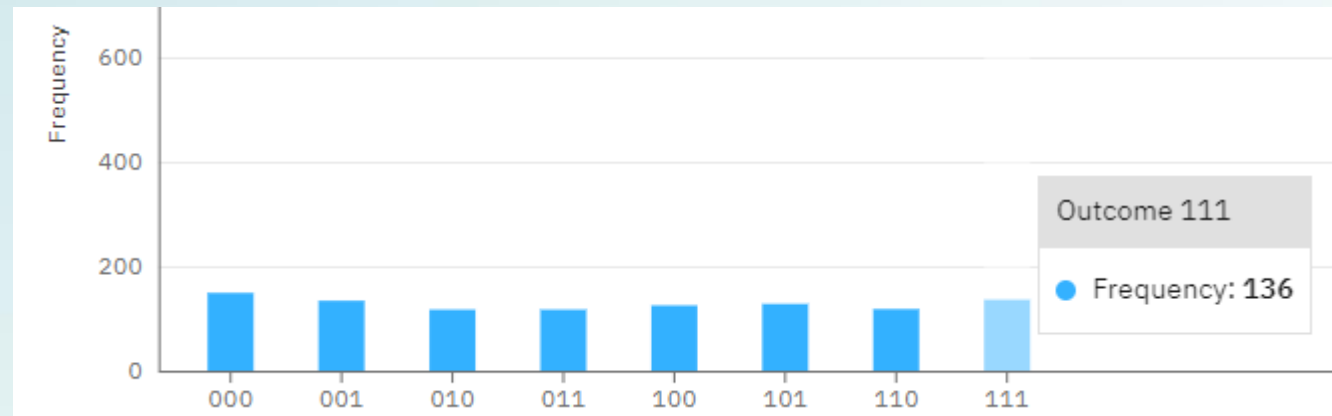
- Nous savons encoder l'opération SOMME sur des entiers
 - Addition comme à l'école (en BASE 2)
 - sur 1 QuBit
 - sur 2 QuBits
 - sur n Qubits
- Nous allons encoder l'opération SOUSTRACTION
- Nous allons encoder l'opération INFÉRIEUR (pour pouvoir minimiser)
 $f(x) < k$?

Représenter des entiers (simultanément)

Exemple : 3 qubits, 8 états



- Un registre de 3 QuBits s'écrit $|q_2q_1q_0\rangle$ (représentation binaire d'un entier)
- Chaque Qubit superpose l'état $|0\rangle$ et $|1\rangle$ équitablement (porte H)
- La lecture des 3 QuBits (portes grises) fournit aléatoirement un chiffre entre 0 et 7
- L'exécution 1000 fois de ce circuit fournit une distribution de probabilités

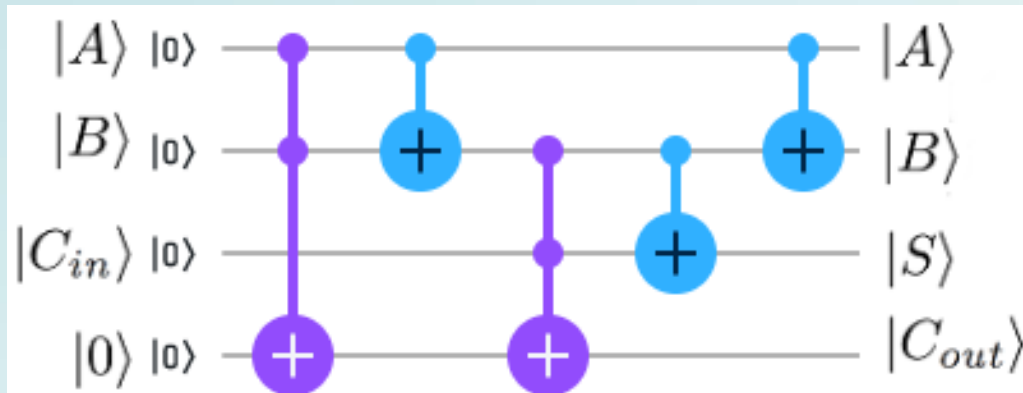


Remarque : 40 QuBits peuvent « stocker » un Tera d'états.

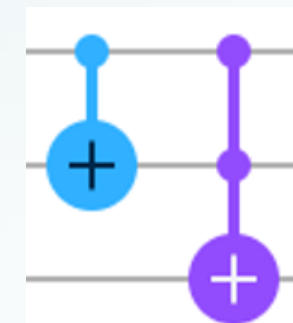
Encoder l'addition binaire (adder)

- Des entiers A, B représentés sur des registres de taille $\max(\lceil \log_2(A) \rceil, \lceil \log_2(B) \rceil)$
- Somme entre deux entiers comme à l'école : $A+B=S$ (c=carry, retenue en anglais)

$$\begin{array}{r}
 c_{n-1} \ c_n \quad c_{n-1} \ \dots \ c_2 \ c_1 \ c_0 \\
 b_n \ b_{n-1} \ \dots \ b_2 \ b_1 \ b_0 \\
 + \quad a_n \ a_{n-1} \ \dots \ a_2 \ a_1 \ a_0 \\
 \hline
 s_{n+1} \ s_n \ s_{n-1} \ \dots \ s_2 \ s_1 \ s_0
 \end{array}$$



| $ A\rangle$ | $ B\rangle$ | $ C_{in}\rangle$ | $ A\rangle$ | $ B\rangle$ | $ S\rangle$ | $ C_{out}\rangle$ |
|-------------|-------------|------------------|-------------|-------------|-------------|-------------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |



Porte Cnot Porte de Toffoli

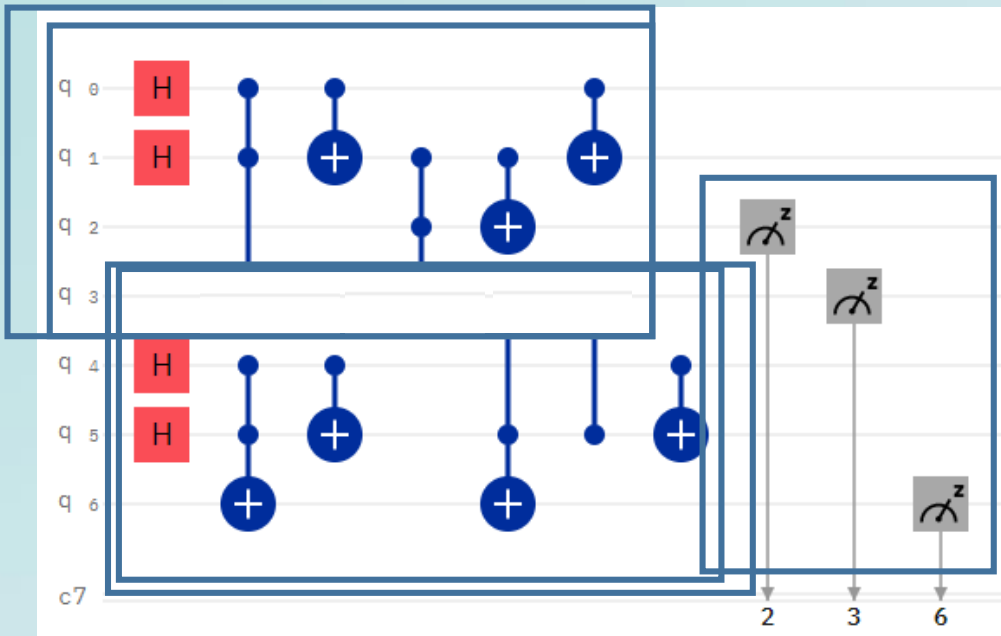
Exercice 1 – encoder ses opérateurs (TODO)

- Exécuter l'opération SOMME sur des entiers
 - sur 1 QuBit (vérifier que la table de vérité fonctionne)
 - sur 2 QuBits (superposer les QuBits de A et B en entrée)
 - Que se passe t'il sur les états des QuBits résultats (combien d'états) ?
 - Que se passe t'il sur les probabilités des états résultats ?
- Déduire comment encoder l'opération SOUSTRACTION
 - *indice* : La Soustraction est l'inverse de l'Addition
- Encoder l'opération INFÉRIEUR $f(x) \leq k$?
 - *indice* : $f(x) \leq k \iff f(x) - k \leq 0$

Correction Somme

Additionnons 2 registres A et B (stockés sur 2 QuBits) dans S (3 QuBits)

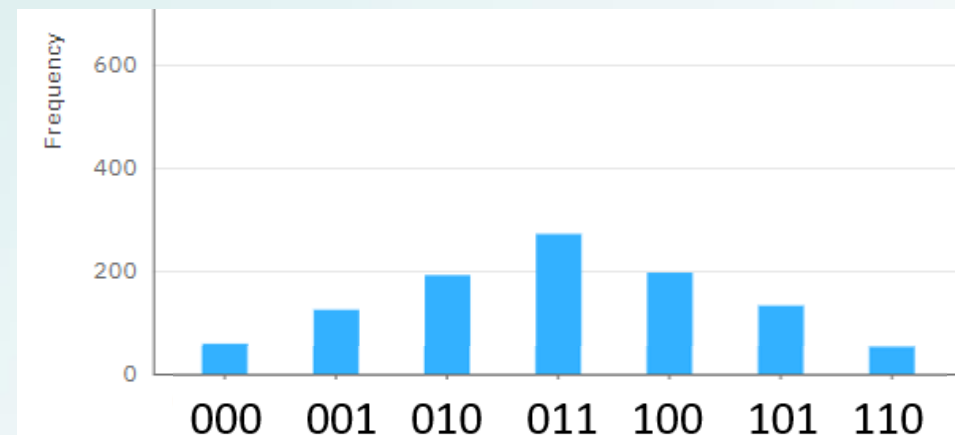
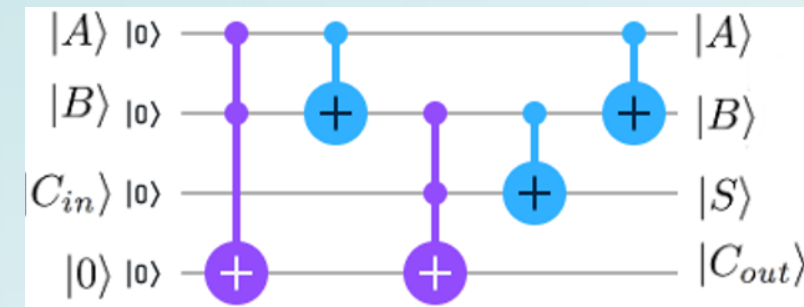
$A = |q_4q_0\rangle$ $B = |q_5q_1\rangle$ $S = |q_6q_3q_2\rangle$



a_0
 b_0
 $0 \dots s_0 \leftarrow$
 $\dots c_1 \dots s_1 \leftarrow$
 a_1
 b_1
 $\dots c_2 \leftarrow$

$$\begin{array}{c}
 c_2 \ c_1 \\
 b_1 b_0 \\
 + a_1 a_0 \\
 \hline
 c_2 s_1 s_0
 \end{array}$$

A
D
D
E
R



- Il « manque » l'état 111
- Les probabilités correspondent à la convolution des deux lois de probabilité uniforme ... en $0(10)$

Indices Bibliographiques

- Circuit itéré de la somme (et bien plus)

- « *Quantum Networks for Elementary Arithmetic Operations* », Vedral et al, *Physical Review* 95

- Circuit optimisé de la somme

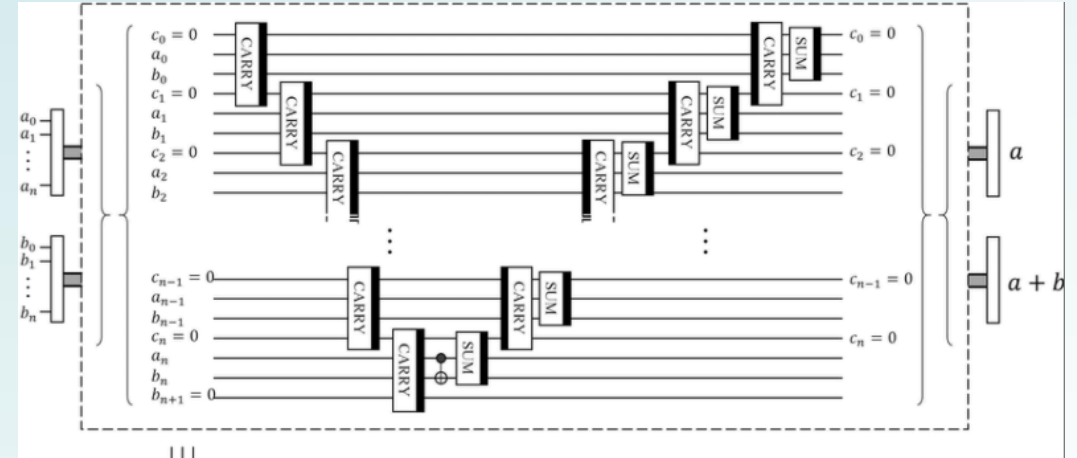
- Steven Cuccaro, Thomas Draper, Samuel Kutin, and David Moulton. *A new quantum ripple-carry addition circuit*. 11 2004. 15
 - $(a,b) \Rightarrow (a, a+b)+1 \text{ carry}$

- Circuit de la soustraction

- *Circuit de la somme de droite à gauche*
 - $(a,b) \Rightarrow (a, b-a)$

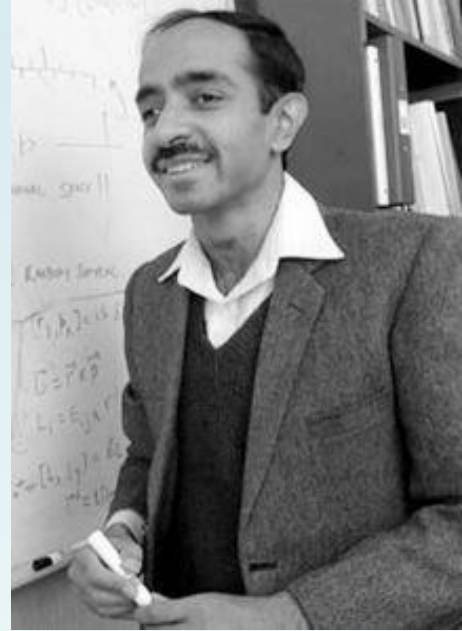
- Circuit de la comparaison

- À partir de la soustraction, on teste le résultat du carry : 0 si $b \geq a$ et 1 sinon, on inverse le carry (car on cherche le booléen $a < b$) puis on refait l'addition
 - $(a,b) \Rightarrow (a, b-a)+\text{carry} \Rightarrow (a,b)+\text{carry}$ « $a < b$ »



Algorithme de Grover

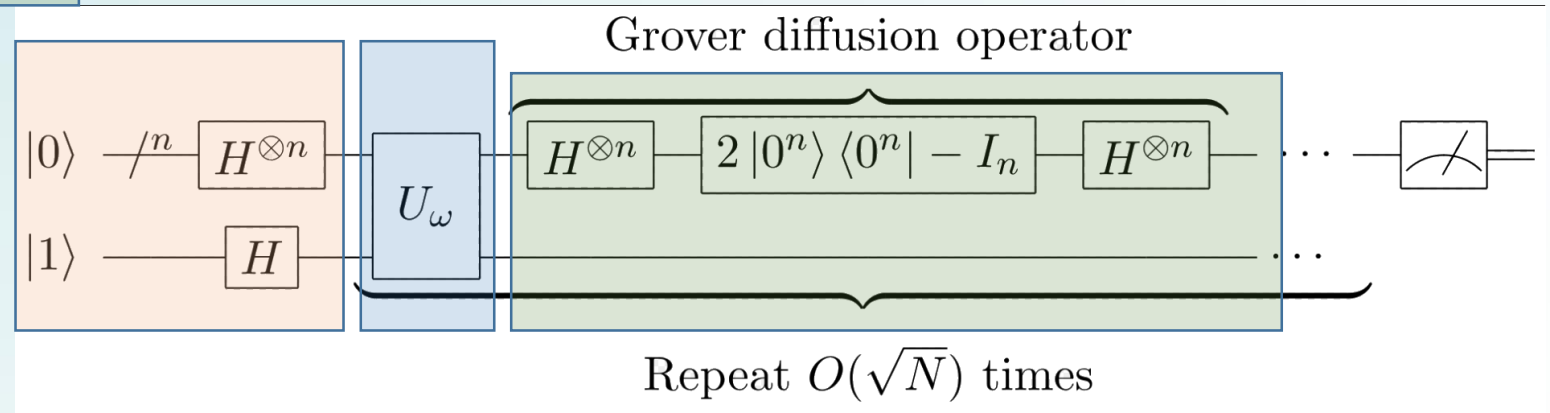
- Lov Grover (Bell Labs) propose un algorithme qui permet de trouver un élément dans une table de taille N non triée en \sqrt{N}
- 3 étapes
 - Initialisation sur n qubits des $2^n=N$ états possibles
 - Demander à un oracle (U_ω) de définir l'élément à trouver
 - Révéler où est l'élément






← Superposition

← Intrication

← Mesure

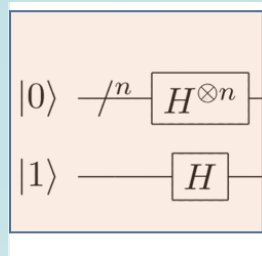


Exercice 2 – démo d'encodage d'un oracle (3-SAT)

- Une formule SAT est une formule logique constituée avec des booléens (appelés littéraux) dans une expression CNF (Conjonctive Normal Form) : un ensemble de ET (\wedge) et de OU (\vee)
- Les opérations nécessaires sont la négation (\neg), \wedge et \vee sachant que :
$$(x_0 \vee x_1) == \neg (\neg x_0 \wedge \neg x_1)$$
- Une formule 3-SAT est construite avec des clauses disjonctives (\vee) avec seulement 3 variables connectées entre elles par des conjonctions (\wedge)
- Exemple : $f = (x_0 \vee x_1 \vee \neg x_2) \wedge (\neg x_0 \vee \neg x_1 \vee \neg x_2) \wedge (\neg x_0 \vee x_1 \vee x_2)$
- Le but est de trouver les valeurs VRAI ou FAUSSE de chaque littéral x_i telle que la formule soit toujours VRAIE.
- On peut réécrire $f = \neg(\neg x_0 \wedge \neg x_1 \wedge x_2) \wedge \neg(x_0 \wedge x_1 \wedge x_2) \wedge \neg(x_0 \wedge \neg x_1 \wedge \neg x_2)$
- Nous possédons la porte NOT grâce à  et la porte ET avec CNOT  et CCNOT 

3-SAT Quantique avec Grover

- Initialisation



chaque x_i est représenté par un qubit donc les valeurs VRAI (1) ou/et FAUX (0) sont superposées

Ecrire 3-SAT



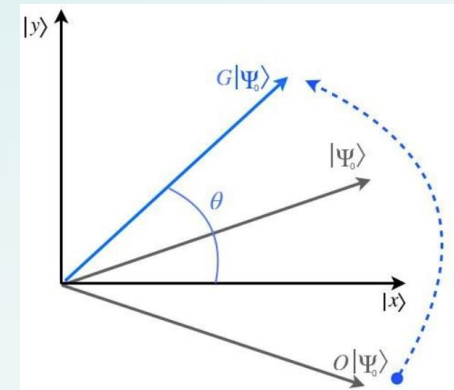
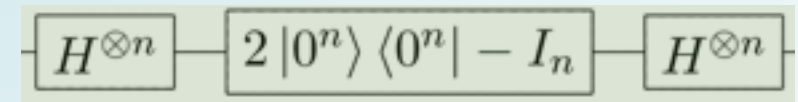
clause 0

clause 1

clause 2

$$\neg(\neg x_0 \wedge \neg x_1 \wedge x_2) \wedge \neg(x_0 \wedge x_1 \wedge x_2) \wedge \neg(x_0 \wedge \neg x_1 \wedge \neg x_2)$$

Opérateur de Grover

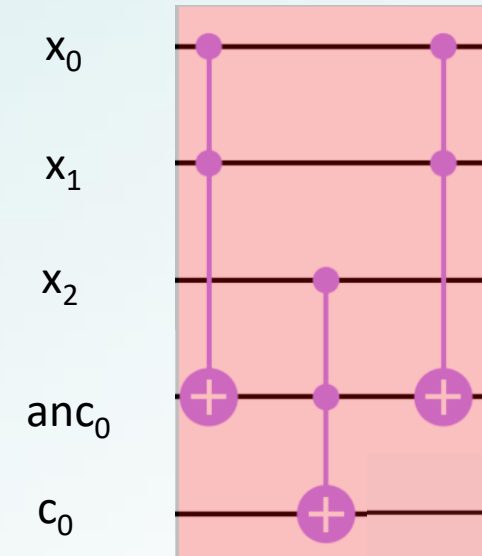
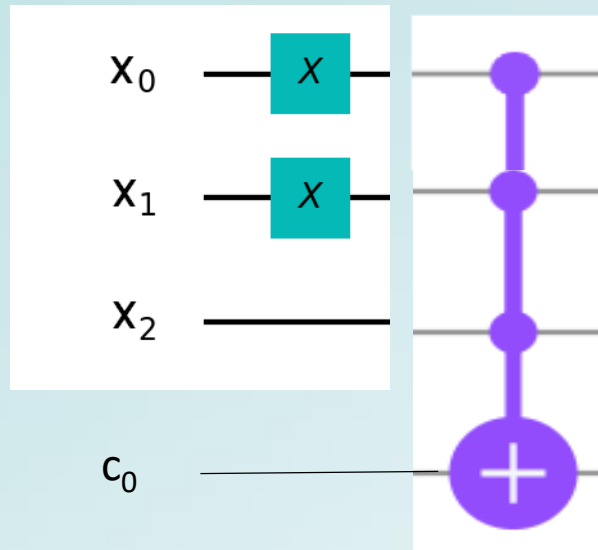


Exercice 2 – 3-SAT et les itérations

- 3-SAT
 - Superposer tous les états possible des littéraux
 - 1 booléen = 1 Qubit, déclarer les QuBits représentants les variables
 - Encoder la formule SAT
 - chaque clause = 1 nouveau Qubit résultat
 - La formule global = 1 QuBit final
 - Ajouter l'opérateur de Grover
(qui révèle la/les solutions SAT, ie QuBit final = 1)
- TODO : Tester différente valeur d'itérations. Que se passe t'il ?

Description d'une clause (disjonction)

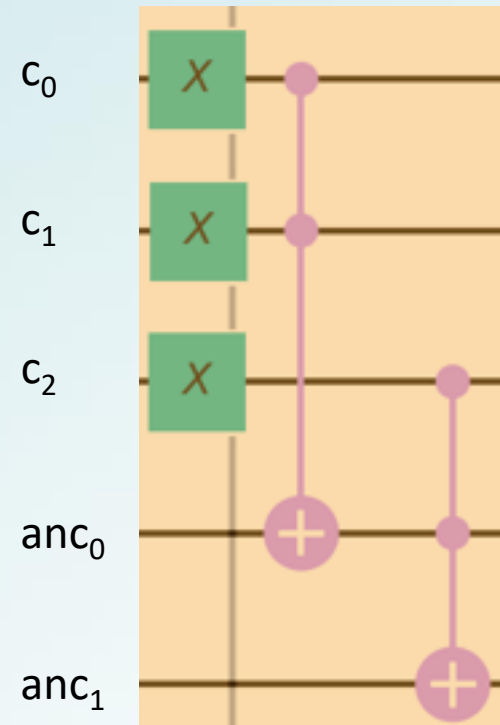
- Clause 0 :
on intrique le résultat du grand ET dans un nouveau qubit auxilliaire c_0
 - $c_0 = (\neg x_0 \wedge \neg x_1 \wedge x_2)$



- On ajoute un nouveau qubit ancilla pour stocker temporairement le résultat du ET

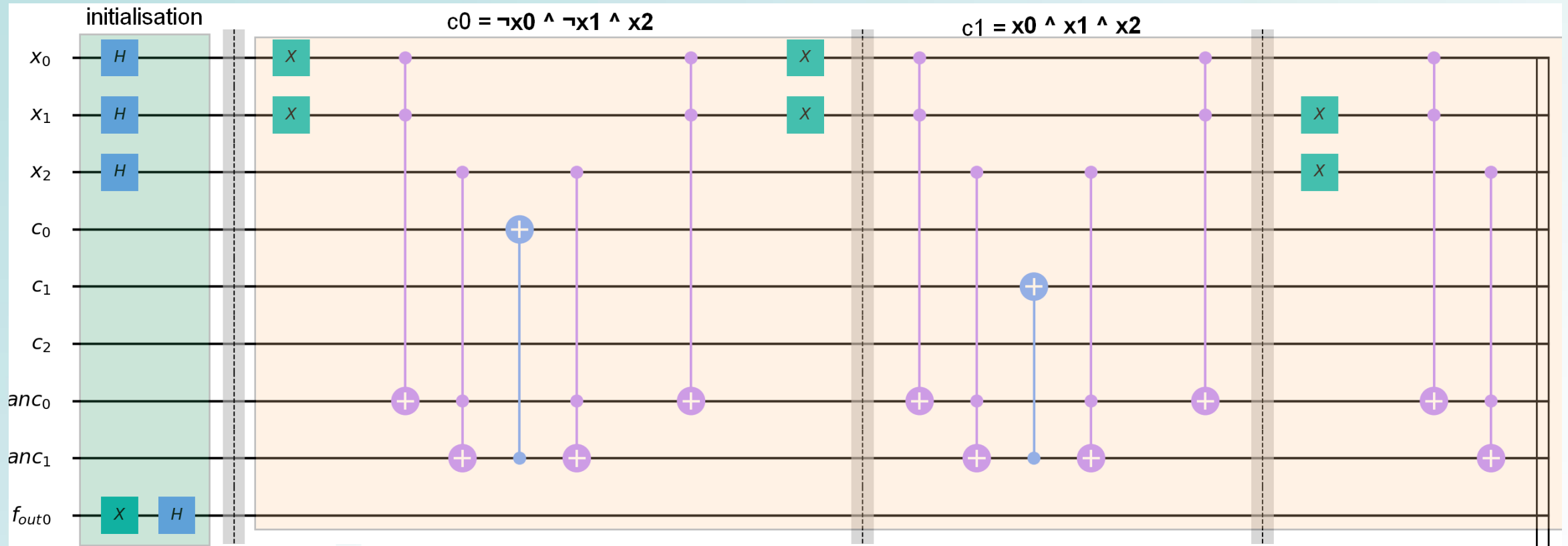
Description de la conjonction

- $f = \neg(c_0) \wedge \neg(c_1) \wedge \neg(c_2)$
- On ajoute des X devant chaque clause
- On utilise anc_0 pour stocker les résultat intermédiaire du ET
- On stocke le résultat des 3 clauses dans un nouveau quBit anc_1



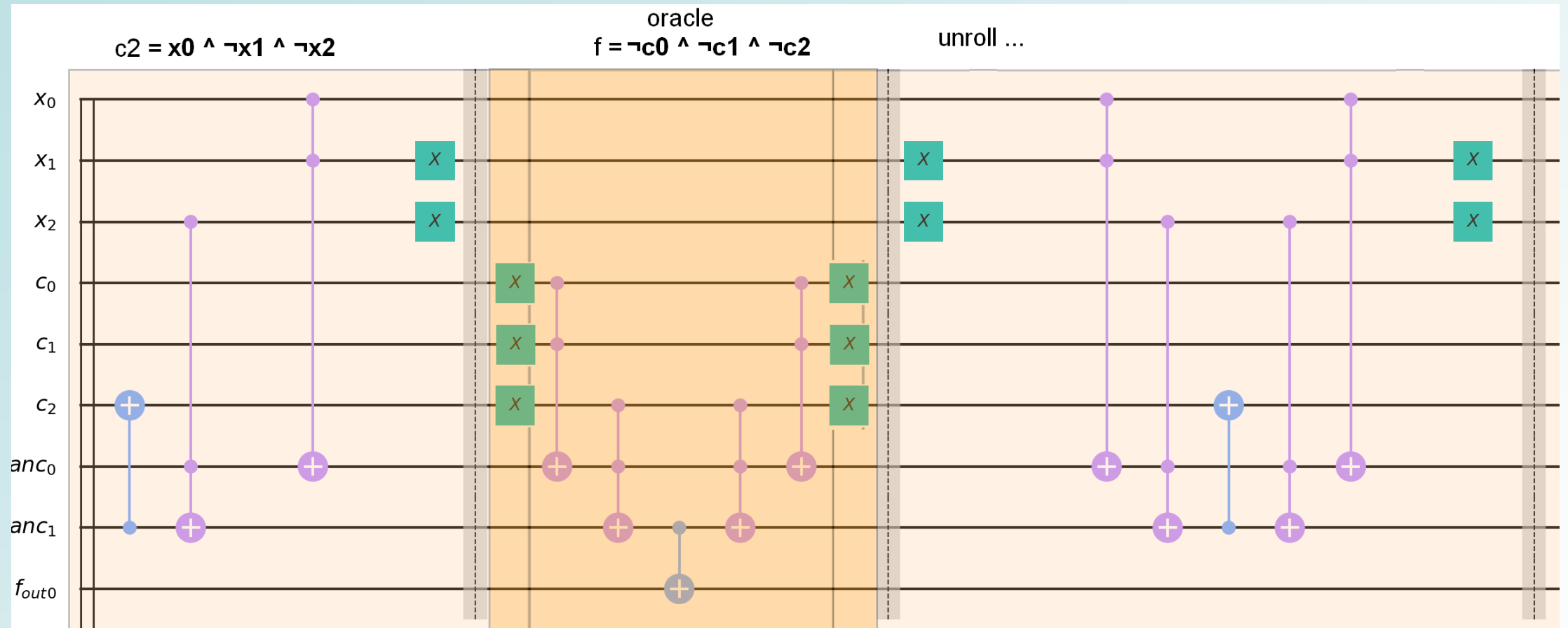
Vue d'ensemble 3-SAT

1/3



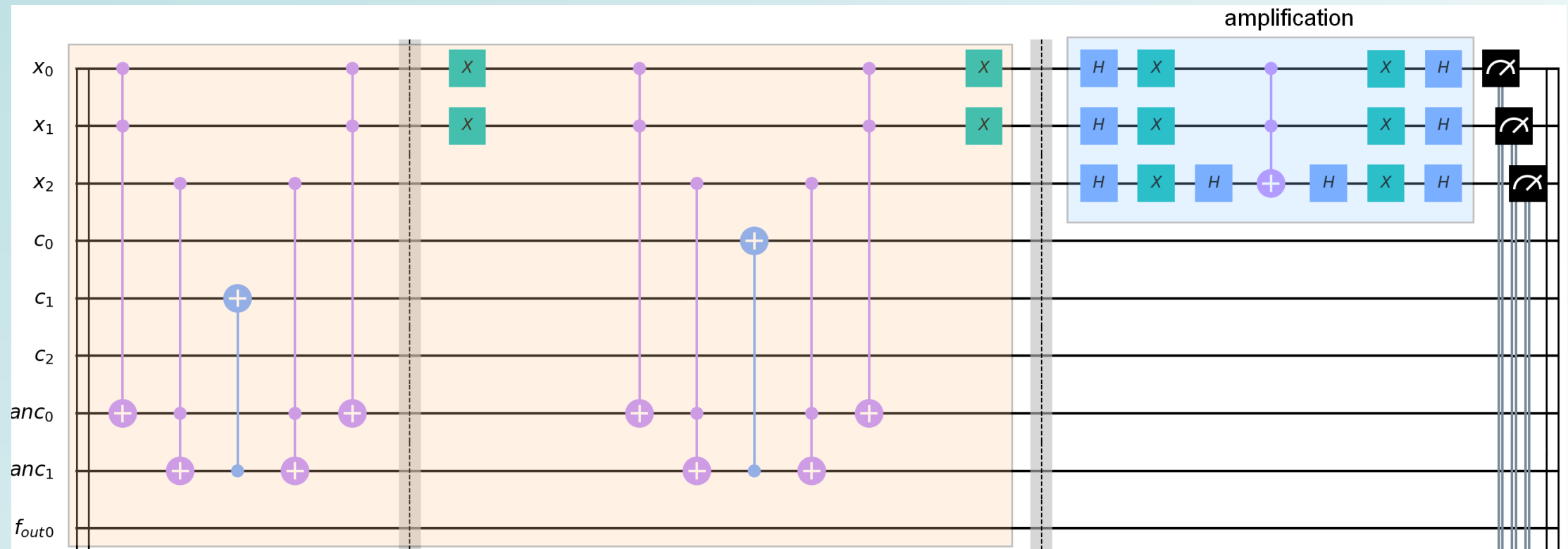
Vue d'ensemble 3-SAT

2/3



Vue d'ensemble 3-SAT

3/3

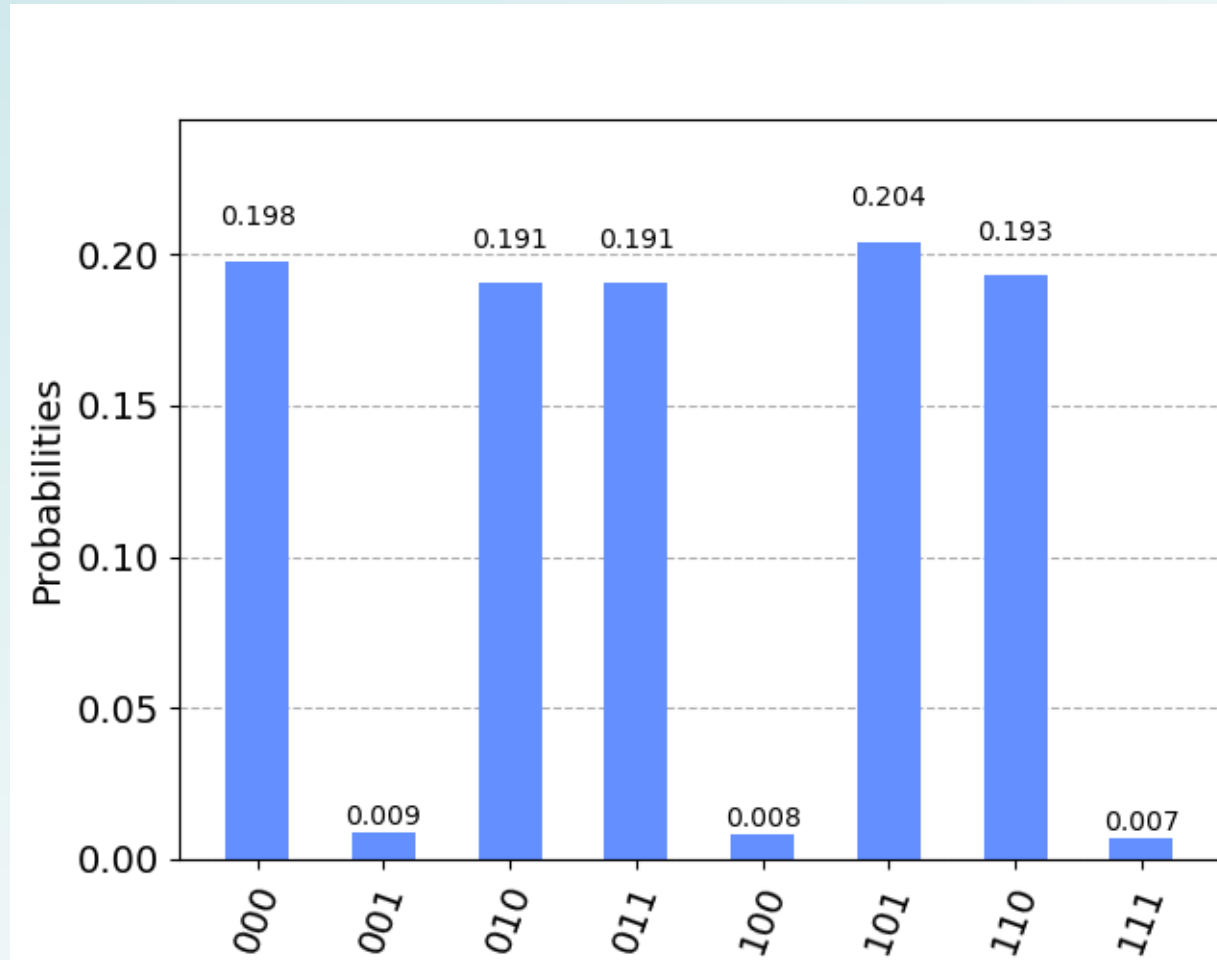


Résultat

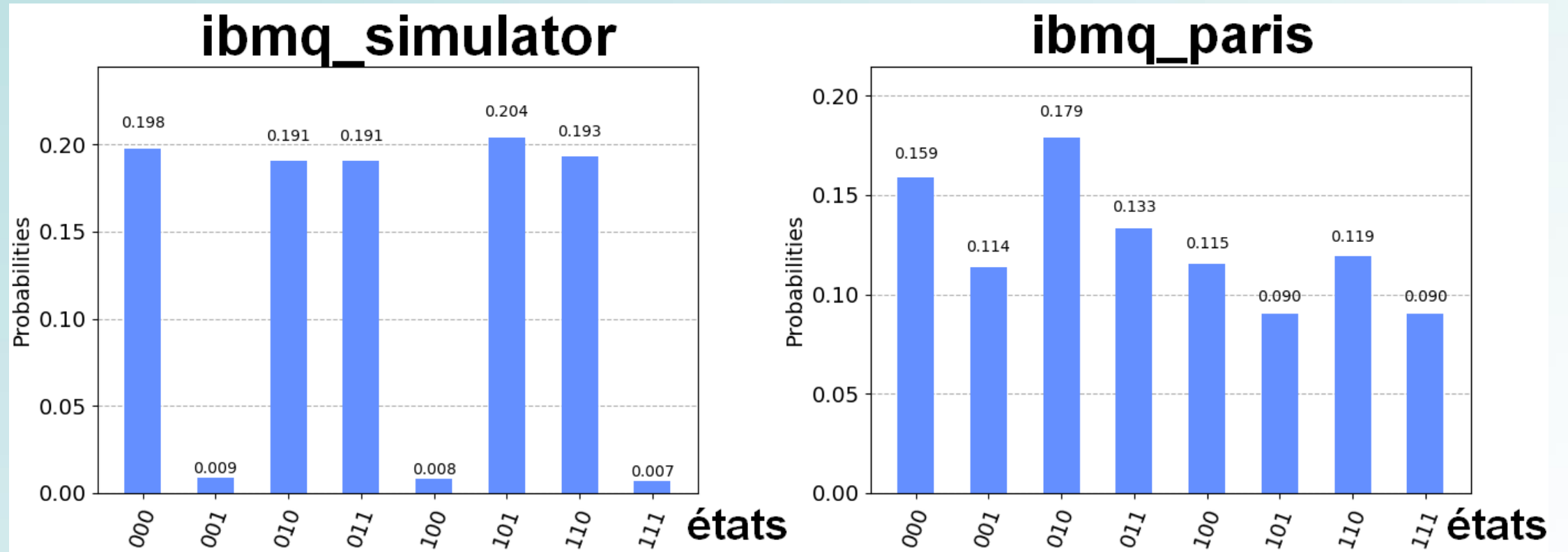
$$f = (x_0 \vee x_1 \vee \neg x_2) \wedge (\neg x_0 \vee \neg x_1 \vee \neg x_2) \wedge (\neg x_0 \vee x_1 \vee x_2)$$

- Les 5 états probables sont effectivement les solutions de la formule 3-SAT
- On ne trouve pas une mais toutes les solutions
- Cela coûte $O(\sqrt{2^n})$ en temps

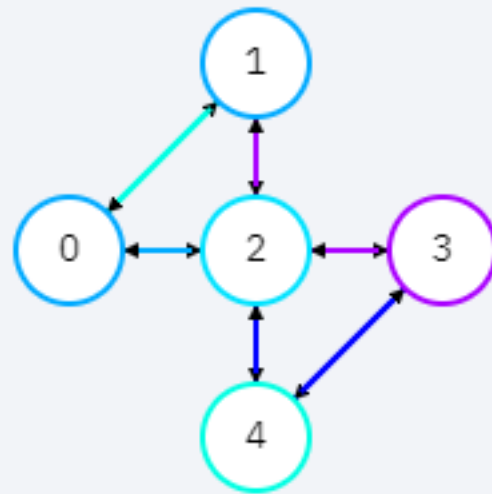
Attention !
le vecteur se lit $|x_2 x_1 x_0\rangle$



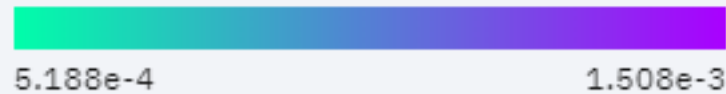
Du simulateur aux machines physiques



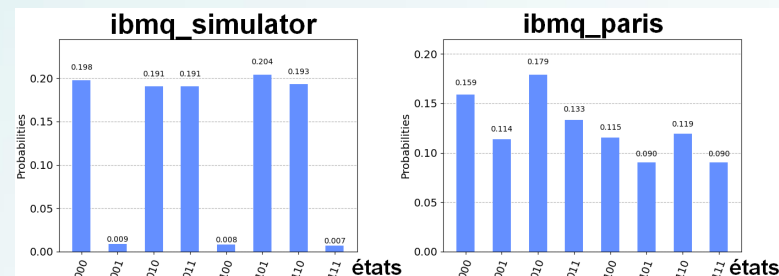
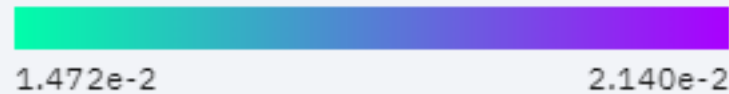
Expérimentation : retour sur terre



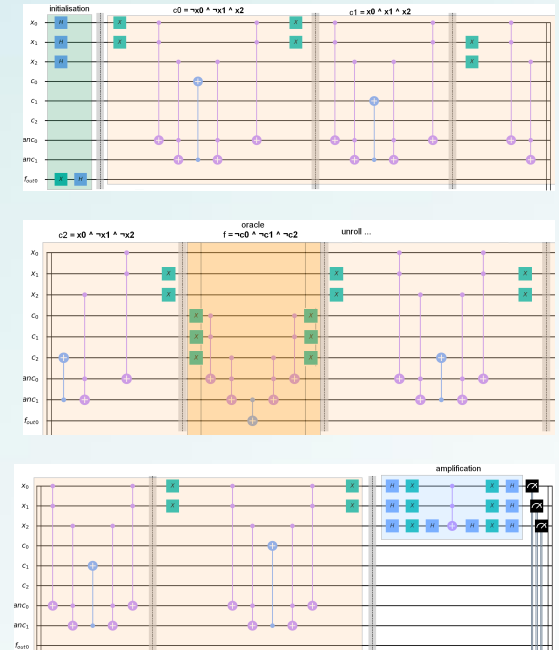
Single-qubit U2 error rate



CNOT error rate



3-SAT



40 portes unaire

94 portes binaires/ternaires
transpilées en 185/375

===

Profondeur:

$$185/3 \approx 60$$

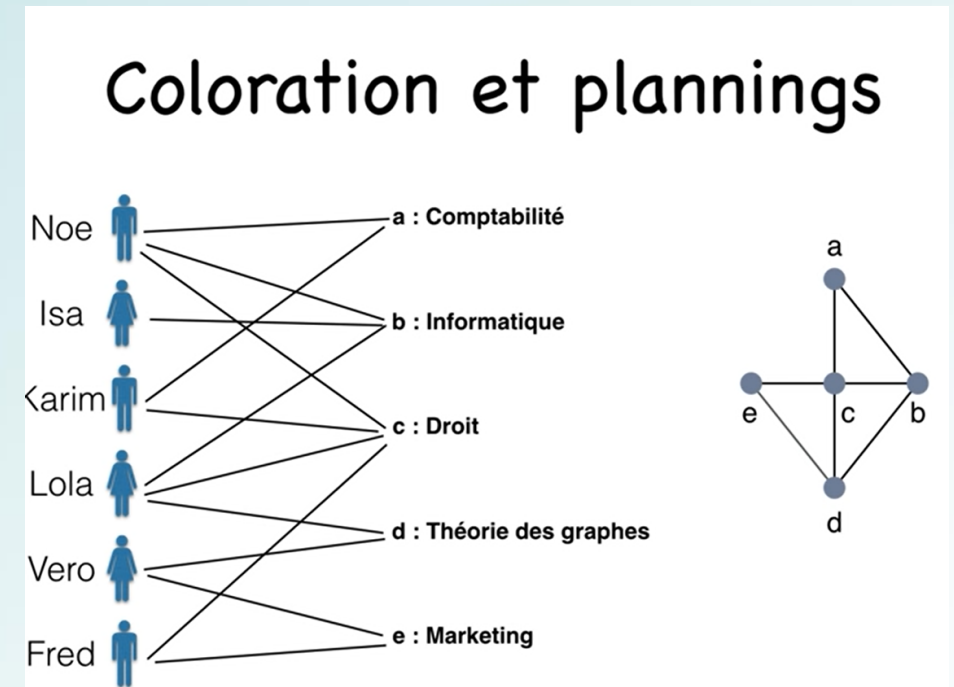
$$375/9 \approx 40$$

===

$$(99,95\%)^{60 \wedge 3} \cdot (98,5\%)^{40 \wedge 9} = 0,4\%$$

Exercice 3 – encoder un problème (coloration)

- Représenter un problème sous forme de graphe
 - Construction d'un emploi du temps
 - Une coloration du graphe des conflits fourni un planning
- Superposition
 - Variables de décision (couleur sur 2 QuBits)
 - H^{2n}
- Oracle (intrication)
 - Contraintes à satisfaire (opérateur 'différent')
 - Fonction de coût à minimiser (opérateur ' \leq ')
- Opérateur de Grover (révélateur)
- Itérer
- Mesurer



https://www.youtube.com/watch?v=CUe7LC3CdH8&ab_channel=%C3%80lad%C3%A9couvertedesgraphes

Circuit Quantique pour la coloration avec Grover

Définitions

n variables de décision (couleur) codées sur $2*n$ Qubits

m booléens exprimant les différences sur les arêtes

Validation des contraintes

Nombre Chromatique

Décision ($\leq k$?)

$|0\rangle - |1\rangle$ Grover

Qubits

c_{a1}
 c_{a2}
 c_{b1}
..
 c_{e2}

$\neq_{a,b}$
 $\neq_{a,c}$
..
 $\neq_{d,e}$

SAT

K_1
 K_2

SAT

Y

Operateurs de l'Oracle

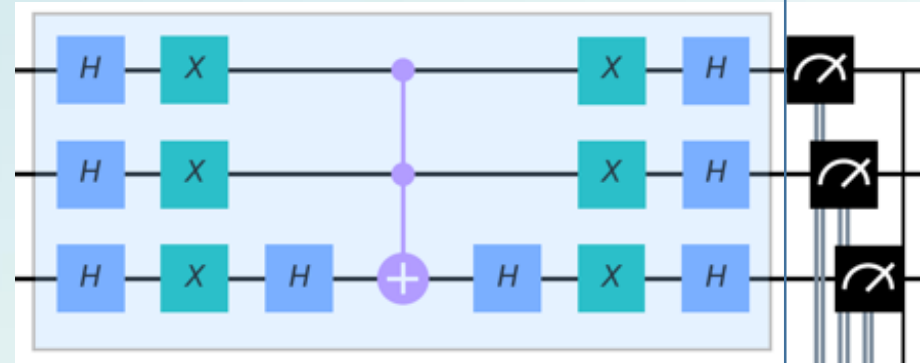
Had

\neq

\leq

\leq

Operateur de Grover



Mesure

Ne pas oublier d'itérer

X H

+

Exercice 3 - TODO

- Encoder l'opérateur 'neq' (not_equal) dans utils.py
 - ($\neq \longleftrightarrow$ not =)
 - (Deux nombres égaux le sont bit à bit)
 - Tester l'égalité de 2 bits
 - Cnot X
- Encoder les contraintes dans color.py
 - Pour toute arête la couleur des nœuds adjacents est différentes (SAT)
 - Chaque couleur est bornée par le nombre chromatique (cost function à minimiser)
- Exécuter (plusieurs fois)

