QUANTUM ORACLES - COMO TRANSFORMAR PROBLEMAS CLÁSSICOS EM QUÂNTICOS

Alexandre Silva
Ciências da Computação
UNIVEM - Centro Universitário Eurípides de Marília

ABSTRACT

1 Introdução

Hoje, não é difícil ver alguém falando sobre computação quântica e como essas máquinas vão mudar o futuro. Contudo, muitas dessas frases acabam se levando por extrapolações e/ou usos indevidos de ficção. Neste artigo, mostrarei que nem tudo é possível ser feito com um computador quântico atual, assim como existem pequenas áreas que se beneficiam ao máximo dessa nova tecnologia.
Para esse feito, serão mostrado alguns testes feitos usando o qiskit, um framework open source da IBM para computação quântica, além de alguns resultados obtidos após executar os algoritmos em simuladores e máquinas reais, assim como seus relativos em computação clássica. Algoritmos dos quais tomam proveito dos quantum oracles, modelos ideias de função que não ajudam a descrever o algoritmo matematicamente, também tomam proveito de alguns efeitos quânticos, como superposição e interferência, para se sobressair à algumas estratégias clássicas.
Com isso, o projeto foi desenvolvido em cima de cinco pequenos problemas, sendo eles: conversão de milhas para quilômetros, torre de Hanoi, explorador de arquivos, Buckshot Roulette e QRAM. Todas as implementações e materiais utilizados podem ser encontrados nesse repositório do GitHub.

2 Início do projeto

Para dar início a pesquisa, foi necessário entender quais os tipos de oracles existem e como eles podem ser usados.
Em computação clássica, temos as Oracle Machines, as quais são maquinas de Turing, das quais implementam alguma função em seu interior, e ao ser chamado/invocado o resultado correto é retornado em tempo constante $O(1)$, podendo ser vista como uma caixa preta, abstraindo completamente o seu funcionamento. Devido a essa definição, as OMs são ideias matemáticos, sendo assim usados apenas para formalismo matemático.
Contudo em computação quântica, podemos de fato implementar certos modelos de Oracles e adiciona-los a um circuito maior, executando certas funções como: encoding de dados, aplicação de $f(x)$, abstração de partes do circuito, etc.

2.1 Tipos de Oracles

2.1.1 Phase Oracle

Um dos primeiros tipos de Oracles usados para a criação de algoritmos como os de: Grover e Deutsch–Jozsa; é comummente conhecido como *Phase Oracle*.
Tal dispositivo, é usado para atribuir uma fase ao circuito, sendo muito usado para configurar valores, explorar a interferência ou se aproveitar de outros efeitos como o *Phase Kickback*. Matematicamente poderiamos descrever ele da seguinte forma: $|x\rangle |-\rangle \rightarrow (-1)^{f(x)} |x\rangle |-\rangle$, do qual $|x\rangle$ é a entrada do oracle e $|-\rangle$ é a ancilla que prove a fase.

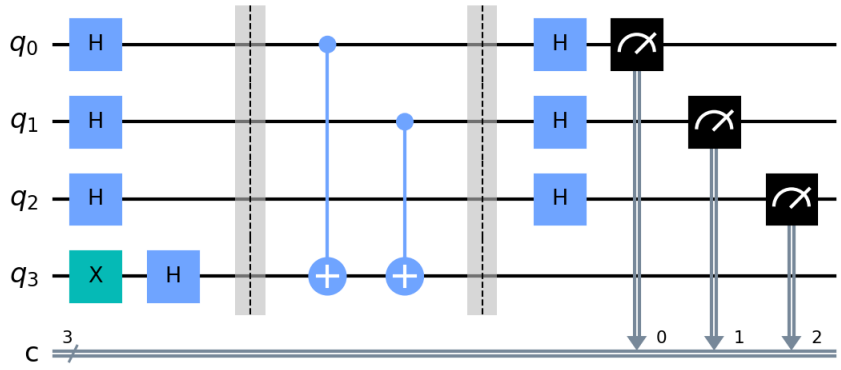


Figura 1: Exemplo de phase oracle usado para o algoritmo de Deutsch–Jozsa

No exemplo acima, utilizamos o *Phase Kickback* para adicionar uma fase nos qubits 0 e 1, transformando seus estados de $|+\rangle$ para $|-\rangle$, fazendo com que ao serem colapsados o resultado $|1\rangle$ apareça na saída.

É possível também criar um phase oracle removendo o qubit adicional (nesse exemplo o Q3), uma vez que podemos utilizar outros gates para introduzir a fase e manter ainda natureza unitária.

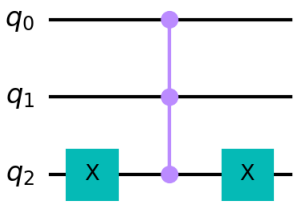
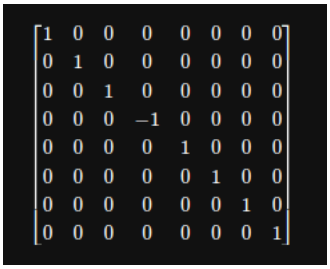


Figura 2: Exemplo fase Oracle sem a Ancilla

Dessa vez, utilizamos o MCP gate para adicionar uma fase global π e dois gates X para dizer quais qubits queremos q tenham o valor 0, codificando assim o valor 011 ou 3 na base decimal.



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 3: Matriz unitária do Phase oracle

É possível verificar então que ao criarmos esse circuito, matemos a matriz identidade e adicionamos a fase -1 no valor da coluna relativa ao 011.
Essa versão pode ser considerada como um minimal oracle, uma vez que a própria função interna se mantém unitária, sem a necessidade de ancilla.

2.2 Boolean Oracle

O Boolean oracle, por sua vez, representa uma função booleana, sem qualquer adição de fases.
Nesse caso, $|x\rangle$ representa a entrada do oracle e $|y\rangle$ represetam os qubits auxiliares que receberam a resposta, $|x\rangle\,|y\rangle \rightarrow |x\rangle\,|y \oplus f(x)\rangle$.

2.3 Minimal oracle

Como já citado anteriormente, o minimal oracle possui uma função que em sua essência é unitária, não requerendo qubits adicionais $|x\rangle \rightarrow |f(x)\rangle$.

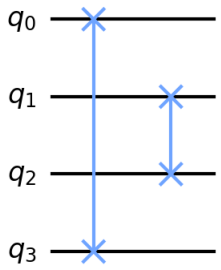


Figura 4: Exemplo de minimal oracle

Lembrando que este pode também adicionar fases ao circuito.

2.4 Simon's Oracle

O Oracle de Simon, é uma instância do Boolean Oracle. Neste visamos encontrar os períodos da função implementada, ou seja, dado x e $f(x) = y$ existe ao menos uma função em que $f(y) = x$?

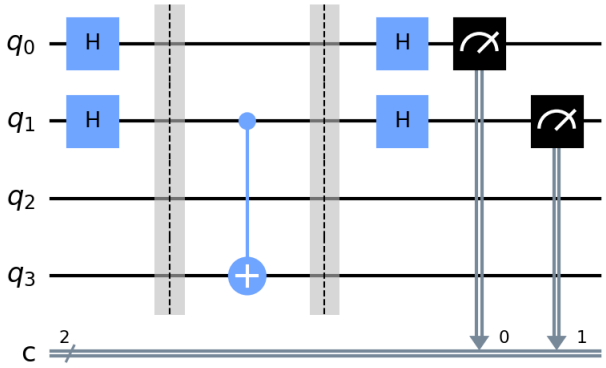


Figura 5: Exemplo do algoritmo de Simon

Nesse algoritmo, configuramos uma chave s dentro do oracle, e ao executar o algoritmo temos os possíveis períodos da função, sendo necessário rotinas de pós processamento para identificar o valor correto.

2.5 QFT(Quantum Fourier Transformation) Oracle

Por fim, o Oracle QFT aplica a versão quântica da transformada de Fourier, projetando os valores de entrada na base X (também conhecido como base de Fourier).

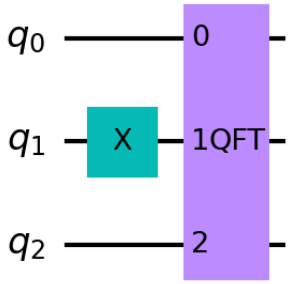


Figura 6: Exemplo do algoritmo de QFT

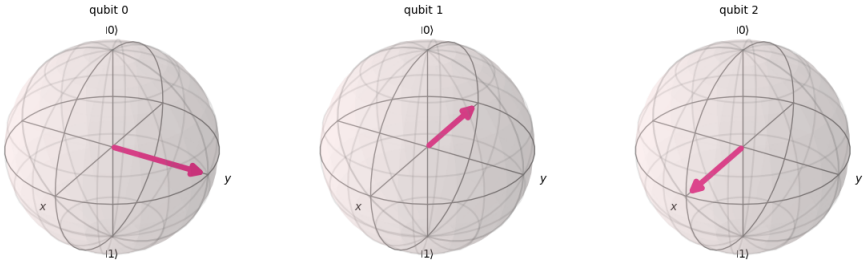


Figura 7: Valores mapeados na base de Fourier

3 Desenvolvimento

3.1 File Explorer

3.2 Miles to Kilometers

3.3 Hanoi Tower

3.4 Buckshot Roulette

3.5 QRAM

Referências