QCOMPPW100: Quantum computing kickoff practical work

Lotus Noir Quantum Computing Research Group

Based on:
MITx: 8.370.1x Quantum Information Science I, Part I

Practical work Outline: During This practical work we will simulate a quantum computer

Contents

1 Generate state vector

Program the function getStateVector.This function takes in input, the two integers: nbQbit and state.This function output a vector of $2^n bQbit$ values, with the value $state^t h$ value being equal to one and the other values must be equal to zeroes. This function must pass this test:

```
nbQbit = 3
state = 5
assert(str(getStateVector(nbQbit,state)) == "[0,0,0,0,0,1,0,0]")
```

2 compute matrix thanks to kronecker product

Program the function computeMatrix. This function takes in input an integers : nbQbit, a 2x2 matrix of complex number: matrix, and an integer fQbit. This function output: $I_2^{\otimes fQbit} \otimes matrix \otimes I_2^{\otimes nbQbit - fQbit}$. This function must pass this test:

```
nbQbit = 2
matrix = np.array([[0,1],[1,0]])
fQbit =1
assert(str(computeMatrix(nbQbit,matrix, fQbit)) == """[[0. 1. 0. 0.]
[1. 0. 0. 0.]
[0. 0. 0. 1.]
[0. 0. 1. 0.]]""")
```

3 compute CNOT

Program the function computeCNOT. This function takes in input an integers: nbQbit, an integers fQbit and integer sQbit. This function must output a such kronecker product: FIX-THAT

4 compute product

Program the function computeProduct. This function takes in input this array Matrixs of square matrixs of the same dimension. This function must ouput their product. This function must pass this test:

5 compute Probability

Program the function computeProbability. This function takes in inputs an array of complex number: tab. This function output an array of the same dimension. Every coefficient of the output is computed this way:

$$output[i] = \frac{|tab[i]|^2}{\Sigma_{k=0}^{tab.length()} |tab[k]|^2}$$

6 package everything

6.1 Quantum gates

Program the class quantum gates

6.2 Quantum computer

Program the function quantumCOmputer, that takes in input an integers: nbQbit, array of integer having on value at 0 and every other values at one: input and an array of QuantumGate: quantumGates. This function must output the computed probability of the product of the input and the product of all this nQbit×nqBit matrices create from the quantum gates.
This function must pass this test