ENG1116 - PUC-Rio, 2020.2

Trabalho 5: ENG1116

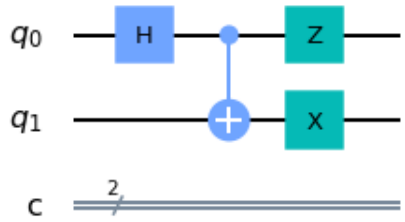
16/11/2020

Professor: Guilherme Temporão e Thiago Guerreiro *Aluno: Rafael Vilela*

1 Questão 1 - Preparação dos Circuitos Quânticos

1.1 1.1

O circuito quântico que gera o estado de Bell correspondente como visto em trabalhos anteriores, $|\psi-\rangle$, mais 2 cbits:



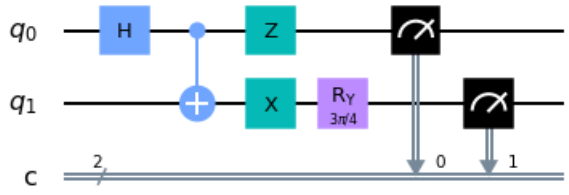
(1)

1.2 1.2

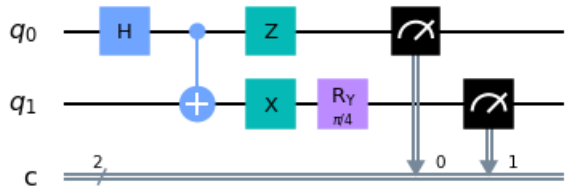
Os operadores unitários necessários (para as medidas) serão: Hadamar e Ry (variando o ângulo). Eles vão levar os respectivos qubits (q0 e q1) para os respectivos observáveis Z, X, $\frac{1}{\sqrt{2}} \cdot (-Z - X)$ ou $\frac{1}{\sqrt{2}} \cdot (Z - X)$, conforme pedido no enunciado. Os circuitos estão no próximo item.

1.3 1.3

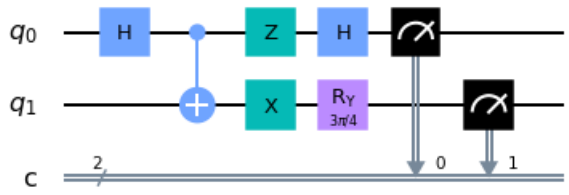
A seguir a representação dos 4 circuitos quânticos com medidas projetivas, respectivamente A1B1, A1B2, A2B1 e A2B2.


q_0
q_1
c
H
Z
X
R_Y
3π/4
2
0
1

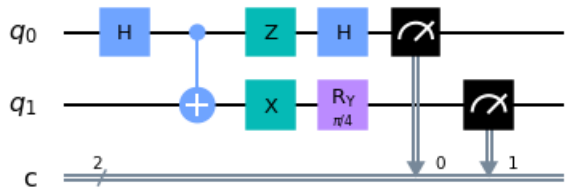
(2)


q_0
q_1
c
H
Z
X
R_Y
π/4
2
0
1

(3)


q_0
q_1
c
H
Z
H
X
R_Y
3π/4
2
0
1

(4)


q_0
q_1
c
H
Z
H
X
R_Y
π/4
2
0
1

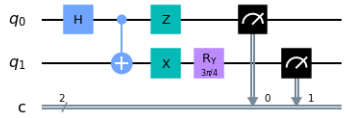
(5)

2 Questão 2 - Medindo o parâmetro S

2.1 2.1

Resultados dos valores médios ($\frac{1}{\sqrt{2}}$) e de S no print. Conforme dito no enunciado, há uma substituição de valores 0 e 1 por -1 e 1. A conversão está no código enviado no outro pdf , que corresponde ao argumento "observable correlated" do "average data".

```
########################################################################

measureZZ2 = QuantumCircuit(2, 2)

eva = randint(2, size=1)  #primeiro qubit

#if (eva==1):
    #nada
if (eva==0):
        measureZZ2.x(0)
        measureZZ2.ry(3*math.pi/4,0)

eva = randint(2, size=1)  #segundo qubit

if (eva==1):
    measureZZ2.ry(3*math.pi/4,1)
if (eva==0):
    measureZZ2.x(1)


measureZZ2.measure(0, 0)
measureZZ2.measure(1, 1)
bellZZ2 = qc+measureZZ2

bellZZ2.draw(output='mpl')
```



(6)

2.2 2.2

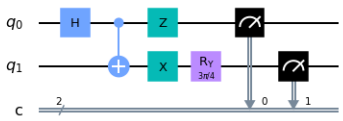
O valor de S (em módulo) calculado foi próximo ao valor na teoria que seria (em módulo) cerca de 2.82, acima do valor de 2 do modelo do Realismo local, comprovando assim a violação da desigualdade de Bell.

3 Questão 3 - Estados separáveis

3.1 3.1

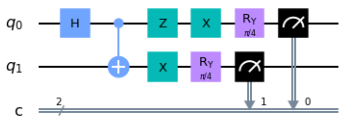
Circuitos modificados com condicionais (if) que dependem do "randint", que representa a ação de Eva, sendo "1" o envio correto e "0" o não correto, ambos com 50 % de chance:

```
########################################################################

measureZZ2 = QuantumCircuit(2, 2)

eva = randint(2, size=1)  #primeiro qubit

#if (eva==1):
    #nada
if (eva==0):
        measureZZ2.x(0)
        measureZZ2.ry(3*math.pi/4,0)

eva = randint(2, size=1)  #segundo qubit

if (eva==1):
    measureZZ2.ry(3*math.pi/4,1)
if (eva==0):
    measureZZ2.x(1)


measureZZ2.measure(0, 0)
measureZZ2.measure(1, 1)
bellZZ2 = qc+measureZZ2

bellZZ2.draw(output='mpl')
```



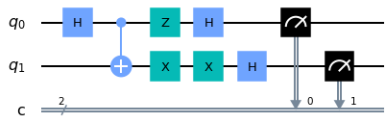
(7)

```
measureXX2 = QuantumCircuit(2, 2)

eva = randint(2, size=1) #primeiro qubit

#if (eva==1):
    #nada
if (eva==0):
        measureXX2.x(0)
        measureXX2.ry(1*math.pi/4,0)


eva = randint(2, size=1) #segundo qubit

if (eva==1):
    measureXX2.ry(1*math.pi/4,1)
if (eva==0):
    measureXX2.x(1)


measureXX2.measure(0, 0)
measureXX2.measure(1, 1)
bellXX2 = qc+measureXX2

bellXX2.draw(output='mpl')
```



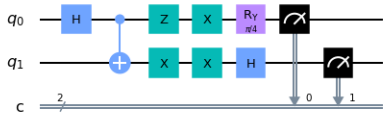
(8)

```
measureZX2 = QuantumCircuit(2, 2)

eva = randint(2, size=1)  #primeiro qubit

if (eva==1):
    measureZX2.h(0)
if (eva==0):
    measureZX2.x(0)
    measureZX2.ry(3*math.pi/4,0)


eva = randint(2, size=1)  #segundo qubit

if (eva==1):
    measureZX2.ry(3*math.pi/4,1)
if (eva==0):
    measureZX2.x(1)
    measureZX2.h(1)


measureZX2.measure(0, 0)
measureZX2.measure(1, 1)
bellZX2 = qc+measureZX2

bellZX2.draw(output='mpl')
```



(9)

```
measureXZ2 = QuantumCircuit(2, 2)

eva = randint(2, size=1)  #primeiro qubit

if (eva==1):
    measureXZ2.h(0)
if (eva==0):
    measureXZ2.x(0)
    measureXZ2.ry(1*math.pi/4,0)


eva = randint(2, size=1) #segundo qubit

if (eva==1):
    measureXZ2.ry(1*math.pi/4,1)
if (eva==0):
    measureXZ2.x(1)
    measureXZ2.h(1)

measureXZ2.measure(0, 0)
measureXZ2.measure(1, 1)
bellXZ2 = qc+measureXZ2

bellXZ2.draw(output='mpl')
```



(10)

```
result22 = job2.result()
result22.get_counts() #equivalem aos seguintes resultados
#result22.get_counts(bellZZ2)
#result22.get_counts(bellXX2)
#result22.get_counts(bellZX2)
#result22.get_counts(bellXZ2)
```

```
[{'00': 4311, '01': 745, '10': 699, '11': 4245},
 {'00': 2539, '01': 2512, '10': 2460, '11': 2489},
 {'01': 5049, '10': 4951},
 {'00': 4099, '01': 757, '10': 784, '11': 4360}]
```

```
res11=(average_data(result22.get_counts(bellZZ2),observable_correlated))
res22=(average_data(result22.get_counts(bellXX2),observable_correlated))
res33=(average_data(result22.get_counts(bellZX2),observable_correlated))
res44=(average_data(result22.get_counts(bellXZ2),observable_correlated))

s2= -res22+res33+res44 +res11;
s2=str(s2)

res11=str(res11)
res22=str(res22)
res33=str(res33)
res44=str(res44)

print('res11 = ' + res11)
print('res22 = ' + res22)
print('res33 = ' + res33)
print('res44 = ' + res44)
print('s2=' + s2)
```

```
res11 = 0.7111999999999999
res22 = 0.005600000000000049
res33 = -1.0
res44 = 0.6918
s2=0.39739999999999986
```

(11)

3.2 3.2

É possível observar que o valor de S não continua acima do valor (em módulo) de 2. Somente estados emaranhados violam a desigualdade de Bell, estados separavéis não violam, vão ser sempre menor ou igual a 2, em módulo. Nesse caso há uma variável escondida que reproduz os resultados da teoria quântica para um estado puro de um qubit.

O código foi baseado e adaptado da primeira referência.

References

[1] *https://github.com/qiskit-community/qiskit-community-tutorials/blob/master/terra/qis*$_i ntro/entanglement_t e$

[2] *http://theory.caltech.edu/ preskill/ph229/notes/chap4*$_0 1.pdf$.

[3] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[3] [1] [2]