

Uma Abordagem Funcional na Implementação do Algoritmo de Teleportação Quântica

M.B. CARDOSO¹, R.H.S. REISER², A.C.R. COSTA³, Escola de Informática, Universidade Católica de Pelotas, 96010-000 Pelotas, RS, Brazil.

Resumo. O desenvolvimento de extensões de bibliotecas da linguagem de programação Haskell, mais intuitiva para programação quântica, visa obter uma compreensão mais abrangente da computação quântica baseada em seus fundamentos matemáticos. Os conceitos gerais da programação funcional são aplicados na implementação de algoritmos quânticos para teleportação. Baseada na metodologia introduzida por Amr Sabry, esta implementação inclui a aplicação da técnica de valores virtuais, a análise de dados quânticos (clássicos, superposição, emaranhados), a especificação de transformações unitárias e a compreensão do processos de medida, que é a observação de valores.

1. Introdução

Neste trabalho, os fundamentos da mecânica quântica são interpretados sob o nível de abstração funcional [1], mais acessível aos programadores, com o objetivo de despertar o estudo e incentivar o desenvolvimento da computação quântica [5, 10, 12].

Segue-se a idéia intuitiva de implementação de algoritmos quânticos utilizando a linguagem Haskell [14] e aplica-se a metodologia sugerida por Amr Sabry [13], com base na técnica de valores virtuais e na utilização de mônadas [15] para suprir a falta de estruturas.

O trabalho implementou o algoritmo de teleportação quântica, utilizando-se de dois bits clássicos para transmitir um único q-bit. Em geral, algoritmos para teleportação são desenvolvidos utilizando a programação imperativa ou o paradigma da programação orientada a objetos, e ainda não são eficientes para modelagem de sistemas reais.

2. Teleportação

A teleportação faz uso de partículas emaranhadas e respeita o princípio da física quântica de não clonagem, que permite a transmissão de um estado quântico desconhecido a outro lugar [5, 10]. Nos próximos parágrafos, uma breve contextualização

¹mbcardo@ucpel.tche.br

²reiser@ucpel.tche.br

³rocha@ucpel.tche.br

da teleportação com base na aplicação de conceitos da mecânica quântica é apresentada, de acordo com [2, 7, 8].

A teoria da teleportação 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, aproximadamente 300 mil quilômetros por segundo.

Einstein, Podolsky e Rosen argumentaram que a teoria da mecânica quântica, que descreve o comportamento físico dos 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. Naquela 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 pensamento de Einstein pode ser posto em prática, pela construção de um experimento científico. Os franceses Michel Paty e Alain Aspect demonstraram a inseparabilidade quântica: dois átomos que estiveram juntos e sofreram um processo de emaranhamento em algum momento mantêm entre si uma relação permanente, independente da distância que os separa, no caso de estarem isolados. Uma medida realizada em um átomo altera, instantaneamente, o estado de outro que está distante.

Em 1993, o físico Charles Bennet e sua equipe, que trabalhavam nos 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 o 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, entretanto, 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 clássico 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 da comunidade científica, pois ela pode ser aplicada no desenvolvimento de novas tecnologias de informação, como as 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 [12]. O ajuste inicial é dado para ambos os processos. Sempre que Alice e Bob desejam se comunicar, é emitido a cada um, uma das partículas emaranhadas. O estado conjunto das partículas é dada pela seguinte expressão:

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

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 do processo de teleportação 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, necessariamente, destruído. O único bit teleportado foi realizado experimentalmente [2, 7].

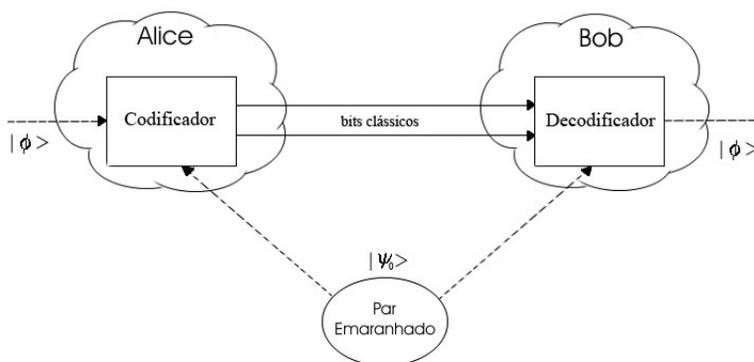


Figura 1: Representação gráfica da teleportação.

Suponha que Alice tem um q-bit cujo o estado é desconhecido e quer enviar a Bob o estado do q-bit

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

através de canais de comunicação clássicos. Sabe-se que cada um dos dois, Alice e Bob, possui um q-bit de um par emaranhado

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

Alice aplica a etapa da codificação ao q-bit $|\phi\rangle$ a ser transmitido e sua metade do par emaranhado. O estado inicial é o estado quântico resultante do produto tensor [12, 8] entre $|\phi\rangle$ e $|\psi_0\rangle$,

$$|\phi\rangle \otimes |\psi_0\rangle = \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle),$$

do qual Alice controla os dois primeiros q-bits e Bob controla o último.

Para dar continuidade à descrição do funcionamento da teleportação, além do operador Identidade (I), consideram-se também os operadores Hadamard (H) e Not Controlado ($CNOT$), definidos a partir da base computacional e correspondentes expressões matriciais:

$$H : \begin{array}{l} |0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{array} \quad \left(\begin{array}{cc} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{array} \right), \quad (4.1)$$

$$CNOT : \begin{array}{l} |00\rangle \rightarrow |00\rangle \\ |01\rangle \rightarrow |01\rangle \\ |10\rangle \rightarrow |11\rangle \\ |11\rangle \rightarrow |10\rangle \end{array} \quad \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right). \quad (4.2)$$

Considerando os operadores definidos nas expressões 4.1 e 4.2 como argumentos para o produto tensor, Alice aplica $CNOT \otimes I$ e $H \otimes I \otimes I$ ao estado $|\phi\rangle \otimes |\psi_0\rangle$:

$$\begin{aligned} (H \otimes I \otimes I)(CNOT \otimes I)(|\phi\rangle \otimes |\psi_0\rangle) &= (H \otimes I \otimes I)(CNOT \otimes I) \\ &\quad \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle) \\ &= (H \otimes I \otimes I) \\ &\quad \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|110\rangle + b|101\rangle) \\ &= \frac{1}{2}(a(|000\rangle + |011\rangle + |100\rangle + |111\rangle) + \\ &\quad b(|010\rangle + |001\rangle - |110\rangle - |101\rangle)) \\ &= \frac{1}{2}(|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(a|1\rangle + b|0\rangle) + \\ &\quad |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(a|1\rangle - b|0\rangle)). \end{aligned}$$

O circuito de teleportação está graficamente representado na Figura 2.

A seguir, Alice mede os primeiros dois q-bits para pegar, com igual probabilidade, um dos q-bits

$$|00\rangle, |01\rangle, |10\rangle, \text{ ou } |11\rangle.$$

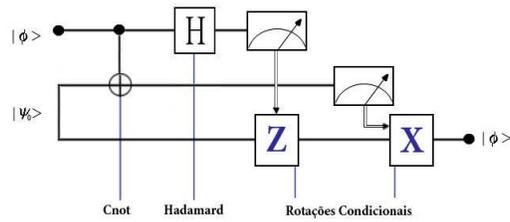


Figura 2: Representação do circuito de teleportação.

Tabela 1: Etapa de Decodificação

Bits Recebidos	Estado Quântico	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	$a 1\rangle - b 0\rangle$	XZ

Dependendo do resultado desta medida, o estado quântico do q-bit de Bob é projetado em um dos seguintes estados:

$$a|0\rangle + b|1\rangle, a|1\rangle + b|0\rangle, a|0\rangle - b|1\rangle \text{ ou } a|1\rangle - b|0\rangle,$$

respectivamente. Alice emite o resultado de sua medida como dois bits clássicos a Bob.

Quando a Alice faz a medição, é alterado, de forma irrecuperável, o estado de seu q-bit original $|\phi\rangle$, cujo estado está no processo emitido a Bob. Esta perda do estado original é a razão da teleportação não violar o princípio que proíbe a clonagem na física quântica.

Logo após o recebimento dos dois bits clássicos de Alice, Bob dá prosseguimento ao processo de teleportação, com a etapa de decodificação.

Na última etapa, Bob pode reconstruir o estado original do q-bit $|\phi\rangle$ de Alice, aplicando a transformação de decodificação apropriada a sua parte do par emaranhado, de acordo com a Tabela 4.

Assim, caso o q-bit seja:

- 00 - aplica-se a matriz identidade,
- 01 - aplica-se a operação X (matriz referente a operação apresentada na Fórmula 4.3),
- 10 - aplica-se a operação Z (matriz referente a operação apresentada na Fórmula 4.4),
- 11 - aplica-se seqüencialmente Z e X .

$$X : \begin{array}{l} |0\rangle \rightarrow |1\rangle \\ |1\rangle \rightarrow |0\rangle \end{array} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (4.3)$$

$$Z : \begin{array}{l} |0\rangle \rightarrow |0\rangle \\ |1\rangle \rightarrow -|1\rangle \end{array} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (4.4)$$

5. Implementação

Seguindo a idéia apresentada anteriormente, foi feita uma implementação utilizando uma extensão da biblioteca introduzida por Amr Sabry em [13], implementada na linguagem Haskell. Esta biblioteca está centrada num modelo quântico funcional que aplica o conceito de mônadas para modelar efeitos colaterais, estados globais, de entrada e saída de dados, introduzindo novos tipos como o valor virtual, que é aplicado neste trabalho para implementar a teleportação.

Um valor virtual é valor que, apesar de estar preso por uma estrutura, e possivelmente emaranhado com outros valores, pode ser operado individualmente, por transformações unitárias. Um valor virtual é especificado pela a estrutura de dados ao qual ele pertence e por 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 estão relacionados por referências componíveis [6] 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 funções como a operação Z e duas outras funções sobre a referência quântica: a primeira função (`tiraQR`) é usada para a retirada de referência, já que apenas havia sido definido a colocação de um dado em uma referência (`mkQR`), e a outra função (`ler QR`) 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 valor virtual. Salienta-se também outra função implementada, denotada por `mostraVirt`, que pega um virtual e utiliza a mônada de IO para apresentá-lo.

O código abaixo mostra a aplicação das funções definidas nos parágrafos anteriores.

```
Z :: Qop Bool Bool
Z = 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))}

```

Considere que o algoritmo de teleportação, já descrito na Seção 4., 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 $|\phi\rangle$ (`qphi`) o qual está sendo teleportado.

```

qemaranhado = qv [((False, False), 1/sqrt(2)),((True, True), 1/sqrt(2))]
qphi = qv [(False, 0), (True, 1)]

```

O código apresentado logo abaixo refere-se à implementação do algoritmo.

```

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

hadamard_op = qop [((False,False),1),
                  ((False,True),1),
                  ((True,False),1),
                  ((True,True),-1)]

qnot_op = qop [((False,True),1)
              ((True,False),1)]

alice qv_ema qv_phi = do putStrLn("Alice")
                        qph <- tiraQR qv_phi
                        ema <- tiraQR qv_ema
                        emara <- mkQR (qph &* ema)
                        let vtriple = virtFromR emara
                            let meio = virtFromV vtriple ad_telep2
                                let tm = virtFromV vtriple ad_telep12
                                    let mb = virtFromV vtriple ad_telep23
                                        appl cnot tm
                                            appl 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 Z ultimo
            mostraVirt(ultimo)
        (True,True) -> do putStrLn ("True True")
            app1 qnot_op ultimo
            app1 Z 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 $|\phi\rangle$ (`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. Plataforma do Sabry

A utilização da linguagem Haskell justificou-se sobretudo pela sua semântica acessível, modelada a partir das funções matemáticas e utilizando a construção lógica do cálculo lambda (λ -calculus). Os compiladores Haskell estão disponíveis para vários sistemas operacionais, permitindo uma maior portabilidade com as demais linguagens. Estas características resultaram em um código elegante e de fácil compreensão, viabilizando o aumento substancial da produtividade nas etapas de programação,

além de mostrar-se como um código de fácil entendimento para a comunidade científica, em especial para a Ciência da Computação.

O poder de definição e especificação de funções especializadas, que operam com componentes de estruturas de dados não-triviais, foram consideradas vantagens na etapa de implementação. A aplicação da técnica de manipulação de estruturas emaranhadas, definida pela construção de valores virtuais e adaptadores, operacionalizou a manipulação de permutações arbitrárias, incluído a noção de registrador simbólico para indicar qualquer coleção de q-bits, possivelmente emaranhados.

Para melhor compreensão e aplicação da programação funcional, e em especial da linguagem Haskell, na implementação de modelos tradicionais da computação quântica, foram relevantes as contribuições obtidas em [16].

Grande parte dos modelos computacionais disponíveis para manipulação de algoritmos quânticos são implementados em linguagens tradicionais (*C*, *C++*, *Java*) e utilizam, essencialmente, apenas instruções de q-bits e portas lógicas quânticas [11, 3, 4]. As abordagens, na forma de circuitos quânticos, exigem maior conhecimento dos fundamentos da mecânica quântica e dificulta a compreensão e implementação dos algoritmos para a comunidade restritas à computação.

Os modelos para computação quântica construídos a partir de linguagens baseadas em programas matemáticos (*Maple*, *MatLab*), são programas proprietários cujas simulações são geralmente executadas sobre específica plataforma, restringindo seu acesso e aplicação [9].

7. Conclusão

O desenvolvimento e a modelagem de algoritmos para a programação quântica, bem como a aplicação e a análise de simulações e testes, independem dos avanços tecnológicos para concretizar a construção de computadores quânticos. Os resultados obtidos incentivam o interesse pelo desenvolvimento da programação quântica e pela compreensão das novas tecnologias para teleportação.

As simulações viabilizaram imediatas aplicações das portas lógicas, das transformações unitárias e das operações de medidas, disponibilizando ainda uma extensão da biblioteca inserida na plataforma introduzida por Sabry.

Baseado nos trabalhos desenvolvidos, pode-se estender a aplicação da programação funcional na implementação de algoritmos quânticos em outras áreas da computação científica, como a criptografia.

Agradecimentos

Este trabalho é parcialmente financiado pela FAPERGS e CNPq.

Abstract. The development of extensions and libraries of the Haskell programming language, which are more intuitive for quantum programming, aims to obtain a more general comprehension of the quantum computation supported by its mathematics basis. The discussion of the general concepts of functional programming is considered in order to implement algorithms related to teleportation. Based on

the methodology introduced by Amr Sabry, this implementation includes the application of the virtual value technique, the analysis of quantum data (classical, superposition, entanglement), the specification of unitary transformations and the understanding of measurement process of observing values.

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] B. Butscher, H. Weimer, “Libquantum”, 2003. (disponível em: <http://www.enyo.de/libquantum>).
- [4] M. Eck, P. Wocjan, “Quantum Circuit Simulator - QuaSi”, Universität Karlsruhe, 2000. (disponível em: <http://iaks-www.ira.uka.de/QIV/QuaSi/aboutquasi.html>).
- [5] M. Hirvensalo, “Quantum Computing”, Springer Verlag, 2001.
- [6] K. Kagawa, Mutable data structures and composable references in pure functional language, em “State in Programming Languages - SIPL’95”, (U.S.Reddy, ed.) pp. 79-94, ACM/SIGPLAN University of Illinois at Urbana-Champaign, 1995.
- [7] M. Nielsen, E. Knill, “Complete quantum teleportation by nuclear magnetic resonance”, 1998. (disponível em: www.citeseer.ist.psu.edu/595490.html).
- [8] M. Nielsen, I. Chuang, “Quantum Computation and Quantum Information”, Cambridge University Press, 2000.
- [9] C. McCubbin, “Open-Source Quantum Computer Simulator in Maple - (OpenQUACS)”, 2000. (disponível em: <http://userpages.umbc.edu/cmc-cub1/quacs/quacs.html>).
- [10] A. Pittenger, “An Introduction to Quantum Computing Algorithms”, Birkhauser Boston, 1999.
- [11] H. Raedt, A.H. Hams, K. Michielsen, Quantum Computer Emulator - QCE, *Computer Physics Communications*, 132 (2000), 1–29. (disponível em: <http://arxiv.org/PS-cache/quant-ph/pdf/9911/9911041.pdf>).
- [12] E. Rieffel, W. Polak, An introduction to quantum computing for non-physicists, *ACM Computing Surveys*, **32**(3)(2000), 300 – 335. (disponível em: <http://arxiv.org/abs/quant-ph/9809016>).
- [13] 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>).

- [14] S. Thompson, “Haskell The Craft of Functional Programming”, Addison-Wesley, 1996.
- [15] P. Wadler, Monads for functional programming, *Lecture Notes in Computer Science*, 925, 1995.
- [16] J. Vizzotto, A. Costa, Concurrent Quantum Programming in Haskell, em “VII Congresso Brasileiro de Redes Neurais”, 2005.