INIQ - TP#2

Philippe Deniel (philippe.deniel@cea.fr)

7 mai 2024

1 Rappel : Décomposition ABC et réduction de Sleathor-Weinfurter

1.1 Décomposition ABC

Tout porte sur 1 qubits U peut s'écrire sous la forme

$$U = e^{i\alpha}.A \times X \times B \times X \times C$$

Si on écrit U sous la forme (ou décomposition ZYZ) on a

$$U = e^{i\alpha} R_Z(\theta_2) R_Y(\theta_1) R_Z(\theta_0)$$

Alors on peut écrire la décomposition ABC en écrivant, en utilisant les angles θ_i précédemment identifiés

$$A = R_Z(\theta_2) R_Y(\frac{\theta_1}{2})$$
$$B = R_Y(\frac{-\theta_1}{2}) R_Z(\frac{-\theta_0 - \theta_2}{2})$$
$$C = R_Z(\frac{\theta_0 - \theta_2}{2})$$

1.2 Opérateur de phase globale

Pour construire le circuit qui implémente une décompositon ABC simple, il faut implémenter un opérateur de phase global, qui va multiplier la fonction d'onde par un facteur $e^{i\theta}$, cela revient à appliquer une matrice égale à l'identité multipliée par $e^{i\theta}$. On utilisera la classe `AbstractGate` de myQLM pour cela.

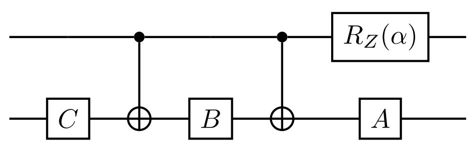
```
# Global Phase gate
from qat.lang.AQASM import AbstractGate
import numpy as np

def Phase_generator(theta):
    return np.array([[np.exp(1j * theta), 0],
                     [0, np.exp(1j * theta)]])

GlobalPhase = AbstractGate("Phase",
                           [float],
                            arity=1,
                            matrix_generator=Phase_generator)
(...)
prog.apply(GlobalPhase(-np.pi/2), qbits)
```

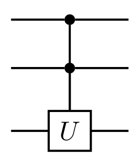
1.3 Construction d'une porte U controlée

Si U s'écrit sous la forme ABC $U = e^{i\alpha} AXBX \times C$ alors on peut écrire une porte CU ainsi

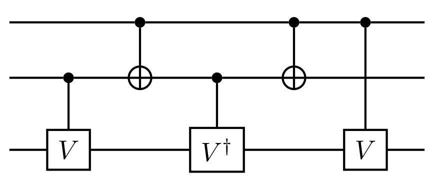


1.4 Construction d'une porte contrôlée sur 2 qubits

La décomposition de Sleathor-Weinfurter[?], utilise les racines carrées de porte.
Soit U un opérateur unitaire, soit V l'opérateur unitaire tel que $V^2 = U$, alors le circuit

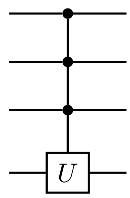


Peut s'écrire sous la forme

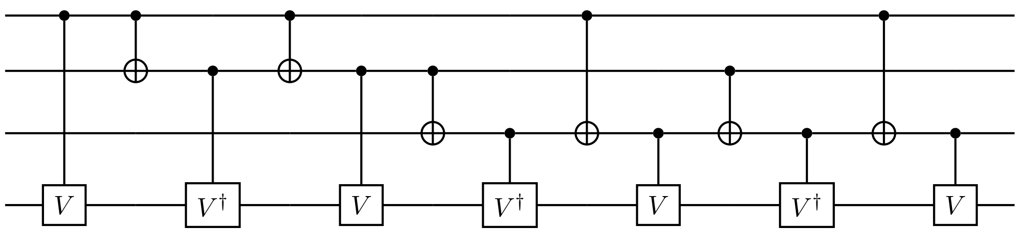


1.5 Construction d'une porte contrôlée sur 3 qubits

Le principe de réduction de Sleathor-Weinfurter peut être gnéralisée à 4 qubits. Soit U un opérateur unitaire, soit V l'opérateur unitaire tel que $V^4 = U$, alors le circuit

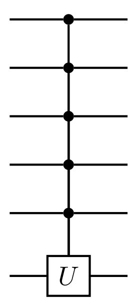


est équivalent à

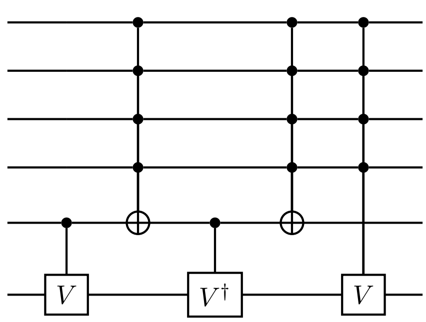


1.6 Construction d'une porte contrôlée sur n qubits

Si l'on sait construit une porte CNOT à m qubits de contrôle, on sait construire toute porte U contrôlée par m qubits pour peu que l'on dispose d'une racine carrée V de U telle que $V^2 = U$.



est équivalent à



2 Travail à faire

Pour la porte X opérant sur un qubit

1. écrire cette porte comme le produit de porte paramétrées $e^{i\alpha} R_Z(\theta_2) R_Y(\theta_1) R_Z(\theta_0)$, ou décomposition ZYZ, en déduire les valeurs des angles θ_i
2. déduire du résultat précédent les porte ABC en appliquant les formules ci-dessus
3. faire un programme myqlm, sur un qubit, qui écrit cette sous forme ABC (en utilisant l'AbstractGate pour écrire l'opérateur de phase globale) et vérifier que le résultat est bien celui attendu
4. faire un programme myqlm sur 2 qubits qui encode une porte contrôlée basée sur cette porte, grâce à la décomposition ABC
5. écrire les racines carrée et quatrième de X sous forme de porte paramétrée
6. en utilisant la racine carrée de X, faire un programme myqlm qui encode une porte doublement contrôlée basée sur cette porte,
7. en utilisant la racine quatrième de X, faire un programme qui encode une contrôlée sur 3 qubits toujours basé sur la même porte
8. écrire un programme générique qui écrit une porte contrôléée sur n qubits à base de NOT contrôlée sur plusieurs qubits. Ecrivez en particulier une sous-routine qui implémente la "CNOT à n qubits de contrôle" de manière récursive.

Pour chaque programme myQLM, donner le nombre de portes à un qubit et de CNOT sur 2 qubits utilisées (y compris pour le dernier). Montrer comment le "coût en portes" évolue avec le nombre de qubits de contrôle.