Computação Quântica
Aula 05

Murilo V. G. da Silva

DINF/UFPR

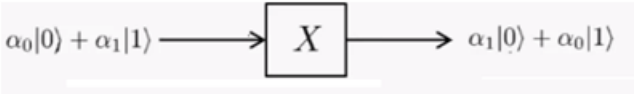
Axiomas da mecânica quântica:

- Axioma da sobreposição
- Axioma da medida
- Axioma da evolução unitária

Portas quânticas de 1-qubit (matrizes unitárias 2x2)

Bit flip (Porta NOT):

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$



Portas quânticas de 1-qubit (matrizes unitárias 2x2)

Phase flip:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Pergunta: O que a porta Z faz com $\alpha |0\rangle + \beta |1\rangle$?

Portas quânticas de 1-qubit (matrizes unitárias 2x2)

Porta de Hadamard:

$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Pergunta: O que a porta H faz com $\alpha |0\rangle + \beta |1\rangle$?

Parando para pensar um pouco

Transformação Unitária:

$$U = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$$

Pergunta: O que a porta U faz com $\alpha |0\rangle + \beta |1\rangle$?

Voltando as nossas portas H, X e Z

Relação entre estas portas:

- $X = HZH$
- $Z = HXH$

Portas quânticas de 2-qubit (matrizes unitárias 4x4)

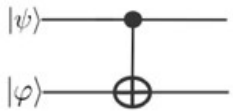
Porta CNOT:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Pergunta: O que a porta *CNOT* faz com $\alpha |00\rangle + \beta |01\rangle + \gamma |10\rangle + \delta |11\rangle$?

Portas quânticas de 2-qubits (matrizes unitárias 4x4)

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ \delta \\ \gamma \end{pmatrix}$$



Exercício: O que a porta *CNOT* faz com o estado $|+0\rangle$?

Portas quânticas de 2-qubit (matrizes unitárias 4x4)

Porta U-controlado:

$$U_C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_1 & u_2 \\ 0 & 0 & u_3 & u_4 \end{pmatrix}$$

Pergunta: O que a porta U_C faz com $\alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$?

Portas quânticas de 3-qubit (matrizes unitárias 8x8)

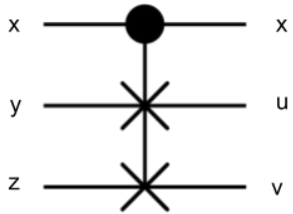
Porta CSWAP:

$$CSWAP = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Pergunta: O que faz esta porta?

Portas quânticas de 3-qubits (matrizes unitárias 8x8)

Diagrama da porta quântica do slide anterior:

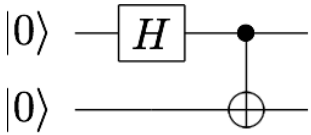


Comportamento da porta:

- Se $x = 0$, então $u = y$ e $v = z$
- Se $x = 1$, então $u = z$ e $v = y$
- Você consegue simular uma porta AND usando a porta CSWAP?
- Faça $z = 0$ e colete $x \wedge y$ na saída v

Circuitos quânticos

- Circuito quântico: sequência de portas quânticas
- Exemplo:



O que o circuito acima faz com a entrada $|00\rangle$?

- Resposta: emaranha dois qubits
- Se ao invés de $|00\rangle$, a entrada fosse $|01\rangle$ ou $|10\rangle$ ou $|11\rangle$?

Atenção: Veja que a direção da evolução do sistema é o oposto da representação em termos de multiplicação de matrizes!

Relembrando: Circuitos Quântico

Tipicamente, encapsulamos várias portas quânticas em caixas
("modularização de código")

Pergunta: Qual é matriz U do circuito encapsulado?
Resposta: $U = U_1 \otimes U_2$

Família universal de portas

Exemplo de família universal de portas clássicas: $\mathcal{S}_c = \{\text{AND}, \text{NOT}\}$

Exemplo de família univeral de portas quânticas: $\mathcal{S}_u = \{\text{C-NOT}, \frac{\pi}{8}, \text{H}\}$

No caso quântico há uma sutileza:

- Dada uma matriz U de dimensão d, para um d constante
  - Podemos construir um circuito que compute U' usando portas de $\mathcal{S}_u$ que aproxima U com precisão arbitrária
  - Mais precisamente, dado $\epsilon > 0$, $\exists U_\epsilon$, tal que $\forall |\psi\rangle$, $||U|\psi\rangle - U'|\psi\rangle|| < \epsilon$
  - Número de portas $O(d^2 log^3 \frac{1}{\epsilon})$ i.e., polinomial
    (Teorema de Solovay-Kitaev)
  - Além disso, isso pode ser feito "tolerante a falhas"

Circuitos quânticos

Parando para pensar um pouco sobre circuitos quânticos...

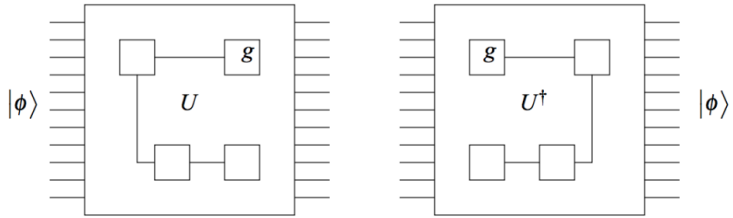
- Um circuito quântico é uma maneira de descrever um algoritmo
- Observe que um algoritmo computa uma função $f(x)$ tal que a entrada x tem tamanho n é arbitrário.
- Portanto quando projetamos um algoritmo quântico, estamos falando de uma família infinita de circuitos quânticos (todos computam a mesma função $f(x)$, mas para entradas de tamanho $n = 1, 2, 3, ...$)

Circuitos quânticos

- Na prática, isso significa que dada uma entrada de tamanho n, para que um computador quântico universal possa rodar o algoritmo, ele deve "construir" o circuito adequado para o tamanho da entrada em questão.
- Para que isto possa ser fisicamente realizável, uma das restrições que temos para esta família infinita de circuitos é que, dado n, o circuito possa ser computado em tempo polinomial.
  - Aqui podemos assumir tempo polinomial em Máquinas de Turing clássicas
  - Isso tudo é equivalente a MT Quânticas rodando a MT Universais (não veremos MTs quânticas, pois é muito trabalhoso descrever algoritmos neste modelo)

Computação reversível

Computação quântica é reversível:



Pergunta: Dado um certo circuito quântico, qual o circuito inverso?

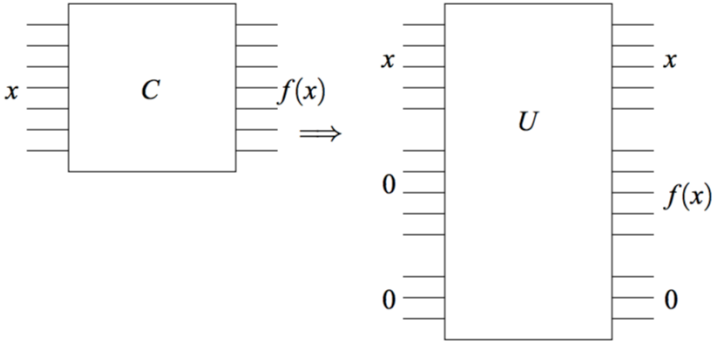
- Se o circuito é descrito por U, o inverso é descrito por $U^\dagger$
- Basta fazer o circuito de trás para frente usando as inversas das portas

Pergunta: Computação clássica é reversível?

- A porta "AND", por exemplo, não é reversível, pois perde informação
- Como vamos lidar com isso?

Encapsulando circuitos clássicos

Dado um circuito clássico C, nosso objetivo será obter um circuito quântico U com a seguinte estrutura:

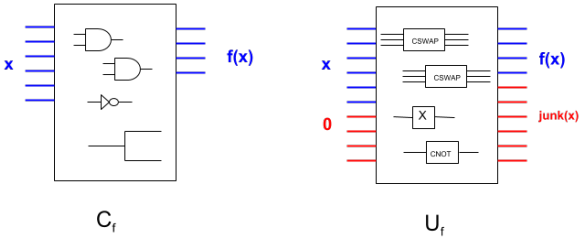


Por que tudo isso?

Encapsulando circuitos clássicos

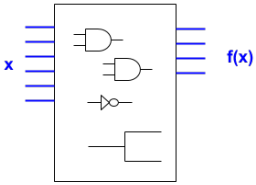
Seja $f : \{0,1\}^n \to \{0,1\}^m$ uma função booleana e C_f o circuito que a computa.

- Usando portas CSWAP, X e CNOT podemos contruir um circuito quântico U_f computando f:
  - Usando CSWAP para simular as portas AND
  - Usando X para simular as portas NOT
  - Usando CNOT para efetuar "fan out"

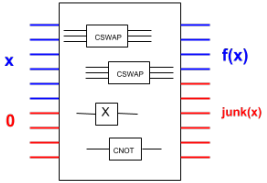


Encapsulando circuitos clássicos

Olhando o nosso circuito com calma:



C_f



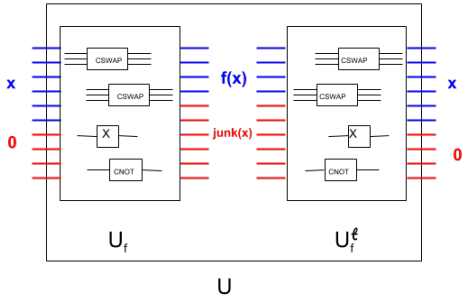
U_f

- Observe que agora o circuito é reversível

  (neste caso em particular usando apenas cswap, x, cnot, pois são auto-inversas)
- Entretanto, veremos a seguir que "lixo(x)" pode gerar problema
- Como se livrar de "lixo(x)"?

Encapsulando circuitos clássicos

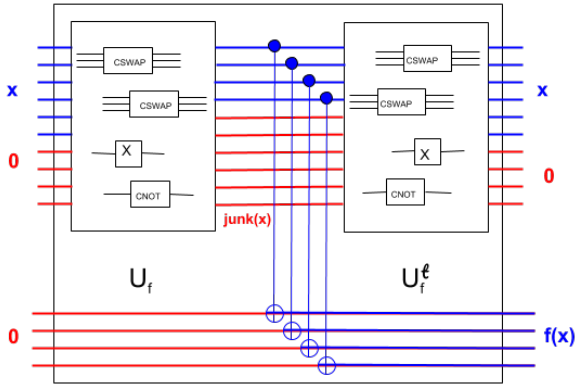
Tentando eliminar o "lixo":



- Nos livramos de "lixo"
- Entretanto, nos livramos de $f(x)$, o resultado da computação!
- Alguém tem uma ideia de como alterar esta ideia para que ela funcione?

Encapsulando circuitos clássicos

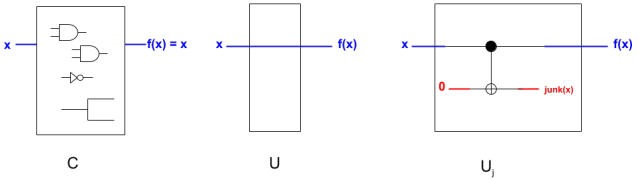
Resultado final:



Encapsulando circuitos clássicos

Por que não apenas ignorar o "lixo"?

Um exemplo extremamente simples: o circuito que calcula a função $f(x) = x$:



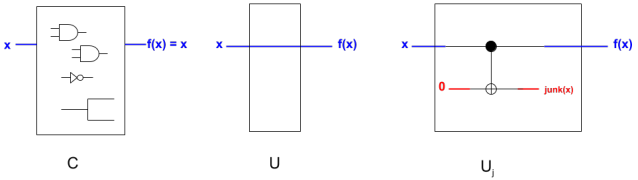
No exemplo acima temos:

- Um circuito clássico C
- Um circuito quântico U sem "lixo" (*)
- Um circuito com "lixo"
- A seguir vamos ver como U e U_j se comportam

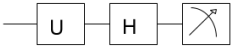
(*) Este circuito não foi produzido usando o método descrito anteriormente (oponto aqui é comparar circuitos com e sem "lixo").

Encapsulando circuitos clássicos

As duas implementações U e U_j do circuito clássico C:



Agora considere o circuito abaixo:



- Se rodarmos o circuito acima com o qubit $|+\rangle$ como entrada, qual é a saída?
- Se substibuirmos U por U_j, qual é a saída?
- O fato de termos qubits emaranhados causa interferência!!