Interacção e Concorrência

Teste - 26 Maio, 2025 (14:00 - 16:00 - Ed. 2, Sala 1.01)

Nota: *O teste é composto por 10 questões, cada uma cotada para 2 valores. Pode ser realizado com consulta de qualquer tipo de material impresso ou manuscrito.*

Questão 1

Relembre as seguintes portas quânticas:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

e considere ainda a porta definida à custa de X e Z pela seguinte expressão:

$$G = \frac{1}{\sqrt{2}}(X + Z)$$

1. A porta G é muito usada em algoritmos quânticos. Identifique-a e indique o seu propósito.

   Sugestão de resolução

$$\begin{aligned} G &= \frac{1}{\sqrt{2}}(X + Z) \\ &= \frac{1}{\sqrt{2}}\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\right) \\ &= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= H \end{aligned}$$

   Trata-se, pois, da porta de Hadamard, essencialmente utilizada para criar estados quânticos em sobreposição uniforme.

2. Mostre que $Z = GXG$.

   Sugestão de resolução

$$GXG = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Z$$

3. Considere agora o operador

$$C_G|x\rangle|y\rangle = |x\rangle \otimes G^x|y\rangle$$

   Explique o seu comportamento e mostre que é unitário.

   Sugestão de resolução

   Trata-se da versão da porta G controlada pelo primeiro qubit: a porta é aplicada na ocorrência de uma componente em $|1\rangle$ no primeiro qubit. Para mostrar que se trata de um operador unitário verifiquemos que $C_G C_G = I$,

$$C_G C_G^\dagger|x\rangle|y\rangle = C_G\left(|x\rangle \otimes (G^\dagger)^x|y\rangle\right) = |x\rangle \otimes G^x(G^\dagger)^x|y\rangle = \begin{cases} |0\rangle \otimes |y\rangle & \text{if } x = 0 \\ |0\rangle \otimes GG^\dagger|y\rangle & \text{if } x = 1 \end{cases} = |x\rangle|y\rangle$$

   porque $GG^\dagger = I$ uma vez que G é unitário.

Questão 2

Considere o seguinte estado quântico

$$|\psi\rangle \; = \; \sqrt{0.2}|00\rangle + \sqrt{0.8}|11\rangle$$

Escreva um programa em pennylane que reproduza este estado e estime as probabilidades de cada um dos estados base.

Sugestão de resolução

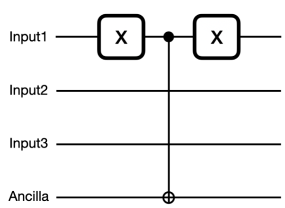
```python
import pennylane as qml

n_qubits = 2

#create device
dev = qml.device("default.qubit", wires=n_qubits)

#binding to python function
@qml.qnode(dev)
def qcircuit(angle):
    #create non uniform superposition
    qml.RY(angle , wires=0)
    #create entanglement
    qml.CNOT(wires=[0,1])

    #return probabilities
    return qml.probs(wires=range(n_qubits))


# we want probabilities [0.2, 0.8]
# Ry(\theta)|00> = cos(\theta / 2) |00> + sin(\theta/2) |10>
# \theta = 2 * arcsin(\sqrt(0.8))

angle = 2*np.arcsin(np.sqrt(0.8))

#execute circuit
probs = qcircuit(angle)

print(probs)


```

✓ 0.0s

```
[0.2 0.  0.  0.8]
```

Questão 3

Considere o seguinte oráculo para o algoritmo de Deutsch-Jozsa.



Identifique o tipo de função e proponha uma implementação em pennylane que classifique a função.

Sugestão de resolução

```python
import pennylane as qml

inputs = ["input1", "input2", "input3"]
ancilla = ["ancilla"]

#create device
dev = qml.device("default.qubit", wires=inputs+ancilla)

#binding to python function
@qml.qnode(dev)
def qcircuit():
    #create uniform superposition on function inputs
    qml.broadcast(qml.Hadamard , wires=inputs, pattern="single")

    #create ancilla in the |-> state
    qml.PauliX(wires="ancilla")
    qml.Hadamard(wires="ancilla")

    qml.Barrier()

    #function oracle
    qml.PauliX(wires="input1")
    qml.CNOT(wires=["input1", "ancilla"])
    qml.PauliX(wires="input1")
    qml.Barrier()

    #create interference
    qml.broadcast(qml.Hadamard , wires=inputs, pattern="single")

    #return probabilities
    return qml.probs(wires="input1")


#draw the circuit
qml.draw_mpl(qcircuit)()

#execute circuit
probs = qcircuit()

print(probs)
```

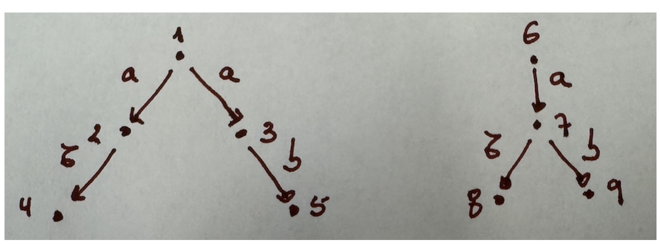
0.1s Python

Questão 4

Construa dois sistemas de transição distintos, com um mínimo de 4 estados cada, ambos com transições τ, tal que os respectivos estados iniciais, s_0 e t_0, reconhecem os mesmos traços mas não satisfazem a propriedade $s_0 \approx t_0$. Justifique a sua construção.

Sugestão de resolução

Consideremos os seguintes sistemas com $s_0 = 1$ e $t_0 = 6$.



Claramente os traços gerados a partir de s_0 ou de t_0 são iguais: $\{\epsilon, a, ab, a\tau\}$. No entanto, $1 \not\approx 6$, para o que basta notar que a transição $6 \stackrel{a}{\Longrightarrow} 7$ sendo correspondida pelas transições $1 \stackrel{a}{\Longrightarrow} 2$ e $1 \stackrel{a}{\Longrightarrow} 4$ conduzem a estados que não são observacionalmente equivalentes, i.e. $7 \not\approx 2$ e $7 \not\approx 4$.

Questão 5

Considere a seguinte especificação de um processo que se comporta como um *buffer* de 1 única posição, com duas portas de entrada e uma de saída:

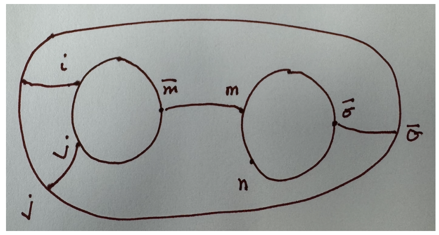
$$D(in_1, in_2, out) \triangleq in_1.\overline{out}.D + in_2.\overline{out}.D$$

Considere agora o processo

$$W \triangleq (D(i,j,m) \mid D(m,n,o))\backslash_{\{m,n\}}$$

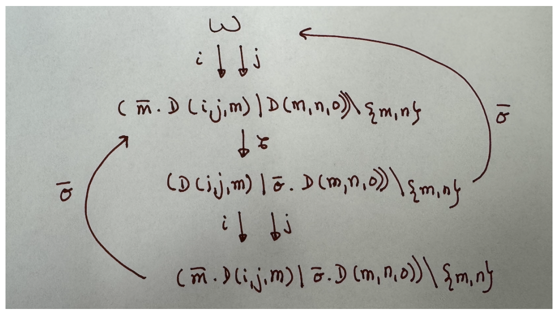
1. Represente o diagrama de sincronização do processo W.

   Sugestão de resolução



2. Desenhe o respectivo grafo de transições.

   Sugestão de resolução



3. Aplique uma vez o teorema da expansão ao processo W. Comente o processo que obteve.

   Sugestão de resolução

   Aplicando uma única vez o teorema da expansão obtemos uma versão equivalente de W que tem como conectivo principal a escolha não determinística entre dois ramos correspondentes a cada uma das entradas i e j, i.e.

$$i.(\overline{m}.(D(i,j,m) \mid D(m,n,o))\backslash_{\{m,n\}}) + j.(\overline{m}.(D(i,j,m) \mid D(m,n,o))\backslash_{\{m,n\}})$$

   Estes dois ramos são exactamente as primeiras derivações do processo W.

Questão 6

Sejam P e Q dois processos fracamente bissimilares, i.e. $P \approx Q$. Mostre ou refute que, para toda a acção observável c, $P\backslash_c \approx Q\backslash_c$

Sugestão de resolução

Para responder a esta questão é necessário construir uma bissimulação fraca R que contenha o para $(P\backslash_c, Q\backslash_c)$. Vamos postular que

$$R \;=\; \{(X\backslash_c, Y\backslash_c) \mid X \approx Y\}$$

é essa bissimulação. Pelas condições do problema o par $(P\backslash_c, Q\backslash_c) \in R$. Suponhamos, agora, que existe uma transição $P\backslash_c \stackrel{a}{\Longrightarrow} P'\backslash_c$. Claramente, por definição do operador restrição, $a \neq c$ e $P \stackrel{a}{\Longrightarrow} P'$. Como, por hipótese, $P \approx Q$, existe igualmente a transição $Q \stackrel{a}{\Longrightarrow} Q'$ e $P' \approx Q'$. Mas então temos ainda que $Q\backslash_c \stackrel{a}{\Longrightarrow} Q'\backslash_c$, estando o par $(P'\backslash_c, Q'\backslash_c)$ em R. O caso simétrico considera inicialmente uma transição $Q\backslash_c \stackrel{a}{\Longrightarrow} Q'\backslash_c$, mas os passos de raciocínio e a conclusão são análogos.