Paraconsistência e Computação Quântica

Walter Carnielli

Centro de Lógica, Epistemologia e História da Ciência – CLE Departamento de Filosofia - IFCH UNICAMP

SQIG- Security and Quantum Information Group, IT- Lisboa

< WECIQ | 2010 >

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

Motivações

Versão forte da Tese de Church-Turing: todo modelo ‘razoável’ de computação calcula, em tempo equivalente, a mesma classe de funções que podem ser calculadas por máquinas de Turing determinísticas.

Os modelos de computação quântica representam o maior desafio para a Versão forte da Tese de Church-Turing.

Questões:

- Qual é a lógica subjacente aos modelos de computação quântica?
- Noções lógicas podem ser úteis no total entendimento da computação quântica?

Motivações

As máquinas de Turing são equivalentes (quanto à computabilidade e complexidade algorítmica) às famílias uniformes de circuitos Booleanos, e podem ser axiomatizadas em FOL.

O modelo de circuitos Booleanos é baseado na lógica clássica.

A lógica subjacente aos modelos de computação clássicos (equivalentes à máquina de Turing) é a lógica clássica.

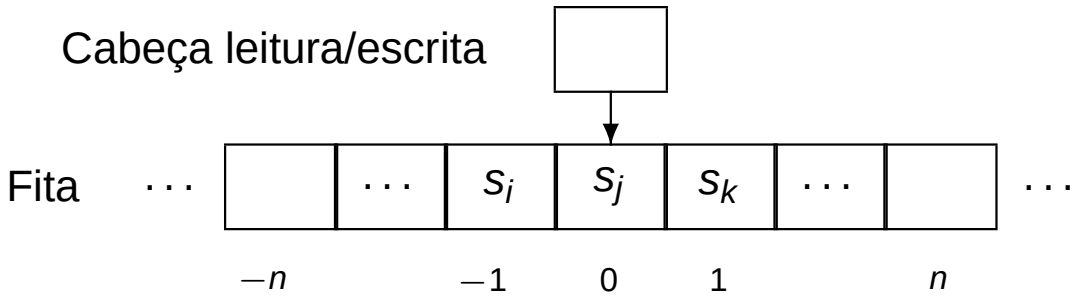
Questões:

- É possível definir modelos de computação baseados em lógicas não-clássicas?
- Há vantagens em modelos de computação baseados em lógicas não-clássicas?

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

Máquinas de Turing (MTs)



- Instruções: (I) $q_i s_j s_k q_l$, (II) $q_i s_j R q_l$, (III) $q_i s_j L q_l$.
- Duas instruções são ambíguas se coincidem nos dois símbolos iniciais.
- Uma máquina de Turing é determinística (MTD) se não têm instruções ambíguas, caso contrario é não-determinística (MTND).
- As MTNDs são mais eficientes do que as MTDs? (**P** $=?$ **NP**)

Circuitos Booleanos

Definição

Um circuito booleano é uma coleção finita de variáveis de entrada e portas lógicas conectadas de maneira direcionada e acíclica, onde as variáveis de entrada tomam valores em $\{0,1\}$ e cada porta calcula uma operação booleana (usualmente AND, OR ou NOT).

- Circuitos booleanos computam funções da forma $f\colon \{0,1\}^n \to \{0,1\}^m$ (com tamanho da entrada n fixo).
- Para computar uma função com tamanho da entrada variável deve-se definir uma família uniforme de circuitos booleanos.

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

Máquinas de Turing quânticas (MTQs)

Generalização das MTs considerando os elementos da máquina (símbolos nas células da fita, estado e posição da máquina) como sendo observáveis de um sistema microscópico (Deutsch, 1985):

- Uma MTQ pode estar num estado superposto, que é uma superposição linear de estados de MTs clássicas.
- A evolução da máquina é descrita através de um operador unitário.
- Os estados superpostos e a linearidade dos operadores unitários permitem processamento em paralelo: paralelismo quântico.
- No final da computação deve ser realizada uma medição, obtendo-se um único elemento da superposição de maneira probabilística.
- O paralelismo pode ser aproveitado através da interferência quântica.

Computando em MTQs

Computações em modelos de computação não-determinísticos são usualmente representadas através de árvores onde os nós representam configurações da máquina e as arestas representam transições entre configurações.

- Nas MTNDs uma computação específica corresponde a um caminho da árvore.
- Nas MTQs uma única computação percorre todos os caminhos simultaneamente. Diferentes caminhos podem interferir (construtiva ou destrutivamente).

Circuitos quânticos

Generalização do modelo de circuitos booleanos através das leis da mecânica quântica (Deutsch, 1989):

- Os valores de entrada/saída são vetores unitários num espaço de estados bidimensional (chamados qubits). Um qubit pode estar numa superposição dos estados $|0\rangle$ e $|1\rangle$ (estados superpostos).
- As portas são operadores unitários.
- Os estados superpostos e a linearidade dos operadores de evolução permitem processamento em paralelo.
- No final da computação deve ser realizada uma medição, obtendo-se um resultado de maneira probabilística.
- O paralelismo pode ser aproveitado através da interferência quântica.

MARCOS DA COMPUTAÇÂO QUÂNTICA

- 1982, Richard Feynman: propôs que efeitos quânticos poderiam oferecer algo realmente novo em computação

- 1985, David Deutsch: primeiro modelo teórico da Máquina de Turing Quântica mostrando que qualquer processo físico poderia ser modelado por um computador quântico.

- 1989, David Deutsch: introduziu o modelo de circuitos quânticos.

MARCOS DA COMPUTAÇÂO QUÂNTICA

- 1993, Charles Bennett e pesquisadores da IBM: teleportação é possível, desde que se destrua a amostraoriginal.

- 1996, Lov K. Grover: propossta de um algoritmo quântico para busca em bancos de dados, quadraticamente mais rápido que os análogos clássicos.

- 1998, Isaac Chuang: desenvolvimento do primeiro computador quântico de 1 q-bit

- 2001, IBM: computador quântico de 7q-bits que fatora o número 15 usando o algoritmo de Shor

O Computador Quântico - O bit quântico (q-bit)

- A Mecânica Quântica "mora" no Espaço de Hilbert que é um espaço vetorial dotado de um produto interno chamado norma.
- Agora teremos estados quânticos para representar os bits e portanto o chamaremos **q-bit**. Os valores 0 e 1 serão substituídos pelos **vetores** $|0\rangle$ e $|1\rangle$ representados por

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{e} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

- Grande diferença! Agora podemos ter um q-bit genérico $|\psi\rangle$ resultado da combinação linear dos estados $|0\rangle$ e $|1\rangle$

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

- α e β são números complexos.

O Computador Quântico - O bit quântico (q-bit)

- Os vetores $|0\rangle$ e $|1\rangle$ formam uma base ortonormal do espaço vetorial $\mathbb{C}^2$. Base computacional.
- $|\psi\rangle$ é chamado de **superposição** dos vetores da base com amplitudes α e β.
- $|\psi\rangle$ está simultaneamente nos estados $|0\rangle$ e $|1\rangle$!!!!
- Podemos armazenar uma quantidade de informação infinita em $|\psi\rangle$!! Essa informação está no mundo quântico.
- Para trazê la ao mundo clássico precisamos fazer uma medida. A medida altera o estado do q-bit!!
- Teremos então:
  - $|0\rangle$ com probabilidade $|\alpha|^2$;
  - $|1\rangle$ com probabilidade $|\beta|^2$.
- α e β não podem ser conhecidos através de uma medida. Além disso

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

O Computador Quântico - O bit quântico (q-bit)

Há duas interações básicas de um computador quântico com os dados de entrada:

- Transformação unitária - evolução temporal atua no nível quântico, não temos acesso! Qualquer matriz unitária de ordem 2×2 pode ser usada! É um processo **reversível**.
- Medida. Faz a ligação entre o mundo quântico e clássico!
- Como fazer para aproveitarmos toda essa informação armazenada em um q-bit?

O Computador Quântico - Produto tensorial

Se quisermos tratar de sistemas de mais de um q-bit temos que introduzir o conceito de **produto tensorial**. Sejam dois estados

$$|\psi\rangle = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_m \end{bmatrix} \quad \text{e} \quad |\varphi\rangle = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_p \end{bmatrix}, \tag{6}$$

O produto tensorial entre eles resultará no vetor $|\chi\rangle = |\psi\rangle \otimes |\varphi\rangle$ com $mp-$linhas. Também podemos usar as seguintes notações para o produto tensorial $|\psi\rangle \otimes |\varphi\rangle$, $|\psi\rangle\,|\varphi\rangle$, $|\psi, \varphi\rangle$ e $|\psi\varphi\rangle$.

O Computador Quântico - Produto tensorial

$$|\chi\rangle = \begin{bmatrix} \psi_1\varphi_1 \\ \psi_1\varphi_2 \\ \vdots \\ \psi_1\varphi_p \\ \psi_2\varphi_1 \\ \psi_2\varphi_2 \\ \vdots \\ \psi_2\varphi_p \\ \vdots \\ \psi_m\varphi_1 \\ \psi_m\varphi_2 \\ \vdots \\ \psi_m\varphi_p \end{bmatrix}, \tag{7}$$

onde $\psi_i\varphi_j$ é o produto usual dos números complexos.

O Computador Quântico - Produto tensorial

Vejamos alguns exemplos:

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

e

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

O produto tensorial **não** é comutativo!

O Computador Quântico - Produto tensorial

Podemos generalizar o produto tensorial para matrizes quaisquer. Sejam as matrizes $A \in \mathbb{C}^{m \times n}$ e $B \in \mathbb{C}^{p \times q}$, a matriz $A \otimes B \in \mathbb{C}^{mp \times nq}$ é dada por:

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \cdots & A_{mn}B \end{bmatrix}, \tag{8}$$

onde A_{ij} é o elemento da coluna i e linha j da matriz A.

O Computador Quântico - Produto tensorial

Exemplo:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{e} \quad \mathrm{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

então

$$A \otimes B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

O Computador Quântico - 2 q-bits

Seja $|\psi\rangle$ o estado genérico de 2 q-bits. Ele será uma superposição dos estados $|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$, isto é,

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle\,, \tag{10}$$

onde

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1.$$

Podemos tentar simplificar a notação considerando os zeros e uns que aparecem nos vetores,$|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$ como números binários e escrevê- los em sua notação decimal

$$|0\rangle\,, |1\rangle\,, |2\rangle \quad \text{e} \quad |3\rangle\,.$$

O Computador Quântico - 2 q-bits

Em geral, um estado de n q-bits é uma superposição dos 2^n estados da base computacional $\{|0\rangle, |1\rangle, \ldots |2^n - 1\rangle\}$, dada por

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \tag{11}$$

e as amplitudes α_i obedecendo à conservação de probabilidade

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1. \tag{12}$$

O Computador Quântico - Emaranhamento

Um estado de 2 q-bits pode ou não ser o resultado do produto tensorial de 2 estados de 1 q-bit! Sejam dois estados de 1 q-bits

$$|\varphi\rangle = a\,|0\rangle + b\,|1\rangle$$

e

$$|\psi\rangle = c\,|0\rangle + d\,|1\rangle\,,$$

onde a, b, c e d $\in \mathbb{C}$. Então

$$\begin{aligned} |\varphi\rangle \otimes |\psi\rangle &= (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) \\ &= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle\,. \end{aligned} \tag{13}$$

Temos então que um estado de 2 q-bits genérico (10)só é da forma (13)se, e somente se,

$$\alpha = ac,$$

$$\beta = ad,$$

$$\gamma = bc,$$

$$\delta = bd.$$

O Computador Quântico - Emaranhamento

Dessas relações de igualdade, temos que

$$\frac{\alpha}{\beta} = \frac{c}{d} \quad \text{e} \quad \frac{\gamma}{\delta} = \frac{\mathrm{c}}{\mathrm{d}}.$$

Portanto,

$$\alpha\delta = \beta\gamma. \tag{14}$$

Em geral, um estado de 2 q-bits **não** é o produto tensorial de dois q-bits!! Um estado de 2 q-bits é dito estado **emaranhado** quando este estado não pode ser escrito como produto tensorial de dois estados de 1 q-bit.

O Computador Quântico - Emaranhamento

Vejamos:

$$|01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Já o estado

$$|\zeta\rangle = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

é um estado **emaranhado**, pois não pode ser escrito como produto de dois q-bits.

Exercício: Prove que $|\zeta\rangle$ é um estado emaranhado!

O Computador Quântico - Produtos interno e externo

O produto interno entre dois estados $|\varphi\rangle$ e $|\psi\rangle \in \mathbb{C}^n$ é denotado por $\langle\varphi|\psi\rangle$ e definido como sendo o produto matricial entre $|\varphi\rangle^\dagger$ e $|\psi\rangle$, isto é,

$$\langle\varphi|\psi\rangle = (|\varphi\rangle)^\dagger |\psi\rangle = \sum_{i=1}^{n} \varphi_i^* \psi_i. \tag{15}$$

Propriedades:

1. $\langle\varphi|\psi\rangle = \langle\psi|\varphi\rangle^*$,
2. $\langle\psi|(a|u\rangle + b|v\rangle)\rangle = a\langle\psi|u\rangle + b\langle\psi|u\rangle$,
3. $\langle\psi|\psi\rangle > 0$, se $|\psi\rangle \neq 0$,

com a,b $\in \mathbb{C}$ e $|\psi\rangle, |\varphi\rangle, |u\rangle, |v\rangle \in \mathbb{C}^n$.
Exercício: Demonstre as propriedades acima!

Circuitos Quânticos - Porta NOT Quântica

- No caso clássico a porta NOT troca 0 por 1 e vice versa;
- No caso quântico temos o operador (transformação linear unitária) X
  - $X|0\rangle = |1\rangle$
  - $X|1\rangle = |0\rangle$
- Sua representação matricial é dada por

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

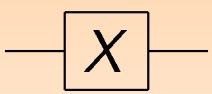


Figura: Porta X (NOT quântica).

Exercício: Prove que o operador X é unitário.

Circuitos Quânticos - Porta NOT Quântica

Situação sem contrapartida no caso clássico!

Seja o estado

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

aplicando o operador X a saída será

$$X\,|\psi\rangle = X\alpha\,|0\rangle + X\beta\,|1\rangle = \alpha\,|1\rangle + \beta\,|0\rangle\,.$$

As probabilidades foram trocadas!!!

Circuitos Quânticos - Porta Hadamard

Esta é uma porta muito utilizada.

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

- $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Figura: Porta Hadamard.

Exercício: Prove que a porta H é unitária.

Circuitos Quânticos - Porta Hadamard

$$\begin{aligned} H^{\otimes 2}|00\rangle &= H|0\rangle \otimes H|0\rangle \\ &= \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle). \end{aligned}$$

Em notação decimal,

$$H^{\otimes 2}|00\rangle = \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle).$$

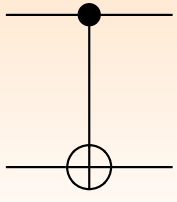
Generalizando para estados com n q-bits, obtemos:

$$\begin{aligned} H^{\otimes n}|0\ldots0\rangle &= (H|0\rangle)^{\otimes n} \\ &= \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)^{\otimes n} \\ &= \frac{1}{\sqrt{2^n}} \sum^{2^n-1} |i\rangle. \end{aligned}$$

Circuitos Quânticos - Porta CNOT Quântica

Podemos usar as diversas portas de 1 q-bit para transformar o estado $|0\ldots\rangle$ de n q-bits em qualquer estado do tipo $|\psi_1\rangle\,|\psi_2\rangle\ldots|\psi_n\rangle$, onde cada um desses $|\psi_i\rangle$ é uma superposição arbitrária $\alpha\,|0\rangle + \beta\,|1\rangle$. Mas todos estes estados são do tipo produto, ou seja, não emaranhados. Para obtermos estados emaranhados precisamos de portas que atuem sobre os múltiplos q-bits.

- 2 q-bits de entrada **controle** e **alvo**;
- Se o bit de controle está no estado $|1\rangle$ ela é ativada caso contrário não;
- Os bits alvo e controle podem estar superpostos ou emaranhados.
- A porta CNOT é universal. Qualquer operador unitário pode ser representado usando portas CNOT e portas de um q-bit.



$$\begin{aligned} |00\rangle &\rightarrow |00\rangle \\ |01\rangle &\rightarrow |01\rangle \\ |10\rangle &\rightarrow |11\rangle \\ |11\rangle &\rightarrow |10\rangle \end{aligned}$$

Circuitos Quânticos - Porta Toffoli Quântica

Esta é uma porta controlada por dois q-bits

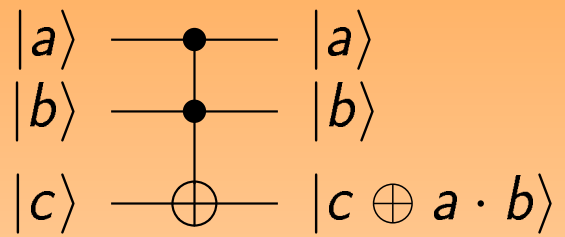


Figura: Porta Toffoli Quântica.

Sua ação na base computacional pode ser representada por:

$$|i, j, k\rangle \rightarrow |i, j, k \oplus ij\rangle \,, \tag{18}$$

onde $i, j \in \{0, 1\}$ e $\oplus$ é a adição módulo 2. A base computacional possui 8 elementos. Em geral ela é muito utilizada para simplificar a representação de circuitos quânticos.

Circuitos Quânticos - Porta Toffoli Quântica

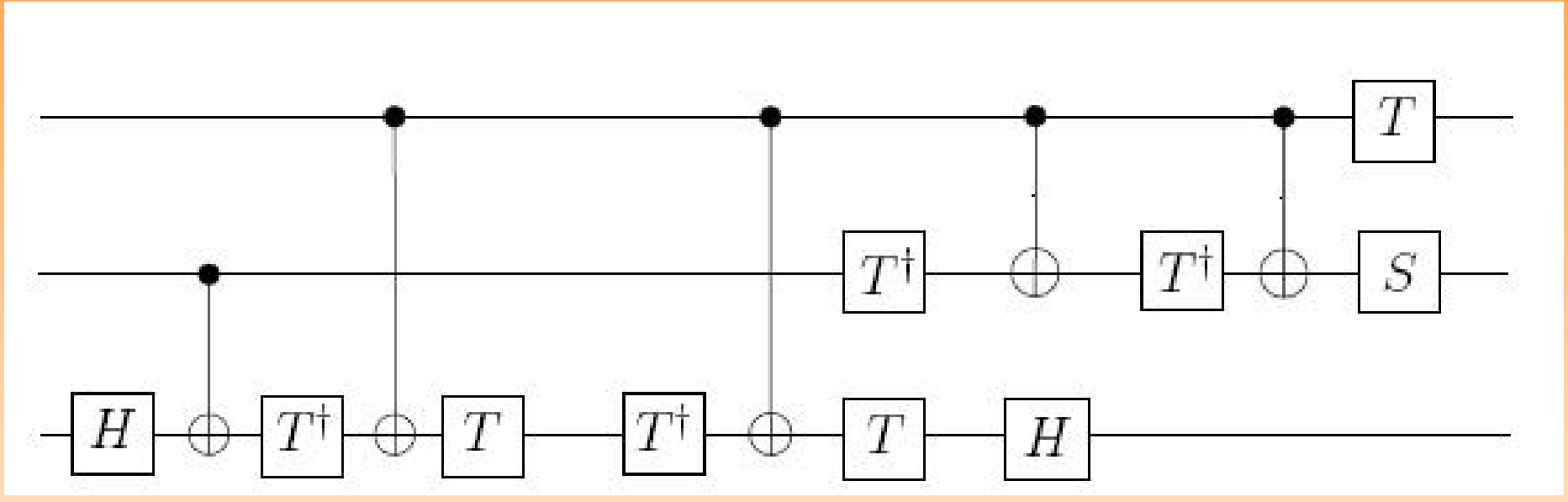


Figura: Decomposição da porta Toffoli em portas de 1 q-bit e portas CNOT.

Circuitos Quânticos - Paralelismo Quântico

Uma operação quântica é sempre unitária e portanto reversível. Então, um computador quântico precisa de dois registradores para realizar uma computação:

- Um registrador para guardar o estado de entrada;
- Um registrador para guardar o estado de saída.

A computação de uma função f é determinada por uma operação unitária U_f que deve atuar sobre os dois registradores preservando a entrada e efetuando a operação no segundo.

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle . \tag{19}$$

Se $y = 0$, então,

$$U_f |x\rangle |0\rangle = |x\rangle |0 \oplus f(x)\rangle = |x\rangle |f(x)\rangle . \tag{20}$$

Circuitos Quânticos - Paralelismo Quântico

Suponha agora que preparamos um registrador com m q-bits no estado $|\psi\rangle$ de superposição igualmente distribuída e um registrador com n q-bits no estado $|0\rangle$

$$|\psi\rangle\,|0\rangle = \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle\,|0\rangle.$$

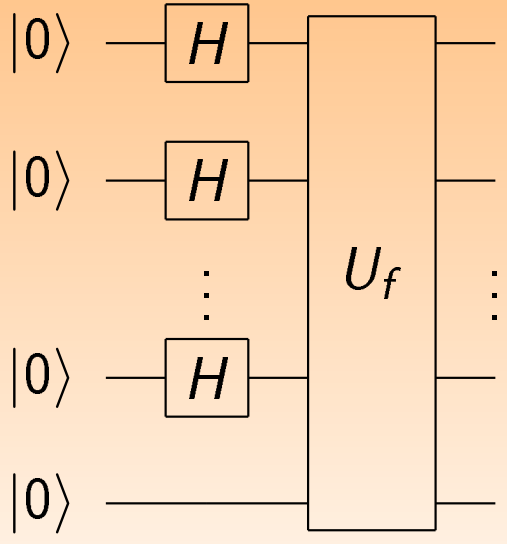


Figura: Circuito que calcula o valor de f para todos os valores de x.

Circuitos Quânticos - Paralelismo Quântico

Aplicando U_f a este estado obtemos:

$$\begin{aligned} U_f |\psi\rangle |0\rangle &= U_f \left(\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle |0\rangle \right) \\ &= \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} U_f |x\rangle |0\rangle \\ &= \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle |f(x)\rangle. \end{aligned} \tag{21}$$

Este circuito realiza o cálculo de todos os 2^m valores $f(0), f(1), \ldots, f(2^m - 1)$ ao mesmo tempo com uma única aplicação da operação unitária U_f. Este é o **Paralelismo Quântico**.

Exemplos de portas quânticas

- Hadamard (H): serve para criar estados superpostos.

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \qquad \begin{aligned} |0\rangle &\mapsto \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle &\mapsto \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

- Negação (X): inverte os valores da base.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \begin{aligned} |0\rangle &\mapsto |1\rangle \\ |1\rangle &\mapsto |0\rangle \end{aligned}$$

- Negação controlada (X_c): nega o segundo qubit se o primeiro é 1.

$$X_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \begin{aligned} |0,0\rangle &\mapsto |0,0\rangle \\ |0,1\rangle &\mapsto |0,1\rangle \\ |1,0\rangle &\mapsto |1,1\rangle \\ |1,1\rangle &\mapsto |1,0\rangle \end{aligned}$$

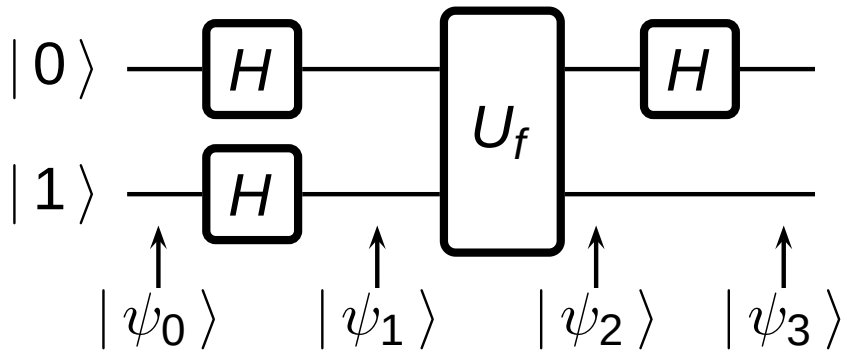
- Toffoli (T): negação duplamente controlada.

Algoritmos quânticos

- Algoritmo de Deutsch: determina se uma função $f\colon \{0,1\} \to \{0,1\}$ é constante ou balanceada, usando uma chamada ao oráculo que computa f.
- Algoritmo de Deutsch-Josza: generaliza o algoritmo anterior para funções $f\colon \{0,1\}^n \to \{0,1\}$.
- Algoritmo de Simon: determina o período de uma função $f\colon \{0,1\}^n \to \{0,1\}^n$ tal que, para todo $x \neq x'$, $f(x) = f(x')$ se e somente se $x' = x \oplus s$, com um número polinomial de portas.
- Algoritmo de Shor: permite fatorar números inteiros com um número polinomial de portas.
- Algoritmo de Grover: permite encontrar um dado, numa base de dados desordenada, com um número $O(\sqrt{n})$ de portas.

Algoritmo de Deutsch

$$U_f|x,y\rangle = |x, y \oplus f(x)\rangle$$



$$|\psi_0\rangle = |0,1\rangle$$

$$|\psi_1\rangle = \frac{1}{2}(|0,0\rangle - |0,1\rangle + |1,0\rangle - |1,1\rangle)$$

$$|\psi_2\rangle = \frac{1}{2}(|0,0\oplus f(0)\rangle - |0,1\oplus f(0)\rangle + |1,0\oplus f(1)\rangle - |1,1\oplus f(1)\rangle)$$

$$|\psi_3\rangle = \begin{cases} \pm|0\rangle \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) & \text{se } f(0) = f(1), \\ \pm|1\rangle \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) & \text{se } f(0) \neq f(1). \end{cases}$$

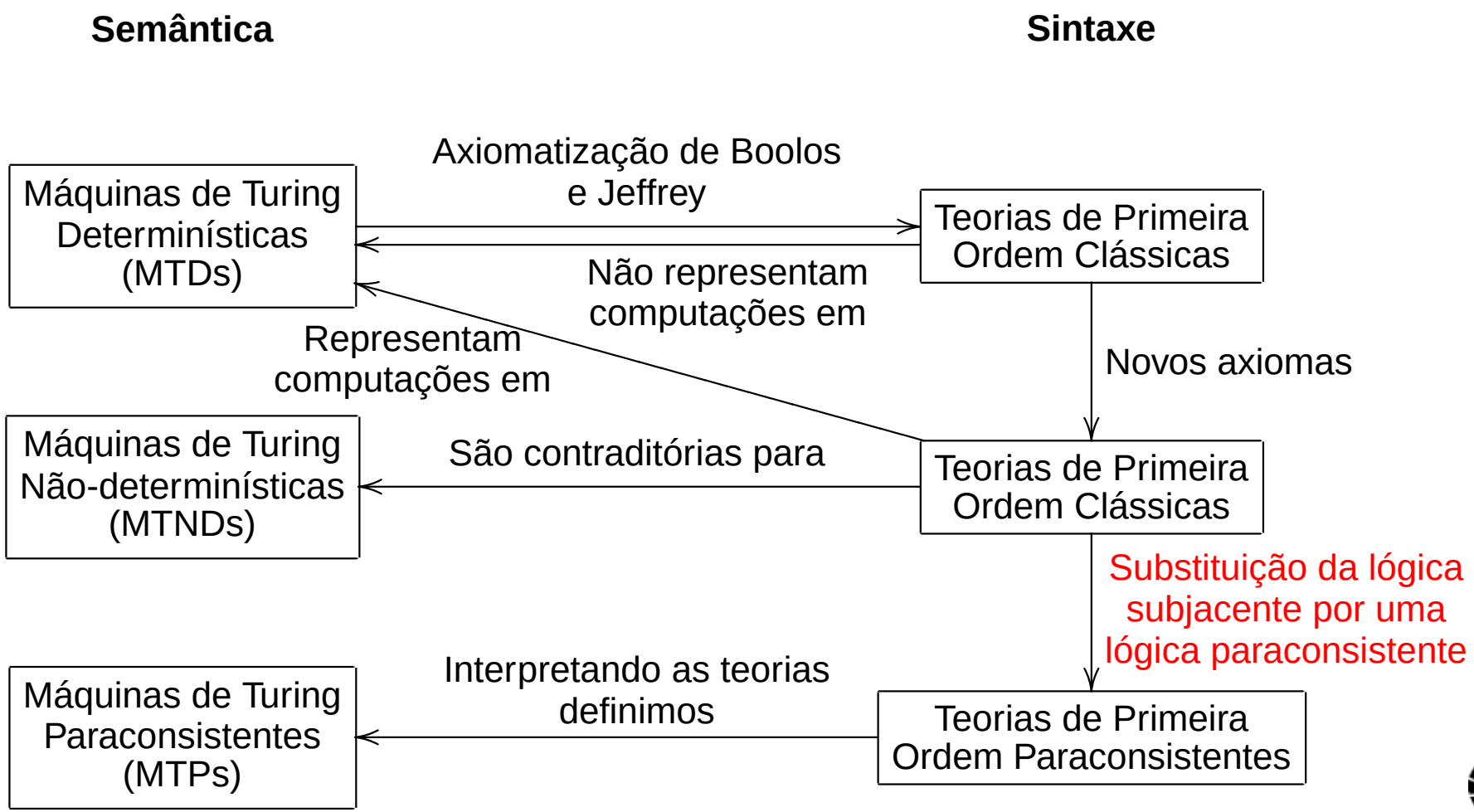
Algoritmo de Deutsch

- As primeiras portas H geram um estado superposto.
- A porta U_f computa $f(0)$ e $f(1)$ em paralelo.
- A ultima porta H gera interferência, de tal maneira que o primeiro qubit fica em $|0\rangle$ se $f(0) = f(1)$ ou fica em $|1\rangle$ em caso contrario.
- Na medição do primeiro qubit obtém-se 0 (com probabilidade 1) se $f(0) = f(1)$ ou 0 (com probabilidade 1) em caso contrario.

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

Definindo as Máquinas de Turing Paraconsistentes



Máquinas de Turing Paraconsistentes

Máquinas de Turing Paraconsistentes (MTPs): definidas através da lógica $LFI1^*$, lógica paraconsistente de primeira ordem na hierarquia de Lógicas da Inconsistência Formal (LFIs).

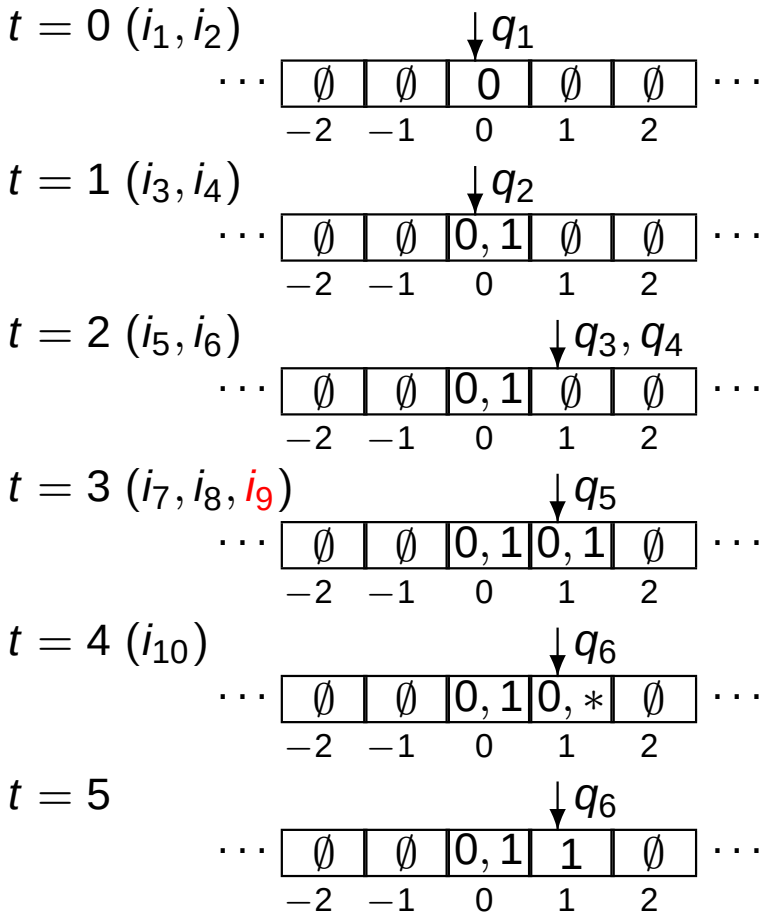
Definição

Uma MTP é uma MTND tal que:

- *As instruções ambíguas são executadas simultaneamente, gerando multiplicidade de símbolos em células da fita, e multiplicidade de estados ou posições da máquina.*
- *Permite adicionar condições de inconsistência (ou multiplicidade) nos primeiros dois símbolos da instrução: $q^\bullet$ (resp. $s^\bullet$) significa que a instrução é executada somente se o estado (resp. o símbolo lido) é múltiplo.*

Exemplo MTP

Exemplo: seja $\mathcal{M}$ uma MTP com instruções: i_1: $q_1 00 q_2$, i_2: $q_1 01 q_2$, i_3: $q_2 0 R q_3$, i_4: $q_2 1 R q_4$, i_5: $q_3 \emptyset 0 q_5$, i_6: $q_4 \emptyset 1 q_5$, i_7: $q_5 00 q_6$, i_8: $q_5 10 q_6$, i_9: $q_5 0^\bullet * q_6$ and i_{10}: $q_6 * 1 q_6$.



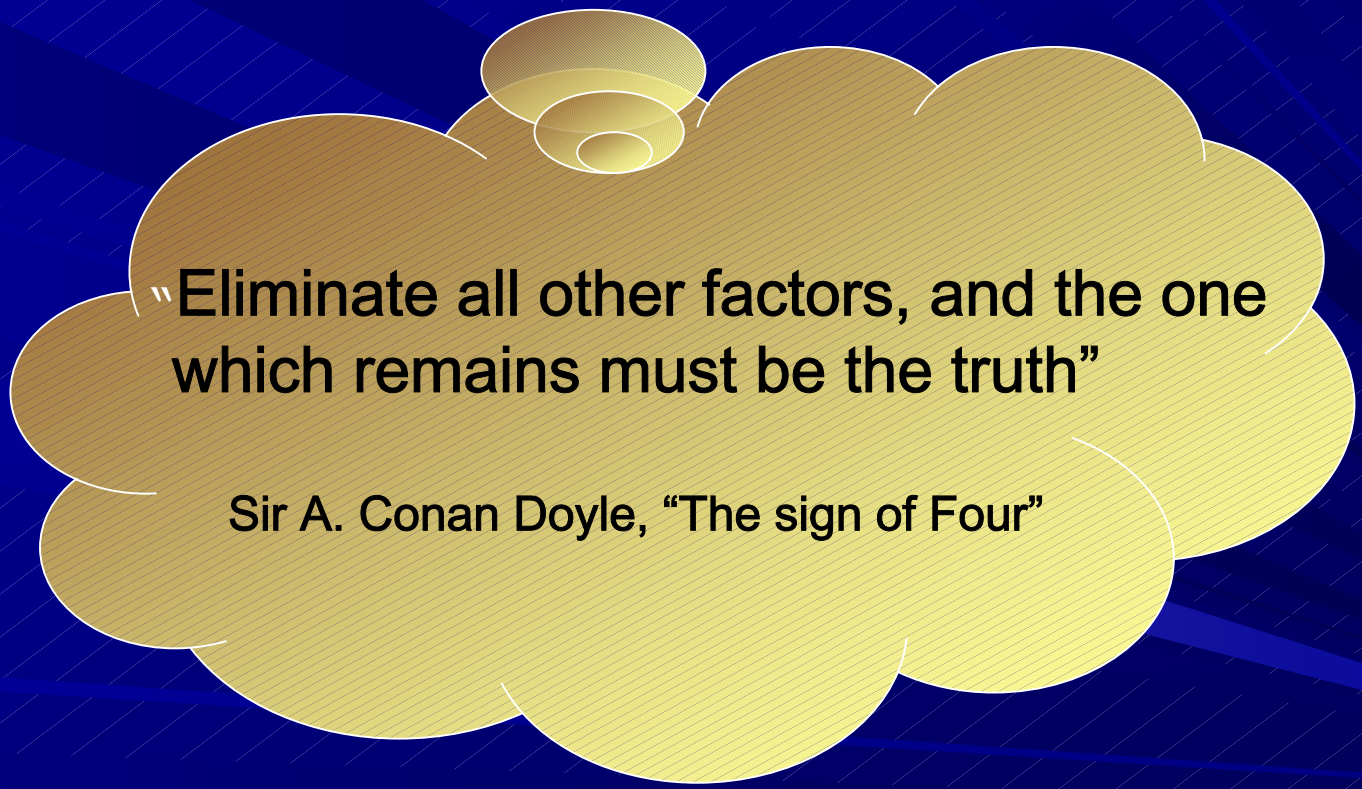
Exemplo MTP

A máquina $\mathcal{M}$ pode ser interpretada da seguinte maneira:

- As instruções i_1 e i_2 geram uma multiplicidade de símbolos na posição 0 da fita.
- As instruções i_3 a i_6 computam a função identidade (para os valores 0 e 1 em paralelo)
- As instruções i_7 a i_{10} determinam, fazendo uso de condições de inconsistência nas instruções, se o valor obtido é múltiplo ou não.

Considerando as instruções i_3 a i_6 como sendo um oráculo, tais instruções poderiam computar qualquer outra função $f\colon \{0,1\} \to \{0,1\}$, portanto $\mathcal{M}$ é uma solução paraconsistente para o problema de Deutsch.

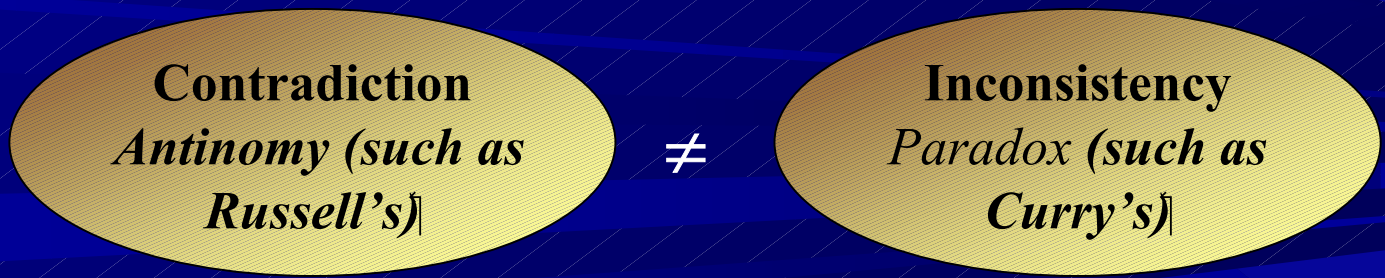
Do not let any contradiction stop you, and truth will appear …



How to take profit of a contradiction?

- Finance authorities are happy to find a contradiction in your IRS statements…
- Police investigation too;

  "Have you and your friend been in the same place last week ?"
- Only if contradiction appears, we can be *sure* to have received wrong information from one of your sources!



Inconsistency: $\circ\alpha$ as imaginary numbers

Softned hermemeutics

- $\alpha \wedge \neg\alpha$ can be seen as true
- $(\alpha).(-\alpha)=1 \Rightarrow \alpha^2 = -1 \Rightarrow \alpha = \pm\sqrt{-1}$
- view $\circ\alpha$ (α is consistent) as "$\alpha \in \Re$" and there is *no* solution
- view $\bullet\alpha$ (α is inconsistent) as "$\alpha \notin \Re$" and there are *two* solutions

LFI's very quickly

- **LFI**'s are non-explosive:
  $\alpha, \neg\alpha \nvdash \beta$, but yet are

- *finitely gently explosive*:
  $\circ\alpha, \alpha, \neg\alpha \vdash \beta$, for all β;

What is the meaning?

$\circ\alpha$ is a "safe" or "uncontroversial" statement, which cannot support contradictions

The scope of negation

- α = "*n* is an odd number"

  Negation as "hard": From : α , $\neg\alpha \vdash \beta$, all β because *"odd number"* is an uncontroversial predicate: so $\circ\alpha$ holds, and we are in the case

  $\circ\alpha$, α , $\neg\alpha \vdash \beta$, all β

- α "*n* is a big number"

Negation as "soft" α , $\neg\alpha \nvdash \beta$ for all β because "*big number* " is controversial, unsafe, debatable!

What do we gain?

- **LFI**'s separate contradictoriness $\{\alpha, \neg\alpha\}$, from inconsistency $\bullet\alpha$
- non-consistency $\neg\circ\alpha$ from inconsistency $\bullet\alpha$
- non-inconsistency $\neg\bullet\alpha$ from consistency $\circ\alpha$;
- **LFI**s do *not* validate contradictions or invalidate the 'Principle of Non-Contradiction' ("there are non-contradictory theories").

The basic logic of (in)consistency: bC

bC: add to C_{min} a new rule, realizing the Gentle Principle of Explosion:

(bc1) $\circ\alpha, \alpha, \neg\alpha \vdash_{\mathbf{bC}} \beta$.

- If α is consistent and contradictory, then it explodes'
- Contradiction $\neq$ inconsistency

Ci: a stronger LFI

Ci* = *bC + the axiom:

$$\bullet\alpha \vdash (\alpha \wedge \neg\alpha),$$

(defining $\bullet\alpha =_{def} \neg\circ\alpha$).

Now, In *Ci*, inconsistency and contradiction ***are*** equivalent: we had before $(\alpha \wedge \neg\alpha) \vdash_{bC} \neg\circ\alpha$

Nice properties of Ci

- $\circ\alpha, \bullet\alpha \vdash_{Ci} \beta$;
- $\circ\alpha, \neg\circ\alpha \vdash_{Ci} \beta$;
- $\bullet\alpha, \neg\bullet\alpha \vdash_{Ci} \beta$
- $\vdash_{Ci} \circ\circ\alpha$

Any classical reasoning can be simulated within Ci (and bC)!!

In ***Ci*** we can recover "classical negation"

defined as $\sim\alpha =_{def} \neg\alpha \wedge \circ\alpha$, or $\dot{\sim}\alpha =_{def} \alpha \rightarrow (\alpha \wedge \sim\alpha)$

Thus Classical Logic can be "simulated" inside ***Ci***
by means of $\wedge$, $\vee$, $\rightarrow$ and $\sim$ (or $\dot{\sim}$)

Ci is a subclassical logic, but *PC* can be translated inside *Ci* (and also in the weaker *bC*)

Máquinas de Turing Paraconsistentes

As MTPs permitem um certo tipo de paralelismo, porém:

Teorema

Toda MTP pode ser simulada em tempo polinomial por uma MTD.

As MTPs (definidas usando a lógica $LFI1^*$) não apresentam vantagens quanto à computabilidade e complexidade algorítmica em relação aos modelos de computação clássicos. Contudo, permitem dequantizar o algoritmo de Deutsch.

Máquinas de Turing Paraconsistentes Não-Separáveis

MTPs Não-Separáveis (MTPNSs): definido usando um fragmento da lógica modal $S5$, com negação paraconsistente ($\neg_{\Diamond} A \stackrel{\text{def}}{=} \Diamond \neg A$) e conjunção não-separável ($A \wedge_{\Diamond} B \stackrel{\text{def}}{=} \Diamond(A \wedge B)$).

Definição

Uma MTPNS é uma MTND tal que:

- *quando alcança uma configuração ambígua, com n possíveis instruções a executar, a máquina se divide em n cópias, executando uma instrução diferente em cada cópia.*
- *Permite adicionar condições de inconsistência: $q^{\bullet}$ (resp. $s^{\bullet}$) significa que a instrução é executada somente se o estado (resp. o símbolo lido) é diferente em diferentes cópias da máquina.*

Máquinas de Turing Paraconsistentes Não-Separáveis

Uma MTPNS pode ser vista como uma MTND onde todos os caminhos de computação são executados simultaneamente e onde podem ser executadas instruções levando em consideração configurações em caminhos diferentes.

Teorema

SAT pode ser computado em tempo polinomial, de maneira determinística, por uma MTPNS.

Se **P** $\neq$ **NP**, então a complexidade algorítmica é relativa a lógica.

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

MTPs vs. MTQs

- Situações de multiplicidade (que correspondem a inconsistências) nas MTPs podem ser vistas como estados superpostos uniformes.
- Os algoritmos de Deutsch e Deutsch-Josza podem ser 'simulados' por MTPs, preservando eficiência.
- As MTPs não permitem uma representação direta de estados emaranhados (i.e. estados onde valores de diferentes elementos estão correlacionados, como é o caso do estado $|\psi\rangle = \frac{1}{\sqrt{2}}(|0,0\rangle + |1,1\rangle)$).
- As MTPs não permitem uma representação direta da noção de fase relativa.

MTPNSs vs. MTQs

- As MTPNSs se aproximam mais às MTQs: permitem uma representação direta de estados emaranhados.
- As MTPNSs também não permitem uma representação direta da noção de fase relativa.
- O mecanismo provido pelas MTPNSs para aproveitar o paralelismo (as condições de consistências nas instruções) parece ser mais eficiente do que o mecanismo provido pelas MTQs (a interferência quântica).

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

Circuitos sobre Lógicas Não-Clássicas

- Generalizar os circuitos booleanos para lógicas finitamente valoradas é simples: valores de entrada/saída correspondem aos valores de verdade e as portas realizam as funções associadas aos conetivos lógicos.
- Como generalizar para lógicas não caraterizáveis por matrizes finitas (como é o caso de varias lógicas paraconsistentes)? Quais seriam os valores de entrada? Quais seriam as operações das portas?

Cálculo de Anéis de Polinômios

- O Cálculo de Anéis de Polinômios (CAP) consiste basicamente em traduzir fórmulas de uma lógica em polinômios com coeficientes num corpo finito (ou corpo de Galois), e realizar deduções através de operações sobre esses polinômios.
- Através da introdução de variáveis ocultas podem ser definidos CAPs para lógicas não-caracterizáveis por matrizes finitas, como é o caso da lógica paraconsistente *mbC* e a lógica modal *S*5.

L-circuitos

Definição

Seja L uma lógica provida de CAP sobre um corpo F. Um L-circuito é uma coleção finita de variáveis de entrada e portas lógicas conectadas de maneira direcionada e acíclica, onde as variáveis de entrada tomam valores em F e cada porta calcula o polinômio associado ao conetivo no CAP.

- Variáveis ocultas podem ser introduzidas por conetivos lógicos.
- As variáveis ocultas produzem indeterminismo, mesmo que as entradas sejam determinísticas.
- O indeterminismo pode ser usado para simular certo tipo de processamento em paralelo.

Circuitos Paraconsistentes

mbC-circuitos:

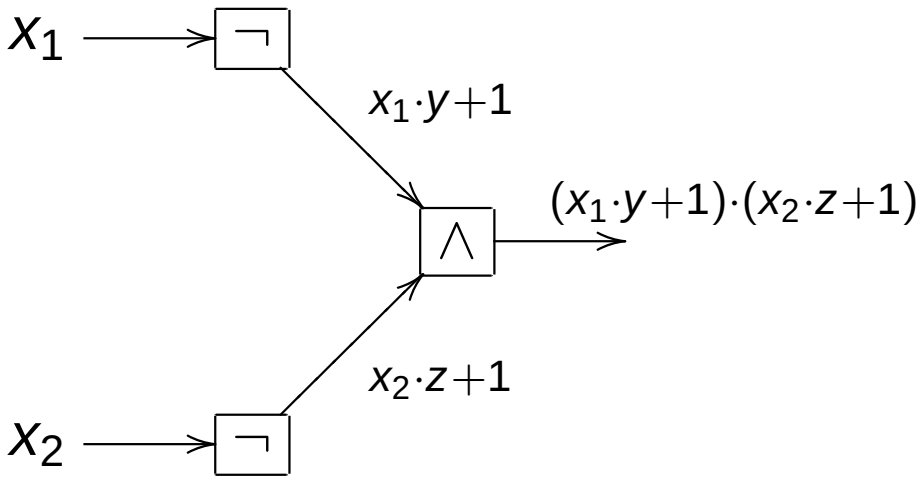
- Permitem trocar variáveis de entrada por variáveis ocultas, dando a possibilidade de traduzir computações determinísticas a não-determinísticas sem considerar entradas aleatórias.
- Não existe nenhum tipo de correlação entre variáveis.

*S*5-circuitos:

- Existem fortes correlações entre diferentes variáveis que permitem determinar a satisfatibilidade de fórmulas de CPL com um número polinomial de portas.
- Num fragmento paraconsistente de *S*5 a satisfatibilidade de fórmulas de CPL com um número polinomial de portas é ainda possível.

Exemplo de *mbC*-circuito

mbC-circuito correspondente à fórmula $\neg p_1 \wedge \neg p_2$ (y e z são variáveis ocultas)



x_1	x_2	$\neg x_1$	$\neg x_2$	$\neg x_1 \wedge \neg x_2$
0	0	1	1	1
0	1	1	$z+1$	$z+1$
1	0	$y+1$	1	$y+1$
1	1	$y+1$	$z+1$	$(y+1)\cdot(z+1)$

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

L-circuitos vs. CQs

- O indeterminismo introduzido pelas variáveis ocultas nos *L*-circuitos pode ser usado para simular o indeterminismo nos CQs.
- A porta de Hadamard pode ser parcialmente simulada pela porta de negação nos *mbC*-circuitos e pela porta de inconsistência no fragmento paraconsistente dos *S*5-circuitos.
- Relações entre variáveis algébricas podem ser usadas para representar estados emaranhados.
- O paralelismo pode ser obtido através do indeterminismo controlado.

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

Modelos de computação não-local

- No modelo de MTs clássicas todas as operações são realizadas de maneira local (a execução de uma instrução só pode mudar o símbolo na posição atual).
- Os estados emaranhados da mecânica quântica permitem certo tipo de ação a distância, portanto, os modelos de computação quântica podem ser considerados intrinsecamente não-locais.
- O modelo de MTPNS permite representar estados emaranhados, pode portanto também ser considerado um modelo de computação não-local.
- *L*-circuitos com variáveis ocultas correlacionadas também podem ser legitimamente considerados como modelos de computação não-local.

Sumário

1. Motivações
2. Modelos de computação clássicos
3. Modelos de computação quânticos
4. Máquinas de Turing Paraconsistentes
5. Relações entre MTQs e MTPs/MTPNSs
6. Circuitos Paraconsistentes
7. Relações entre circuitos quânticos e paraconsistentes
8. Comentários referentes a não-localidade
9. Conclusões

Conclusões

- Propomos noções lógicas para interpretar propriedades quânticas: estados superpostos correspondem a estados inconsistentes e estados emaranhados correspondem a conjunções não-separáveis.
- Mostramos como alguns algoritmos quânticos podem ser dequantizados.
- Abrimos um caminho para interpretar circuitos quânticos através das variáveis ocultas intruduzidas por conetivos não-classicos.
- MTPNS e *L*-circuitos são modelos abstratos que permitem estudar, desde um ponto de vista lógico, as características da computação não-local.

REFERÊNCIAS

Paraconsistent Machines and their Relation to Quantum Computing. Juan C. Agudelo; Walter Carnielli
Journal of Logic and Computation 20(2):573-595,2010.
doi: 10.1093/logcom/exp072

Introdução à Computação Quântica. Amanda Castro Oliveira; Renato Portugal. LNCC/MCT, slides de mini-curso, 28 a 31/01/2008
www.lncc.br/pdf_consultar.php?idt_arquivo=2445&mostrar=1

Introduction to the logics of formal inconsistency and possible-translations semantics: A tutorial
Walter Carnielli, Slides- IRIT- LILAC, May 2010, TOulouse.

A Taxonomy of C-Systems. W.A. Carnielli; João Marcos In: Paraconsistency-the logical way the inconsistent, Lecture Notes in Pure and Applied Mathematics, Vol. 228, pp. 01-94 2002. (Eds. W.A. Carnielli, M. E. Coniglio and I.M.L. D'Ottaviano) Marcel Dekker, New York.
Pre-print available from CLE e-Prints
http://www.cle.unicamp.br/e-prints/abstract_5.htm

Logics of Formal Inconsistency. W.A. Carnielli, M.E. Coniglio and J. Marcos. HANDBOOK OF PHILOSOPHICAL LOGIC, 2nd edition, 14, p.15-107 (Eds. D. Gabbay and F. Guenthner), Springer, 2007.
Pre-print available from IST server
http://www.cs.math.ist.utl.pt/ftp/pub/MarcosJ/03-CCM-lfi.pdf