

Uma Abordagem Funcional na Implementação do Algoritmo de Teleportação Quântica

Marcos B. Cardoso, Renata H. S. Reiser, Antônio Carlos R. Costa

Escola de Informática, UCPel - Universidade Católica de Pelotas,

Rua Felix da Cunha, 412

96010-000, Pelotas, RS

E-mail: mbcardo@atlas.ucpel.tche.br, reiser@atlas.ucpel.tche.br, rocha@atlas.ucpel.tche.br

1 Introdução

Este trabalho aborda a pesquisa em Programação Quântica centrada na busca pela análise e implementação de algoritmos capazes de utilizar as vantagens do paralelismo quântico. Em geral, estes algoritmos são desenvolvidos utilizando a programação imperativa ou paradigma da programação orientada a objetos, e ainda não são eficientes para modelagem de sistemas reais.

Neste trabalho, segue-se a idéia intuitiva de implementação de algoritmos quânticos utilizando a Programação Funcional [1]. O trabalho utiliza a metodologia sugerida por Amr Sabry [7], com base na técnica de valores virtuais e na utilização de Mônadas [8] para suprir a falta de estruturas. O trabalho atual implementou o algoritmo de teleportação quântica, utilizando-se de dois bits clássicos para transmitir um único q-bit.

2 Teleportação

A teleportação faz uso de partículas emaranhadas e respeita o princípio da Mecânica Quântica de não clonagem, que permite a transmissão de um estado quântico desconhecido a outro lugar [3, 5]. Nos próximos parágrafos, uma breve contextualização da teleportação com base na aplicação de conceitos da Mecânica Quântica é apresentada, de acordo com [2, 6, 4].

A teoria da teleportação na física começou em 1935, quando os físicos Einstein, Podolsky e Rosen publicaram um artigo afirmando que a então recém-surgida teoria da Mecânica Quântica era incompleta, pois conduzia a um paradoxo, que era a propagação instantânea de informação, violando a teoria da relatividade do próprio Einstein, que proíbe a propagação de energia e informação com velocidades superiores à da luz, ou seja 300 mil quilômetros por segundo.

Einstein, Podolsky e Rosen argumentaram em 1935 que a teoria da Mecânica Quântica, que descreve os átomos, era inconsistente, pois ela aparentemente previa a propagação instantânea

de informações, contrariando a noção de que a velocidade da luz é um limite que não pode ser ultrapassado. A resposta a esse argumento foi dada por um dos criadores da Mecânica Quântica, o físico dinamarquês Niels Bohr. Na época foi um consenso que Bohr havia vencido o argumento, mas a situação havia ficado extremamente confusa.

Foi somente no início dos anos 80 que o experimento de pensamento de Einstein pode ser posto em prática. Os franceses Michel Paty e Alain Aspect demonstraram a inseparabilidade quântica: dois átomos que estiveram juntos em algum momento mantêm entre si uma relação permanente, independente da distância que os separa. Uma medida realizada em átomo altera instantaneamente o estado de outro que está distante.

Em 1993, o físico Charles Bennet e sua equipe, dos laboratórios da IBM, propuseram um modo de utilizar o fenômeno da inseparabilidade quântica para realizar o sonho da teleportação. Ou seja, viajar simplesmente reaparecendo em algum outro lugar. Em 1997, na universidade de Innsbruck, na Áustria, realizaram a primeira teleportação.

Nesse experimento, um fóton, que é uma partícula de luz, foi teleportado. A pergunta que naturalmente surge é: como conciliar o fato da transmissão instantânea realizada no experimento de Innsbruck com a fato de que a luz dá a velocidade limite para a propagação de informações. Caso essa premissa básica fosse violada, seria possível violar o princípio da causalidade, que nos diz que um efeito não pode preceder sua causa.

O que ocorre é que no experimento de teleportação existe uma componente de incerteza no que se refere ao sucesso da emissão do sinal. Embora a transmissão seja realmente instantânea, caracterizando a teleportação, é necessário que o receptor saiba de alguma forma que o evento de emissão teve sucesso. Isso é feito através de um meio normal de comunicação.

Aparentemente a natureza de uma forma ou outra proíbe a violação do princípio de causa e efeito. Mesmo assim, a teleportação quântica tem despertado grande interesse dos governos dos

países industrializados, pois ela pode ser aplicada no desenvolvimento de redes de computadores quânticos.

3 Idéia

A teleportação usa dois bits clássicos para transmitir um único q-bit. Teleportação respeita o princípio da Mecânica Quântica da não clonagem, que permite a transmissão de um estado quântico desconhecido.

A chave da teleportação é o uso de partículas emaranhadas [6]. O ajuste inicial é dado para ambos os processos. Alice e Bob desejam comunicar-se, é emitido a cada um, uma das partículas emaranhadas, dada pela expressão

$$\psi_0 = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Assim até que uma partícula esteja transmitida, apenas Alice pode executar transformações em sua partícula, e somente Bob pode executar transformações em sua partícula.

4 Funcionamento

O objetivo é transmitir o estado quântico de uma partícula usando bits clássicos e reconstruir o exato estado quântico no receptor (Figura 1). Desde que o estado quântico não pode ser copiado, o estado quântico da partícula dada será destruído necessariamente. O único bit teleportado foi realizado experimentalmente [2, 4].

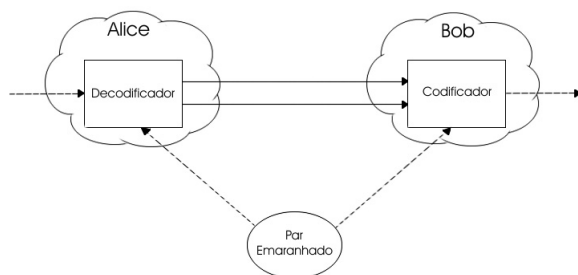


Figura 1: Representação gráfica da teleportação.

Alice. Alice tem um q-bit cujo o estado é desconhecido. Quer enviar o estado do q-bit

$$\phi = a|0\rangle + b|1\rangle$$

a Bob através de canais de comunicação clássicos. Cada um dos dois, Alice e Bob, possui um q-bit de um par emaranhado

$$\psi_0 = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Alice aplica a etapa da decodificação ao q-bit ϕ a ser transmitido e sua metade do par emaranhado. O estado inicial é o estado quântico

$$\begin{aligned} \phi \otimes \psi_0 &= \frac{1}{\sqrt{2}}(a|0\rangle \otimes (|00\rangle + |11\rangle) + \\ &\quad b|1\rangle \otimes (|00\rangle + |11\rangle)) \\ &= \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle), \end{aligned}$$

os quais Alice controla os dois primeiros bits e Bob controla o último. Alice aplica $C_{not} \otimes I$ e $H \otimes I \otimes I$ (representado na Figura 2) a este estado:

$$\begin{aligned} (H \otimes I \otimes I)(C_{not} \otimes I)(\phi \otimes \psi_0) &= (H \otimes I \otimes I)(C_{not} \otimes I) \\ &\quad \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle) \\ &= (H \otimes I \otimes I) \\ &\quad \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|110\rangle + b|101\rangle) \\ &= \frac{1}{2}(a(|000\rangle + |011\rangle + |100\rangle + |111\rangle) + \\ &\quad b(|010\rangle + |001\rangle - |110\rangle - |101\rangle)) \\ &= \frac{1}{2}(|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(a|1\rangle + b|0\rangle) + \\ &\quad |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(a|1\rangle - b|0\rangle)) \end{aligned}$$

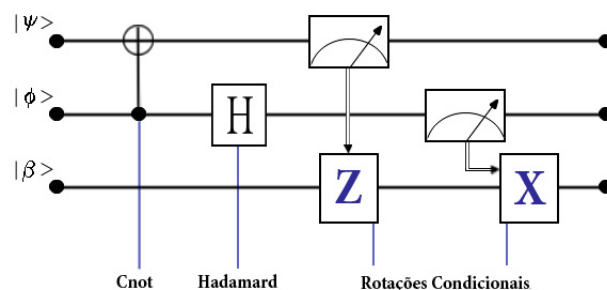


Figura 2: Representação do circuito de teleportação.

Alice mede os primeiros dois q-bits para pegar um dos $|00\rangle, |01\rangle, |10\rangle$, ou $|11\rangle$ com probabilidade igual. Dependendo do resultado da medida, o estado quântico do q-bit de Bob é projetado a $a|0\rangle + b|1\rangle, a|1\rangle + b|0\rangle, a|0\rangle - b|1\rangle$, ou $a|1\rangle - b|0\rangle$ respectivamente. Alice emite o resultado de sua medida como dois bits clássicos a Bob.

Quando a Alice fez a medição, é alterado irreversivelmente o estado de seu q-bit original ϕ , cujo estado está no processo emitido a Bob. Esta

perda do estado original é a razão da teleportação não violatar o princípio que proíbe a clonagem na física quântica.

Quando Bob recebe os dois bits clássicos de Alice, ele sabe o estado de sua metade do par emaranhado e compara ao estado original do q-bit de Alice.

bits recebidos	estado	decodificação
00	$a 0\rangle + b 1\rangle$	I
01	$a 1\rangle + b 0\rangle$	X
10	$a 0\rangle - b 1\rangle$	Z
11	$a 1\rangle - b 0\rangle$	X, Z

Bob pode reconstruir o estado original do q-bit de Alice, ϕ , aplicando a transformação de decodificação apropriada a sua parte do par emaranhado. Caso o q-bit seja:

- 00 - aplica-se a matriz identidade,
- 01 - aplica-se a operação X (C_{not}) (matriz referente a operação apresentada na Fórmula 1),
- 10 - aplica-se a operação Z ($Shift$) (matriz referente a operação apresentada na Fórmula 2),
- 11 - aplica-se seqüencialmente C_{not} e $Shift$.

$$X : \begin{array}{l} |0\rangle \rightarrow |1\rangle \\ |1\rangle \rightarrow |0\rangle \end{array} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (1)$$

$$Z : \begin{array}{l} |0\rangle \rightarrow |0\rangle \\ |1\rangle \rightarrow -|1\rangle \end{array} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2)$$

5 Implementação

Seguindo a idéia apresentada anteriormente, foi feita uma implementação utilizando da programação funcional. Este trabalho segue como base o modelo introduzido por Amr Sabry em [7], onde ele defini um modelo quântico funcional que introduz alguns exemplos e novos conceitos como o do valor virtual que é utilizado para implementar a teleportação.

Um valor virtual se trata de um valor que, apesar de estar profundamente embutido dentro de uma estrutura e possivelmente emaranhada com outros valores, pode ser operada individualmente. Um valor virtual é especificado dando toda a estrutura de dados ao qual ele pertence e um adaptador, que define o mapeamento de toda essa estrutura de dados para o valor em questão, e vice-versa. Mais especificamente, na linguagem, tem-se a expressão:

```
data Adaptor l g =
  Adaptor { dec :: g -> l,
            cmp :: l -> g }

data Virt a na u = Virt (QR u)
```

(Adaptor (a,na) u)

Considerando a definição de valores virtuais, o tipo (**Virt a na u**) define um valor virtual do tipo **a** que é emaranhado com valores de tipo **na**. O tipo **u** é o tipo da estrutura de dados total, que contém tanto **a** quanto **na**. O adaptador mapeia nas duas direções entre o tipo **u** e sua decomposição.

Valores virtuais são relacionados a referências componíveis [14], que fornecem acessos a um campo ou a uma subestrutura relativa a uma tupla ou registro maior, usado como estado.

A definição de adaptadores (pelo menos para estruturas de dados como tuplas) é tão regular que deveríamos ser capazes de automatizar sua geração apenas a partir do tipo da informação. Embora tenham sido implementados outros adaptadores, a seguir descreve-se apenas alguns adaptadores básicos:

```
ad_pair1 :: Adaptor (a1,a2) (a1,a2)
ad_pair1 = Adaptor {dec = \(a1,a2) ->
                    (a1,a2),
                    cmp = \(a1,a2) ->
                    (a1,a2)}
```

```
ad_pair2 :: Adaptor (a2,a1) (a1,a2)
ad_pair2 = Adaptor {dec = \(a1,a2) ->
                    (a2,a1),
                    cmp = \(a2,a1) ->
                    (a1,a2)}
```

Para a implementação da teleportação faz-se necessário a implementação de algumas operações como a funções a **shift** e duas outras funções sobre a referência quântica: a primeira usada para a retirada de referência já que apenas havia sido definido a colocada de um dado em uma referência (**mkQR**), e outra que lê uma referência quântica e põe o valor em um outro dado. Funções sobre os valores virtuais também foram definidas. Dentre estas, tem-se a função **pegaVirt** a qual tira o valor quântico de um valor virtual para que ele possa ser utilizado fora do virtual. Outra função implementada, e indicada por **mostraVirt**, pega um virtual e utiliza a mônada de IO para apresentá-lo.

```
shift :: Qop Bool Bool
shift = qop[((False,False),1),
            ((True,True),(-1))]
```

```
tiraQR :: QR a -> IO(QV a)
tiraQR (QR ptr) = do v <- readIORef ptr
                    return v
```

```
lerQR :: QR a -> QV a -> IO ()
lerQR (QR ptr) qv = writeIORef ptr qv
```

```

pegaVirt :: (Basis a, Basis na,
            Basis u, Show a) =>
            Virt a na u -> IO (QV a)
pegaVirt (Virt (QR r) (Adaptor dec = dec,
                    cmp = cmp)) =
do v <- readIORef r
writeIORef r v
let progT a = sum[pr v (cmp(a,na)) |
                na <- basis]
let res = qv[(a, progT a) |
             a <- basis]
return(res)

```

```

mostraVirt :: (Basis a, Basis na,
              Basis u, Show a) =>
              Virt a na u -> IO ()
mostraVirt (Virt (QR r)
            (Adaptor dec = dec, cmp = cmp)) =
do v <- readIORef r
writeIORef r v
let progT a = sum[pr v (cmp(a,na)) |
                na <- basis]
let res = qv[(a, progT a) |
             a <- basis]
print("Bob:", res)

```

Como a implementação aplica a técnica de valores virtuais, foram definidos adaptadores, de acordo com a codificação abaixo:

```

ad_telep2 :: Adaptor (a2,(a1,a3))
            (a1,(a2,a3)))
ad_telep2 = Adaptor {dec = \ (a1,(a2,a3)) ->
                    (a2,(a1,a3)),
                    cmp = \ (a2,(a1,a3)) ->
                    (a1,(a2,a3))}

ad_telep12 :: Adaptor ((a1,a2),a3)
            (a1,(a2,a3)))
ad_telep12 = Adaptor {dec = \ (a1,(a2,a3)) ->
                    ((a1,a2),a3),
                    cmp = \ ((a1,a2),a3) ->
                    (a1,(a2,a3))}

ad_telep23 :: Adaptor ((a2,a3),a1)
            (a1,(a2,a3)))
ad_telep23 = Adaptor {dec = \ (a1,(a2,a3)) ->
                    ((a2,a3),a1),
                    cmp = \ ((a2,a3),a1) ->
                    (a1,(a2,a3))}

```

Como o algoritmo de teleportação, descrito na primeira seção deste capítulo, faz uso do emaranhamento de dados quânticos em Haskell, foi contruído um dado com 2 q-bits emaranhados. Este q-bit foi chamado de **qemaranhado**. Da mesma forma, defini-se o dado ϕ (qphi) o qual está sendo teleportado.

```

qemaranhado = qv [(False, False),
                 1/sqrt(2)],

```

```

((True, True),
 1/sqrt(2))]

```

```

qphi = qv [(False, 0), (True, 1)]

```

A implementação do algoritmo foi feita conforme mostra o código apresentado logo abaixo.

```

teleporta = do qv_ema <- mkQR qemaranhado
qv_phi <- mkQR qphi
alice qv_ema qv_phi
print "Acabou"

```

```

alice qv_ema qv_phi = do putStrLn("Alice")
qv <- tiraQR qv_phi
ema <- tiraQR qv_ema
emara <- mkQR
                (qv &* ema)
let vtriple =
                virtFromR
                emara
let meio = virtFromV
                vtriple
                ad_telep2
let tm = virtFromV
                vtriple
                ad_telep12
let mb = virtFromV
                vtriple
                ad_telep23
app1 cnot tm
app1 hadamard_op meio
medicao <- observeVV
                tm
print(medicao)
dado <- pegaVirt mb
putStrLn("AcabouAlice")
canal medicao dado
                qv_ema

canal medicao dado qv_ema = do putStrLn("Canal")
lerQR qv_ema dado
bob qv_ema
                medicao

bob qv_ema medicao = do putStrLn("Bob Inicia")
vepr <- virtFromRR
                qv_ema
let ultimo = virtFromV
                vepr ad_pair2
case medicao of
(False,False) ->
do putStrLn
                ("False False")
mostraVirt(ultimo)
(False,True) ->
do putStrLn
                ("False True")
app1 qnot_op
                ultimo
mostraVirt(ultimo)
(True,False) ->

```

```

do putStrLn
  ("True False")
  app1 shift
    ultimo
  mostraVirt(ultimo)
(True,True) ->
do putStrLn
  ("True True")
  app1 qnot_op
    ultimo
  app1 shift
    ultimo
  mostraVirt(ultimo)

```

A implementação foi desenvolvida de maneira que a função `teleporta` gera referências quânticas tanto para o dado emaranhado (`qemaranhado`) quanto para o dado ϕ (`qphi`). Após ter criado estas referências, chama-se Alice passando as referências geradas. Alice por sua vez, faz transformações nas referências, criando dados virtuais a partir das referências recebidas.

Assim que Alice acaba esta etapa, ela passa suas informações para um canal, o qual leva as informações até Bob. Bob recebe estas informações e posteriormente, executa medições e tratamento nos dados, caso seja necessário.

A execução desta função pode ser vista logo abaixo.

```

> teleporta
Alice
(False,False)
AcabouAlice
Canal
Bob Inicia
False False
("Bob:"[(False,0 :+ 0),(True,1 :+ 0)])
"Acabou"

```

6 Conclusão

O desenvolvimento e a modelagem de algoritmos quânticos, bem como a aplicação e análise de simulações e testes, impulsionam o desenvolvimento da Programação Quântica, independentemente dos avanços tecnológicos para concretizar a construção de computadores quânticos.

Os resultados obtidos neste trabalho buscaram incentivar o conhecimento básico da Computação Quântica visando o desenvolvimento da Programação Quântica e a compreensão das novas Tecnologias de Informação, atualmente relevantes para a Ciência da Computação. Neste sentido, o Projeto apresentou um estudo introdutório da aplicação da Programação Funcional na implementação de algoritmos quânticos básicos para teleportação.

A simulação deste algoritmo quântico viabilizou importantes e imediatas aplicações demonstrativas

das portas lógicas, das transformações unitárias e das operações de medidas, anteriormente estudadas e implementadas junto a biblioteca construída no Projeto.

Referências

- [1] R. Bird, "Introduction to Functional Programming", Prentice Hall, 1988.
- [2] G. Brassard, "Teleportation as a quantum computation", 1998. (disponível em: www.citeseer.ist.psu.edu/brassard98teleportation.html)
- [3] M. Hirvensalo, "Quantum Computing", Springer Verlag, 2001.
- [4] M. Nielsen, E. Knill, "Complete quantum teleportation by nuclear magnetic resonance", 1998. (disponível em: www.citeseer.ist.psu.edu/595490.html)
- [5] A. Pittenger, "An Introduction to Quantum Computing Algorithms", Birkhauser Boston, 1999.
- [6] E. Rieffel, W. Polak, "An introduction to quantum computing for non-physicists", 2000. (disponível em: <http://arxiv.org/abs/quant-ph/9809016>)
- [7] A. Sabry, Modeling quantum computing in Haskell. In: *ACM SIGPLAN 2003 Haskell Workshop*, 2003. (disponível em: <http://www.cs.indiana.edu/sabry/papers/quantum.pdf>)
- [8] P. Wadler, Monads for functional programming, *Lecture Notes in Computer Science*, 925, 1995.