

QCEdit: MODELAGEM DE CIRCUITOS QUÂNTICOS

de AVILA, Anderson B.¹; MARON, Adriano K.², REISER, Renata H. S.³

¹UFPEL, Graduação em Ciência da Computação, Pelotas abdavila@inf.ufpel.edu.br;

²UFPEL, Mestrado em Ciência da Computação, Pelotas, akmaron@inf.ufpel.edu.br

³UFPEL, PPGC/CDTEC, Pelotas, Brasil reiser@inf.ufpel.edu.br

1 INTRODUÇÃO

A Computação Quântica (CQ) idealiza o processamento da informação através de um meio físico (atômico/subatômico) que se comporta de acordo com as leis da Mecânica Quântica [KNILL, (2002)]. A unidade de informação, o *bit* quântico (*qubit*), modela um sistema que tem dois estados básicos, geralmente escritos na notação de Dirac, como $|0\rangle$ e $|1\rangle$. Entretanto, diferentemente do bit, o *qubit* pode estar em uma superposição dos seus estados básicos $\alpha|0\rangle + \beta|1\rangle$, onde α , β são números complexos cujas amplitudes satisfazem a condição de normalidade $|\alpha|^2 + |\beta|^2 = 1$. Intuitivamente, pode-se dizer que o *qubit* pode existir nos estados $|0\rangle$, $|1\rangle$ e também em um estado que está simultaneamente em ambos, $\alpha|0\rangle + \beta|1\rangle$, onde $|\alpha|^2$ e $|\beta|^2$ representam as probabilidades de ocorrência dos estados $|0\rangle$ e $|1\rangle$, respectivamente. Formalmente, um *qubit* é modelado como um vetor normalizado em um espaço de Hilbert bi-dimensional [KNILL, (2002); NIELSEN, CHUANG, (2003)].

Devido às características especiais dos qubits, a computação quântica impõe novas fronteiras com relação à velocidade das computações [HANNEKE et. al., (2009)], se comparadas com o seu processamento clássico. Exemplificando, o algoritmo quântico de Shor fatora números inteiros em tempo polinomial sobre o número de dígitos; o algoritmo quântico de Grover [GROVER, (1968)], realiza uma pesquisa eficiente em banco de dados. Algoritmos quânticos estão sendo propostos, porém exigem plataformas para interpretação e testes, visto a simplicidade do hardware quântico atual. Nesse contexto, modelos e simuladores quânticos introduzem novas abordagens para estudo e desenvolvimento desses algoritmos [WATANABE, (2002), GUTIERREZ, (2010)]. Entretanto, tais simuladores nem sempre estão disponíveis para uso e desenvolvimento de novas aplicações.

O ambiente VPE-qGM (*Visual Programming Environment for the qGM Model*) [MARON et. Al., (2010); MARON, et. Al., (2011)] e o editor QCEdit (*Quantum Circuit Editor*), implementados na linguagem Python sob a filosofia de software livre, estão em desenvolvimento visando o suporte à implementação e simulação distribuída de algoritmos da CQ. O QCEdit é uma ferramenta de edição gráfica de circuitos quânticos, que tem por objetivo tornar mais fácil e intuitiva a implementação de algoritmos quânticos. Posteriormente pretende-se realizar a integração destas duas ferramentas, fazendo-se assim a implementação dos algoritmos quânticos no QCEdit e a simulação no VPE-qGM. Considera-se a filosofia de software livre, permitindo um contínuo desenvolvimento do ambiente, expandindo as funcionalidades de acordo com as necessidades encontradas para construção de novos algoritmos quânticos. Como principal motivação, destaca-se a necessidade de cooperação para extensão das capacidades do ambiente, visto a complexidade associada a implementação de uma ferramenta para simulação quântica. Mais

especificamente, este artigo tem por objetivo descrever com mais detalhes a ferramenta de edição de circuitos quânticos *QCEdit*.

2 MÉTODOS, FERRAMENTAS E TECNOLOGIAS APLICADAS

Para desenvolvimento das extensões exigidas pelas ferramentas, foi realizado um estudo teórico para aprendizagem de conceitos básicos da CQ. Considerou-se, também, estudo dos fundamentos da linguagem *Python* [ASCHER, LUTZ, (2007)] e sua biblioteca gráfica *wxPython*, possibilitando a implementação das operações necessárias e construção e modificação das interfaces das ferramentas.

Salienta-se, ainda, a relevância da investigação de estruturação e descrição de algoritmos quânticos, tanto na forma de circuitos quânticos segundo modelo de Circuitos Quânticos, quanto na forma de processos elementares proposta no modelo qGM [REISER, (2010)].

3 RESULTADOS E DISCUSSÃO

O editor gráfico de circuitos quânticos *QCEdit* está em desenvolvimento, e sua interface gráfica está ilustrada na Figura 1.

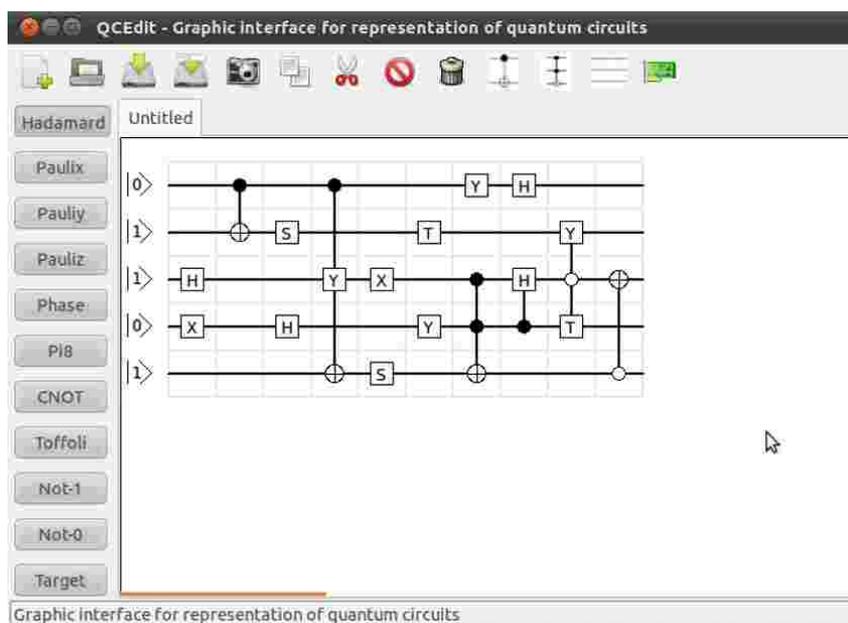


Figura 1: Interface gráfica do editor de circuitos quânticos *QCEdit*.

As principais funções para modelagem de circuitos quânticos são brevemente descritas na sequência:

(i) *Criação simultânea de circuitos*: O *QCEdit* permite que vários circuitos sejam abertos e/ou criados simultaneamente, para que possam ser editados ou até mesmo servir como auxílio na criação de outros circuitos. Gerenciando o caso de circuitos cuja descrição gráfica ocupa uma área maior que a tela do computador, tem-se barras de rolagem na área de edição para suporte a tal situação.

(ii) *Inserção e seleção de portas quânticas*: Seleciona-se a porta desejada, clicando no botão referente à porta, localizado na parte esquerda do editor, e utiliza-se o duplo-clique com o botão esquerdo do mouse na parte de edição. Assim, se não houver uma porta ocupando a posição mais próxima em relação ao clique, a porta escolhida será colocada nesta posição.

(iii) *Construção, inserção e seleção de portas controladas*: Selecionam-se as portas sobre as quais se deseja associar um operador de controle e, através de um clique com o botão esquerdo em cada uma delas, tem-se a opção de agrupamento e geração de novo operador. Existem duas formas para realizar o agrupamento: o “group” agrupa somente as portas selecionadas; o “groups”, que além das portas selecionadas, agrupa também seus respectivos controles, como no caso de iteração de operadores de controle. Ressalta-se que, para agrupar um conjunto de portas, todas precisam estar no mesmo nível da construção do sistema. A seleção sempre considera o intervalo entre a primeira e a última selecionada. Caso contrário, a operação de agrupamento não é realizada.

(iii) *Construção e inserção de portas personalizadas ou de portas padrões*: Portas e transformações como C-NOT e Tofolli, apesar de serem portas controladas, estão disponibilizadas na interface do editor.

(iv) *Deslocamento e remoção das portas quânticas*: Feita a partir de operações de *drag and drop*. Assim, antes de mudar a posição das portas, o gerenciador do sistema verifica se já não há nenhuma porta no local de destino. Caso contrário, é exibida uma mensagem de erro.

(v) *Operação de cópia*: Se dá a partir do menu ou barra de ferramentas. A operação é realizada pela seleção das portas a serem copiadas, sendo que controles e portas associadas também são copiadas. A realização da operação utiliza os mesmos princípios da inserção, bastando além do duplo-clique segurar a tecla *Control*.

(vi) *Operação de deleção de portas*: O editor disponibiliza duas opções: (a) a operação “*delete gate*”, que apaga somente as portas selecionadas, preservando seus controles e fazendo os ajustes necessários; (b) a operação “*delete gates*”, que além das portas selecionadas, deleta os controles e portas associadas.

(vii) *Recorte de portas e operadores*: Para recortar, aplica-se a composição de duas operações. Primeiramente, considera-se a operação de cópia e, posteriormente, a operação “*delete gates*”. Em ambas operações, recortar e copiar, a cópia que é mantida internamente no editor é compartilhada entre as abas de edição, facilitando assim a reutilização de partes (ou a totalidade) dos circuitos quânticos que já foram implementados, acelerando a criação de novos circuitos.

(viii) *Expansão e ajuste automáticos no dimensionamento gráfico dos circuitos*: O número de *qubits* e *steps* pode ser alterado de acordo com a aplicação, através da opção de ajustar circuito. Caso seja realizada alguma operação que exija mais *qubits/steps* do que foi inicialmente definido, a expansão do circuito ocorre automaticamente, evitando constantes ações por parte do usuário. Exemplos destas operações são: (a) arrastar um controle para uma parte mais inferior do circuito, ocasionando que uma parte dele acabe ficando além da área atual de edição; (b) uma operação de *copy/paste* que apresente as mesmas necessidades, por exemplo.

(ix) *Configuração de estados iniciais*: O estado inicial de um *qubit* pode ser alterado, podendo assumir, por exemplo, valor zero ou um, com a realização de um duplo-clique com o botão esquerdo na área na parte de edição referente ao qubit que se deseja alterar. O ambiente irá abrir outra janela onde poderá ser feita a alteração.

(xii) *Edição gráfica do conjunto de portas universais*: Todos os comandos de operações quânticas elementares, que definem conjuntos de portas universais, encontram-se no menu do editor e na barra de ferramentas, facilitando o acesso e agilizando a edição, com possível uso de teclas de atalho.

(x) *Geração de arquivo XML*: Para salvar o circuito que está sendo editado, e eventualmente abri-lo novamente, o *QCEdit* fornece a opção de salvá-lo no formato XML e recuperá-lo através do mesmo código. Este arquivo XML é utilizado

posteriormente para carregar o circuito no *VPE-qGM* e efetuar a simulação, quando as ferramentas forem integradas.

Algumas funcionalidades ainda devem ser adicionadas ao editor, com destaque para a opção de criação, por parte do usuário, de portas quânticas e adicioná-las ao editor, o que contribui para geração de circuitos mais eficientes e maior agilidade no envio para simulação (seqüencial ou paralela) no *VPE-qGM*. A opção criação de macros também viabiliza a geração de *portas genéricas*, sem necessidade de visualizar e/ou saber a complexidade das operações envolvidas. Além, é claro, das opções de *desfazer* e *refazer*, as quais muito colaboram na realização e edição dos circuitos.

4 CONCLUSÃO

A integração do *QCEdit* ao ambiente *VPE-qGM* permite a modelagem gráfica de algoritmos quânticos, não somente a partir das abstrações do modelo *qGM*, mas também pelo consolidado modelo de circuitos quânticos. O trabalho descreve a extensão de funcionalidades do *QCEdit*, viabilizando a descrição dos algoritmos em uma representação amplamente difundida, de mais alto nível. Assim, usufruindo de operações que facilitam a descrição de algoritmos quânticos, tem-se uma solução voltada para a simplificação desse processo de edição, tornando mais intuitiva a análise e interpretação dos algoritmos. Por fim, busca-se agregar a facilidade de criação do editor *QCEdit* às possibilidades de simulação distribuída do ambiente *VPE-qGM*.

5 REFERÊNCIAS

MARON, Adriano; PINHEIRO, Anderson; REISER, Renata; PILLA, Maurício. Distributed Quantum Simulation on the VPE-qGM. In: **WSCAD-SSC**, 2010. p.128-135.

MARON, Adriano; REISER, Renata; PILLA, Maurício. Distributed Quantum Simulation on the VPE-qGM. In: Anais **DINCON 2011**, 2011. p.1-6.

HANNEKE, D; HOME, J. P.; JOST, J. D.; AMINI, J. M.; LEIBFRIED, D.; WINELAND, D. J. Realizations of a Programmable Two-Qubit Quantum Processor. **Nature Physics**, v. 6, p. 13-16, 2009.

GUTIERREZ, Eladio; ROMERO, Sergio; TRENAS, Maria; ZAPATA, Emilio. Quantum Computer Simulation Using the CUDA Programming Model. **Computer Physics Communications**, p. 283-300, 2010.

KNILL, Emanuel; NIELSEN, Michael. **Theory of quantum computation**. 2002.

WATANABE, Hiroshi. **QCAD: GUI environment for Quantum Computer Simulator**. Super Computing Division, Information Technology Center, University of Tokyo, 2002.

GROVER, Lov. A Fast Quantum Mechanical Algorithm for Database Search. In: **Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing**, p. 212-219, 1968.

NIELSEN, Michael; CHUANG, Isaac. **Computação Quântica e Informação Quântica**. Local de Edição: Bookman, 2003.

REISER, Renata; AMARAL, Rafael. The quantum states space in the qGM Model. In: **Anais Workshop Escola de Comp. e Inf. Quântica**, Petrópolis/RJ, 2010, p.92-101.

ASCHER, David; LUTZ, Mark. **APRENDENDO PYTHON**. Local de Edição: Bookman, 2007.