JUAN CARLOS AGUDELO AGUDELO

DA COMPUTAÇÃO PARACONSISTENTE À COMPUTAÇÃO QUÂNTICA

Dissertação de Mestrado apresentada ao Departamento de Filosofia do Instituto de Filosofia e Ciências Humanas da Universidade Estadual de Campinas sob a orientação do Prof. Dr. Walter Carnielli.

Este exemplar corresponde à redação final da dissertação defendida e aprovada pela Comissão Julgadora em 08/05/2006.

BANCA
Prof. Dr. Walter Carnielli (Orientador)
Prof. Dr. Marcelo Finger
Prof. Dr. Marcelo E. Coniglio
Profa. Dra. Itala M. Loffredo D'Ottaviano (Suplente)
Prof. Dr. Silvio Seno Chibeni (Suplente)

MAIO/2006

JUAN CARLOS AGUDELO AGUDELO

DA COMPUTAÇÃO PARACONSISTENTE À COMPUTAÇÃO QUÂNTICA

Dissertação de Mestrado apresentada ao Departamento de Filosofia do Instituto de Filosofia e Ciências Humanas da Universidade Estadual de Campinas sob a orientação do Prof. Dr. Walter Carnielli.

Este exemplar corresponde à redação final da dissertação defendida e aprovada pela Comissão Julgadora em 08/05/2006.

BANCA
Prof. Dr. Walter Carnielli (Orientador)
Prof. Dr. Marcelo Finger
Prof. Dr. Marcelo E. Coniglio
Profa. Dra. Itala M. Loffredo D'Ottaviano (Suplente)
Prof. Dr. Silvio Seno Chibeni (Suplente)

MAIO/2006

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IFCH - UNICAMP

Agudelo, Juan Carlos Agudelo

Ag91d Da computação paraconsistente à computação quântica / Juan Carlos Agudelo Agudelo. - - Campinas, SP : [s. n.], 2006.

Orientador: Walter Carnielli.
Tese (Mestrado) - Universidade Estadual de Campinas, Instituto de Filosofia e Ciências Humanas.

1. Turing - máquinas. 2. Funções computáveis. 3. Lógica. 4. Circuitos Lógicos. 5. Computação quântica. I. Carnielli, Walter A. (Walter Alexandre), 1952- II. Universidade Estadual de Campinas. Instituto de Filosofia e Ciências Humanas. III.Título.

Palavras-chave em inglês (Keywords): Turing machines
Computable functions
Logic
Logical circuits
Quantum computation

Área de Concentração: Lógica

Titulação: Mestre em Filosofia

Banca examinadora: Prof. Dr. Walter Carnielli (Orientador)
Prof. Dr. Marcelo Finger
Prof. Dr. Marcelo E. Coniglio
Profa. Dra. Itala M. Loffredo D'Ottaviano (Suplente)
Prof. Dr. Silvio Seno Chibeni (Suplente)

Data da defesa: 08 de maio de 2006.

A minha mãe e à memória de meu pai,
por tudo e mais um pouco!

Agradecimentos

Foram várias as pessoas que participaram de maneiras diversas da realização do presente trabalho, e é portanto meritório e oportuno expressar meus agradecimentos àqueles e àquelas que colaboraram de modo influente no que há de bom nesta dissertação, esperando não deixar ninguém de fora.

Sem dúvida a quem mais devo pela elaboração desta dissertação e, em geral, pela feliz realização do meu mestrado, é a meu orientador, o Professor Walter Carnielli. Foi em boa parte pelo seu apoio que vim ao Brasil para estudar na UNICAMP, e foi ele quem me desafiou a relacionar o modelo de máquinas de Turing paraconsistentes (idéia que tinha começado a desenvolver na Colômbia, numa especialização em lógica, sob orientação do Professor Andrés Sicard) com modelos de computação quântica, intuição que se mostrou bastante acertada. Além disso, várias das idéias aqui apresentadas, e muito do que possa ser considerado de bom estilo na apresentação das idéias e na redação em português são a ele devidas.

Ao Professor Marcelo Coniglio e à Professora Itala D'Ottaviano, por seus ensinamentos e seu apoio constante.

Aos meus colegas pela amizade e por comigo compartilhar seus conhecimentos, em particular a Evandro Gomes, Rodrigo de Alvarenga Freire, Juliana Bueno-Soler, Paulo Petrillo e Samir Gorsky. Ao Samir, também por me ajudar com algumas correções de português.

Aos membros da banca por terem levado a sério sua função com a maior competência e seriedade, e por suas sugestões e comentários muito pertinentes.

Aos funcionarios do CLE e do IFCH por me disponibilizar todos os recursos necessários (ou talvez suficientes) e pela ajuda na elaboração dos trâmites burocráticos.

À minha mãe, às minhas irmãs, aos meus familiares e aos meus amigos por seu apoio incondicional e por tantos momentos belos que têm dado sentido à vida.

Resumo

As diferentes interpretações da mecânica quântica levanta sérios problemas filosóficos a respeito da natureza do mundo físico e do estatuto das teorias físicas. Tais interpretações desempenham um papel importante na compreensão dos modelos de computação quântica, e por sua vez os modelos de computação quântica abrem a possibilidade de se confrontar as teses filosóficas que se atrevem a responder a tais problemas. Apesar das relevantes e surpreendentes promessas de uso pragmático e tecnológico da computação quântica, não é por essa vereda que caminha este trabalho: o que aqui se oferece é um novo paradigma de computação (um modelo de computação baseado no paradigma paraconsistente), e se propõe uma nova interpretação da computação quântica através desse novo paradigma, dessa forma colaborando simultaneamente na discussão filosófica a respeito da noção de computabilidade e da mecânica quântica.

O presente trabalho introduz a definição do que será chamado de modelo de *máquinas de Turing paraconsistentes* (MTPs). Tal modelo de computação é uma generalização do modelo clássico de máquinas de Turing. No modelo de máquinas de Turing paraconsistentes, diferentemente do modelo clássico, permite-se a execução de múltiplas instruções de maneira simultânea, dando lugar a multiplicidade de símbolos em diferentes casas da fita, multiplicidade de estados e multiplicidade de posições da máquina. Considerando que tal multiplicidade de configurações, embora essencial nas MTPs, pode ser interpretado como incoerências com respeito às máquinas de Turing clássicas, permite-se acrescentar condições de consistência e inconsistência na execução das instruções nas MTPs, o que servirá então para controlar o estado de incoerência do sistema.

Depois de apresentar o modelo de MTPs, são descritos os modelo de máquinas de Turing quânticas (MTQs) e o modelo de circuitos quânticos (CQs), ambos introduzidos inicialmente por David Deutsch, os quais são, respectivamente, generalizações do modelo de máquinas de Turing clássicas

e de circuitos booleanos clássicos usando as leis da mecânica quântica.

Finalmente, estabelecem-se relações entre o modelo de MTPs e os modelos de computação quântica, simulando algoritmos quânticos simples (um CQ que soluciona o chamado problema de Deutsch e um CQ que soluciona o chamado problema de Deutsch-Josza) e mostrando que o paralelismo quântico, uma característica essencial da computação quântica, pode em alguns casos ser simulado por meio de MTPs. Dessa forma, apesar de o particular modelo de MTPs aqui apresentado ter algumas restrições na simulação de certas características da computação quântica, abre-se a possibilidade de se definir outros modelos de MTPs de maneira a simular tais características.

Em resumo, o presente trabalho, na medida em que oferece um novo paradigma de computação (a saber, a computação paraconsistente) e uma nova interpretação dos modelos de computação quântica (a saber, a interpretação da computação quântica através da computação paraconsistente) contribui para a discussão filosófica a respeito da interpretação dos modelos de computação quântica, e possivelmente da interpretação da própria mecânica quântica. Contudo, não menos importante é o fato de que, apesar de o presente trabalho não pretender se dedicar a questões puramente técnicas da computabilidade, ele de fato abre um imenso campo de investigação a respeito da computação relativizada à lógica e suas implicações – no caso presente, relativizada à lógica paraconsistente.

Abstract

The interpretations of quantum mechanics open serious philosophical questions about the nature of the physical world and about the status of physical theories. Such interpretations play an important role in the understanding of models of quantum computing, and models of quantum computing, by their turn, open possibilities to confront and test philosophical theses that dare to address such problems. Although the promising pragmatic and technological applications of quantum computing, this work goes in another way: what is here offered is a new paradigm of computation based upon the paraconsistency paradigm, and a new interpretation of quantum computing through this model, in this way simultaneously collaborating in the philosophical discussion about the concepts of computability and of quantum mechanics.

This work introduces the definition of what will be called the model of *paraconsistent Turing machines* (PTMs). Such computational model is a generalization of the classical model of Turing machines. In the PTMs model, differently from the classical Turing machines model, simultaneous execution of multiple instructions is allowed, giving rise to a multiplicity of symbols on different cells of the tape, multiplicity of machine states and multiplicity of machine positions. Such multiplicity of configurations, though essential in the PTMs, can be seen as incoherencies with respect to classical Turing machines; to compensate this, the PTMs permit to operate with consistency and inconsistency conditions to control the global state of incoherence of the system.

After introducing the PTMs, the model of quantum Turing machines (QTMs) and the model of quantum circuits (QCs) are presented. This models are due to David Deutsch and are, respectively, generalizations of the model of classical Turing machines and the model of classical boolean circuits, using the laws of quantum mechanics.

Finally, relations between the model of PTMs and models of quantum

computing are established, which permits to simulate simple quantum algorithms (a QC to solve the so called Deutsch problem and a QC to solve the so called Deutsch-Jozsa problem) by PTMs. It is also shown that quantum parallelism, an essential characteristic of quantum computation, may be simulated in some cases by PTMs. Although the particular model of PTMs here presented has some restrictions in the capacity to simulate certain quantum computing characteristics, our work opens the possibility to define other PTM models which could simulate such characteristics.

To sum up, the present work, while offering a new paradigm of computation (namely, paraconsistent computation) and a new interpretation of quantum computing (namely, interpretation of quantum computation by means of paraconsistent computation) contributes to the philosophical discussion about the interpretation of quantum computation and of the quantum mechanics itself. However, not less important is the fact that, besides its lack of explicit intention towards tecnical questions of computability theory, opens a new line of research about the possibilities of logic-relativized computation and its implications- in the present case, relativized to paraconsistent logics.

Sumário

Introdução e justificativas

O presente trabalho apresenta uma definição e oferece uma fundamentação lógica do que será chamado de modelo de *máquinas de Turing paraconsistentes* (MTPs). Tal modelo é uma generalização do modelo de máquinas de Turing clássico. O modelo de MTPs é obtido através da proposta de um método de axiomatização que produz teorias em primeira ordem, as quais representam formalmente computações por meio de máquinas de Turing determinísticas (MTDs). Quando tal método é utilizado para axiomatizar computações de máquinas de Turing não determinísticas (MTNDs) produz-se, em alguns casos, teorias contraditórias, e portanto triviais, supondo como lógica subjacente destas teorias a lógica de primeira ordem clássica. Para evitar o problema da trivialização destas teorias existem pelo menos duas possibilidades: a *solução clássica* que consiste em modificar o procedimento de axiomatização para evitar o surgimento de contradições e a *solução paraconsistente* que consiste em substituir a lógica subjacente dessas teorias por uma lógica paraconsistente, permitindo suportar as contradições sem levar à trivialização das teorias. Ambas as soluções são investigadas, com maior ênfase à solução paraconsistente, que permite definir o modelo de MTPs a partir da interpretação de teorias inconsistentes mas não triviais.

Na definição do método de axiomatização para MTDs (Capítulo 1, Seção 1.2.1), considera-se inicialmente um método de axiomatização para máquinas de Turing proposto por George Boolos e Richard Jeffrey em [13], por meio do qual eles demonstram que a lógica de primeira ordem é não decidível. Se a lógica de primeira ordem clássica fosse decidível resultaria que o problema da parada de uma máquina de Turing também o seria; contudo, é conhecido que o problema da parada não é decidível, e portanto a lógica de primeira ordem também não o é.

Para determinar "boas propriedades" das teorias que axiomatizam as computações das MTDs, definem-se estruturas matemáticas para expressar suas computações (Capítulo 1, Definição 1.10), e com base em tais estruturas, adotando as definições de representação de funções e relações em

teorias arbitrárias consideradas em [79] (ver também [42] ou [21]), define-se formalmente o que significa uma computação ser representada numa teoria (Capítulo 1, Definição 1.13). Tal definição é introduzida de maneira original no presente trabalho para justificar formalmente a adequação das teorias de primeira ordem na axiomatização do comportamento das máquinas de Turing.

O método de axiomatização de MTDs definido por Boolos e Jeffrey é adequado para o objetivo que eles perseguem, mas, usando a definição formal fornecida de representação de uma computação numa teoria, demonstra-se que as teorias produzidas com o método de Boolos e Jeffrey não representam de fato as computações das MTDs (Capítulo 1, Teorema 1.6). Como saída, para produzir teorias em primeira ordem que representem as computações das MTDs, acrescentam-se novos esquemas axiomáticos ao método de axiomatização (Capítulo 1, Seção 1.2.3). A Figura 1 esquematiza os passos seguidos na definição do método de axiomatização de MTDs (onde "axiomatização B-J" significa o método de axiomatização de Boolos e Jeffrey).

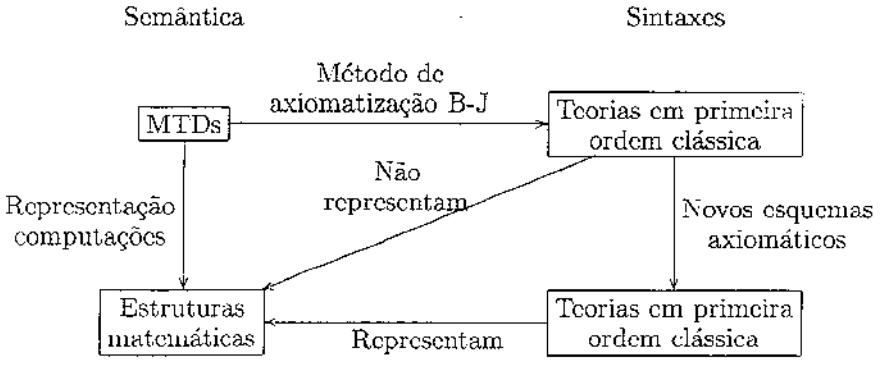

Figura 1: Axiomatização das MTDs

Uma vez definido o procedimento para se obter teorias que representem computações das MTDs, demonstra-se que utilizando tal método para MTNDs obtém-se, em alguns casos, teorias contraditórias, portanto triviais, supondo a lógica de primeira ordem clássica como a lógica subjacente dessas teorias, o que leva a um fracasso na tentativa de axiomatização das MTNDs. Mostra-se então como é possível modificar o método de axiomatização de tal forma a se obter teorias que evitam as contradições e representam com-

putações legítimas em MTNDs (Capítulo 1, Seção 1.3).

Ainda que a tentativa de axiomatizar o comportamento das MTNDs através do método obtido para MTDs leve a um fracasso, devido ao surgimento de contradições e a trivialização das teorias, tal fracasso pode ser evitado substituindo a lógica subjacente das teorias obtidas por uma lógica paraconsistente. O Capítulo 2 dedica-se então a definir o modelo de MTPs usando o método de axiomatização para MTDs descrito no capítulo anterior para axiomatizar MTNDs, substituindo a lógica subjacente pela lógica paraconsistente de primeira ordem **LFI1*** apresentada em [20]. Desse modo é possível se obter um arcabouço conceitual que suporta inconsistências e evita a trivialização das teorias. Através da análise do mecanismo de inferência dedutiva das teorias paraconsistentes obtidas define-se o modelo de MTPs.

Aproveitando-se do fato de que **LFI1*** é não somente axiomatizável, mas é caracterizada (isto é, é correta e completa) com respeito a uma noção adequada de estrutura, adaptam-se as definições de representação numa teoria para as teorias cuja lógica subjacente seja **LFI1***, o que possibilita demonstrar formalmente que o modelo de MTPs definido é realmente representado pelas teorias paraconsistentes em consideração, desta maneira se justificando lógicamente o modelo construído.

A partir da definição do modelo de MTPs, estuda-se a possibilidade de se definir condições de consistência e inconsistência na execução de instruções neste modelo, permitindo assim tirar um maior proveito das MTPs. Por último, descrevem-se outras possíveis formas de se obter outros modelos de MTPs. A Figura 2 esquematiza os passos seguidos na definição do modelo de MTPs.

A teoria da computação paraconsistente é, de certa forma, uma noção que nasce aqui. A única menção prévia a essa idéia de que se tem notícia é apenas uma rápida menção em [78], onde foi aventada a idéia de *máquinas dialéticas*: estas seriam máquinas de Turing que agiriam sob uma certa "lógica dialética" quando encontrassem contradições; os autores contudo não oferecem uma definição clara[1] para tais máquinas, nem estudam a questão

[1] Os autores alegam (cf. [78, p. 196]) por exemplo que "não é difícil descrever como uma máquina encontra contradições: para alguma sentença A, ambos A e $\neg A$ aparecem na entrada ou na saída"; mas o que significa "aparecer" A e $\neg A$ na entrada ou na saída da máquina? Que sentido tem "aparecer" $\neg A$ na entrada ou na saída da máquina? Seria "aparecer" $\neg A$ equivalente a não ocorrer A? Logo depois afirmam que "[quando uma contradição aparece], em contraste [com a máquina clássica], as máquinas programadas com uma lógica dialética continuam satisfatoriamente com a computação "; mas como continuariam?

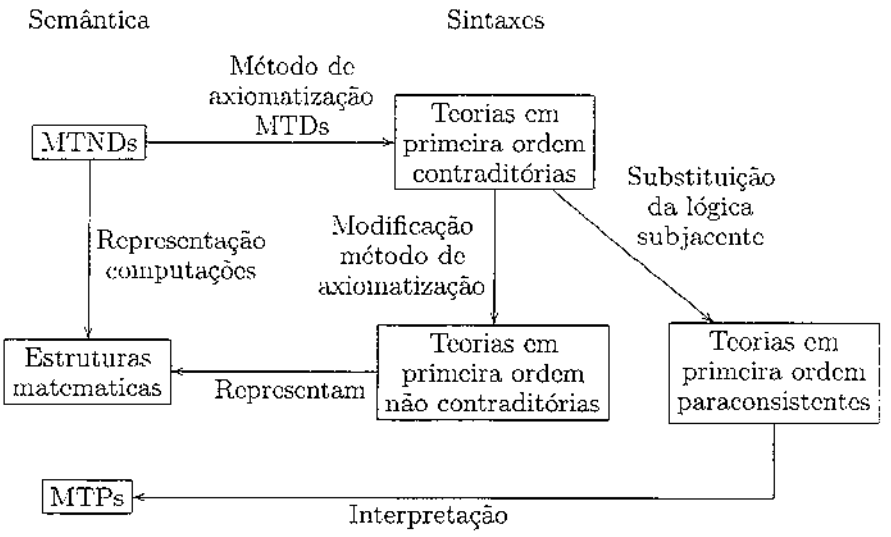

Figura 2: Definição do modelo de MTPs

em profundidade. A primeira definição precisa de um modelo de MTPs foi apresentada em [1] (mais tarde publicada em [2]). O presente trabalho retoma os resultados de [1], justificando de uma melhor maneira a proposta do modelo de MTPs e alcançando uma fundamentação lógica mais adequada do modelo obtido, graças à noção de representação de uma computação numa máquina numa teoria formal de primeira ordem. Além disso estabelecem-se relações surpreendentes entre o modelo de MTPs e certos modelos de computação quântica, abrindo a possibilidade de se conseguir simular algoritmos quânticos através de MTPs (Capítulo 4). Para conseguir compreender tais relações é necessário partir de uma idéia precisa dos modelos de computação quântica, e é para esta finalidade que o Capítulo 3 apresenta uma descrição das características essenciais destes modelos.

À primeira vista, pode parecer estranho tentar relacionar modelos de computação quântica e modelos de MTPs, devido ao fato de que, em princípio, a mecânica quântica e as lógicas paraconsistentes tratam de objetos totalmente diferentes com enfoques distintos: a mecânica quântica trata das leis do mundo microfísico, enquanto as lógicas paraconsistentes tratam das possibilidades de fundamentar o raciocínio na presença de contradições, e até

mesmo tirar proveito das contradições. Contudo, algumas interpretações da mecânica quântica, em particular a interpretação de Copenhagen e a interpretação de múltiplos mundos, permitem vislumbrar relações entre modelos de computação quânticos e MTPs. Existem ainda outras interpretações, como a interpretação por variáveis ocultas e as teorias com ontologias de potência (cf. [23], Capítulo 7). Todas estas interpretações são de alguma forma controversas, e todas levantam questões sofisticadas do ponto de vista da filosofia da ciência; em especial, as teorias com ontologias de potência partiram de propostas de Karl Popper com base na formulação da interpretação da probabilidade como propensão; suas propostas cristalizaram-se em [69] (ver também [45]). Para nossos propósitos, é conveniente ater-se apenas à interpretação de Copenhagen e a interpretação de múltiplos mundos; para tornar claras a influência que estas interpretações exercem na relação entre os modelos de computação quânticos e os modelos de computação paraconsistente apresenta-se primeiro uma breve descrição destas duas interpretações.

A formalização da mecânica quântica, tal como é hoje conhecida, é o resultado de duas formulações independentes, a *mecânica matricial* proposta por Werner Heisenberg em 1925 e a *mecânica ondulatória* proposta por Erwin Schrödinger em 1926 (cf. [51, p. 344]). Nesse mesmo ano, o próprio Schrödinger demonstrou a equivalência entre as duas formulações. Posteriormente as duas formulações foram imersas numa teoria mais geral e compreensível, em boa parte graças ao esforço de Paul Dirac. Outro passo crucial na formalização da mecânica quântica foi dado por John von Neumann ao propor o formalismo dos espaços de Hilbert como fundamento matemático de uma formulação da teoria. Para mais detalhes históricos sobre os primódios da mecânica quântica recomenda-se [55].

A partir da primeira formulação da mecânica quântica foram várias as interpretações que surgiram para justificar tais formulações, interpretações estas com pontos de divergência muito fortes, e cada uma delas com enfoques muito discutidos. É por isso que ainda hoje, 80 anos após o primeiro formalismo da mecânica quântica, a questão da interpretação destes formalismos continua ainda um tema de acalorada discussão.

Entre as diferentes interpretações dos formalismos da mecânica quântica a de maior aceitação (embora recentemente outras interpretações estejam ganhando aceitação, segundo [80]), é a chamada de *interpretação de Copenhagen*, também chamada de *interpretação da complementaridade*. Tal interpretação, porém, não consiste em um conjunto de idéias únicas e claramente definidas, mostrando-se como um denominador comum de vários pontos de vista divergentes em múltiplos aspectos, baseada principalmente na noção de *complementaridade* de Niels Bohr.

Como descrito por Max Jammer em [55, p. 104], mesmo que não seja fácil se definir a noção de complementaridade sugerida por Bohr, a noção de interpretação da complementaridade parece ser menos difícil de se definir. Pode-se dizer que uma teoria T admite uma interpretação da complementaridade se as seguintes condições são satisfeitas:

1. T contém (pelo menos) duas descrições D_1 e D_2 de seus aspectos substanciais.

2. D_1 e D_2 referem-se ao mesmo universo de discurso U (no caso de Bohr, a microfísica).

3. Nem D_1 nem D_2, isoladamente, dão conta exaustiva dos fenômenos de U.

4. D_1 e D_2 são mutuamente exclusivas, no sentido em que sua combinação, numa descrição única, leva a contradições lógicas.

No caso da mecânica quântica, D_1 e D_2 podem ser tomadas como as descrições que interpretam os objetos microfísicos como ondas e como partículas, respectivamente, levando à chamada *dualidade onda-partícula*.

A noção de interpretação da complementaridade sugere, naturalmente, a possibilidade de se utilizar alguma lógica paraconsistente como lógica subjacente a teorias que admitam tal tipo de interpretação, em particular à mecânica quântica. Esta é precisamente a idéia que trabalham Newton da Costa e Decio Krause em [31] (ver também [30]), onde eles chamam de *c-theories* às teorias com estas características. A noção de interpretação da complementaridade sugere também que se lance mão da idéia de *semântica de traduções possíveis* (introduzida por Walter Carnielli em [17] e reelaborada em [18] e em [35]): aspectos complementares poderiam ser pensados como diferentes traduções do mesmo estado de coisas.

Ainda a respeito da descrição da interpretação de Copenhagen, é importante ressaltar dois princípios fundamentais de tal interpretação:

1. Um sistema quântico, enquanto não é medido, encontra-se num estado autenticamente indeterminado (estado de superposição); não faz sentido afirmar que o sistema está num estado específico mas não conhecido.

2. Quando é realizada uma medição, o sistema é forçado a adotar um estado, de acordo com uma probabilidade dada pela função de onda do sistema. É a isto que se chama de *colapso da função de onda*.

Outro tipo de interpretação da mecânica quântica conhecida como *interpretação de múltiplos mundos*, da qual existem múltiplas versões (ver [82]), consiste de variações, reinterpretações ou aprimoramentos da *interpretação de estados relativos* proposta por Hugh Everett em [43]. A interpretação de múltiplos mundos afirma que a função de onda nunca colapsa. Antes da medição todos os possíveis estados coexistem no *mundo* descrito pela função de onda $|\psi\rangle$. A medição divide o mundo em múltiplos ramos ou múltiplos mundos igualmente reais, cada um dos quais corresponde a um componente clássico da superposição $|\psi\rangle$. A interpretação de múltiplos mundos baseia-se no postulado de que todo sistema isolado evolui deterministicamente, de acordo com a famosa equação de Schrödinger (ver por exemplo [68, Cap. 6]), e portanto, como o universo é um sistema isolado, deve necessariamente evoluir de acordo com tal equação. Um resultado único para uma medição violaria tal postulado.

Uma objeção de cunho verificacionista à interpretação de múltiplos mundos é que não é possível se ter acesso a esses múltiplos mundos, e por não ser experimentalmente testável esta não pode se diferenciar das outras interpretações.

Por um longo tempo a interpretação de múltiplos mundos não teve muitos seguidores, mas recentemente o interesse por esta interpretação voltou a crescer consideravelmente. Um partidário particularmente relevante desta interpretação é o físico inglês David Deutsch, um dos fundadores da computação quântica, que pensa que na explicação da operação dos modelos de computação quântica, em particular da característica do paralelismo quântico, é necessário assumir a interpretação de múltiplos mundos. Deutsch defende ainda que os modelos de computação quântica podem servir como teste experimental a tal interpretação (cf. [36, p. 113-114]).

Como é possível que as interpretações da mecânica quântica, anteriormente descritas, permitam vislumbrar relações entre modelos de computação quântica e MTPs? Por um lado, a interpretação de Copenhagen sugere a utilização de uma lógica paraconsistente na fundamentação da mecânica quântica, e conseqüentemente, na fundamentação dos modelos de computação quântica. Faz sentido então tentar relacionar modelos de computação quântica e modelos de computação paraconsistente. Por outro lado, a interpretação de múltiplos mundos da mecânica quântica, assumida por David Deutsch na interpretação da operação dos computadores quânticos, permite avançar na idéia de simular o paralelismo quântico por meio de MTPs, como será apresentado no Capítulo 4.

A interpretação dos diferentes formalismos da mecânica quântica levanta sérios problemas filosóficos a respeito da natureza do mundo físico e do esta-

tuto das teorias físicas. Por exemplo, a questão de se existe ou não uma realidade física independente do processo de observação levou a um intenso debate principalmente entre Bhor e Einstein. Para Bhor a realidade física está determinada pela maneira como ela é observada, o que constitui uma característica essencial na chamada interpretação de Copenhagen anteriormente descrita. Contrariamente, para Einstein, o mundo físico teria uma realidade objetiva independente do processo de observação. Outros problemas com indiscutível relevância filosófica levantadas pela mecânica quântica têm a ver com o caráter probabilístico desta teoria, e com a questão de se tal teoria formaliza ou não de maneira completa o mundo microfísico. Estes são só alguns dos sérios problemas filosóficos levantados pela teoria quântica. No presente trabalho não se pretende aprofundar em tais questões filosóficas; para uma introdução nesta direção sugerem-se as seguintes referências: [55], [15] e [23]. Contudo, as diferentes interpretações da mecânica quântica desempenham um papel importante na interpretação dos modelos de computação quântica, e os modelos de computação quântica abrem a possibilidade de se testar experimentalmente e validar essas teses filosóficas. Isto se faz evidente com as idéias de Deutsch de que na explicação da operação dos modelos de computação quântica, em particular da característica do paralelismo quântico, é necessário assumir a interpretação de múltiplos mundos, e de que os modelos de computação quântica podem servir como teste experimental a tal interpretação. Deste modo, essa discussão filosófica leva a uma nova noção de computabilidade, e mas ainda, no sentido contrario, esta nova noção de computabilidade participa da discussão filosófica, oferecendo a possibilidade de validar experimentalmente aquilo que a teoria prega. È neste sentido que se espera que o presente trabalho (na medida em que oferece novas possibilidades de interpretação dos modelos de computação quântica) contribua para a discussão filosófica a respeito da interpretação dos modelos de computação quântica, e possivelmente ainda na interpretação da mecânica quântica.

Capítulo 1

O substrato lógico da noção de máquinas de Turing

O presente capítulo apresenta um procedimento de axiomatização de máquinas de Turing, por meio do qual, dada uma máquina de Turing $\mathcal{M}$ e um dado de entrada n (o que será denotado por $\mathcal{M}(n)$), obtém-se teorias $\Delta'(\mathcal{M}(n))$, na lógica de predicados de primeira ordem, que expressam formalmente a computação de $\mathcal{M}(n)$, sempre que $\mathcal{M}$ seja uma MTD. Mostra-se que o procedimento de axiomatização definido, quando utilizado para axiomatizar MTNDs, em alguns casos produz teorias $\Delta'(\mathcal{M}(n))$ contraditórias e portanto triviais, tendo em conta que a lógica subjacente é a lógica de primeira ordem clássica. Por último, modifica-se o método de axiomatização para se obter teorias $\Delta''(\mathcal{M}(n))$ que representam formalmente a computação de $\mathcal{M}(n)$ para MTNDs. O critério para se determinar quando uma teoria Δ representa formalmente a computação de $\mathcal{M}(n)$ é estabelecido a partir das definições de representação de funções e relações apresentadas em [79] (ver também [42] ou [21]).

Antes de começar a definir o procedimento de axiomatização, apresenta-se a definição de máquina de Turing e outras definições que serão necessárias.

1.1 Máquinas de Turing

O modelo teórico de computação, hoje chamado de *máquina de Turing*, foi definido por Alan Turing em 1936, na tentativa de dar uma definição formal da noção intuitiva de "número computável" e oferecer uma resposta negativa ao chamado *Enstcheidugsproblem* ou *problema da decisão*, mostrando que não existe um procedimento geral para determinar quando uma

fórmula A é demonstrável num cálculo funcional K (ver [81]). No mesmo artigo, Turing demonstrou a equivalência de sua definição de "computável" com a definição de "efetivamente calculável" de Alonso Church (por meio do lambda cálculo), e notou que Church já tinha chegado a conclusões similares com respeito ao problema da decisão (cf. [26] e [25]). O modelo de Turing tem como vantagem o fato de possibilitar um maior apelo à intuição do que seja ser calculável ou computável. Mais tarde foi mostrado que a definição de Turing é equivante a diversas tentativas de formalizar a noção intuitiva de função computável (lambda cálculo (ver [26]), funções parcialmente recursivas (ver [57]), máquina de Post (ver [70]), entre outras); dessa forma, todas as formalizações conhecidas da noção de computabilidade definem exatamente a mesma classe de funções. Assim, se uma função é computável por uma máquina de Turing, pode ser também computada em qualquer outro sistema; este fato notável passou a ser conhecido como "Tese de Church" (ver [42, cap. 10] ou [21, cap. 9]).

Há diversas críticas contemporâneas à Tese de Church que discutem a questão de se a noção de computabilidade deveria coincidir com uma classe maior ou menor do que a de funções computáveis por máquina de Turing. Em particular, a proposta dos modelos conhecidos como de *hipercomputação* tem a pretensão de computar funções não computáveis por máqinas de Turing (ver [29]). É conveniente esclarecer desde já que a abordagem proposta neste trabalho é completamente independente de tais discussões, já que o modelo de MTP aqui investigado computa exatamente as mesmas funções que uma máquina de Turing. Contudo, isso não significa que as MTPs e as máquinas de Turing sejam equivalentes quanto à complexidade algorítmica ou quanto à capacida de processamento de informação. De fato, o que se pretende é que as MTPs possam ser em princípio muito mais eficientes, tirando partido das propriedades que compartem com os modelos de computação quântica.

Informalmente, pode-se descrever uma máquina de Turing como uma máquina constituída por uma cabeça de leitura e escrita e uma fita unidimensional infinita em ambos os sentidos, dividida em casas, cada uma das quais com capacidade de conter um único símbolo. Em todo instante de tempo a máquina encontra-se num *estado* (ou *configuração*), e com a cabeça sobre uma casa da fita, a qual é chamada de *posição atual* da máquina (ver Figura 1.1). O comportamento da máquina está determinado por um conjunto de instruções, as quais indicam a operação a se realizar dependendo do estado e do símbolo na posição atual da máquina. As únicas operações que a máquina pode realizar são: escrever um novo símbolo na posição atual da máquina (substituindo o símbolo anterior) ou realizar um movimento da

cabeça à esquerda ou direita, passando a uma casa subseqüente da atual. No máximo uma instrução é executada em cada instante discreto de tempo. A máquina pára quando não tem nenhuma instrução a ser executada para o estado e símbolo atual da máquina. Os estados, os símbolos de leitura e escrita e as instruções são conjuntos finitos (ver [81, 42, 21, 34] ou [65]).

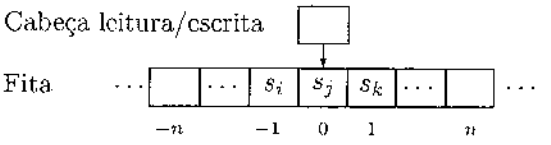

Figura 1.1: Máquina de Turing

Uma máquina de Turing pode ser definida formalmente da seguinte maneira (ver [65]):

Definição 1.1 (Máquina de Turing). *Uma máquina de Turing $\mathcal{M}$ é uma estrutura matemática $\langle Q, \Sigma, M, I\rangle$ onde:*

- $Q = \{q_1, q_2, \ldots, q_n\}$ *(com $Q \neq \emptyset$) é um conjunto finito de estados.*
- $\Sigma = \{s_0, s_1, \ldots, s_{m-1}\}$ *(com $(\Sigma - \{s_0\} \neq \emptyset)$) é um conjunto finito de símbolos de entrada/saída. Por convenção s_0 representa o símbolo vazio.*
- $M = \{R, L\}$ *é o conjunto de movimentos, onde R indica movimento à direita e L movimento à esquerda.*
- $I = \{i_1, i_2, \ldots, i_k\}$ *(com $I \neq \emptyset$) é um conjunto finito de instruções, onde cada i_j é uma quádrupla*[1] *de alguma das formas:*

$$q_i s_j s_k q_l. \qquad \text{(I)}$$

$$q_i s_j R q_l. \qquad \text{(II)}$$

$$q_i s_j L q_l. \qquad \text{(III)}$$

O tipo de instrução (I) *indica que quando a máquina estiver no estado q_i, lendo o símbolo s_j, vai escrever o símbolo s_k e vai passar ao estado*

[1] Na definição original de Turing as instruções definem-se por meio de quíntuplas (cf. [81]). Contudo, pode-se demonstrar que ambas a definições são equivalentes. Utiliza-se aqui a definição por meio de quádruplas para facilitar o processo de axiomatização que será definido na seguinte seção.

q_l. O tipo de instrução (II) *indica que quando a máquina estiver no estado q_i, lendo o símbolo s_j, vai se mover uma casa à direita e vai passar ao estado q_l. O tipo de instrução* (III) *é similar ao* (II), *com a diferença que o movimento se dá à esquerda.*

Por convenção a máquina começa a execução no estado q_1, lendo a posição 0 da fita.

Exemplo 1.1. *Seja $\mathcal{M}_1 = \langle Q, \Sigma, M, I\rangle$ tal que $Q = \{q_1, q_2\}$, $\Sigma = \{s_0, s_1\}$ e $I = \{i_1, i_2\}$, onde $i_1 = q_1 s_1 R q_1$ e $i_2 = q_1 s_0 s_1 q_2$. $\mathcal{M}_1$ é uma máquina de Turing que percorre uma cadeia de símbolos s_1, acrescenta um s_1 à cadeia e pára.*

A definição de uma máquina de Turing $\mathcal{M}$ pode ser representada esquematicamente por meio de um grafo:

Definição 1.2 (Grafo associado a uma máquina de Turing). *Seja $\mathcal{M} = \langle Q, \Sigma, M, I\rangle$ uma máquina de Turing, o grafo para representar graficamente $\mathcal{M}$ tem como nós os estados $q_i \in Q$, e existe uma aresta do nó q_i ao nó q_l, etiquetado com "s_j : ação", se $\mathcal{M}$ tiver uma instrução $i_n \in I$ tal que $i_n = q_i s_j$ ação q_l, onde "ação" pode tomar os valores s_k, R ou L, dependendo de se i_n é de tipo* (I), (II) *ou* (III), *respectivamente.*

Exemplo 1.2. *O grafo que representa a máquina de Turing $\mathcal{M}_1$ do Exemplo 1.1 é:*

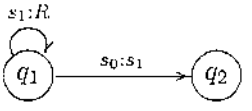

Figura 1.2: Grafo da máquina $\mathcal{M}_1$

De acordo com o conjunto de instruções, as máquinas de Turing são classificadas em *determinísticas* (MTDs) e *não determinísticas* (MTNDs). Uma máquina de Turing é chamada determinística se para cada combinação de estado e símbolo de entrada existe no máximo uma instrução a ser executada. Por outro lado, é chamada não determinística se para ao menos uma combinação de estado e símbolo de entrada existe mais de uma instrução a ser executada, em cujo caso a máquina escolhe e executa somente uma delas. A escolha da instrução a ser executada pode ser realizada aleatoriamente ou

através de uma função de distribuição de probabilidades. No presente trabalho não é relevante o método de escolha da instrução a ser executada (caso existam varias possíveis), por isso, daqui para frente se referirá a MTNDs sem indicar qualquer método de escolha. Formalmente:

Definição 1.3 (Máquina de Turing determinística (MTD)). *Seja $\mathcal{M} = \langle Q, \Sigma, M, I \rangle$ uma máquina de Turing, $\mathcal{M}$ é* determinística *se I é uma função de um subconjunto de $Q \times \Sigma$ em $(\Sigma \cup M) \times Q$.*

Definição 1.4 (Máquina de Turing não determinística (MTND)). *Seja $\mathcal{M} = \langle Q, \Sigma, M, I \rangle$ uma máquina de Turing, $\mathcal{M}$ é* não determinística *se $I \subseteq Q \times \Sigma \times \{\Sigma \cup M\} \times Q$ não cumpre a propriedade de unicidade para ser função de um subconjunto de $Q \times \Sigma$ em $(\Sigma \cup M) \times Q$, isto é, se I contém ao menos duas instruções i_j e i_k ($i_j \neq i_k$) tais que os primeiros dois símbolos delas são iguais. As instruções i_j e i_k são chamadas* instruções em conflito. *Quando $\mathcal{M}$ chega a uma situação na qual há instruções em conflito a serem executadas, escolhe e executa somente uma delas.*

Outro conceito que precisa ser formalizado, para os propósitos do presente trabalho, é o conceito de computação numa máquina de Turing. Para isso, é necessário se definir uma maneira de descrever a situação da máquina (símbolos na fita, estado e posição atual) num instante de tempo t, e logo se definir como esta situação vai evoluindo de um instante de tempo para outro, de acordo com as instruções da máquina. Seguindo [34], define-se como descrever a situação da máquina por meio do que será chamado de *descrição instantânea*, definem-se as regras de mudança de uma descrição instantânea para outra, indica-se quando uma descrição instantânea é *terminal*, e a partir destes elementos define-se como descrever formalmete uma computação numa máquina de Turing.

Definição 1.5 (Descrição instantânea). *Seja $\mathcal{M} = \langle Q, \Sigma, M, I \rangle$ uma máquina de Turing, uma* descrição instantânea *α de $\mathcal{M}$ é uma sucessão finita de símbolos pertencentes a $Q \cup \Sigma$, que contém pelo menos um símbolo $s_j \in \Sigma$ e exatamente um símbolo $q_i \in Q$, e cujo q_i não pode ser o último símbolo da esquerda para direita.*

A título de ilustração, supondo que $\mathcal{M}$, no instante de tempo t, encontra-se na situação representada na Figura 1.3, a descrição instantânea para descrever tal situação é $\alpha_t = s_{i_{r-m}} \cdots s_{i_{r-1}} q_i s_{i_r} s_{i_{r+1}} \cdots s_{i_{r+n}}$.

Definição 1.6 (Mudança de descrição instantânea). *Seja $\mathcal{M}$ uma máquina de Turing e sejam α e β descrições instantâneas de $\mathcal{M}$. Diz-se*

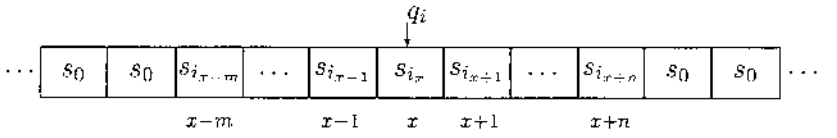

Figura 1.3: Situação máquina $\mathcal{M}$ no tempo t

que há uma mudança de α para β em $\mathcal{M}$, denotada por $\alpha \xrightarrow{\mathcal{M}} \beta$, quando existem sucessões finitas (possivelmente vazias) de símbolos $s_i \in \Sigma$, denotadas por P e Q, tais que alguma das seguintes condições é satisfeita:

- $\alpha = Pq_is_jQ$, $\beta = Pq_ls_kQ$ *e* $\mathcal{M}$ *contem uma instrução do tipo* (I);
- $\alpha = Pq_is_js_kQ$, $\beta = Ps_jq_ls_kQ$ *e* $\mathcal{M}$ *contem uma instrução do tipo* (II);
- $\alpha = Pq_is_j$, $\beta = Ps_jq_ls_0$ *e* $\mathcal{M}$ *contem uma instrução do tipo* (II);
- $\alpha = Ps_kq_is_jQ$, $\beta = Pq_ls_ks_jQ$ *e* $\mathcal{M}$ *contem uma instrução do tipo* (III);
- $\alpha = q_is_jQ$, $\beta = q_ls_0s_jQ$ *e* $\mathcal{M}$ *contem uma instrução do tipo* (III).

Definição 1.7 (Descrição instantânea terminal). *Seja $\mathcal{M}$ uma máquina de Turing; uma descrição instantânea α é* terminal, *com respeito a $\mathcal{M}$, se não existe uma descrição instantânea β tal que $\alpha \xrightarrow{\mathcal{M}} \beta$.*

Definição 1.8 (Computação de uma máquina de Turing). *Uma* computação de uma máquina de Turing $\mathcal{M}$ *é uma seqüência finita de descrições instantâneas $\alpha_0\alpha_1 \ldots \alpha_n$, tal que $\alpha_i \xrightarrow{\mathcal{M}} \alpha_{i+1}$, para $0 \leq i \leq n-1$, e α_n é uma descrição instantânea terminal com respeito a $\mathcal{M}$. Neste caso escreve-se $\alpha_n = Res_{\mathcal{M}}(\alpha_0)$ e diz-se que α_n é o resultado de $\mathcal{M}$ com respeito a α_0. O resultado da computação é obtido desprezando-se o símbolo de estado de α_n.*

Exemplo 1.3. *Seja $\mathcal{M}_1$ a máquina de Turing do Exemplo 1.1. Para a seqüência de entrada $n_1 = s_1s_1$ a seqüência de descrições instantâneas $\alpha_0\alpha_1\alpha_2\alpha_3$, onde $\alpha_0 = q_1s_1s_1$, $\alpha_1 = s_1q_1s_1$, $\alpha_2 = s_1s_1q_1s_0$ e $\alpha_3 = s_1s_1q_2s_1$, descreve formalmente a computação de $\mathcal{M}_1(n_1)$ (ver Figura 1.4). Neste exemplo, α_3 é o resultado de $\mathcal{M}_1$ com respeito a α_0. O resultado da computação é $s_1s_1s_1$.*

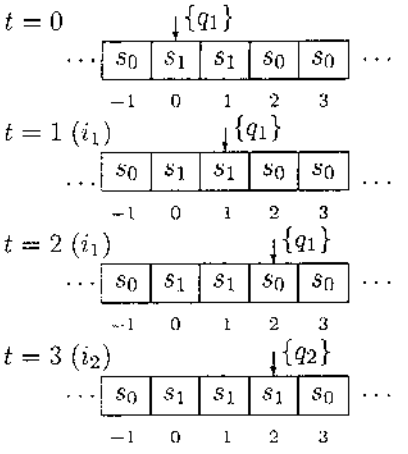

Figura 1.4: Computação de $\mathcal{M}_1(n_1)$

1.2 Axiomatização de MTDs

Com o objetivo de demonstrar a indecidibilidade da lógica de predicados de primeira ordem, George Boolos e Richard Jeffrey em [13] definem um procedimento para construir uma teoria axiomática Δ e uma fórmula H, a partir das instruções de uma máquina de Turing $\mathcal{M}$ e um dado de entrada n. O procedimento é definido de tal modo que, sob uma interpretação $\mathcal{I}$, a teoria Δ descreve a operação de $\mathcal{M}(n)$ e H indica que a máquina eventualmente pára. Assim, $\Delta \vdash H$ se e somente se $\mathcal{M}(n)$ pára. Portanto, supondo-se que exista um procedimento efetivo para determinar se $\Delta \vdash H$, ter-se-ia como consequência que o problema da parada de uma máquina de Turing é decidível. Como é conhecido que o problema da parada de uma máquina de Turing é indecidível (cf. em [81], [42] ou [21]), conclui-se que determinar se $\Delta \vdash H$ é também indecidível. Portanto, a lógica de predicados de primeira ordem é indecidível.[2] Embora os propósitos do presente trabalho não digam respeito diretamente às relações entre indecidibilidade em uma lógica e resultados sobre parada em máquinas, nossas definições são suficientemente naturais para que resultados desse tipo possam ser estendidos para outras lógicas. Esse ponto será retomado no capítulo de Considerações Finais.

[2]Como Δ é uma teoria finita, $\Delta \vdash H$ se e somente se $\vdash (\bigwedge_{i=1}^{n} \delta_i) \rightarrow H$, onde $\bigwedge_{i=1}^{n} \delta_i$ é a conjunção de todas as fórmulas $\delta_i \in \Delta$. Portanto, a partir da suposição que a lógica de predicados de primeira ordem é decidível, conclui-se que $\vdash (\bigwedge_{i=1}^{n} \delta_i) \rightarrow H$ é decidível e consequentemente $\Delta \vdash H$ também seria decidível.

Para o objetivo perseguido por Boolos e Jeffrey, o procedimento de axiomatização que definem (o qual será chamado de *método de axiomatização B-J*) resulta ser adequado. Contudo, se se deseja ter um procedimento de axiomatização para MTDs, de tal modo que a teoria resultante represente formalmente a computação da máquina de Turing em consideração para um dado de entrada n, esbarra-se no problema que o método de axiomatização B-J produz teorias que não atingem este objetivo. A seguir descreve-se o método de axiomatização B-J, mostrando como tal procedimento produz teorias que não representam formalmente as computações das MTDs. Isto justifica alguns ajustes ao procedimento de forma a atingir tal objetivo.

1.2.1 Método de axiomatização B-J

Para definir o método de axiomatização de máquinas de Turing, Boolos e Jeffrey (ver [13]) imaginam as casas da fita como numeradas por meio dos números inteiros, e o tempo dividido momentos discretos t, em cada um dos quais a máquina executa exatamente uma instrução. Existe um momento 0 no qual a máquina começa a funcionar lendo a casa 0 da fita. Supõe-se que os momentos de tempo estendem-se infinitamente ao futuro e ao passado. De maneira similar, a fita estende-se infinitamente à direita e à esquerda. Além disso, eles assumem que a máquina é *ligada* no momento 0 e *desligada* no primeiro momento (se existir algum) depois de a máquina parar; e assumem que enquanto a máquina estiver desligada não está em nenhum estado, não está lendo nenhuma casa da fita e não tem nenhum símbolo (nem sequer o vazio) em nenhuma casa da fita[3] (Figura 1.5).

Estabelecidos os pressupostos, Boolos e Jeffrey definem a linguagem de primeira ordem $\mathcal{L} = \{Q_1, Q_2, \ldots, Q_n, S_0, S_1, \ldots, S_{m-1}, <, ', 0\}$, a qual utilizam para construir a teoria[4] $\Delta_{LPC}(\mathcal{M}(n))$, que axiomatiza a computação de $\mathcal{M}(n)$. Em $\mathcal{L}$, os símbolos Q_i, S_j e $<$ são símbolos de predicado de aridade dois, $'$ é um símbolo de função de aridade um e o simbolo 0 é um símbolo de constante.

Sob a interpretação intencional[5] $\mathcal{I}$, das sentenças presentes em

[3]Este último pressuposto não é levado em conta no método de axiomatização B-J, o que é uma das causas para que as teorias obtidas não representem formalmente as computações das respectivas $\mathcal{M}(n)$, como vai se evidenciar na demonstração do Teorema 1.4.

[4]Acrescenta-se o sub-índice LPC a Δ para ressaltar o fato de a lógica subjacente ser a lógica de primeira ordem clássica (LPC), e se acrescenta o parâmetro $\mathcal{M}(n)$ para indicar a máquina de Turing e o dado de entrada em consideração.

[5]Ao considerar os predicados desde o ponto de vista *extensional* diz-se que aludem a, ou melhor, que denotam *classes* ou *conjuntos*. Mas, quando são considerados desde o ângulo *intencional* diz-se que designam *propriedades* dos objetos (cf. [47, p. 163]).

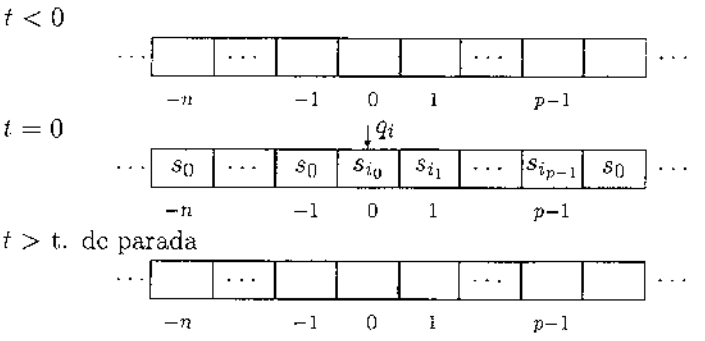

Figura 1.5: Pressupostos da axiomatização B-J ($s_0, s_{i_k} \in \Sigma$)

$\Delta_{LPC}(\mathcal{M}(n))$, as variáveis interpretam-se como números inteiros e os símbolos de $\mathcal{L}$ interpretam-se da seguinte maneira:

- $Q_i(t, x)$ interpreta-se como o fato da máquina de Turing $\mathcal{M}$ se encontrar no estado q_i, no tempo t, na posição x;
- $S_j(t, x)$ interpreta-se como o fato da máquina de Turing $\mathcal{M}$ se encontrar lendo o símbolo s_j, no tempo t, na posição x;
- $<$ interpreta-se como a relação "menor que" nos números inteiros;
- $'$ interpreta-se como a função "sucessor";
- 0 interpreta-se como o número 0.

A teoria $\Delta_{LPC}(\mathcal{M}(n))$ é construída incluindo axiomas para estabelecer as propriedades dos símbolos $<$ e $'$, um axioma por cada uma das instruções de $\mathcal{M}$, e um axioma para estabelecer a situação inicial de $\mathcal{M}$ (isto é, estado inicial, posição sobre a fita e dado de entrada n).

Definição 1.9 (Teoria $\Delta_{LPC}(\mathcal{M}(n))$). *Seja $\mathcal{M} = (Q, \Sigma, M, I)$ uma máquina de Turing e $n = s_{i_0}s_{i_1}\ldots s_{i_{p-1}}$ com $s_{i_k} \in \Sigma$ a seqüência ou dado de entrada para $\mathcal{M}$. A teoria $\Delta_{LPC}(\mathcal{M}(n))$ é o conjunto que contem os axiomas:*

- *Axiomas para estabelecer as propriedades (nos números inteiros) dos*

símbolos $<$ *e* $'$*:*

$$\forall z \exists x (z = x'), \tag{A1}$$

$$\forall z \forall x \forall y (((z = x') \wedge (z = y')) \rightarrow (x = y)), \tag{A2}$$

$$\forall x \forall y \forall z (((x < y) \wedge (y < z)) \rightarrow (x < z)), \tag{A3}$$

$$\forall x (x < x'), \tag{A4}$$

$$\forall x \forall y ((x < y) \rightarrow (x \neq y)). \tag{A5}$$

O axioma (A1) *estabelece a existência de sucessor para todo número. O axioma* (A2) *estabelece a injetividade da função sucessor. O axioma* (A3) *estabelece a propriedade transitiva da relação* $<$*. O axioma* (A4) *estabelece a relação entre a função* $'$ *e a relação* $<$*, especificando que o sucessor de um número sempre é maior do que ele. O axioma* (A5) *estabelece a propriedade anti-reflexiva da relação* $<$*.*

- *Para cada uma das instruções* $i_j \in I$*, sendo* m *o cardinal de* Σ*, constrói-se um axioma de acordo com algum dos seguintes esquemas:*

  - *Se* i_j *é da forma* (I) *(isto é,* $i_j = q_i s_j s_k q_l$*), então o esquema axiomático é:*

$$\forall t \forall x \Bigg(\Big(Q_i(t,x) \wedge S_j(t,x) \Big) \rightarrow \Big(Q_l(t',x) \wedge S_k(t',x) \wedge \forall y \Big((y \neq x) \rightarrow \Big(\bigwedge_{i=0}^{m-1} \big(S_i(t,y) \rightarrow S_i(t',y) \big) \Big) \Big) \Big) \Bigg), \quad (\mathrm{A}i_j\ (\mathrm{I}))$$

  tal esquema indica que se a máquina de Turing $\mathcal{M}$*, no instante de tempo* t*, estiver na posição* x*, no estado* q_i *e lendo o símbolo* s_j*, no seguinte instante de tempo* t'*, a máquina estará na mesma posição* x*, no estado* q_l *e lendo o símbolo* s_k*. Além disso, em todas as posições* $y \neq x$ *o símbolo no instante de tempo* t' *será o mesmo do instante anterior* t*.*

  - *Se* i_j *é da forma* (II) *(isto é,* $i_j = q_i s_j R q_l$*), então o esquema axiomático é:*

$$\forall t \forall x \Bigg(\Big(Q_i(t,x) \wedge S_j(t,x) \Big) \rightarrow \Big(Q_l(t',x') \wedge \forall y \Big(\bigwedge_{i=0}^{m-1} \big(S_i(t,y) \rightarrow S_i(t',y) \big) \Big) \Big) \Bigg), \quad (\mathrm{A}i_j\ (\mathrm{II}))$$

tal esquema indica que se a máquina de Turing $\mathcal{M}$, no instante de tempo t, estiver na posição x, no estado q_i e lendo o símbolo s_j, no seguinte instante de tempo t', a máquina estará na seguinte posição a direita x', no estado q_l. Além disso, em todas as casas da fita os símbolos no instante de tempo t' serão os mesmo do instante anterior t.

- *Se i_j é da forma* (III) *(isto é, $i_j = q_i s_j L q_l$), então o esquema axiomático é:*

$$\forall t \forall x \left(\left(Q_i(t,x') \wedge S_j(t,x') \right) \rightarrow \left(Q_l(t',x) \wedge \forall y \left(\bigwedge_{i=0}^{m-1} \left(S_i(t,y) \rightarrow S_i(t',y) \right) \right) \right) \right), \quad (\mathrm{A}i_j \text{ (III)})$$

a interpretação deste esquema é similar ao caso anterior.

- *O axioma para especificar a situação inicial da máquina, supondo que a máquina começa no estado q_1, sobre a posição 0 da fita, com a entrada $n = s_{i_0} s_{i_1} \ldots s_{i_{p-1}}$, constrói-se de acordo com o esquema:*

$$Q_1(0,0) \wedge \left(\bigwedge_{j=1}^{p} S_{i_{j-1}}(0, 0^{j-1}) \right) \wedge \forall y \left(\left(\bigwedge_{j=1}^{p} y \neq 0^{j-1} \right) \rightarrow S_0(0,y) \right), \quad (\mathrm{A}n)$$

onde 0^{j-1} indica a aplicação de $j-1$ vezes a função sucessor à constante 0.

A sentença H, que indica que $\mathcal{M}$ pára, define-se tomando a disjunção de todas as sentenças que indicam que existe um tempo t e uma posição x, nos quais a máquina encontra-se no estado q_i lendo o símbolo s_j e não existe nenhuma instrução $i_k \in I$ que comece pelos símbolos $q_i s_i$. O esquema para a construção de H é:

$$\bigvee_{q_i s_j \text{ não inic.}} (\exists t \exists x (Q_i(t,x) \wedge S_j(t,x))), \quad (\mathrm{H})$$

o sub-índice "$q_i s_j$ não inic." indica que na disjuntória são consideradas somente as combinações $q_i s_j$ que não são os dois primeiros símbolos de nenhuma instrução. Embora H seja de vital importância na demonstração de Boolos e Jeffrey, não é relavante para os objetivos do presente trabalho.

A seguir apresenta-se um exemplo de uma teoria de máquina de Turing construída usando o método de Boolos e Jeffrey:

Exemplo 1.4. *Seja $\mathcal{M}_1$ a máquina de Turing do Exemplo 1.1 e $n_1 = s_1 s_1$ a seqüência de entrada para $\mathcal{M}_1$. A teoria $\Delta_{LPC}(\mathcal{M}_1(n_1)) = \{(\mathrm{A1}), \ldots, (\mathrm{A5}), (\mathrm{A}i_1), (\mathrm{A}i_2), (\mathrm{An}_1)\}$, onde (A1) a (A5) são os axiomas da Definição 1.9 e:*

$$\forall t \forall x \Big(\Big(Q_1(t,x) \wedge S_1(t,x) \Big) \rightarrow \Big(Q_1(t',x') \wedge \forall y \Big(\big(S_0(t,y) \rightarrow S_0(t',y) \big) \wedge \big(S_1(t,y) \rightarrow S_1(t',y) \big) \Big) \Big) \Big), \quad (\mathrm{A}i_1)$$

$$\forall t \forall x \Big(\Big(Q_1(t,x) \wedge S_0(t,x) \Big) \rightarrow \Big(Q_2(t',x) \wedge S_1(t',x) \wedge \forall y \Big((y \neq x) \rightarrow \big((S_0(t,y) \rightarrow S_0(t',y)) \wedge (S_1(t,y) \rightarrow S_1(t',y)) \big) \Big) \Big) \Big), \quad (\mathrm{A}i_2)$$

$$Q_1(0,0) \wedge \big(S_1(0,0) \wedge S_1(0,0') \big) \wedge \forall y \left((y \neq 0 \wedge y \neq 0') \rightarrow S_0(0,y) \right). \quad (\mathrm{An}_1)$$

1.2.2 Representabilidade das MTDs nas teorias $\Delta_{LPC}(\mathcal{M}(n))$

Definindo-se uma estrutura matemática $\mu(\mathcal{M}(n))$ para expressar a computação de $\mathcal{M}(n)$, e com base na noção clássica de representação de funções e relações numa teoria introduzida por Alfred Tarski, Andrzej Mostowski e Raphael M. Robinson com finalidade de investigar a aritmética (ver [79]), apresentadas nas Definições 1.11 e 1.12, propõe-se na Definição 1.13 a importante noção de representação de uma computação numa teoria; este novo conceito permite determinar se uma certa teoria está ou não axiomatizando uma computação numa máquina de Turing de maneira adequada. Vale ressaltar que esta noção será estendida para o caso das máquinas de Turing paraconsistentes na Definição 2.10.

Com a definição de representação de uma computação numa teoria demonstra-se que as teorias $\Delta_{LPC}(\mathcal{M}(n))$, obtidas através do método de axiomatização B-J, não conseguem representar a computação da $\mathcal{M}(n)$ respectiva, sendo necessário então ajustar o método de axiomatização B-J incluindo novos axiomas nas teorias $\Delta_{LPC}(\mathcal{M}(n))$, para se obter teorias

$\Delta'_{LPC}(\mathcal{M}(n))$ que conseguem representar a computação da $\mathcal{M}(n)$ respectiva.

Definição 1.10 (Estrutura que expressa uma computação numa MTD). *Seja $\alpha_0\alpha_1\alpha_2\ldots$ a seqüência (possivelmente infinita) de descrições instantâneas de $\mathcal{M}(n)$, tal que α_0 descreve a situação inicial de $\mathcal{M}(n)$ e $\alpha_i \xrightarrow{\mathcal{M}} \alpha_{i+1}$ $(i = 0, 1.2\ldots)$. A computação de $\mathcal{M}(n)$ é expressada matemáticamente pela estrutura*[6] $\mu(\mathcal{M}(n)) = \langle A, Q_1^\mu, Q_2^\mu, \ldots, Q_n^\mu, S_0^\mu, S_1^\mu, \ldots, S_{m-1}^\mu, <^\mu, '^\mu, 0^\mu\rangle$, *onde:*

- $A = \mathbb{Z}$,
- $Q_i^\mu = \{(t, x) \mid q_i$ *esta em* α_t *antes do símbolo de entrada/saída correspondente à posição* x *da fita* $(s_{i_x})\}$,
- $S_j^\mu = \{(t, x) \mid s_j$ *esta em* α_t *e é o símbolo de entrada/saída correspondente à posição* x *da fita* $(s_{i_x})\} \cup \{(t, y) \mid j = 0$, *existe descrição instantânea para o tempo* t (α_t) *e* y *é uma posição não descrita em* $\alpha_t\}$.
- $<^\mu =$ *relação menor que em* $\mathbb{Z}$,
- $'^\mu =$ *função sucessor em* $\mathbb{Z}$,
- $0^\mu = 0$.

Sob a função de interpretação δ para $\Delta'_{LPC}(\mathcal{M}(n))$, tal que $\delta(Q_i(t, x)) = Q_i^\mu$, $\delta(S_j(t, x)) = S_j^\mu$, $\delta(<) =<^\mu$, $\delta(') ='^\mu$ e $\delta(0) = 0^\mu$ demonstra-se que:

Teorema 1.1. *Se $\mathcal{M}$ é uma MTD então $\mu(\mathcal{M}(n))$ é modelo de $\Delta'_{LPC}(\mathcal{M}(n))$.*

Demonstração.

1. A relação menor que $(<)$ e a função sucessor $(')$ nos números inteiros cumprem as propriedades estabelecidas pelos axiomas (A1) - (A5) da Definição 1.9. portanto, $\mu(\mathcal{M}(n)) \models$ (A1) - (A5).
2. Se $\mathcal{M}$ contem uma instrução da forma (I) então em $\Delta'_{LPC}(\mathcal{M}(n))$ existe um axioma construído sob o esquema (Ai_j (I)), neste caso:

[6] Definição adaptada de [83, p. 13]

2.1. Para toda descrição instantânea $\alpha_t = Pq_i s_j Q$, com P e Q seqüências finitas (possivelmente vazias) de símbolos $s_k \in \Sigma$, por definição de mudança de descrição instantânea e por ser $\mathcal{M}$ uma MTD, a descrição instantânea seguinte será obrigatoriamente $\alpha_{t+1} = Pq_l s_k Q$.

2.2. Por definição de Q_i^μ e S_j^μ em 2.1, tem-se que $(t, x) \in Q_i^\mu$ e $(t, x) \in S_j^\mu$ em $\mu(\mathcal{M}(n))$.

2.3. Por definição de Q_i^μ e S_j^μ em 2.1, tem-se que $(t + 1, x) \in Q_l^\mu$ e $(t + 1, x) \in S_k^\mu$ em $\mu(\mathcal{M}(n))$.

2.4. P e Q permanecem iguais, portanto, por definição de S_j^μ em 2.1, para todo $y \neq x$, se $(t, y) \in S_i^\mu$ então $(t+1, y) \in S_i^\mu$ em $\mu(\mathcal{M}(n))$.

2.5. Por definição de δ em 2.2, 2.3 e 2.4, tem-se que $\mu(\mathcal{M}(n)) \vDash (\mathrm{A}i_\mathrm{j}\ (\mathrm{I}))$.

3. Se $\mathcal{M}$ contem uma instrução da forma (II) então em $\Delta'_{LPC}(\mathcal{M}(n))$ existe um axioma construído sob o esquema $(\mathrm{A}i_\mathrm{j}\ (\mathrm{II}))$, neste caso $\mu(\mathcal{M}(n)) \vDash (\mathrm{A}i_\mathrm{j}\ (\mathrm{II}))$. A demonstração é similar à realizada no passo 2.

4. Se $\mathcal{M}$ contem uma instrução da forma (III) então em $\Delta'_{LPC}(\mathcal{M}(n))$ existe um axioma construído sob o esquema $(\mathrm{A}i_\mathrm{j}\ (\mathrm{III}))$, neste caso $\mu(\mathcal{M}(n)) \vDash (\mathrm{A}i_\mathrm{j}\ (\mathrm{III}))$. A demonstração é similar à realizada no passo 2.

5. Para o tempo $t = 0$ a descrição instantânea de $\mathcal{M}(n)$ é $\alpha_0 = q_1 s_{i_0} s_{i_1} \ldots s_{i_{p-1}}$. Pela definição de Q_i^μ e S_j^μ, em $\mu(\mathcal{M}(n))$ tem-se que $(0.0) \in Q_1^\mu$, $(0,0) \in S_{i_0}^\mu$, $(0,1) \in S_{i_1}^\mu$, $\ldots$, $(0, p-1) \in S_{i_{p-1}}^\mu$ e $(0, y) \in S_0^\mu$ para todo $y < 0$ ou $y > p - 1$. Portanto, por definição de δ, tem-se que $\mu(\mathcal{M}(n)) \vDash (\mathrm{A}n)$.

□

Definição 1.11 (Representação de uma função numa teoria). *Seja f uma função de k variáveis, Δ uma teoria arbitrária e $\varphi(x_1, \ldots, x_k, x)$ uma fórmula bem formada (fbf) em Δ, com $k+1$ variáveis livres. Dize-se que f* é representada por φ em Δ *se $f(m_1, \ldots, m_k) = n$ implica que:*

1. $\Delta \vdash \varphi(\bar{m_1}, \ldots, \bar{m_k}, \bar{n})$,

2. *Se $n \neq p$ então* $\Delta \vdash \neg\varphi(\bar{m_1}, \ldots, \bar{m_k}, \bar{p})$, *e*

3. $\Delta \vdash \varphi(\bar{m}_1, \ldots, \bar{m}_k, \bar{q}) \rightarrow \bar{q} = \bar{n}$.

Onde os símbolos $\bar{m}$ indicam numerais unários.

Dize-se que f é representável em Δ *se existir alguma fbf que a represente. Será dito também que o símbolo de função $\mathcal{F}$* representa a função f em Δ *se $\mathcal{F}(x_1, \ldots, x_k) = x$ representa a f em Δ.*

Definição 1.12 (Representação de uma relação numa teoria). *Seja R uma relação de aridade k, Δ uma teoria arbitrária e $\varphi(x_1, \ldots, x_k)$ uma fbf em Δ, com k variáveis livres. Diz-se que R* é representada por φ em Δ *se:*

1. $(m_1, \ldots, m_k) \in R$ *implica* $\Delta \vdash \varphi(\bar{m}_1, \ldots, \bar{m}_k)$, *e*

2. $(m_1, \ldots, m_k) \notin R$ *implica* $\Delta \vdash \neg\varphi(\bar{m}_1, \ldots, \bar{m}_k)$.

Onde os símbolos $\bar{m}$ indicam de novo numerais unarios.

Dize-se que R é representável em Δ *se existir alguma fbf que a represente. Será dito também que o símbolo de predicado $\mathcal{R}$* representa a relação R em Δ *se $\mathcal{R}(x_1, \ldots, x_k)$ representa a R em Δ.*

Definição 1.13 (Representação de uma computação numa teoria). *Seja $\mathcal{M}$ uma máquina de Turing, n o dado e entrada para $\mathcal{M}$ e $\mu(\mathcal{M}(n))$ a estrutura que expressa matematicamente a computação de $\mathcal{M}(n)$. Uma teoria Δ, na linguagem $\mathcal{L} = \{Q_1, Q_2, \ldots, Q_n, S_0, S_1, \ldots, S_{m-1}, <, ', 0\}$, representa a computação de $\mathcal{M}(n)$ se em Δ cada símbolo de predicado Q_i representa a respectiva relação Q_i^μ de $\mu(\mathcal{M}(n))$, cada símbolo de predicado S_j representa a respectiva relação S_j^μ de $\mu(\mathcal{M}(n))$, o símbolo de predicado $<$ representa a relação $<^\mu$ de $\mu(\mathcal{M}(n))$, e o símbolo de função $'$ representa a função $'^\mu$ de $\mu(\mathcal{M}(n))$.*

Teorema 1.2. *Nas teorias $\Delta_{LPC}(\mathcal{M}(n))$, o símbolo de predicado $<$ representa a relação $<^\mu$ da estrutura $\mu(\mathcal{M}(n))$ respectiva.*

Demonstração.

1. Condição 1 da Definição 1.12: suponha que $m < n$, logo $n = m + p$, o que é equivalente a $n = m'^{\mu^p}$ (onde $'^{\mu^p}$ denota a aplicação de p vezes a função $'^\mu$), então, os termos $\bar{n}$ e $\bar{m}'^p$, que representam n e m'^{μ^p} respectivamente, são idênticos. Pelas propriedades da igualdade em LPC, para todo termo t, $\vdash_{LPC} t = t$. Portanto, $\vdash_{LPC} \bar{n} = \bar{m}'^p$. Pelos axiomas (A1) e (A4), da Definição 1.9, tem-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{m} < \bar{m}', \Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{m}' < \bar{m}'', \ldots, \Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{m}'^{p-1} < \bar{m}'^p$.

Aplicando $p-1$ vezes instanciação[7] em (A3), adjunção e modus ponens (MP) obtém-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{m} < \bar{m}'^{p}$. Pelas propriedades da igualdade em LPC tem-se que $\vdash_{LPC} (\bar{n} = \bar{m}'^{p}) \rightarrow ((\bar{m} < \bar{m}'^{p}) \rightarrow (\bar{m} < \bar{n}))$. Aplicando MP duas vezes obtém-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{m} < \bar{n}$.

2. Condição 2 da Definição 1.12: suponha que $m \not< n$, então $n = m$ ou $n < m$. Se $n = m$ então $\bar{n}$ e $\bar{m}$ são idênticos, e pelas propriedades da igualdade $\vdash_{LPC} \bar{m} = \bar{n}$. Por instanciação em (A5) $\Delta_{LPC}(\mathcal{M}(n)) \vdash (\bar{m} < \bar{n}) \rightarrow (\bar{m} \neq \bar{n})$. Portanto, por contraposição e MP obtém-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \neg(\bar{m} < \bar{n})$. Se $n < m$, pelo passo 1. tem-se que (*) $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{n} < \bar{m}$. Instanciando em (A3) tem-se que (**) $\Delta_{LPC}(\mathcal{M}(n)) \vdash ((\bar{n} < \bar{m}) \wedge (\bar{m} < \bar{n})) \rightarrow (\bar{n} < \bar{n})$. Instanciando em (A5) tem-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{n} < \bar{n} \rightarrow \bar{n} \neq \bar{n}$, mas $\vdash_{LPC} \bar{n} = \bar{n}$, logo, por contraposição e MP obtém-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \neg(\bar{n} < \bar{n})$. Por contraposição em (**) e MP obtém-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \neg((\bar{n} < \bar{m}) \wedge (\bar{m} < \bar{n}))$. Pelas Leis de Demorgan ($\neg(A \wedge B) \equiv \neg A \vee \neg B$ e $\neg(A \vee B) \equiv \neg A \wedge \neg B$) e pelo silogismo disjuntivo em (*) obtém-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \neg(\bar{m} < \bar{n})$.

□

Teorema 1.3. *Nas teorias $\Delta_{LPC}(\mathcal{M}(n))$, o símbolo de função $'$ representa a função $'^{\mu}$ da estrutura $\mu(\mathcal{M}(n))$ respectiva.*

Demonstração.

1. Condição 1 da Definição 1.11: suponha que m é o sucessor de n na estrutura $\mu(\mathcal{M}(n))$, isto é $n'^{\mu} = m$; logo os termos $\bar{n}'$ e $\bar{m}$, que representam n'^{μ} e m respectivamente, são idênticos. Pelas propriedades da igualdade em LPC, para todo termo t, $\vdash_{LPC} t = t$. Portanto, $\vdash_{LPC} \bar{n}' = \bar{m}$, e por monotonicidade $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{n}' = \bar{m}$.

2. Condição 2 da Definição 1.11: suponha que m é o sucessor de n na estrutura $\mu(\mathcal{M}(n))$, isto é $n'^{\mu} = m$; pelo passo 1 tem-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{n}' = \bar{m}$. Seja $p \neq m$, logo $p <^{\mu} m$ ou $m <^{\mu} p$. Se $p <^{\mu} m$ então $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{p} < \bar{m}$ (Teorema 1.2). Por instanciação em (A5) tem-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{p} < \bar{m} \rightarrow \bar{p} \neq \bar{m}$. Por MP obtém-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{p} \neq \bar{m}$. Pelas propriedades da igualdade em LPC tem-se que $\vdash_{LPC} \bar{p} \neq \bar{m} \rightarrow \bar{m} \neq \bar{p}$ e que $\vdash_{LPC} ((\bar{n}' = \bar{m}) \wedge (\bar{m} \neq \bar{p})) \rightarrow \bar{n}' \neq \bar{p}$. Portanto, por monotinicidade,

[7] No apêndice A apresentam-se as regras de dedução usadas nas demonstrações.

adjunção e MP obtém-se que $\Delta_{LPC}(\mathcal{M}(n)) \vdash \bar{n}' \neq \bar{p}$. Se $m <^{\mu} p$ a demonstração é similar.

3. Condição 3 da Definição 1.11: esta condição corresponde exatamente ao axioma (A2).

□

Teorema 1.4. *Nas teorias $\Delta_{LPC}(\mathcal{M}(n))$, os símbolos de predicado Q_i não representam as relações Q_i^{μ} da estrutura $\mu(\mathcal{M}(n))$ respectiva.*

Demonstração. Para demonstrar este teorema, mostra-se que a condição 2 da Definição 1.12 não é satisfeita, isto é, para toda $\mathcal{M}(n)$ existem $t, x \in \mathbb{N}$ tais que $(t, x) \notin Q_i^{\mu}$ e $\Delta_{LPC}(\mathcal{M}(n)) \nvdash \neg Q_i(\bar{t}, \bar{n})$. Para isso constroem-se modelos de máquinas de Turing "não estândar" $\mathcal{M}'$, cujas estruturas $\mu(\mathcal{M}'(n))$ validam os axiomas de $\Delta_{LPC}(\mathcal{M}(n))$, mas não validam a fórmula $\neg Q_i(\bar{t}, \bar{x})$ para alguns $\bar{t}$ e $\bar{x}$ específicos.

Os primeiros modelos não-estândar para as teorias que axiomatizam máquinas de Turing podem ser construídos aproveitando-se do fato que, no método de axiomatização B-J, não são incluídas sentenças nas teorias $\Delta_{LPC}(\mathcal{M}(n))$ para estabelecer a suposição de que a máquina, quando desligada, não está em nenhum estado, não está lendo nenhuma casa da fita e não tem nenhum símbolo (nem sequer o vazio) em nenhuma casa da fita. Pode-se então definir modelos de máquinas de Turing $\mathcal{M}'$, onde $\mathcal{M}'$ constrói-se acrescentando-se à definição da máquina de Turing $\mathcal{M}$ a condição de que quando esteja desligada vai se encontrar no estado q_i, na posição 0, e com o símbolo S_j em todas as casas da fita. A definição das estruturas $\mu(\mathcal{M}'(n))$, para expressar matematicamente a computação de $\mathcal{M}'(n)$, seria similar à da Definição 1.10, só que se acrescentariam à relação Q_i^{μ} os pares $(t, 0)$, tais que $t < 0$ ou $t >$ tempo-de-parada (se existir); e se acrescentariam à relação S_j^{μ} os pares (t, x), t com as mesmas condições anteriores e $x \in \mathbb{Z}$. Assim, pela demonstração do Teorema 1.1, $\mu(\mathcal{M}'(n))$ valida todos os axiomas de $\Delta_{LPC}(\mathcal{M}(n))$, mas, $\mu(\mathcal{M}'(n))$ não valida $\neg Q_i(\bar{t}, \bar{x})$ para os valores de $\bar{t}$ e $\bar{x}$ especificados.

Outros modelos não-estândar, quiçá mais interessantes, podem ser construídos aproveitando-se ainda do fato que, no método de axiomatização B-J, não são incluídas sentenças nas teorias $\Delta_{LPC}(\mathcal{M}(n))$ para estabelecer a unicidade de estado, posição e símbolo em cada casa da fita, em cada instante de tempo. Podem-se definir, por exemplo, máquinas de Turing $\mathcal{M}'$ acrescentando-se à definição da máquina de Turing $\mathcal{M}$ a condição de que no começo da computação, tempo $t = 0$, a máquina além de estar no estado

inicial q_1, na posição 0 da fita e lendo o símbolo s_l, a máquina esteja também no estado q_i (com $q_i \neq q_1$), na posição 0 da fita e lendo o símbolo s_j (com $s_j \neq s_l$). Note que para construir estes modelos é necessario que a máquina tenha mais de um símbolo de estado e mais de um símbolo de entrada-saída. De maneira similar aos modelos não-estândar anteriores, estes outros modelos não-estândar também validam todos os axiomas de $\Delta_{LPC}(\mathcal{M}(n))$, mas não validam $\neg Q_i(\bar{t}, \bar{x})$ para $\bar{t} = \bar{x} = 0$ e possivelmente para mais valores, se se pensar que $\mathcal{M}'$, nas situações simultâneas em que estiver, a máquina possa executar todas as possíveis instruções simultâneamente.

□

Teorema 1.5. *Nas teorias $\Delta_{LPC}(\mathcal{M}(n))$, os símbolos de predicado S_j não representam as relações S_j^{μ} da estrutura $\mu(\mathcal{M}(n))$ respectiva.*

Demonstração. Similar à demonstração do teorema anterior, usando-se os mesmos modelos não-estândar. □

Teorema 1.6. *As teorias $\Delta_{LPC}(\mathcal{M}(n))$ não representam a computação da MTD $\mathcal{M}(n)$ respectiva.*

Demonstração. Pela Definição 1.13 e os Teoremas 1.4 e 1.5. □

1.2.3 Ajustes ao método de axiomatização B-J

Para conseguir representar as relações Q_i^{μ} e S_j^{μ}, das estruturas $\mu(\mathcal{M}(n))$, por meio dos símbolos de predicado Q_i e S_j, acrescentam-se novos axiomas as teorias $\Delta_{LPC}(\mathcal{M}(n))$, obtendo-se teorias $\Delta'_{LPC}(\mathcal{M}(n))$. Os novos axiomas impedem a construção de modelos não-estândar como os apresentados na demonstração do Teorema 1.4, estabelecendo os supostos de Boolos e Jeffrey a respeito de quando a máquina está desligada e as unicidades de estado, posição e símbolo sobre cada casa da fita em todo instante de tempo.

Definição 1.14 (Teoria $\Delta'_{LPC}(\mathcal{M}(n))$). *Seja $\mathcal{M} = \langle Q, \Sigma, M, I\rangle$ uma máquina de Turing e $n = s_{i_0}s_{i_1}\ldots s_{i_{p-1}}$ com $s_{i_j} \in \Sigma$ a seqüência ou dado de entrada para $\mathcal{M}$. Sendo n o cardinal de Q e m o cardinal de Σ, a teoria $\Delta'_{LPC}(\mathcal{M}(n))$ estende a teoria $\Delta_{LPC}(\mathcal{M}(n))$ (Definição 1.9) acrescentando os axiomas:*

- *Um axioma para estabelecer o pressuposto de que $\mathcal{M}(n)$, antes do tempo $t = 0$, não está em nenhum estado, não esta lendo nenhuma*

casa da fita e não tem nenhum símbolo (nem sequer o vazio) em nenhuma casa da fita; este axioma é construído de acordo com o esquema:

$$\forall t \forall x \left((t < 0) \rightarrow \left(\left(\bigwedge_{i=1}^{n} \neg Q_i(t.x) \right) \wedge \left(\bigwedge_{j=0}^{m-1} \neg S_j(t.x) \right) \right) \right). \qquad (\mathrm{A}t < 0)$$

- *Um axioma para estabelecer o pressuposto de que $\mathcal{M}(n)$, depois do tempo de parada (se existir), não está em nenhum estado, não esta lendo nenhuma casa da fita e não tem nenhum símbolo (nem sequer o vazio) em nenhuma casa da fita; este axioma é construído de acordo com o esquema:*

$$\forall t \forall x \left(\neg \left(\bigvee_{q_i s_j \ inic.} \left(Q_i(t.x) \wedge S_j(t,x) \right) \right) \rightarrow \right.$$
$$\left. \forall u \forall y \left(t < u \rightarrow \left(\left(\bigwedge_{i=1}^{n} \neg Q_i(u,y) \right) \wedge \left(\bigwedge_{j=0}^{m-1} \neg S_j(u,y) \right) \right) \right) \right).$$
$$(\mathrm{A}t > t.parada)$$

O subíndice "$q_i s_j$ inic" na disjuntória indica que devem ter-se em conta somente os pares $q_i s_j$ que aparecem como iniciais nas instruções de $\mathcal{M}$.

- *Um axioma por cada símbolo de estado $q_i \in Q$, para estabelecer a unicidade do estado em todo instante de tempo t; este axioma é construído de acordo com o esquema:*

$$\forall t \forall x \left(Q_i(t,x) \rightarrow \bigwedge_{i \neq j} \neg Q_j(t,x) \right). \qquad (\mathrm{A}q_i)$$

- *Um axioma para estabelecer a unicidade de posição em cada instante de tempo t; este axioma é construído de acordo como o esquema:*

$$\forall t \forall x \left(\bigvee_{i=1}^{n} Q_i(t,x) \rightarrow \forall y \left(x \neq y \rightarrow \bigwedge_{j=1}^{n} \neg Q_j(t.y) \right) \right). \qquad (\mathrm{A}x)$$

- *Um axioma por cada símbolo de entrada-saída $s_j \in \Sigma$, para estabelecer a unicidade do símbolo em cada casa da fita em todo instante de tempo t: este axioma é construído de acordo como o esquema:*

$$\forall t \forall x \left(S_i(t,x) \rightarrow \bigwedge_{i \neq j} \neg S_j(t,x) \right). \qquad (\mathrm{A}s_j)$$

Exemplo 1.5. *Seja $\mathcal{M}_1$ e n_1 a máquina de Turing e o dado de entrada do Exemplo 1.1, então $\Delta'_{LPC}(\mathcal{M}_1(n_1))$ estende a teoria $\Delta_{LPC}(\mathcal{M}_1(n_1))$ (do Exemplo 1.4) acrescentando os axiomas:*

$$\forall t \forall x \Big((t < 0) \rightarrow \big(\neg Q_1(t,x) \wedge \neg Q_2(t,x) \wedge \neg S_0(t,x) \wedge \neg S_1(t,x) \big) \Big). \qquad (\mathrm{A}t < 0\ (\mathcal{M}_1))$$

$$\forall t \forall x \Bigg(\neg \Bigg(\Big(Q_1(t,x) \wedge S_1(t,x) \Big) \vee \Big(Q_1(t,x) \wedge S_0(t,x) \Big) \Bigg) \rightarrow \forall u \forall y \Bigg(t < u \rightarrow \Big(\neg Q_1(u,y) \wedge \neg Q_2(u,y) \wedge \neg S_0(u,y) \wedge S_1(u,y) \Big) \Bigg) \Bigg), \qquad (\mathrm{A}t > t.parada\ (\mathcal{M}_1))$$

$$\forall t \forall x (Q_1(t,x) \rightarrow \neg Q_2(t,x)), \qquad (\mathrm{A}q_1)$$

$$\forall t \forall x (Q_2(t,x) \rightarrow \neg Q_1(t,x)). \qquad (\mathrm{A}q_2)$$

$$\forall t \forall x \Big(\big(Q_1(t,x) \vee Q_2(t,x) \big) \rightarrow \forall y \Big(x \neq y \rightarrow \big(\neg Q_1(t,y) \wedge \neg Q_2(t,y) \big) \Big) \Big), \qquad (\mathrm{A}x\ (\mathcal{M}_1))$$

$$\forall t \forall x (S_0(t,x) \rightarrow \neg S_1(t,x)), \qquad (\mathrm{A}s_0)$$

$$\forall t \forall x (S_1(t,x) \rightarrow \neg S_0(t,x)). \qquad (\mathrm{A}s_1)$$

1.2.4 Representabilidade das MTDs nas teorias $\Delta'_{LPC}(\mathcal{M}(n))$

Para demonstrar que as teorias $\Delta'_{LPC}(\mathcal{M}(n))$ representam a computação da MTD $\mathcal{M}(n)$ respectiva, apresenta-se primeiro uma definição e demonstra-se um teorema auxiliar; logo, demonstra-se a representação nas teorias $\Delta'_{LPC}(\mathcal{M}(n))$ da função $'\mu$ e das relações Q_i, S_j e $<$.

Definição 1.15 (Fórmula associada a uma descrição instantânea[8]). *Seja $\alpha_t = s_{j_1} \dots s_{j_n} q_i s_j \dots s_{j_p}$ a descrição instantânea de $\mathcal{M}(n)$ no tempo t, com $\mathcal{M}$ localizada na posição x da fita. Utilizando para os valores negativos de p, para abreviar as fórmulas, as seguintes convenções:*

$$\begin{aligned} &Q_i(t, 0^p) : \ Q_i(t, s) \wedge s^p = 0, \\ &S_j(t, 0^p) : \ S_j(t, s) \wedge s^p = 0, \\ &y \neq 0^p : \ y \neq s \wedge s^p = 0; \end{aligned}$$

define-se a fórmula:

$$\begin{aligned} D_t = Q_i(t, 0^x) \wedge S_{j,1}(t, 0^{x-n}) \wedge \dots \wedge S_{j,n}(t, 0^{x-1}) \wedge S_j(t, 0^x) \wedge \dots \\ \wedge S_{j,p}(t, 0^{(x+p)-(n+1)}) \wedge \forall y \Big(\big(y \neq 0^{x-n} \wedge \dots \wedge y \neq 0^{x-1} \\ \wedge y \neq 0^x \wedge \dots \wedge y \neq 0^{(x+p)-(n+1)} \big) \rightarrow S_0(t, y) \Big) \quad (D_t) \end{aligned}$$

como a fórmula associada a α_t.

Teorema 1.7. *Seja $\alpha_0 \alpha_1 \alpha_2 \dots$ a seqüência (possivelmente infinita) de descrições instantâneas de $\mathcal{M}(n)$, então $\Delta'_{LPC}(\mathcal{M}(n)) \vdash D_t$, para $t = 0, 1, 2 \dots$*

Demonstração. Utilizando indução matemática:

1. Passo base $t = 0$:
   1.1. O axioma (An), o qual codifica a situação inicial de $\mathcal{M}(n)$, é precisamente a fórmula associada a α_0.
2. Passo indutivo:
   2.1. Seja $\alpha_t = s_{j_1} \dots s_{j_n} q_i s_j \dots s_{j_p}$ a descrição instantânea de $\mathcal{M}(n)$ no tempo t, com $\mathcal{M}$ localiçada na posição x.
   2.2. Por hipóteses indutiva $\Delta'_{LPC}(\mathcal{M}(n)) \vdash D_t$.

[8] Definição adaptada de [13, p. 117].

2.3. Se α_t não é a última na seqüência e ($\mathcal{M}(n)$ ainda não parou no tempo t), então existe em $\mathcal{M}$ uma instrução que produz a mudança de descrição instantânea $\alpha_t \xrightarrow{\mathcal{M}} \alpha_{t+1}$.

2.4. Se a instrução que produz a mudança de descrição instantânea $\alpha_t \xrightarrow{\mathcal{M}} \alpha_{t+1}$ é da forma (I):

2.4.1. Por 2.1, 2.4 e definição de mudança de descrição instantânea tem-se que $\alpha_{t+1} = s_{j,1} \dots s_{j,n} q_l s_k \dots s_{j,p}$.

2.4.2. Por 2.2 e simplificação tem-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash Q_i(t, 0^x) \wedge S_j(t, 0^x)$.

2.4.3. Por 2.4 e definição de $\Delta'_{LPC}(\mathcal{M}(n))$, em $\Delta'_{LPC}(\mathcal{M}(n))$ existe um axioma construído sob o esquema (Ai_j (I)).

2.4.4. Aplicando instanciação em (Ai_j (I)), MP com 2.4.2 e simplificação obtém-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash Q_l(t', 0^x) \wedge S_k(t', 0^x)$.

2.4.5. Aplicando instanciação em (Ai_j (I)), MP com 2.4.2 e simplificação obtém-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash \forall y\Big((y \neq 0^x) \rightarrow \Big(\bigwedge_{i=0}^{m}(S_i(t, y) \rightarrow S_i(t', y))\Big)\Big)$.

2.4.6. Aplicando simplificação em 2.2 para cada um dos símbolos $S_{j_k}(t, 0^y)$, logo aplicando instanciação em 2.4.5, MP e adjunção obtém-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash S_{j,1}(t', 0^{x-n}) \wedge \dots \wedge S_{j,n}(t', 0^{x-1}) \wedge S_{j,n+2}(t', 0^{x+1}) \wedge \dots \wedge S_{j,p}(t', 0^{(x+p_l-(n+1)})$.

2.4.7. Aplicando simplificação em 2.2 e 2.4.5 obtém-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash \forall y\Big(\big(y \neq 0^{x-n} \wedge \dots \wedge y \neq 0^{x-1} \wedge y \neq 0^x \wedge \dots \wedge y \neq 0^{(x+p)-(n+1)}\big) \rightarrow S_0(t', y)\Big)$.

2.4.8. Aplicando adjunção de 2.4.4, 2.4.6 e 2.4.7, obtém-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash D_{l+1}$.

2.5. Para as mudanças de descrição instantânea produzidos por instruções da forma (II) e (III), a demostração é similar à do passo 2.4.

□

Teorema 1.8. *Nas teorias $\Delta'_{LPC}(\mathcal{M}(n))$, o símbolo de predicado $<$ representa a relação $<^{\mu}$ da estrutura $\mu(\mathcal{M}(n))$ respectiva.*

Demonstração. Igual à demonstração do Teorema 1.2. □

Teorema 1.9. *Nas teorias $\Delta'_{LPC}(\mathcal{M}(n))$, o símbolo de função $'$ representa a função $'^{\mu}$ da estrutura $\mu(\mathcal{M}(n))$ respectiva.*

Demonstração. Igual à demonstração do Teorema 1.3. □

Teorema 1.10. *Nas teorias $\Delta'_{LPC}(\mathcal{M}(n))$, os símbolos de predicado Q_i representam as relações Q_i^{μ} da estrutura $\mu(\mathcal{M}(n))$ respectiva.*

Demonstração.

1. Condição 1 da Definição 1.12: seja $(t, x) \in Q_i^{\mu}$ em $\mu(\mathcal{M}(n))$, pela definição de Q_i^{μ}, q_i está na descrição instantânea α_t, e pelo Teorema 1.7 tem-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash D_t$, portanto, pela definição de D_t e simplificação tem-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash Q_i(t, 0^x)$.

2. Condição 2 da Definição 1.12: seja $(t, x) \notin Q_i^{\mu}$, se $t < 0$ ou $t >$ t. de parada (se existir), na teoria $\Delta'_{LPC}(\mathcal{M}(n))$ existem axiomas construídos sob os esquemas $(\mathrm{A}t < 0)$ e $(\mathrm{A}t > t.parada)$, aplicando instanciação, MP e simplificação obtém-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash \neg Q_i(t, x)$ para os valores de t especificados e para todo x. Se $0 \leq t \leq$ t. de parada (se existir), pela definição de Q_i^{μ} tem-se que q_i não está na descrição instantânea α_t, logo, o símbolo de estado presente em α_t é um $q_j \neq q_i$; pelo Teorema 1.7 tem-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash D_t$ e por definição de D_t e simplificação obtém-se que (*) $\Delta'_{LPC}(\mathcal{M}(n)) \vdash Q_j(t, 0^x)$; além disso, na teoria $\Delta'_{LPC}(\mathcal{M}(n))$ existe um axioma construído sob o esquema $(\mathrm{A}q_i)$, portanto, aplicando instanciação, MP com (*) e simplificação obtém-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash \neg Q_i(t, x)$ para os valores de t especificados e para a posição atual x sobre a fita. Para as posições $y \neq x$, na teoria $\Delta'_{LPC}(\mathcal{M}(n))$ existe um axioma construído sob o esquema $(\mathrm{A}x)$, aplicando instanciação para os tempos $0 \leq t \leq$ t. de parada (se existir) e para a posição x; aplicando adição em (*), MP e simplificação obtém-se que $\Delta'_{LPC}(\mathcal{M}(n)) \vdash \neg Q_i(t, y)$ para toda posição $y \neq x$ e para os tempos especificados.

□

Teorema 1.11. *Nas teorias $\Delta'_{LPC}(\mathcal{M}(n))$, os símbolos de predicado S_j representam as relações S_j^{μ} da estrutura $\mu(\mathcal{M}(n))$ respectiva.*

Demonstração. Similar à demonstração do teorema anterior. □

Teorema 1.12. *As teorias $\Delta'_{LPC}(\mathcal{M}(n))$ representam a computação da MTD $\mathcal{M}(n)$ respectiva.*

Demonstração. Pela Definição 1.13 e os Teoremas 1.8, 1.9, 1.10 e 1.11. □

1.3 Axiomatização de MTNDs

Na tentativa de se axiomatizar computações em MTNDs, se simplesmente se utilizar o método de axiomatização definido na seção anterior para axiomatizar MTDs obtém-se, em alguns casos, teorias contraditórias (como é mostrado no Exemplo 1.6) e portanto triviais, tendo em conta que a lógica subjacente é o calculo de predicados clássico. As contradições surgem devido ao fato de que o método de axiomatização não leva em conta que as MTNDs, ao se encontrar numa situação na qual existem varias possíveis instruções a serem executadas, escolhe e executa somente uma delas. Para solucionar tal problema, fazem-se algumas modificações ao método de axiomatização de forma a obter teorias $\Delta''_{LPC}(\mathcal{M}(n))$ que formalizem o comportamento de MTNDs.

Exemplo 1.6. *Seja $\mathcal{M}_2$ a MTND obtida ao acrescentar à MTD $\mathcal{M}_1$ do Exemplo 1.1 uma nova instrução $i_3 = q_1 s_1 L q_1$ (Figura 1.6), e seja $n_2 = s_1$ a seqüência de entrada para $\mathcal{M}_2$.*

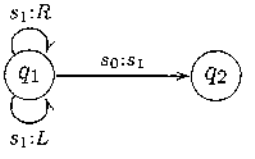

Figura 1.6: Grafo da máquina $\mathcal{M}_2$

Utilizando o método de axiomatização definido na seção anterior, tem-se que $\Delta'_{LPC}(\mathcal{M}_2(n_2)) = \{\Delta'_{LPC}(\mathcal{M}_1(n_1)) - \{\ An_1\}\} \cup \{An_2, Ai_3\}$, onde $\Delta'_{LPC}(\mathcal{M}_1(n_1))$ é a teoria do Exemplo 1.5 e:

$$Q_1(0,0) \wedge S_1(0,0) \wedge \forall y\,(y \neq 0 \rightarrow S_0(0,y))\,. \qquad (An_2)$$

$$\forall t \forall x \Big(\Big(Q_1(t.x') \wedge S_1(t.x') \Big) \rightarrow \Big(Q_1(t',x) \wedge$$

$$\forall y \Big(\big(S_0(t,y) \rightarrow S_0(t',y) \big) \wedge \big(S_1(t,y) \rightarrow S_1(t',y) \big) \Big) \Big) \Big), \quad (\mathrm{Ai}_3)$$

Pelo axioma (An_1) *(Exemplo 1.4) e simplificação tem-se que (*)* $\Delta'_{LPC}(\mathcal{M}_2(n_2)) \vdash Q_1(0,0) \wedge S_1(0,0)$, *aplicando instanciação em* (Ai_1) *(Exemplo 1.4), MP com (*) e simplificação obtém-se que* $\Delta'_{LPC}(\mathcal{M}_2(n_2)) \vdash Q_1(0',0')$. *Aplicando instanciação em* (Ai_3), *MP com (*) e simplificação obtém-se que (**)* $\Delta'_{LPC}(\mathcal{M}_2(n_2)) \vdash Q_1(0',-1)$, *assumindo que* $-1' = 0$. *Aplicando instanciação em* ($\mathrm{Ax}\ (\mathcal{M}_1)$) *(Exemplo 1.5), adição em (**). MP e simplificação obtém-se que (***)* $\Delta'_{LPC}(\mathcal{M}_2(n_2)) \vdash \forall y\,((y \neq -1) \rightarrow (\neg Q_1(0',y) \wedge \neg Q_2(0',y)))$. *Como* $-1' = 0$, *aplicando instanciação no axioma* (A4) *(Definição 1.9) tem-se que* $\Delta'_{LPC}(\mathcal{M}_2(n_2)) \vdash -1 < 0$ *e* $\Delta'_{LPC}(\mathcal{M}_2(n_2)) \vdash 0 < 0'$, *aplicando instanciação no axioma* (A3) *(Definição 1.9), adjunção e MP obtém-se que* $\Delta'_{LPC}(\mathcal{M}_2(n_2)) \vdash -1 < 0'$, *e aplicando instanciação no axioma* (A5) *(Definição 1.9) e MP obtém-se que* $\Delta'_{LPC}(\mathcal{M}_2(n_2)) \vdash -1 \neq 0'$; *logo, por instanciação em (***), MP e simplificação obtém-se que* $\Delta'_{LPC}(\mathcal{M}_2(n_2)) \vdash \neg Q_1(0',0')$. *Portanto* $\Delta'_{LPC}(\mathcal{M}_2(n_2))$ *é contraditória.*

Note que as inconsistências no Exemplo 1.6 surgem devido a que, na teoria $\Delta'_{LPC}(\mathcal{M}_2(n_2))$, não é estabelecida a condição de que só uma das instruções (i_1, i_3) seja executada quando a máquina estiver no estado q_1 lendo o símbolo s_1. Substituindo os conseqüentes dos axiomas (Ai_1) (Exemplo 1.4) e (Ai_3) por α e β respectivamente, com o objetivo de abreviar as fórmulas, tem-se que:

$$\forall t \forall x((Q_1(t,x) \wedge S_1(t,x)) \rightarrow \alpha), \quad (\mathrm{Ai}_1\text{'})$$

$$\forall t \forall x((Q_1(t,x) \wedge S_1(t,x)) \rightarrow \beta). \quad (\mathrm{Ai}_3\text{'})$$

Donde se deduz:

$$\forall t \forall x((Q_1(t.x) \wedge S_1(t,x)) \rightarrow (\alpha \wedge \beta)), \quad (\mathrm{Ai}_1\text{'-Ai}_3\text{'})$$

que interpretado, sob a interpretação intencional $\mathcal{I}$, indica que a máquina executa ambas instruções simultâneamente.

Em [81, p. 232] Turing faz diferença entre "máquinas automáticas" (*a-machines*) e "máquinas de escolha" (*c-machines*). Indicando que uma

a-machine é aquela na qual a operação da máquina é completamente determinada pela configuração (estado atual e símbolo que está lendo a máquina), em quanto que numa *c-machine* a operação da máquina está só parcialmente determinada pela configuração, quando a máquina chega a uma configuração "ambígua" (configuração na qual varias instruções podem ser executadas), a máquina não pode continuar até que um operador externo realize uma escolha. As MTDs correspondem então às *a-machines* e as MTNDs correspondem às *c-machines*. Sob a definição de configuração ambígua dada por Turing e a interpretação intencional $\mathcal{I}$ das teorias $\Delta'_{LPC}(\mathcal{M}(n))$, a causa das contradições nas teorias $\Delta'_{LPC}(\mathcal{M}(n))$ é a falta de controle (escolha de uma única instrução) sobre as situações ambíguas.

Em [65, p. 48], ao definir o conjunto de instruções da máquina de Turing, Odifreddi estabelece a condição de que devem ser "consistentes", indicando com isto que não podem existir pares de instruções "contraditórios", isto é, com as mesmas *premissas* $q_i s_j$ e com diferentes *conclusões*[9]. O que corresponde à definição de MTD. Odifreddi define então as MTNDs eliminando da definição de máquina de Turing a condição de "consistência" nas instruções. A noção de consistência para máquinas de Turing definida por Odifreddi corresponde à noção de consistência das teorias $\Delta'_{LPC}(\mathcal{M}(n))$, o que é uma característica adicional (além do teorema de representação, Teorema 1.12), que indica de forma veemente que as teorias $\Delta'_{LPC}(\mathcal{M}(n))$ são adequadas para formalizar o comportamento das MTDs.

Para construir teorias $\Delta''_{LPC}(\mathcal{M}(n))$, nas quais se estabeleça a condição de escolher e executar só uma instrução quando $\mathcal{M}$ estiver numa situação com varias possíveis instruções a serem executadas, deve-se modificar em $\Delta'_{LPC}(\mathcal{M}(n))$ (Definição 1.14) a forma de construir os axiomas para as instruções em conflito (isto é, para as instruções i_j para as quais existe pelo menos uma outra instrução $i_k \neq i_j$ com os mesmos dois símbolos iniciais $q_i s_j$). Tal modificação se faz incluindo-se um predicado de escolha $E_j(t)$ no antecedente do axioma correspondente à instrução i_j, o qual indica que das múltiplas possíveis instruções a serem executadas no tempo t, a instrução escolhida foi a i_j:

Definição 1.16 (Teoria $\Delta''_{LPC}(\mathcal{M}(n))$). *Seja $\mathcal{M}$ uma MTD e n o dado de entrada para $\mathcal{M}$, a teoria $\Delta''_{LPC}(\mathcal{M}(n))$ constrói-se substituindo em $\Delta'_{LPC}(\mathcal{M}(n))$ (Definição 1.9) os axiomas correspondentes as instruções i_j em conflito por axiomas de acordo com o esquema:*

$$\forall t \forall x \left((Q_i(t,x) \wedge S_j(t,x) \wedge E_j(t)) \rightarrow \alpha \right), \qquad (\mathrm{A}i_j")$$

[9]Odifreddi denomina *premissas* aos dois primeiros símbolos das instruções e *conclusões* aos símbolos seguintes.

onde α é o conseqüente de algum dos esquemas ($\mathrm{Ai_j}$ (I)). ($\mathrm{Ai_j}$ (II)) *ou* ($\mathrm{Ai_j}$ (III)), *dependendo de se i_j é de tipo* (I), (II) *ou* (III) *respectivamente.*

Exemplo 1.7. *Para a máquina de Turing $\mathcal{M}_2$ e a entrada n_2 do Exemplo 1.6, a teoria $\Delta''_{LPC}(\mathcal{M}_2(n_2))$ obtém-se substituindo em $\Delta'_{LPC}(\mathcal{M}_2(n_2))$ (Exemplo 1.6) os axiomas* ($\mathrm{Ai_1}$) *e* ($\mathrm{Ai_3}$) *pelos axiomas:*

$$\forall t \forall x \Big(\Big(Q_1(t.x) \wedge S_1(t,x) \wedge E_1(t) \Big) \rightarrow \Big(Q_1(t',x') \wedge \forall y \Big(\big(S_0(t,y) \rightarrow S_0(t',y) \big) \wedge \big(S_1(t,y) \rightarrow S_1(t',y) \big) \Big) \Big) \Big), \quad (\mathrm{Ai_1''})$$

$$\forall t \forall x \Big(\Big(Q_1(t,x') \wedge S_1(t,x') \wedge E_3(t) \Big) \rightarrow \Big(Q_1(t',x) \wedge \forall y \Big(\big(S_0(t,y) \rightarrow S_0(t',y) \big) \wedge \big(S_1(t,y) \rightarrow S_1(t',y) \big) \Big) \Big) \Big). \quad (\mathrm{Ai_3''})$$

Note que, nas teorias $\Delta''_{LPC}(\mathcal{M}(n))$, para demonstrar as fórmulas D_t associadas a uma computação particular c_i de $\mathcal{M}(n)$, é necessario acrescentar as escolhas das instruções em conflito feitas nos instantes de tempo t em que $\mathcal{M}(n)$ esteve em situações ambíguas; tais teorias serão denotadas por $\Delta''_{LPC}(\mathcal{M}(n))_{c_i}$ e são definidas assim:

Definição 1.17 (Teoria $\Delta''_{LPC}(\mathcal{M}(n))_{c_i}$). *Seja $\mathcal{M}$ uma MTND, n o dado de entrada para $\mathcal{M}$, e $c_i = \alpha_0\alpha_1\alpha_2\ldots$ uma seqüência (possivelmente infinita) de descrições instantâneas associada à computação*[10] *particular c_i de $\mathcal{M}(n)$. A teoria $\Delta''_{LPC}(\mathcal{M}(n))_{c_i} = \Delta''_{LPC}(\mathcal{M}(n)) \cup \{E_j(t) \mid$ a instrução em conflito i_j foi escolhida no tempo t na computação $c_i\}$.*

Exemplo 1.8. *Para a MTND $\mathcal{M}_2$ e a entrada n_2 do Exemplo 1.6, e para a computação $c_i = \alpha_0\alpha_1\alpha_2$, onde $\alpha_0 = q_1s_1$, $\alpha_1 = q_1s_0s_1$ e $\alpha_2 = q_2s_1s_1$, a teoria $\Delta''_{LPC}(\mathcal{M}_2(n_2))_{c_i} = \Delta''_{LPC}(\mathcal{M}_2(n_2)) \cup \{E_3(0)\}$, onde $\Delta''_{LPC}(\mathcal{M}_2(n_2))$ é a teoria do Exemplo 1.7.*

[10]Para uma seqüência de descrições instantâneas ser considerada uma "computação" a seqüência tem que ser finita (Definição 1.8), mas vai se abusar do termo "computação" para denotar também seqüências infinitas de descrições instantâneas que descrevem o comportamento das $\mathcal{M}(n)$ que não param.

Denotando por $\mu(\mathcal{M}(n))_{c_i}$ a estrutura matemática[11] que representa a computação particular c_i de $\mathcal{M}(n)$, e fazendo alguns pequenos ajustes nas demonstrações dos Teoremas 1.7 - 1.11, pode-se demonstrar que as teorias $\Delta''_{LPC}(\mathcal{M}(n))_{c_i}$ representam a computação particular c_i, da MTND $\mathcal{M}$ e o dado de entrada n respectivos.

[11] A definição de $\mu(\mathcal{M}(n))_{c_i}$ é igual a definição de $\mu(\mathcal{M}(n))$ (Definição 1.10), com a diferença de que a seqüência de descrições instantâneas $\alpha_0\alpha_1\alpha_2\ldots$ corresponde a uma computação particular c_i.

Capítulo 2

Máquinas de Turing paraconsistentes

Como apresentado no capítulo anterior (Seção 1.3), ao axiomatizar o comportamento de MTNDs usando o método de axiomatização definido para MTDs (ver Seção 1.2), obtém-se em alguns casos teorias $\Delta'_{LPC}(\mathcal{M}(n))$ que são contraditórias, e portanto triviais tendo-se em conta que a lógica subjacente é a lógica proposicional clássica. Para evitar a trivialização das teorias $\Delta'_{LPC}(\mathcal{M}(n))$, na Seção 1.3 modificou-se o método de axiomatização para MTDs, acrescentando-se condições aos axiomas correspondentes às instruções de $\mathcal{M}$, de tal forma a obter teorias $\Delta''_{LPC}(\mathcal{M}(n))_{c_i}$ que representem o comportamento de MTNDs para computações específicas. No presente capítulo escolhe-se um caminho diferente para evitar a trivialização das teorias $\Delta'_{LPC}(\mathcal{M}(n))$ (quando usadas para axiomatizar MTNDs), o qual consiste em deixar os axiomas de tais teorias intactos e substituir a lógica subjacente pela lógica de primeira ordem paraconsistente **LFI1***. Desse modo, obtém-se teorias $\Delta'_{\text{LFI1*}}(\mathcal{M}(n))$ contraditórias mas não triviais, e mediante a interpretação destas teorias constrói-se o modelo das que serão chamadas de *máquinas de Turing paraconsistentes* (MTPs) (descrito na Seção 2.2). Para poder demonstrar que as teorias $\Delta'_{\text{LFI1*}}(\mathcal{M}(n))$ de fato representam formalmente as computações das MTPs definidas, adaptam-se as definições de representação de funções, relações e computações dadas no capítulo anterior tendo como base a lógica clássica para teorias com **LFI1*** como lógica subjacente (Definições 2.2 a 2.10). A seguir definem-se mecanismos para controlar as contradições nas teorias $\Delta'_{\text{LFI1*}}(\mathcal{M}(n))$, e interpretando-se tais mecanismos permite-se definir condições de execução das instruções nas MTPs, condições estas essenciais para tirar proveito do modelo de MTPs (Seção

2.2.2). Finalmente, descreve-se a possibilidade de construir outros modelos de MTPs (Seção 2.2.3).

Na definição do modelo de MTPs é utilizada a lógica **LFI1*** por ser uma lógica paraconsistente já estendida ao caso da primeira ordem, e pelo fato de que ela preserva o fragmento positivo da lógica proposicional clássica, o que facilita a definição do modelo de MTPs. Além disso, **LFI1*** conta com uma definição de modelo (estrutura) que já foi demonstrada em [20] ser correta e completa com respeito à axiomatização de **LFI1***, o que permite redefinir as noções de representação de funções, relações e computações numa teoria, e ainda demonstrar que as computações das MTPs são representadas pelas respectivas teorias. Desta maneira justifica-se o modelo de MTPs enquanto um modelo de computação baseado numa lógica paraconsistente. Além dessa justificativa lógica, **LFI1*** foi originalmente pensada para manipular o tratamento da informação (enquanto sistemas de bancos de dados). Contudo, na definição de modelo de MTPs em princípio poder-se-ia usar qualquer outra lógica paraconsistente que fosse estensível à primeira ordem, possivelmente dando como resultado modelos de MTPs diferentes, como ilustrado na Seção 2.2.3.

Antes de definir o modelo de MTPs apresenta-se uma breve descrição da lógica **LFI1***, seguindo [20]:

2.1 A lógica LFI1*

Em [20] Walter Carnielli, João Marcos e Sandra de Amo apresentam detalhadamente a lógica de primeira ordem paraconsistente **LFI1*** e a utilizam na fundamentação lógica dos sistemas de bancos de dados evolutivos. **LFI1*** é a extensão a primeira ordem da lógica proposicional **LFI1**, a qual faz parte de uma grande família de lógicas proposicionais paraconsistentes chamadas de "Lógicas da inconsistência formal" (LFIs) (cf. [20]). As LFIs caracterizam-se como lógicas paraconsistentes que internalizam as noções metateóricas de consistência e inconsistência ao nível da linguagem objeto. Nas LFIs os conceitos de contradição e inconsistência não são necessariamente identificados, mas na lógica **LFI1** contradição e inconsistência são de fato identificados por meio da equivalência $\bullet A \leftrightarrow (A \wedge \neg A)$, onde $\bullet$ é o operador de inconsistência. À diferença de muitas lógicas da família das LFIs, as quais não podem ser caracterizadas por meio de matrizes finitas, **LFI1** é uma lógica 3-valorada, cujas tabelas de verdade para os operadores $\wedge, \vee, \rightarrow, \neg$ e $\bullet$ são:

$\wedge$	1	$1/2$	0
1	1	$1/2$	0
$1/2$	$1/2$	$1/2$	0
0	0	0	0

$\vee$	1	$1/2$	0
1	1	1	1
$1/2$	1	$1/2$	$1/2$
0	1	$1/2$	0

$\rightarrow$	1	$1/2$	0
1	1	$1/2$	0
$1/2$	1	$1/2$	0
0	1	1	1

	$\neg$	$\bullet$
1	0	0
$1/2$	$1/2$	1
0	1	0

onde 1 e $1/2$ são os valores distinguidos. Os operadores $\vee, \neg$ e $\bullet$ podem ser tomados como primitivos e $A \wedge B$ e $A \rightarrow B$ podem ser definidos como $\neg(\neg A \vee \neg B)$ e $B \vee \neg(A \vee \bullet A)$ respectivamente. Usando $A \leftrightarrow B$ para abreviar $(A \rightarrow B) \wedge (B \rightarrow A)$ e usando $\circ A$ para abreviar a fórmula $\neg \bullet A$. **LFI1** pode ser axiomatizada por:

$$A \rightarrow (B \rightarrow A). \tag{LFI1-1}$$
$$(A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C)), \tag{LFI1-2}$$
$$A \rightarrow (B \rightarrow (A \wedge B)), \tag{LFI1-3}$$
$$(A \wedge B) \rightarrow A, \tag{LFI1-4}$$
$$(A \wedge B) \rightarrow B, \tag{LFI1-5}$$
$$A \rightarrow (A \vee B), \tag{LFI1-6}$$
$$B \rightarrow (A \vee B), \tag{LFI1-7}$$
$$(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \vee B) \rightarrow C)), \tag{LFI1-8}$$
$$A \vee \neg A, \tag{LFI1-9}$$
$$\neg\neg A \leftrightarrow A, \tag{LFI1-10}$$
$$\circ A \rightarrow (A \rightarrow (\neg A \rightarrow B)), \tag{LFI1-11}$$
$$\bullet A \rightarrow (A \wedge \neg A), \tag{LFI1-12}$$
$$\bullet (A \wedge B) \leftrightarrow ((\bullet A \wedge B) \vee (\bullet B \wedge A)), \tag{LFI1-13}$$
$$\bullet (A \vee B) \leftrightarrow ((\bullet A \wedge \neg B) \vee (\bullet B \wedge \neg A)). \tag{LFI1-14}$$
$$\bullet (A \rightarrow B) \leftrightarrow (A \wedge \bullet B). \tag{LFI1-15}$$

e a regra de modus ponens (MP): $A, A \rightarrow B / B$.

Como demonstrado em [20], a axiomática para a parte proposicional **LFI1** apresentada acima é correta e completa com respeito as tabelas de verdade apresentadas anteriormente.

A escolha de **LFI1*** é natural em razão de que seu fragmento proposicional **LFI1** foi várias vezes redescoberto por pesquisadores preocupados com questões completamente ditintas; isso significa que a caracterização de **LFI1** como uma soluçao lógica a várias questões onde o raciocínio com contradições é problemático surge de forma bastante espontânea, e daí a naturalidade à qual nos referimos. De fato, **LFI1** já havia sido introduzida

trinta anos antes de [20] como a lógica paraconsistente trivalente maximal **J3** em [40], e dez anos antes em [74] com propósitos de justificação de certos argumentos da teoria da demonstração. Ainda mais, essa mesma lógica coincide com o sistema chamado **CLuNs** tratado em [4].

Para estender **LFI1** à lógica de primeira ordem **LFI1*** acrescentam-se os axiomas:

$A(t) \rightarrow \exists x A(x)$, onde t é um termo livre para x em $A(x)$; (LFI1*-ax16)

$\forall x A(x) \rightarrow A(t)$, onde t é um termo livre para x em $A(x)$; (LFI1*-ax17)

$\neg \forall x A(x) \leftrightarrow \exists x \neg A(x)$, (LFI1*-ax18)

$\neg \exists x A(x) \leftrightarrow \forall x \neg A(x)$, (LFI1*-ax19)

$\bullet\, \forall x A(x) \leftrightarrow (\exists x \bullet A(x) \wedge \forall x A(x))$, (LFI1*-ax20)

$\bullet\, \exists x A(x) \leftrightarrow (\exists x \bullet A(x) \wedge \forall x \neg A(x))$. (LFI1*-ax21)

e as regras:

$A(x) \rightarrow B / \exists x A(x) \rightarrow B$, (introdução-$\exists$)

$B \rightarrow A(x) / B \rightarrow \forall x A(x)$. (introdução-$\forall$)

Para que se esclareça o caráter natural dos axiomas (LFI1*-ax18) e (LFI1*-ax19) vale constatar que na parte proposicional de **LFI1*** (ou seja, na lógica proposicional **LFI1**) valem as seguintes versões de lei de De Morgan:

- $\neg(A \wedge B) \equiv \neg A \vee \neg B$
- $\neg(A \vee B) \equiv \neg A \wedge \neg B$

o que pode ser facilmente verificado usando-se as tabelas de verdade de **LFI1**.

Os modelos para interpretar teorias **LFI1*** (onde a lógica subjacente é **LFI1***) são similares aos modelos clássicos, com a diferença de que são acrescentadas as constantes "não estândar" $\checkmark$ e $\times$ no universo dos modelos e a interpretação de termos e predicados é adaptada para incluir as novas constantes. Os símbolos de predicado R de aridade n são interpretados nos novos modelos por relações $R^{I+} \subseteq R^I \times \{\checkmark, \times\}$, onde R^I representa a interpretação estândar (isto é, $R^I \subseteq |I|^n$, sendo $|I|$ o universo de I), e impõe-se a condição de que $(r, \checkmark)$ e $(r, \times)$ não ocorrem simultâneamente para $r \in R^I$. Os símbolos de função f de aridade n são interpretados nos novos modelos por funções $f^I : |I|^n \rightarrow |I| \times \{\checkmark, \times\}$.

Uma interpretação $\models$, numa estrutura I, para sentenças de teorias **LFI1*** é indutivamente definida por:

- Cláusulas para literais "estendidos", com $c_1, c_2, \ldots, c_n$ sendo termos fechados:

$$I \vDash R(c_1, c_2, \ldots, c_n) \iff (c_1^I, c_2^I, \ldots, c_n^I, \checkmark) \in R^{I+} \text{ ou } (c_1^I, c_2^I, \ldots, c_n^I, \times) \in R^{I+}. \tag{2.1}$$

$$I \vDash \neg R(c_1, c_2, \ldots, c_n) \iff (c_1^I, c_2^I, \ldots, c_n^I, \checkmark) \notin R^{I+} \text{ e } (c_1^I, c_2^I, \ldots, c_n^I, \times) \notin R^{I+}, \text{ ou } (c_1^I, c_2^I, \ldots, c_n^I, \times) \in R^{I+}. \tag{2.2}$$

$$I \vDash \bullet R(c_1, c_2, \ldots, c_n) \iff (c_1^I, c_2^I, \ldots, c_n^I, \times) \in R^{I+}. \tag{2.3}$$

- Cláusulas para o fragmento proposicional:

$$I \vDash A \wedge B \iff I \vDash A \text{ e } I \vDash B, \tag{2.4}$$

$$I \vDash A \vee B \iff I \vDash A \text{ ou } I \vDash B, \tag{2.5}$$

$$I \vDash A \rightarrow B \iff I \nvDash A \text{ ou } I \vDash B, \tag{2.6}$$

$$I \vDash \neg\neg A \iff I \vDash A, \tag{2.7}$$

$$I \nvDash \bullet\bullet A, \tag{2.8}$$

$$I \vDash \bullet A \Longrightarrow I \vDash A, \tag{2.9}$$

$$I \vDash \neg A \iff I \nvDash A \text{ ou } I \vDash \bullet A, \tag{2.10}$$

$$I \vDash \bullet(A \wedge B) \iff I \vDash \bullet A \wedge B \text{ ou } I \vDash \bullet B \wedge A, \tag{2.11}$$

$$I \vDash \bullet(A \vee B) \iff I \vDash \bullet A \wedge \neg B \text{ ou } I \vDash \bullet B \wedge \neg A, \tag{2.12}$$

$$I \vDash \bullet(A \rightarrow B) \iff I \vDash A \wedge \bullet B. \tag{2.13}$$

- Cláusulas para os quantificadores:

$$I \vDash \forall x A(x) \iff I \vDash A(t) \text{ para todo termo } t \text{ livre para } x \text{ em } A(x), \tag{2.14}$$

$$I \vDash \exists x A(x) \iff I \vDash A(t) \text{ para algum termo } t \text{ livre para } x \text{ em } A(x), \tag{2.15}$$

$$I \vDash \neg(\forall x A(x)) \iff I \vDash \exists x \neg A(x), \tag{2.16}$$

$$I \vDash \neg(\exists x A(x)) \iff I \vDash \forall x \neg A(x), \tag{2.17}$$

$$I \vDash \bullet(\forall x A(x)) \iff I \vDash \forall x A(x) \text{ e } I \vDash \exists x \bullet A(x), \tag{2.18}$$

$$I \vDash \bullet(\exists x A(x)) \iff I \vDash \forall x \neg A(x) \text{ e } I \vDash \exists x \bullet A(x). \tag{2.19}$$

Como demonstrado em [20], a axiomática apresentada anteriormente para **LFI1*** é completa e correta com respeito à interpretação $\vDash$.

2.2 Modelo de MTPs

Havendo apresentado a lógica **LFI1***, apresentam-se agora alguns exemplos de teorias $\Delta'_{\text{LFI1*}}(\mathcal{M}(n))$, analisam-se suas conseqüências e define-se o modelo de MTPs.

Exemplo 2.1. *Seja $\Delta'_{LFI1^*}(\mathcal{M}_2(n_2))$ a teoria que se obtém ao substituir a lógica subjacente da teoria $\Delta'_{LPC^*}(\mathcal{M}_2(n_2))$, do Exemplo 1.7, pela lógica LFI1*. As conseqüências de $\Delta'_{LFI1^*}(\mathcal{M}_2(n_2))$, que interpretadas sob a interpretação intencional I (Seção 1.2.1), permitem definir o processo de computação de $\mathcal{M}_2(n_2)$ são:*

- *Instante de tempo $t = 0$:*

$$\Delta'_{LFI1^*}(\mathcal{M}_2(n_2)) \vdash Q_1(0,0) \wedge S_1(0.0) \wedge \forall y\,(y \neq 0 \rightarrow S_0(0,y))\,. \quad (\text{An}_2)$$

  (An_2) é axioma de $\Delta'_{LPC^*}(\mathcal{M}_2(n_2))$.

- *Instante de tempo $t = 1$:*

$$\Delta'_{LFI1^*}(\mathcal{M}_2(n_2)) \vdash Q_1(0',0'). \quad (\text{Con1})$$
$$\Delta'_{LFI1^*}(\mathcal{M}_2(n_2)) \vdash Q_1(0',-1). \quad (\text{Con2})$$
$$\Delta'_{LFI1^*}(\mathcal{M}_2(n_2)) \vdash \neg Q_1(0'.0'), \quad (\text{Con3})$$
$$\Delta'_{LFI1^*}(\mathcal{M}_2(n_2)) \vdash \neg Q_1(0',-1), \quad (\text{Con4})$$
$$\Delta'_{LFI1^*}(\mathcal{M}_2(n_2)) \vdash \bullet Q_1(0'.0'), \quad (\text{Con5})$$
$$\Delta'_{LFI1^*}(\mathcal{M}_2(n_2)) \vdash \bullet Q_1(0',-1), \quad (\text{Con6})$$
$$\Delta'_{LFI1^*}(\mathcal{M}_2(n_2)) \vdash S_0(0',-1) \wedge S_0(0',0'). \quad (\text{Con7})$$

  (Con1) é obtida de (An_2) e (Ai_1) (Exemplo 1.4), usando simplificação, instanciação e MP. (Con2) é obtida de maneira similar a (Con1). (Con3) é obtida de (Con2) e $(\text{Ax}\ (\mathcal{M}_1))$ (Exemplo 1.5), usando instanciação, adição, MP e simplificação. (Con4) é obtida de maneira similar a (Con3), usando (Con1) em vez de (Con2). (Con5) é obtida de (Con1), (Con3), (LFI1-3) e a equivalência $\bullet A \leftrightarrow (A \wedge \neg A)$ (que pode ser validada facilmente com as tabelas de verdade de **LFI1**, *portanto, pela completude $LFI1 \vdash \bullet A \leftrightarrow (A \wedge \neg A)$). (Con6) é obtida de maneira similar a (Con5). (Con7) é obtida de (An_2) e (Ai_1) (ou de (Ai_3)), aplicando instanciação, simplificação, MP e conjunção.*

  É de se notar que os axiomas correspondentes às instruções i_1 e i_3 ((Ai_1) e (Ai_3)) foram usados para o instante de tempo $t = 0$, o que indica que ambas instruções foram executadas simultâneamente, fazendo

com que $\mathcal{M}_2(n_2)$, no instante de tempo $t = 1$ ficasse simultaneamente no estado q_1 na posição -1 e no estado q_1 na posição 1, o que dá lugar a inconsistências na teoria $\Delta'_{LFII^*}(\mathcal{M}_2(n_2))$.

- *Instante de tempo $t = 2$:*

$$\Delta'_{LFII^*}(\mathcal{M}_2(n_2)) \vdash Q_2(0'', -1) \wedge S_1(0'', -1), \quad \text{(Con8)}$$
$$\Delta'_{LFII^*}(\mathcal{M}_2(n_2)) \vdash S_0(0'', -1), \quad \text{(Con9)}$$
$$\Delta'_{LFII^*}(\mathcal{M}_2(n_2)) \vdash Q_2(0'', 0') \wedge S_1(0'', 0'). \quad \text{(Con10)}$$
$$\Delta'_{LFII^*}(\mathcal{M}_2(n_2)) \vdash S_0(0'', 0'), \quad \text{(Con11)}$$
$$\Delta'_{LFII^*}(\mathcal{M}_2(n_2)) \vdash \neg Q_2(0'', -1) \wedge \neg S_1(0'', -1) \wedge \neg S_0(0'', -1), \quad \text{(Con12)}$$
$$\Delta'_{LFII^*}(\mathcal{M}_2(n_2)) \vdash \neg Q_2(0'', 0') \wedge \neg S_1(0'', 0') \wedge \neg S_0(0'', 0'), \quad \text{(Con13)}$$
$$\Delta'_{LFII^*}(\mathcal{M}_2(n_2)) \vdash \bullet Q_2(0'', -1) \wedge \bullet S_1(0'', -1) \wedge \bullet S_0(0'', -1). \quad \text{(Con14)}$$
$$\Delta'_{LFII^*}(\mathcal{M}_2(n_2)) \vdash \bullet Q_2(0'', 0') \wedge \bullet S_1(0'', 0') \wedge \bullet S_0(0'', 0'). \quad \text{(Con15)}$$

(Con8) *é obtida de* (Ai$_2$), *instanciando-se para $t = 0'$ e $x = 0'$, e* (Con7). *Note que, sob a interpretação intencional I, as instruções levam para o seguinte instante de tempo os valores das casas que não estão sendo modificadas. Para o caso de $\mathcal{M}_2(n_2)$, a execução da instrução i_2, na posição $x = 1$ e no instante de tempo $t = 1$, leva o símbolo s_0 da posição $x = -1$ para a mesma posição no instante de tempo $t = 2$. Por outro lado, a instrução i_2 também é executada na posição $x = -1$ no instante de tempo $t = 1$, escrevendo o símbolo s_1 na posição $x = -1$, ficando tanto o símbolo s_0 como o símbolo s_1 na posição $x = -1$ da fita, no instante de tempo $t = 2$. Uma situação análoga ocorre na posição $x = 1$, com a instrução i_2 executada na posição $x = -1$. De este modo, a execução da instrução i_2 na posição $x = 1$ está influindo no conteúdo no seguinte instante de tempo da posição $x = -1$, e a execução da instrução i_2 na posição $x = -1$ está influindo no conteúdo no seguinte instante de tempo da posição $x = 1$.*

Nenhum dos axiomas (Ai$_1$) - (Ai$_3$) *de $\Delta'_{LFII^*}(\mathcal{M}_2(n_2))$ podem ser usados para $t = 0''$, o que indica que $\mathcal{M}_2(n_2)$ pára no instante de tempo $t = 2$. A computação de $\mathcal{M}_2(n_2)$ obtida através da interpretação da teoria $\Delta'_{LFII^*}(\mathcal{M}_2(n_2))$ é apresentada na seguinte figura:*

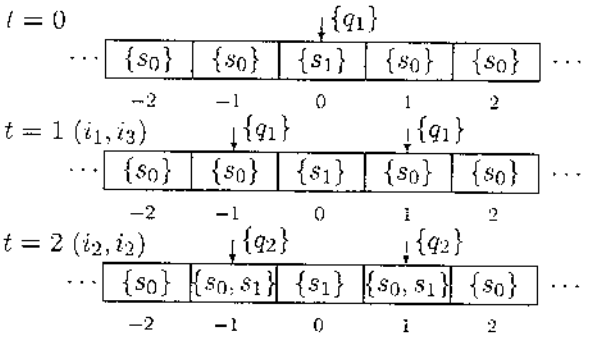

Figura 2.1: Computação sob a interpretação da teoria $\Delta'_{\text{LFI1*}}(\mathcal{M}_2(n_2))$

Exemplo 2.2. *Seja* $\mathcal{M}_3 = \langle Q, \Sigma, M, I\rangle$ *uma MTND tal que* $Q = \{q_1, q_2\}$, $\Sigma = \{s_0, s_1\}$ *e* $I = \{i_1, i_2, i_3\}$, *onde* $i_1 = q_1 s_1 s_0 q_2$, $i_2 = q_1 s_1 s_1 q_2$ *e* $i_3 = q_1 s_1 R q_1$. $\mathcal{M}_3$ *é representada gráficamente por:*

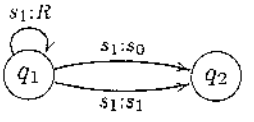

Figura 2.2: Grafo da máquina $\mathcal{M}_3$

Para a entrada $n_3 = s_1 s_1$ *a teoria* $\Delta'_{LFI1^*}(\mathcal{M}_3(n_3)) = \{(\text{A1}), \ldots, (\text{A5}), (\text{A}i_1), (\text{A}i_2), (\text{A}i_3), (\text{A}n_3)\}$, *onde* (A1) - (A5) *são os axiomas da Definição 1.9 e:*

$$\forall t \forall x \Big(\Big(Q_1(t,x) \wedge S_1(t,x) \Big) \rightarrow \Big(Q_2(t',x) \wedge S_0(t',x) \wedge$$

$$\forall y \Big((y \neq x) \rightarrow \big((S_0(t,y) \rightarrow S_0(t',y)) \wedge (S_1(t,y) \rightarrow S_1(t',y)) \big) \Big) \Big) \Big). \quad (\text{A}i_1)$$

$$\forall t \forall x \Big(\Big(Q_1(t,x) \wedge S_1(t,x) \Big) \rightarrow \Big(Q_2(t',x) \wedge S_1(t',x) \wedge$$

$$\forall y \Big((y \neq x) \rightarrow \big((S_0(t.y) \rightarrow S_0(t',y)) \wedge (S_1(t,y) \rightarrow S_1(t',y)) \big) \Big) \Big) \Big). \quad (\mathrm{Ai}_2)$$

$$\forall t \forall x \Big(\Big(Q_1(t.x) \wedge S_1(t.x) \Big) \rightarrow \Big(Q_1(t',x') \wedge$$

$$\forall y \Big(\big(S_0(t,y) \rightarrow S_0(t',y) \big) \wedge \big(S_1(t,y) \rightarrow S_1(t',y) \big) \Big) \Big) \Big). \quad (\mathrm{Ai}_3)$$

$$Q_1(0.0) \wedge \big(S_1(0,0) \wedge S_1(0,0') \big) \wedge \forall y \left((y \neq 0 \wedge y \neq 0') \rightarrow S_0(0,y) \right). \quad (\mathrm{An}_3)$$

As conseqüências de $\Delta'_{LFII^*}(\mathcal{M}_3(n_3))$, que interpretadas sob a interpretação intencional I (Seção 1.2.1), permitem definir o processo de computação de $\mathcal{M}_3(n_3)$ são:

- *Instante de tempo $t = 0$:* (An_3).
- *Instante de tempo $t = 1$:*

$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash Q_2(0',0) \wedge S_0(0',0) \wedge S_1(0',0). \quad (\mathrm{Con1})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash Q_1(0',0') \wedge S_1(0',0'), \quad (\mathrm{Con2})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash \neg Q_2(0'.0) \wedge \neg S_0(0'.0) \wedge \neg S_1(0'.0). \quad (\mathrm{Con3})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash \neg Q_1(0',0'). \quad (\mathrm{Con4})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash \bullet Q_2(0',0) \wedge \bullet S_0(0'.0) \wedge \bullet S_1(0',0) \wedge \bullet Q_1(0',0'). \quad (\mathrm{Con5})$$

- *Instante de tempo $t = 2$:*

$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash S_0(0'',0) \wedge S_1(0''.0), \quad (\mathrm{Con6})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash Q_2(0'',0') \wedge S_0(0'',0') \wedge S_1(0''.0'). \quad (\mathrm{Con7})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash Q_1(0''.0'') \wedge S_0(0'',0''), \quad (\mathrm{Con8})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash \neg S_0(0''.0) \wedge \neg S_1(0'',0), \quad (\mathrm{Con9})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash \neg Q_2(0''.0') \wedge \neg S_0(0'',0') \wedge \neg S_1(0'',0'), \quad (\mathrm{Con10})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash \neg Q_1(0''.0''). \quad (\mathrm{Con11})$$
$$\Delta'_{LFII^*}(\mathcal{M}_3(n_3)) \vdash \bullet S_0(0'',0) \wedge \bullet S_1(0'',0) \wedge \bullet Q_2(0'',0') \wedge \bullet S_0(0''.0') \wedge \bullet S_1(0'',0') \wedge \bullet Q_1(0'',0''). \quad (\mathrm{Con12})$$

Nenhum dos axiomas (Ai_1) - (Ai_3) *de* $\Delta'_{LFI1^*}(\mathcal{M}_3(n_3))$ *podem ser usados para* $t = 0''$, *o que indica que* $\mathcal{M}_3(n_3)$ *pára no instante de tempo* $t = 2$. *A computação de* $\mathcal{M}_3(n_3)$ *obtida através da interpretação da teoria* $\Delta'_{LFI1^*}(\mathcal{M}_3(n_3))$ *é apresentada na seguinte figura:*

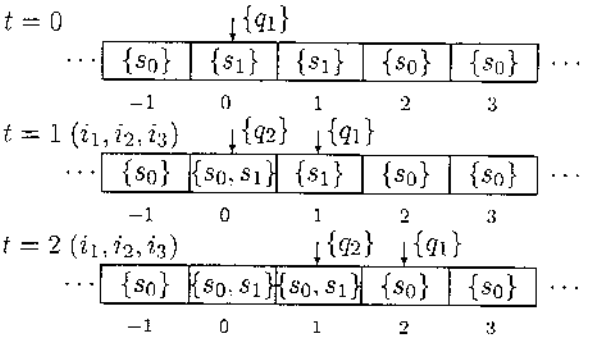

Figura 2.3: Computação sob a interpretação da teoria $\Delta'_{\mathrm{LFI1^*}}(\mathcal{M}_3(n_3))$

Definição 2.1 (Máquina de Turing paraconsistente (MTP)). *Uma* máquina de Turing paraconsistente *é uma máquina de Turing tal que:*

- *Permitem-se instruções contraditórias (instruções com os mesmos dois símbolos iniciais* $q_i s_j$*);*
- *Perante uma situação ambígua (situação na qual a máquina pode executar varias instruções) a máquina executa simultâneamente todas as possíveis instruções, dando lugar a multiplicidade de estados, posições e símbolos em algumas casas da fita;*
- *Cada instrução é executada numa posição específica da fita (onde um dos estados e um dos símbolos nessa casa da fita se correspondam com os primeiros dois símbolos da instrução), e cada instrução leva os símbolos atuais das posições da fita as quais ela não modifica para as mesmas posições no seguinte instante de tempo;*
- *Ao final da computação, cada posição da fita pode conter múltiplos símbolos, tal que cada escolha destes símbolos representa um resultado da computação.*

O que se vê da definição acima é que uma MTP pode produzir vários resultados para uma única entrada. Trazendo à baila a noção de *multifunção* de um conjunto A num conjunto B, definida como uma função

$f\colon A \rightarrow \mathcal{P}(B) - \{\emptyset\}$ (onde $\mathcal{P}(B)$ denota o conjunto de partes de B), pode-se dizer que uma MTP computa multifunções. Além disso, o conjunto de instruções de uma MTP pode ser expresso por meio de uma multifunção de um subconjunto de $Q \times \Sigma$ no conjunto $(\Sigma \cup M) \times Q$.

Os Exemplos 2.1 e 2.2 são exemplos de MTPs e suas respectivas computações.

Para descrever formalmente o conceito de computação de uma MTP é necessário adaptar as Definições 1.5 a 1.8, é isso que se faz a seguir.

Definição 2.2 (Descrição instantânea para uma MTP). *Seja $\mathcal{M} = \langle Q, \Sigma, M, I\rangle$ uma MTP, uma* descrição instantânea *α de $\mathcal{M}$ é uma sucessão finita de subconjuntos não vazios de Q e Σ, que contém pelo menos um subconjunto de Q e um subconjunto de Σ, e cujo último subconjunto da esquerda para direita é subconjunto de Σ.*

A título de ilustração, supondo que $\mathcal{M}$ no instante de tempo t encontra-se na situação representada na Figura 2.4, a descrição instantânea para descrever tal situação é $\alpha_t = \{q_{i_x}, \ldots, q_{k_x}\}\{s_{j_x}, \ldots, s_{l_x}\} \ldots \{q_{i_y}, \ldots, q_{k_y}\}\{s_{j_y}, \ldots, s_{l_y}\} \ldots \{s_{j_z}, \ldots, s_{l_z}\}$.

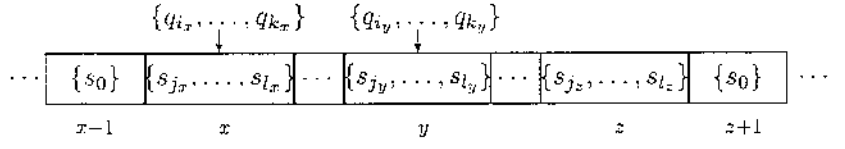

Figura 2.4: Situação MTP $\mathcal{M}$ no instante de tempo t

Definição 2.3 (Mudança de descrição instantânea relativa a uma instrução). *Sejam $\mathcal{M} = \langle Q, \Sigma, M, I\rangle$ uma MTP, α e β descrições instantâneas de $\mathcal{M}$, $\mathcal{Q}_i \subseteq Q$ tal que $q_i \in \mathcal{Q}_i$ e $\mathcal{S}_j \subseteq \Sigma$ tal que $s_j \in \mathcal{S}_j$. Dizemos que há uma* mudança de α para β relativa à instrução i_n *($i_n \in I$), denotada por $\alpha \xrightarrow{\mathcal{M}i_n} \beta$, quando existem sucessões finitas (possivelmente vazias) de conjuntos de estados $\mathcal{Q}_l \subseteq Q$ ($\mathcal{Q}_l \neq \emptyset$) e conjuntos de símbolos de entrada/saída $\mathcal{S}_k \subseteq \Sigma$ ($\mathcal{S}_k \neq \emptyset$), denotadas por P e Q, tais que P' e Q' obtém-se eliminando os conjuntos de estados de P e Q respectivamente e alguma das seguintes condições é satisfeita:*

- *$\alpha = P\mathcal{Q}_i\mathcal{S}_j Q$, $\beta = P'\{q_l\}\{s_k\}Q'$ e i_n é uma instrução do tipo* (I)*;*
- *$\alpha = P\mathcal{Q}_i\mathcal{S}_j\mathcal{S}_k Q$, $\beta = P'\mathcal{S}_j\{q_l\}\mathcal{S}_k Q'$ e i_n é uma instrução do tipo* (II)*;*

- $\alpha = PQ_iS_j$, $\beta = P'S_j\{q_l\}\{s_0\}$ *e* i_n *é uma instrução do tipo* (II);
- $\alpha = PS_kQ_iS_jQ$, $\beta = P'\{q_l\}S_kS_jQ'$ *e* i_n *é uma instrução do tipo* (III);
- $\alpha = Q_iS_jQ$, $\beta = \{q_l\}\{s_0\}S_jQ'$ *e* i_n *é uma instrução do tipo* (III).

Definição 2.4 (Mudança de descrição instantânea para uma MTP). *Seja* $\mathcal{M}$ *uma MTP, e sejam* α, β, β_1, β_2, ... β_k *descrições instantâneas para MTPs tais que* $\alpha \xrightarrow{\mathcal{M}i_{n_1}} \beta_1$, $\alpha \xrightarrow{\mathcal{M}i_{n_2}} \beta_2$, ..., $\alpha \xrightarrow{\mathcal{M}i_{n_k}} \beta_k$ *são todas as mudanças de descrições instantâneas de* α *relativas as instruções* i_n *de* $\mathcal{M}$. *Dizemos que há uma* mudança de α para β, *denotada por* $\alpha \xrightarrow{\mathcal{M}} \beta$, *quando* β *é a união (posição a posição) dos respectivos conjuntos de estados e símbolos de entrada/saída das descrições instantâneas* β_i, *com* $1 \leq i \leq k$.

Definição 2.5 (Descrição instantânea terminal para uma MTP). *Seja* $\mathcal{M}$ *uma MTP; uma descrição instantânea* α *(para uma MTP) é* terminal *com respeito a* $\mathcal{M}$ *se não existe uma descrição instantânea* β *tal que* $\alpha \xrightarrow{\mathcal{M}} \beta$.

Definição 2.6 (Computação de uma MTP). *Uma* computação de uma MTP $\mathcal{M}$ *é uma seqüência finita de descrições instantâneas para MTPs* $\alpha_0\alpha_1 \ldots \alpha_n$, *tal que* $\alpha_i \xrightarrow{\mathcal{M}} \alpha_{i+1}$, *para* $0 \leq i \leq n-1$, *e* α_n *é uma descrição instantânea terminal com respeito a* $\mathcal{M}$. *Nesse caso escreve-se* $\alpha_n = Res_{\mathcal{M}}(\alpha_0)$ *e diz-se que* α_n *é o resultado de* $\mathcal{M}$ *com respeito a* α_0. *Os resultados da computação obtém-se eliminando-se de* α_n *os conjuntos de estados e fazendo a escolha dos conjuntos restantes (conjuntos de símbolos de entrada/saída).*

Exemplo 2.3. *Seja* $\mathcal{M}_2$ *a máquina de Turing do Exemplo 2.1, a seqüência de descrições instantâneas* $\alpha_0\alpha_1\alpha_2$, *onde* $\alpha_0 = \{q_1\}\{s_1\}$, $\alpha_1 = \{q_1\}\{s_0\}\{s_1\}\{q_1\}\{s_0\}$ *e* $\alpha_2 = \{q_2\}\{s_0, s_1\}\{s_1\}\{q_2\}\{s_0, s_1\}$, *descreve formalmente a computação representada gráficamente na Figura 2.1. Neste exemplo,* α_2 *é o resultado de* $\mathcal{M}_2$ *com respeito a* α_0. *Os resultados da computação são* $s_0s_1s_0$, $s_0s_1s_1$, $s_1s_1s_0$ *e* $s_1s_1s_1$.

Note que $\alpha_1 \xrightarrow{\mathcal{M}_2i_2} \alpha_{2_1}$, *onde* $\alpha_{2_1} = \{q_2\}\{s_1\}\{s_1\}\{s_0\}$; *que* $\alpha_1 \xrightarrow{\mathcal{M}_2i_2} \alpha_{2_2}$, *onde* $\alpha_{2_2} = \{s_0\}\{s_1\}\{q_2\}\{s_1\}$; *e que* α_2 *é a união (posição a posição) dos respectivos conjuntos de estados e símbolos de entrada/saída de* α_{2_1} *e* α_{2_2}.

2.2.1 Representabilidade das MTPs

Como descrito anteriormente, os modelos para interpretar teorias com **LFI1*** como lógica subjacente são modelos clássicos aos quais foram acrescentadas constantes não estándar ✓ e ×, adaptados de forma a incluir tais

constantes na interpretação de predicados e funções. Com base nesta nova definição de modelo adapta-se a definição de estrutura matemática para expressar a computação de $\mathcal{M}(n)$ (Definição 1.10), e as definições de representação de funções, relações e computações numa teoria (Definições 1.11 a 1.13). Com as novas definições é fácil (ainda que trabalhoso e repleto de detalhes) adaptar as demonstrações dos Teoremas 1.8 a 1.12 para demonstrar que as teorias $\Delta'_{\mathrm{LFII}^*}(\mathcal{M}(n))$ representam as computações das MTPs respectivas.

Definição 2.7 (Estrutura que expressa uma computação numa MTP). *Seja $\mathcal{M}$ uma MTP, n o dado de entrada para $\mathcal{M}$, $Q_l \subseteq Q$ um conjunto de estados de $\mathcal{M}$, $S_k \subseteq \Sigma$ um conjunto de símbolos de entrada/saída de $\mathcal{M}$ e $\alpha_0\alpha_1\alpha_2\ldots$ a seqüência (possivelmente infinita) de descrições instantâneas de $\mathcal{M}(n)$, tal que α_0 descreve a situação inicial de $\mathcal{M}(n)$ e $\alpha_i \xrightarrow{\mathcal{M}} \alpha_{i+1}$ ($i = 0, 1, 2\ldots$). A estrutura que expressa matematicamente a computação de $\mathcal{M}(n)$ é a seguinte: $\mu(\mathcal{M}(n)) = \langle A, Q_1^\mu, Q_2^\mu, \ldots, Q_n^\mu, S_0^\mu, S_1^\mu, \ldots, S_{m-1}^\mu, <^\mu, {}'^\mu, 0^\mu\rangle$, onde:*

- $A = \mathbb{Z} \cup \{\checkmark, \times\}$,
- $Q_i^\mu = \{(t, x, \checkmark) \mid q_i \in Q_l, Q_l$ *é um singleton, e* Q_l *está em* α_t *antes do conjunto de símbolo de entrada/saída correspondente à posição* x *da fita* $(\{s_{j_x}, \ldots, s_{l_x}\})\} \cup \{(t, x, \times) \mid q_i \in Q_l, Q_l$ *tem mais de um elemento, e* Q_l *está em* α_t *antes do conjunto de símbolo de entrada/saída correspondente à posição* x *da fita* $(\{s_{j_x}, \ldots, s_{l_x}\})\}$,
- $S_j^\mu = \{(t, x, \checkmark) \mid s_j \in S_k, S_k$ *é um singleton, e* S_l *está em* α_t *e é o conjunto de símbolo de entrada/saída correspondente à posição* x *da fita* $(\{s_{j_x}, \ldots, s_{l_x}\})\} \cup \{(t, y, \checkmark) \mid j = 0$, *existe descrição instantânea para o instante de tempo* t *(*α_t*) e* y *é uma posição não descrita em* $\alpha_t\} \cup \{(t, x, \times) \mid s_j \in S_k, S_k$ *tem mais de um elemento, e* S_k *está em* α_t *e é o conjunto de símbolo de entrada/saída correspondente à posição* x *da fita* $(\{s_{j_x}, \ldots, s_{l_x}\})\}$,
- $<^\mu = \{(x, y, \checkmark) \mid x < y$, *onde* $<$ *é a relação "menor que" em* $\mathbb{Z}$,
- ${}'^\mu$: $\mathbb{Z} \to \mathbb{Z} \times \{\checkmark, \times\}$ *tal que* $x'^\mu = (x', \checkmark)$, *onde* $'$ *é a função sucessor em* $\mathbb{Z}$,
- $0^\mu = 0$.

Definição 2.8 (Representação de uma função numa teoria LFI1*). *Seja f uma função de k variáveis, Δ uma teoria arbitraria com* **LFI1*** *como lógica subjacente, $\varphi(x_1, \ldots, x_k, x)$ uma fbf em Δ, com $k+1$ variáveis livres. Diz-se que f* é representada por φ em Δ *se:*

1. $f(m_1, \ldots, m_k) = (n, \checkmark)$ *implica:*

   (a) $\Delta \vdash \varphi(\bar{m}_1, \ldots, \bar{m}_k, \bar{n})$,

   (b) *Se* $n \neq p$ *então* $\Delta \vdash \neg\varphi(\bar{m}_1, \ldots, \bar{m}_k, \bar{p})$, *e*

   (c) $\Delta \vdash \varphi(\bar{m}_1, \ldots, \bar{m}_k, \bar{q}) \rightarrow \bar{q} = \bar{n}$.

   Onde os símbolos $\bar{m}$ indicam numerais unários; e

2. $f(m_1, \ldots, m_k) = (n, \times)$ *implica* $\Delta \vdash \varphi(\bar{m}_1, \ldots, \bar{m}_k, \bar{n}) \wedge \neg\varphi(\bar{m}_1, \ldots, \bar{m}_k, \bar{n})$.

Diz-se que f é representável em Δ *se existir alguma fbf que a representa. Será dito também que o símbolo de função $\mathcal{F}$* representa a função f em Δ *se $\mathcal{F}(x_1, \ldots, x_k) = x$ representa a f em Δ.*

Definição 2.9 (Representação de uma relação numa teoria LFI1*). *Seja R uma relação de aridade k, Δ uma teoria arbitrária com* **LFI1*** *como lógica subjacente e $\varphi(x_1, \ldots, x_k)$ uma fbf em Δ, com k variáveis livres. Diz-se que R* é representada por φ em Δ *se:*

1. $(m_1, \ldots, m_k, \checkmark) \in R$ *implica* $\Delta \vdash \varphi(\bar{m}_1, \ldots, \bar{m}_k)$,

2. $(m_1, \ldots, m_k, \times) \in R$ *implica* $\Delta \vdash \varphi(\bar{m}_1, \ldots, \bar{m}_k) \wedge \neg\varphi(\bar{m}_1, \ldots, \bar{m}_k)$, *e*

3. $(m_1, \ldots, m_k, \checkmark) \notin R$ *e* $(m_1, \ldots, m_k, \times) \notin R$ *implica* $\Delta \vdash \neg\varphi(\bar{m}_1, \ldots, \bar{m}_k)$.

Onde os símbolos $\bar{m}$ indicam novamente numerais unários.

Diz-se que R é representável em Δ *se existir alguma fbf que a representa. Será dito também que o símbolo de predicado $\mathcal{R}$* representa a relação R em Δ *se $\mathcal{R}(x_1, \ldots, x_k)$ representa a R em Δ.*

Definição 2.10 (Representação de uma computação numa teoria LFI1*). *Seja $\mathcal{M}$ uma MTP, n o dado e entrada para $\mathcal{M}$ e $\mu(\mathcal{M}(n))$ a estrutura que expressa matematicamente a computação de $\mathcal{M}(n)$. Uma teoria Δ, na linguagem $\mathcal{L} = \{Q_1, Q_2, \ldots, Q_n, S_0, S_1, \ldots, S_{m-1}, <, ', 0\}$, com* **LFI1*** *como lógica subjacente, representa a computação de $\mathcal{M}(n)$ se em Δ cada símbolo de predicado Q_i representa a respectiva relação Q_i^{μ} de $\mu(\mathcal{M}(n))$.*

cada símbolo de predicado S_j representa a respectiva relação S_j^μ de $\mu(\mathcal{M}(n))$. o símbolo de predicado $<$ representa a relação $<^\mu$ de $\mu(\mathcal{M}(n))$ e o símbolo de função $'$ representa a função $'^\mu$ de $\mu(\mathcal{M}(n))$.

Teorema 2.1. *As teorias $\Delta'_{LFI1^*}(\mathcal{M}(n))$ representam a computação da MTP $\mathcal{M}(n)$ respectiva.*

Demonstração. Devem ser demonstrados teoremas similares aos Teoremas 1.8 a 1.11 usando as definições anteriores. As demonstrações dos novos teoremas são similares as demonstrações dos Teoremas 1.8 a 1.11, mas se devem ter em conta as novas constantes ✓ e ✗. Além disso, se deve ter em conta que as regras de contraposição e silogismos disjuntivo (usadas na demonstração do Teorema 1.2, conseqüentemente também na demonstração do Teorema 1.8) não são válidas em **LFI1***, portanto tem que se fazer uso de outras propriedades da lógica **LFI1***, como as enunciadas em [20, Prop. 2.3]. □

2.2.2 Condições de consistência/inconsistência nas MTPs

Tendo em conta que **LFI1*** internaliza (pelo menos parte da essencialidade das) noções de consistência e inconsistência na linguagem objeto, podem-se usar os operadores de consistência ($\circ \equiv \neg\bullet$) e inconsistência ($\bullet$) para impor condições na execução de instruções. Estes operadores podem ser acrescentados no antecedente dos axiomas correspondentes às instruções, no símbolo de predicado Q_i ou no símbolo de predicado S_j. Tem-se então 9 possíveis combinações de condições de consistência/inconsistência nos antecedentes dos axiomas correspondentes a instruções:

$$
\begin{array}{lll}
Q_i(t,x) \wedge S_j(t,x), & \circ Q_i(t,x) \wedge S_j(t,x), & \bullet Q_i(t,x) \wedge S_j(t,x), \\
Q_i(t,x) \wedge \circ S_j(t,x), & \circ Q_i(t,x) \wedge \circ S_j(t,x), & \bullet Q_i(t,x) \wedge \circ S_j(t,x), \\
Q_i(t,x) \wedge \bullet S_j(t,x). & \circ Q_i(t,x) \wedge \bullet S_j(t,x), & \bullet Q_i(t,x) \wedge \bullet S_j(t,x).
\end{array}
$$

Nas teorias $\Delta'_{\mathrm{LFI1}^*}(\mathcal{M}(n))$ as inconsistências nos predicados $Q_i(t,x)$ são produzidos devido à dedução de múltiplos $Q_i(t,x)$, para o mesmo instante de tempo t e possivelmente para diferentes posições x; as inconsistências dos predicados $S_j(t,x)$ são produzidas devido à dedução de múltiplos $S_j(t,x)$, para o mesmo instante de tempo t e para a mesma posição x. Portanto, as condições de consistência/inconsistência nos predicados $Q_i(t,x)$ e $S_j(t,x)$ correspondem nas MTPs, respectivamente, a condições de unicidade/multiplicidade de estados e símbolos de leitura/escrita. Para poder ter

controle das inconsistências e tirar maior proveito do modelo de MTPs, vai-se permitir então acrescentar condições de consistência/inconsistências aos primeiros dois símbolos das instruções das MTPs. O símbolo $^\circ$ será usado para indicar a condição de consistência (unicidade), enquanto que o símbolo $^\bullet$ será usado para indicar a condição de inconsistência (multiplicidade). Estes símbolos serão escritos logo após o primeiro símbolo da instrução, se a condição se referir ao estado, ou logo após o segundo símbolo da instrução, se a condição se referir ao símbolo de leitura/escrita.

Exemplo 2.4. *Seja $\mathcal{M}_4$ uma MTP com as instruções:*

$$i_1 = q_1 s_1^\circ s_0 q_1, \qquad i_2 = q_1^\circ s_1 s_1 q_2, \qquad i_3 = q_1 s_1^\bullet R q_3,$$
$$i_4 = q_2^\bullet s_0^\bullet R q_2, \qquad i_5 = q_3 s_0 s_1 q_3.$$

Para o dado de entrada $n_4 = s_1$, a computação de $\mathcal{M}_4(n_4)$ é representada pela seguinte figura:

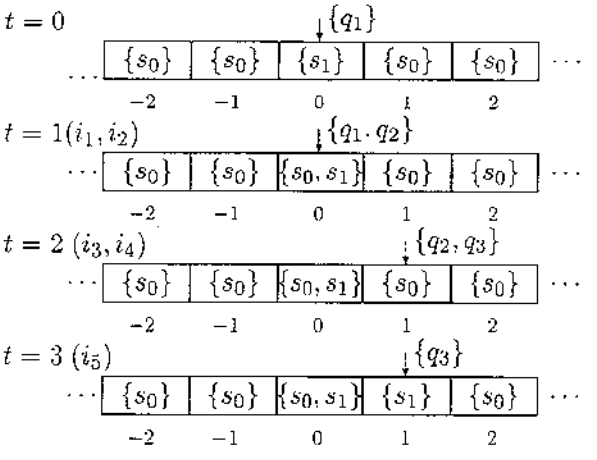

Figura 2.5: Computando numa MTP com condições de consistência

No instante de tempo $t = 0$, $\mathcal{M}_4(n_4)$ executa a instrução i_1 na posição 0 da fita, pois nessa posição a máquina está no estado q_1, lendo o símbolo s_1 e s_1 é o único símbolo nessa posição. Também no instante de tempo $t = 0$, $\mathcal{M}_4(n_4)$ executa a instrução i_2 na posição 0 da fita, pois nessa posição a máquina está no estado q_1, lendo o símbolo s_1 e q_1 é o único estado nesse instante de tempo. No instante de tempo $t = 0$, $\mathcal{M}_4(n_4)$ não executa a instrução i_3, pois ainda que na posição 0 a máquina esteja no estado q_1

lendo o símbolo s_1, o símbolo s_1 é o único símbolo nessa posição e i_3 exige estar lendo múltiplos símbolos para que possa ser executada. No instante de tempo $t = 1$, $\mathcal{M}_4(n_4)$ executa a instrução i_3 na posição 0 da fita, pois nessa posição a máquina está no estado q_1, lendo o símbolo s_1 e s_1 não é o único símbolo nessa posição. Também no instante de tempo $t = 1$, $\mathcal{M}_4(n_4)$ executa a instrução i_4 na posição 0 da fita, pois nessa posição a máquina está no estado q_2, lendo o símbolo s_0, q_1 não é o único estado nesse instante de tempo e s_0 não é o único símbolo nessa posição. No instante de tempo $t = 1$, $\mathcal{M}_4(n_4)$ não executa as instruções i_1 nem i_2, pois ainda que na posição 0 esteja no estado q_1 lendo o símbolo s_1, o símbolo s_1 não é o único símbolo nessa posição e q_1 não é o único estado nesse instante de tempo. Finalmente, no instante de tempo $t = 2$, $\mathcal{M}_4(n_4)$ executa a instrução i_5 que não exige nenhuma condição de consistência/inconsistência.

2.2.3 Outros possíveis modelos de MTPs

O modelo de MTPs definido anteriormente, apenas substituindo a lógica subjacente das teorias $\Delta'_{LPC}(\mathcal{M}(n))$ pela lógica **LFI1*** e interpretando adequadamente tais teorias, ostenta a característica de que as instruções, quando executadas, levam os símbolos presentes nas posições da fita que ela não modifica para as mesmas posições no seguinte instante de tempo. Esta característica pode resultar indesejável para alguns propósitos computacionais, e pode ainda parecer ir contra as noções intuitivas de procedimento algorítmico, devido ao fato de que uma instrução executada numa determinada posição pode influenciar o conteúdo, no seguinte instante de tempo, de uma quantidade arbitrária (porém finita) de casas da fita, as quais podem estar arbitrariamente (porém finitamente) distantes. Pode-se então definir um novo modelo de máquinas de Turing simplesmente eliminando-se tal característica da definição de MTP; com esse procedimento, contudo não se teria justificação suficiente para chamar o novo modelo de modelo de MTPs. Porém, analisando-se as causas do surgimento da característica em questão, pode-se ver que esta é devida à execução simultânea de instruções e a que no método de axiomatização definido para MTDs os axiomas correspondentes as instruções da máquina, além de axiomatizar a operação realizada pela instrução, indicam que os símbolos que a instrução não modifica são preservados.

É possível modificar o método de axiomatização para MTDs, de forma a eliminar dos axiomas correspondentes a instruções a parte que indica a preservação dos símbolos não modificados pela instrução; a idéia é substituir

os esquemas (Ai_j (I)), (Ai_j (II)) e (Ai_j (III)) pelos esquemas seguintes:

$$\forall t \forall x \left(\Big(Q_i(t,x) \wedge S_j(t,x) \Big) \rightarrow \Big(Q_l(t',x) \wedge S_k(t',x) \Big) \right), \qquad (\mathrm{A}i'_j \text{ (I)})$$

$$\forall t \forall x \left(\Big(Q_i(t,x) \wedge S_j(t,x) \Big) \rightarrow Q_l(t',x') \right), \qquad (\mathrm{A}i'_j \text{ (II)})$$

$$\forall t \forall x \left(\Big(Q_i(t,x') \wedge S_j(t,x') \Big) \rightarrow Q_l(t',x) \right); \qquad (\mathrm{A}i'_j \text{ (III)})$$

e acrescentar um novo axioma para estabelecer tal preservação enquanto a máquina esteja em execução, sem depender de nenhuma instrução em particular, por meio do esquema:

$$\forall t \forall x \Bigg(\Bigg(\Big(\bigwedge_{i=1}^{n} \big(\neg Q_i(t,x) \wedge \neg \bullet Q_i(t,x) \big) \Big) \wedge \Big(\bigvee_{q_i s_j \ inic.} \big(Q_i(t,x) \wedge S_j(t,x) \big) \Big) \Bigg) \rightarrow \big(S_j(t,x) \rightarrow S_j(t',x) \big) \Bigg), \qquad (2.20)$$

Dessa forma obtém-se teorias que ainda continuam representando as computações das MTDs correspondentes. Com tais teorias pode-se levar a cabo o mesmo processo de substituição da lógica subjacente pela lógica **LFI1***, de definição de modelo de MTPs a partir da interpretação de tais teorias, e de demonstração de que as computações de tais modelos são representadas formalmente pelas respectivas teorias. Assim, pode se construir um novo modelo de MTPs onde as instruções não levam os símbolos não modificados para o seguinte instante de tempo, modelo este que teria suficiente justificação para ser chamado de modelo de MTPs.

Outra característica que pode resultar indesejável no modelo de MTPs é que os resultados da execução de diferentes instruções podem se misturar indiscriminadamente. Para ilustrar isso, pode-se pensar por exemplo numa MTP $\mathcal{M}$ com as instruções $i_n = q_i s_j s_k q_l$ e $i_m = q_i s_j s_{k'} q_{l'}$ (com $k \neq k'$ e $l \neq l'$). Quando $\mathcal{M}$ chegar a uma situação na qual na posição x esteja no estado q_i, lendo o símbolo s_j, a máquina vai executar i_n e i_m simultâneamente. A instrução i_n vai fazer com que $\mathcal{M}$, na posição x, escreva o símbolo s_k e passe ao estado q_l, enquanto a instrução i_m vai fazer com que $\mathcal{M}$, na posição x, escreva o símbolo $s_{k'}$ e passe ao estado $q_{l'}$. No seguinte instante de tempo todas as possíveis combinações dos estados q_l e $q_{l'}$ com os símbolos s_k e $s_{k'}$ vão poder ser considerados para a execução de

instruções, isto é, qualquer instrução que começar por $q_l s_k$, $q_l s_{k'}$, $q_{l'} s_k$ ou $q_{l'} s_{k'}$ vai ser executada. Note que os resultados das instruções i_n e i_m acabaram sendo misturados indiscriminadamente. Esta característica é devida ao fato de que na lógica **LFI1*** são validas as regras de simplificação (isto é, $\vdash_{\mathrm{LFI1^*}} A \wedge B$ implica que $\vdash_{\mathrm{LFI1^*}} A$ e $\vdash_{\mathrm{LFI1^*}} B$, por (LFI1-4), (LFI1-5) e MP) e adjunção (isto é, $\vdash_{\mathrm{LFI1^*}} A$ e $\vdash_{\mathrm{LFI1^*}} B$ implica $\vdash_{\mathrm{LFI1^*}} A \wedge B$, por LFI1-3 e MP). Para a MTP $\mathcal{M}$ com as instruções i_n e i_m descritas anteriormente, se $\Delta'_{\mathrm{LFI1^*}}(\mathcal{M}(n)) \vdash Q_i(t,x) \wedge S_j(t,x)$ então do axioma correspondente à instrução i_n, com MP e simplificação obtém-se que $\Delta'_{\mathrm{LFI1^*}} \vdash Q_l(t',x) \wedge S_k(t',x)$. Similarmente, com o axioma correspondente à instrução i_m obtém-se que $\Delta'_{\mathrm{LFI1^*}}(\mathcal{M}(n)) \vdash Q_{l'}(t',x) \wedge S_{k'}(t',x)$. Portanto, usando simplificação e adjunção pode-se deduzir qualquer combinação dos predicados $Q_l(t',x)$ e $Q_{l'}(t',x)$ com os predicados $S_l(t',x)$, $S_{l'}(t',x)$. Se na definição do modelo de MTPs, em lugar de substituir-se a lógica subjacente das teorias $\Delta'_{\mathrm{LFI1^*}}(\mathcal{M}(n))$ pela lógica **LFI1*** se substitui-se por uma lógica paraconsistente "não adjuntiva" (isto é, uma lógica na qual não é valida a regra de adjunção), como por exemplo a lógica "pré-discursiva" **J** de Jaśkowski (cf. [56]) estendida a primeira ordem, a mistura indiscriminada dos resultados da execução de diferentes instruções seria evitada. Denotando por J^* à extensão a primeira ordem da lógica "pré-discursiva" de Jaśkowski, para a máquina $\mathcal{M}$ com as instruções i_n e i_m com que viemos trabalhando, $\Delta'_{J^*}(\mathcal{M}(n)) \vdash Q_i(t,x) \wedge S_j(t,x)$ implica que $\Delta'_{J^*}(\mathcal{M}(n)) \vdash Q_l(t',x) \wedge S_k(t',x)$ e que $\Delta'_{J^*}(\mathcal{M}(n)) \vdash Q_{l'}(t',x) \wedge S_{k'}(t',x)$, mas não implica que $\Delta'_{J^*}(\mathcal{M}(n)) \vdash Q_l(t',x) \wedge S_{k'}(t',x)$ nem que $\Delta'_{J^*}(\mathcal{M}(n)) \vdash Q_{l'}(t',x) \wedge S_k(t',x)$. Com a lógica J^* pode-se então definir um novo modelo de MTPs onde os resultados da execução das diferentes instruções não se misturem indiscriminadamente.

Isso ilustra dramaticamente como o substrato lógico pode ter influência na computação, e como de certa maneira a noção de computação é relativa à lógica subjacente ao modelo de computação.

Além dos novos modelos de MTPs descritos anteriormente, podem-se construir outros modelos de MTPs usando lógicas paraconsistentes diferentes, ou mesmo fazendo-se modificações ao método de axiomatização de máquinas de Turing. No capítulo de considerações finais apresenta-se uma análise desse universo de possibilidades.

Capítulo 3

Computação quântica

A *computação quântica* não é meramente uma nova tecnologia para prover maior velocidade e eficiência aos computadores; nem é o mero suporte do meio físico com características quânticas, uma vez que a tendência tecnológica à miniaturização vai inevitavelmente na direção de fenômenos quânticos, no sentido em que pode-se argumentar que todo processo físico é quântico, e este aspecto é realçado à medida em que se diminui a escala dos componentes tecnológicos. Algo de quântico já existe na própria noção de nanotecnologia. Computação quântica é a teoria da computação baseada nos princípios conceituais da mecânica quântica, principalmente a interferência, a superposição de estados e o emaranhamento de estados, de forma a permitir executar tipos de computação radicalmente novos que seriam impossíveis em qualquer máquina de Turing, e conseqüentemente em qualquer computador clássico.

Paul Benioff, em 1980, foi o primeiro a utilizar os princípios da mecânica quântica na descrição física de máquinas de Turing (ver [8]). No modelo de Benioff a máquina encontra-se num estado intrinsecamente quântico (estado de *superposição*, explicado na Seção 3.1) somente entre cada passo de computação, no fim de cada passo a fita da máquina volta a um estado clássico (uma seqüência de *bits*), portanto este modelo não consegue mais do que simular o comportamento de uma máquina de Turing reversível.[1] Por tal razão este modelo não é aceito como um verdadeiro modelo de computação quântica.

A idéia de que os efeitos da computação quântica poderiam prover

[1] Uma das propriedades da computação quântica é a *reversibilidade*, a qual será explicada nas próximas seções. É preciso esclarecer também que toda máquina de Turing pode ser transformada numa máquina de Turing reversível, como demonstrado por Charles Bennet em [9].

um maior poder computacional foi sugerida por Richard Feynman, num workshop de física e computação em 1981 (Feynman publicou um artigo com estas idéias em [44]).[2] Feynman formulou a pergunta de se os fenômenos da mecânica quântica podem ser simulados eficientemente numa máquina de Turing clássica, e apresentou boas razões para acreditar que a resposta seria negativa, argumentando que tal simulação parece impossível sem incorrer num retardo exponencial. Além disso, especulou que para resolver o problema seriam necessários computadores que funcionassem de acordo com as leis da mecânica quântica, mas não definiu um modelo apropriado.

David Deutsch foi quem formalizou a idéia de Feynman, definindo o modelo de *máquinas de Turing quânticas*[3] (MTQs) em 1985 (ver [36]) e o modelo de *circuitos quânticos* (CQs) em 1989 (ver [37]). Em [36], Deutsch também mostrou a existência de uma MTQ universal, mas com a limitação de estar submetida a um retardo exponencial na simulação de outra MTQ qualquer.

Ethan Bernstein e Umesh Vazirani, em 1993, definiram um modelo de MTQ universal que evita o retardo exponencial, e que na verdade pode simular qualquer outra MTQ em tempo polinomial (ver [10]).[4]

Também em 1993, Andrew Yao demonstrou a equivalência, quanto à complexidade algorítmica, entre os modelos de MTQs e CQs. Mais precisamente, Yao demonstrou que qualquer função computável em tempo polinomial numa MTQ pode ser computada por um CQ de tamanho polinomial (ver [85]). Este resultado legitima o trabalho em algoritmos quânticos, comumente implementados utilizando o modelo de CQs ao invés de programas que rodam sobre uma MTQ universal, o que é muito mais complicado.

O principal algoritmo quântico construído até hoje, que nos leva a acreditar fortemente nas potencialidades da computação quântica, é o algoritmo de fatoração de inteiros em tempo polinomial proposto por Peter Shor (ver [75, 76]), atacando um problema de crucial importância em criptografia.

Neste capítulo apresentam-se os modelos de MTQs (Seção 3.2) e de CQs (Seção 3.3). Para um melhor entendimento destes modelos de computação apresenta-se primeiro uma breve descrição de alguns dos princípios da mecânica quântica, a partir dos postulados da mecânica quântica.

[2] Yuri Manin em seu livro [61] já tinha apontado as possíveis vantagens de construir computadores sob os princípios da mecânica quântica, mas nesse momento a idéia não foi suficientemente divulgada.

[3] O modelo recebe o nome de *máquinas de Turing quânticas* por ser uma adaptação do modelo clássico de máquinas de Turing.

[4] Uma versão completa e melhorada deste artigo é [11].

3.1 Postulados e princípios da mecânica quântica

Como descrito na introdução, a formalização da mecânica quântica, tal como é hoje conhecida, é o resultado de duas formulações independentes: a *mecânica matricial* proposta por Werner Heisenberg em 1925 e a *mecânica ondulatória* proposta por Erwin Schrödinger em 1926 (cf. [51, p. 344]). Nesse mesmo ano, o próprio Schrödinger demonstrou a equivalência entre as duas formulações. Posteriormente as duas formulações foram imersas numa teoria mais geral e compreensível, em boa parte graças ao esforço de Paul Dirac.

A idéia de usar espaços de Hilbert na formalização da mecânica quântica, cujos elementos constituintes são números complexos, partiu de David Hilbert, John von Neumann e Lothar Nordheim em 1927 (cf. [54]), os dois últimos assistentes de Hilbert na época. De acordo com tal formalização, a física da mecânica quântica se reduz à matemática dos operadores lineares Hermitianos sobre espaços de Hilbert.

Uma questão que se coloca é por que escolher axiomaticamente o uso de números complexos na representação matemática das propriedades quânticas – seria possível derivar a natureza do objeto matemático inerente ao fenômeno quântico, ao invés de simplesmente postulá-lo? Esta questão aparentemente é inconclusa; o que parece ser o caso é que os números complexos são usados em mecânica quântica como um artefato conveniente para facilitar e simplificar os cálculos, mas é discutível se outras estruturas matemáticas (como por exemplo os corpos p-ádicos) poderiam ser usadas com igual ou maior sucesso.

Para detalhes históricos sobre os primórdios da mecânica quântica recomenda-se [55].

Na formalização de von Neumann-Dirac a mecânica quântica é apresentada por meio de postulados, e estes postulados oferecem respostas às seguintes questões: Como descrever o estado de um sistema físico? Como evolui um sistema enquanto ele não é observado? Como descrever as observações e seus efeitos? Como descrever o estado de sistemas compostos (sistemas constituídos por dois ou mais sistemas físicos distintos)? A seguir apresentam-se tais postulados (tal como aparecem em [24, Seç. 2.2]):

Postulado 1. *Todo sistema físico tem a si associado um espaço de Hilbert,*[5]

[5]No apêndice B apresentam-se alguns conceitos básicos de álgebra linear necessários à formulação da mecânica quântica. A definição de espaço de Hilbert de dimensão finita é a Definição B.13 (para os modelos de computação quântica apresentados não é necessário considerar espaços de Hilbert infinito-dimensionais). A definição de vetor unitário é a Definição B.18.

conhecido como espaço de estados *do sistema. O sistema é completamente descrito por seu* vetor de estado, *o qual é um vetor unitário no espaço de estados do sistema (cf. [24, p. 80]).*

O sistema quântico mais simples e de maior importância na computação quântica é o *qubit* (ou *bit quântico*), utilizado como a unidade básica de informação quântica, análogo ao *bit* na computação clássica. Formalmente, um qubit é um vetor unitário num espaço de estados bidimensional.

O estado geral de um qubit é usualmente definido usando a notação de Dirac, tal notação é utilizada no formalismo da mecânica quântica com o objetivo de simplificar a notação e facilitar o entendimento das equações. Como apresentado no Apêndice B (Definição B.15), na notação de Dirac os vetores são denotados por $|\cdot\rangle$ e são chamados de *kets* e os vetores duais (isto é, os transpostos conjugados dos kets) são denotados por $\langle\cdot|$ e são chamados de *bras*. Utilizando a notação de Dirac e sendo $|0\rangle$ e $|1\rangle$ uma base ortonormal (Definição B.19) do espaço de estados bidimensional, então, o estado geral de um qubit é definido pela equação:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{3.1}$$

onde α e β são números complexos. A condição de que o vetor ψ seja unitário (Definição B.18) é equivalente à condição de que $|\alpha|^2 + |\beta|^2 = 1$ (onde $|\alpha|$ denota o módulo do número complexo $\alpha = a + bi$, definido por $|\alpha| = \sqrt{\alpha\overline{\alpha}} = \sqrt{a^2 - b^2}$); tal condição é comumente chamada de *condição de normalização*. Os coeficientes α e β são denominados *amplitudes de probabilidade*. Comumente $|0\rangle$ e $|1\rangle$ são tomados como os vetores:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.2}$$

Outra maneira de escrever o vetor $|\psi\rangle$ é:[6]

$$|\psi\rangle = e^{i\gamma}\left(\cos\tfrac{\theta}{2}|0\rangle + e^{i\varphi}\operatorname{sen}\tfrac{\theta}{2}|1\rangle\right), \tag{3.3}$$

onde γ, θ e φ são números reais. Como o fator $e^{i\gamma}$ não tem efeitos observáveis,[7] $|\psi\rangle$ pode ser escrita como:

$$|\psi\rangle = \cos\tfrac{\theta}{2}|0\rangle + e^{i\varphi}\operatorname{sen}\tfrac{\theta}{2}|1\rangle. \tag{3.4}$$

[6] Uma explicação para a troca de coordenadas pode ser encontrada em [63, Ap. A]. Embora simples, essa explicação aparentemente não aparece na literatura mais comumente disponível.

[7] Em [24, p. 93] explica-se porque o fator $e^{i\gamma}$ não tem efeitos observáveis.

Tomando θ e φ como as coordenadas de um ponto numa esfera, ψ pode ser representada geometricamente como na Figura 3.1, comumente chamada de *esfera de Bloch*.

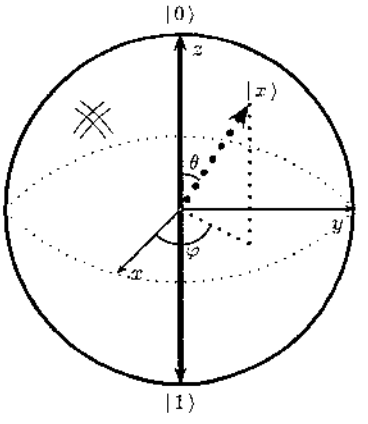

Figura 3.1: Representação do estado $|\psi\rangle$ na esfera de Bloch

Um qubit diferencia-se de um bit clássico pelo fato de que o qubit pode estar numa *superposição* de estados (dada pela equação (3.1)), o qual é fisicamente interpretado (usando a interpretação de múltiplos mundos da mecânica quântica (ver [82])) como a co-existência simultânea do qubit nos estados $|0\rangle$ e $|1\rangle$, fenômeno que não tem análogo clássico.

Existem vários referentes físicos que podem ser usados na implementação de um qubit; um deles é o estado de *spin*, ou direção do momento angular de uma partícula.

Postulado 2. *A evolução de um sistema quântico fechado é descrita por uma* transformação unitária, *isto é, o estado $|\psi\rangle$ de um sistema no tempo t_1 se relaciona com o estado $|\psi'\rangle$ do sistema no tempo t_2 por um operador unitário*[8] *U, o qual depende somente dos tempos t_1 e t_2,*

$$|\psi'\rangle = U|\psi\rangle \tag{3.5}$$

(cf. [24, p. 81]).

O postulado 2 descreve a evolução de um sistema quântico, enquanto não é observado, em intervalos de tempo discretos. Uma versão mais refinada deste postulado pode ser oferecida através da equação de Schrödinger, a qual

[8]Ver Definição B.24, com A^* a transporta conjugada de A.

descreve a evolução de um sistema quântico em tempo contínuo. A equação de Schrödinger e a derivação do postulado 2 (a partir de tal postulado) são apresentados em [24, p. 82-83]. Os modelos de MTQs e CQs são modelos discretos de computação, e é por isso que só se faz uso neste trabalho do postulado 2 na sua versão discreta.[9]

Um operador unitário U pode ser representado, com respeito à base B, por meio de uma matriz unitária M_U^B tal que, para os estados $|i\rangle, |j\rangle \in B$, $M_U^B[i,j]$ é a amplitude de transição do estado $|j\rangle$ ao estado $|i\rangle$. Além disso, como U é um operador unitário, tem a propriedade de que se $U|\psi\rangle = |\psi'\rangle$ então $|\psi\rangle = U^\dagger|\psi'\rangle$ (onde $U^\dagger$ representa a conjugada transposta de U), portanto, todo processo quântico é *reversível* e $U^\dagger$ é o processo inverso a U.

Exemplos de matrizes unitárias para operar sobre qubits são as chamadas de *matrizes de Pauli* [24, p. 65]:

$$\sigma_x \equiv X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \sigma_y \equiv Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \sigma_z \equiv Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{3.6}$$

Postulado 3. *Na mecânica quântica um* observável *é um operador auto-adjunto.*[10] *A saída numérica de uma medição, de um estado $|\psi\rangle$, com respeito a um observável A, é um dos auto-valores de A. O efeito colateral de tal medição é um* colapso *de $|\psi\rangle$ no estado $|\psi'\rangle$. Ao se realizar a medição, o auto-valor λ_i é obtido com probabilidade:*

$$Pr(\lambda_i) = \langle\psi|P_i|\psi\rangle \tag{3.7}$$

(onde P_i é o projetor no auto-espaço de A com respeito ao auto-valor λ_i, isto é, se $|\lambda_i\rangle$ é o auto-vetor associado ao auto-valor λ_i, então $P_i = |\lambda_i\rangle\langle\lambda_i|$). O novo estado $|\psi'\rangle$ no qual $|\psi\rangle$ colapsa é da forma:

$$|\psi'\rangle = \frac{P_i|\psi\rangle}{\sqrt{Pr(\lambda_i)}}. \tag{3.8}$$

Isto significa que a medição de $|\psi\rangle$ com respeito a A destrói irreversivelmente $|\psi\rangle$, a menos que $|\psi\rangle$ seja um auto-vetor de A (cf. [51, p. 26]).

Um *observável* é uma propriedade física mensurável, como por exemplo o *spin* de uma partícula, a posição ou o momento.

Diferentemente da mecânica clássica, na mecânica quântica a observação ou medição de um sistema influi nele de maneira implacável, fazendo-o mudar de estado de uma maneira não determinística. É esta uma das grandes

[9] Existem modelos de computação quântica contínua: ver por exemplo [60, 14, 84].

[10] As definições de operador auto-adjunto, auto-valor, auto-vetor e auto-espaço são apresentadas no apêndice B. Definições B.23 e B.20.

$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle, \tag{3.11}$$

e a condição de normalização é:

$$|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1. \tag{3.12}$$

Para um registro composto por n qubits (n-qubit) o espaço de estados é $H_{2^n} = \otimes^n H_2$ (onde $\otimes^n H_2$ representa o produto tensorial n vezes de H_2. O espaço H_{2^n} é de dimensão 2^n). Para os elementos da base $B = \{|\psi_i\rangle | 0 \leq i < 2^n\}$, o estado geral de um n-qubit é da forma:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle. \tag{3.13}$$

e a condição de normalização é:

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1. \tag{3.14}$$

Quando o estado de um sistema composto tem a propriedade de não poder ser escrito como produto tensorial dos sistemas que o compõem, diz-se que o sistema está num estado *emaranhado*. Por razões que ainda não são muito claras, os estados emaranhados desempenham um papel importante na computação quântica.[14] Um exemplo de estado emaranhado de um 2-qubit é:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}. \tag{3.15}$$

Se se supuser que existem qubits com estados $|\varphi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ e $|\gamma\rangle = \alpha'_0|0\rangle + \alpha'_1|1\rangle$ tais que $|\psi\rangle = |\varphi\rangle \otimes |\gamma\rangle$, então $|\psi\rangle = \alpha\alpha'|00\rangle + \alpha\beta'|01\rangle + \beta\alpha'|10\rangle + \beta\beta'|11\rangle$, logo ter-se-ia que $\alpha\alpha' = 1/\sqrt{2}$, $\alpha\beta' = 0$, $\beta\alpha = 0$ e $\beta\beta' = 1/\sqrt{2}$, o que não é possível, pois como $\alpha\beta' = 0$ então $\alpha = 0$ ou $\beta' = 0$, se $\alpha = 0$ então $\alpha\alpha' = 0$, o que contradiz que $\alpha\alpha' = 1/\sqrt{2}$; e se $\beta' = 0$

[13] $|ij\rangle$ é uma notação usada para abreviar $|i\rangle \otimes |j\rangle$, as vezes é usada também a notação $|i, j\rangle$.

[14] Em [51, p. 76] chega mesmo a afirmar que os estados emaranhados são a principal razão para que os computadores quânticos não possam ser simulados eficientemente pelos computadores clássicos.

então $\beta\beta' = 0$, o que contradiz que $\beta\beta' = 1/\sqrt{2}$. Portanto, $|\psi\rangle$ não pode ser expresso como produto tensorial de dois qubits.

Note que no estado $|\psi\rangle$ ambos os qubits estão em estados embaralhados, de tal forma que ao se realizar uma medição sobre qualquer um dos qubits, baseado no observável padrão, obtém-se o valor 0 com probabilidade $\frac{1}{2}$ ou o valor 1 também com probabilidade $\frac{1}{2}$. Porém, após a realização da medição sobre um dos dois qubits o sistema inteiro irá colapsar ao estado $|00\rangle$ se o valor 0 for obtido na medição, ou ao estado $|11\rangle$ se o valor 1 for obtido na medição. Portanto, ao medir o outro qubit vai-se obter o valor 0 com probabilidade 1, ou o valor 1 com probabilidade 1, respectivamente. Isto é, o valor a se obter na segunda medição (para um qubit) está completamente determinado pelo valor obtido na primeira medição (para o outro qubit); ou seja, os valores de dois qubits em um estado emaranhado estão intimamente relacionados.

Os estados emaranhados não têm equivalente na física clássica, e provocam uma boa quantidade de questões epistemológicas e de debate sobre o mundo quântico e sobre as interpretações e teorias que pretendem explicá-la. Por exemplo, uma primeira crítica é que a interpretação dos sistemas quânticos compostos por meio de produto tensorial em espaços de Hilbert seria semanticamente ambígua, e que novas entidades matemáticas que promovessem a desambiguidade seriam oportunas (cf. [46]).

Outro tipo de abordagem, conforme [16], prega que uma teoria quântica seria melhor compreendida como uma teoria sobre as possibilidades e impossibilidades da transferência de informação, de forma oposta à visão sobre partículas ou ondas. Dessa forma, um estado emaranhado poderia ser pensado como o análogo de um canal de comunicação não-clássico, um tipo de "fio condutor" não-clássico. Tais análogos de canais de comunicação poderiam ser usados para realizar coisas impossíveis classicamente, como teleportação e novas formas de computação. Uma teoria quântica seria então uma teoria sobre tais canais de comunicação, e sobre a representação e manipulação de estados como transmissores de informação. Aparentemente, pouco se sabe sobre a capacidade de informação de um canal quântico: diferentemente do clássico, parece haver mais de uma noção de capacidade de informação, dependendo dos meios auxiliares disponíveis, a classe de protocolos de transmissão de informação permitidos, e do fato de a informação a ser transmitida ser clássica ou quântica. De todo modo, este não é o tema do presente trabalho, embora essa visão a respeito da mecânica quântica seja coerente com nossa proposta de explorar o substrato lógico da computação quântica.

3.2 Máquinas de Turing quânticas

O modelo de MTQ é obtido *quantizando* os elementos do modelo de máquina de Turing clássica, isto é, trocando os elementos da descrição clássica de uma máquina de Turing por observáveis dentro de um sistema quântico (cf. [67, p. 3]).

Antes de realizar a quantização é preciso definir o que é uma *configuração da fita*: sendo $\Sigma = \{\sigma_0, \ldots, \sigma_{\overline{\Sigma}-1}\}$ o conjunto de símbolos de entrada-saída de uma máquina de Turing, uma configuração T da fita é uma seqüência infinita de símbolos de Σ. Assumindo que Σ contem um símbolo B para representar uma casa vazia na fita e denotando o símbolo na posição m da fita por $T(m)$ (para todo inteiro m), toda configuração T da fita deve cumprir com a condição que exige que $T(m) = B$ exceto para um número finito de casas m.

Para realizar a quântização, sendo $Q = \{q_0 \ldots . q_{\overline{Q}-1}\}$ o conjunto de estados de uma máquina de Turing, o estado atual da máquina é substituído pelo observável:

$$\hat{q} = \sum_{n=0}^{\overline{Q}-1} n |\, q_n \,\rangle\langle\, q_n \,|. \tag{3.16}$$

O símbolo na posição m da fita é substituído pelo observável:

$$\hat{T}(m) = \sum_{n=0}^{\overline{\Sigma}-1} n |\, \sigma_n \,\rangle\langle\, \sigma_n \,|. \tag{3.17}$$

E a posição atual da máquina é substituído pelo observável:

$$\hat{\xi} = \sum_{\xi \in \mathbb{Z}} \xi |\, \xi \,\rangle\langle\, \xi \,|. \tag{3.18}$$

Portanto, o estado de uma MTQ $\mathcal{Q}$, no tempo t, pode ser representado por um vetor unitário $|\, \psi(t) \,\rangle$, no espaço de Hilbert $\mathcal{H}(Q, \Sigma)$ gerado pelo espaço de configurações $C(Q, \Sigma) = Q \times \Sigma^{\#} \times \mathbb{Z}$, onde $\Sigma^{\#}$ é o conjunto de todas as possíveis configurações T da fita.

Uma computação em $\mathcal{Q}$ começa no instante $t = 0$; neste instante $\mathcal{Q}$ é preparada no estado inicial:

$$|\, \psi(0) \,\rangle = |\, q_0, T_{inicial}, 0 \,\rangle. \tag{3.19}$$

A computação é realizada em passos de duração fixa τ. A evolução de $\mathcal{Q}$, em cada passo de computação, é descrita por um operador unitário U

em $\mathcal{H}(Q, \Sigma)$. Portanto, depois de n passos de computação o estado de Q é:

$$|\psi(n\tau)\rangle = U^n |\psi(0)\rangle. \tag{3.20}$$

Uma *configuração* C de uma máquina de Turing é representada por uma tripla $C = (q, T, \xi)$ no espaço de configurações $C(Q, \Sigma)$. O operador U pode ser representado como uma matriz unitária M_U, onde o elemento $M_U[i, j]$ é a amplitude de transição da configuração C_j à configuração C_i.

As transições de configurações no processo de computação de uma MTQ, de maneira similar a como se faz para as máquinas de Turing probabilísticas (MTPrs), podem ser representadas por meio de uma árvore, onde os nós são configurações C_i, e tal que existe uma aresta entre os nós C_i e C_j, assinalada com α_{ij}, se a amplitude de probabilidade de passar da configuração C_i à configuração C_j é α_{ij} ($\alpha_{ij} \neq 0$). A raiz da árvore é a configuração inicial $C_0 = (q_0, T_{inicial}, 0)$ (Figura 3.2).

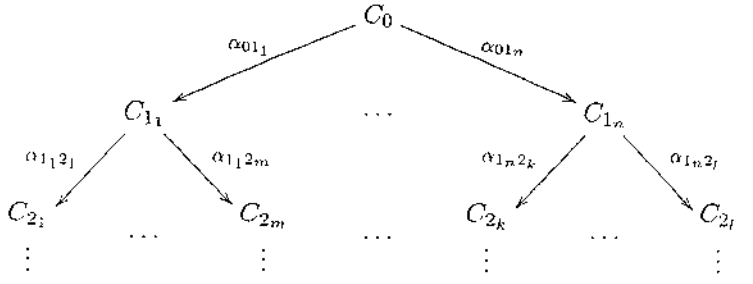

Figura 3.2: Árvore de computação numa MTQ

Diferentemente de uma MTPr, onde cada computação particular percorre somente um caminho da árvore (de acordo com as probabilidades dadas), nas MTQs a computação percorre todos os possíveis caminhos simultâneamente. Como o número de nós vai crescendo exponencialmente em cada nível da árvore, isto significa que a MTQ pode estar, num momento da computação, numa superposição de um número exponencial de configurações, de acordo com o número de passos da computação. É a isto que se chama de *paralelismo quântico*.[15] O problema é que na MTQ a computação não pode ser observada sem afetar o sistema (de acordo com os princípios

[15] O paralelismo quântico é explicado mais claramente no contexto dos CQs, ver Seção 3.3.

da mecânica quântica). Por isso, para se obter o resultado da computação, a medição deve ser feita no final da computação (seguindo o protocolo de parada, Seção 3.2.2), obtendo-se de forma probabilística somente uma das configurações e perdendo irreversivelmente as outras [51].

Note que no estado geral de uma MTQ a máquina encontra-se numa superposição de configurações. É através da interpretação de múltiplos mundos (da qual, como discutimos na Introdução, existem diversas versões; ver [82]), a partir da proposta de reinterpretações ou aprimoramentos da *interpretação de estados relativos* proposta Hugh Everett em [43]) que a superposição de configurações em qualquer instante de tempo é interpretada como a coexistência dessas configurações, e é uma interpretação desse tipo que permite pensar no paralelismo quântico como a execução simultânea das diferentes possíveis computações. O fato de que pode haver outras interpretações na mesma direção só confere maior vigor a um tal entendimento a respeito do paralelismo quântico. Contudo, como ressaltamos na Introdução, há várias outras interpretações para a mecânica quântica, que poderiam ter alguma influência na interpretação da computação quântica; apesar de que este ponto é ainda totalmente aberto e inexplorado, pode ser possível partir de interpretações com significado na filosofia da ciência, como as teorias de variáveis ocultas e as teorias com ontologias de potência devida basicamente a Karl Popper (ver, por exemplo, (cf. [23, Cap. 7], [45] e principalmente [69]) e desembocar em novas noções de computação quântica. Se tal fosse o caso, teríamos como consequência que a noção de computação quântica seria não apenas relativa à lógica, como defendido no Capítulo 2 (Seção 2.2.3), mas também seria relativa à interpretações da mecânica quântica.

Além da definição apresentada acima do modelo de MTQ, Deutsch em [36] exige que a MTQ opere finitamente (cf. [66, p. 4]) impondo as condições:

1. Somente uma parte finita do sistema deve estar em movimento durante cada passo.

2. O movimento deve depender somente do estado quântico de um subsistema finito.

3. As regras que especificam o movimento devem poder ser dadas finitamente.

A exigência destas regras, para o operador de evolução U, estão dadas em termos de uma *função de transição local* δ, a qual é apresentada a seguir:

3.2.1 Função de transição local

De maneira similar a como se faz na definição de uma MTPr, a evolução de uma MTQ Q pode ser definida por meio de uma função de transição local δ, tal que (cf. [66, 67]):

$$\delta: Q \times \Sigma \times Q \times \Sigma \times \{-1, 0, 1\} \to \tilde{C}, \tag{3.21}$$

onde Q e Σ são definidos como anteriormente, $\{-1, 0, 1\}$ representam os movimentos da cabeça de leitura-escritura (a esquerda, não movimento e direita, respectivamente) e $\tilde{C}$ representa o conjunto dos números complexos computáveis. Portanto, $\delta(q, \sigma, q', \tau, d) = c$ é interpretado assim: se Q está no estado q, lendo o símbolo σ, então, com uma amplitude de probabilidade c, a máquina Q escreve o símbolo τ e faz o movimento d.

A função δ deve cumprir as condições (cf. [66, 67]):

1. Para toda $(q, \sigma) \in Q \times \Sigma$:

$$\Sigma_{p,\tau,d} |\delta(q, \sigma, p, \tau, d)|^2 = 1. \tag{3.22}$$

2. Para toda $(q, \sigma), (q', \sigma') \in Q \times \Sigma$, com $(q, \sigma) \neq (q', \sigma')$:

$$\Sigma_{p,\tau,d} \delta(q', \sigma', p, \tau, d) * \delta(q, \sigma, p, \tau, d) = 0. \tag{3.23}$$

3. Para toda $(q, \sigma, \tau), (q', \sigma', \tau') \in Q \times \Sigma^2$:

$$\Sigma_{p \in Q, d \in \{0,1\}} \delta(q', \sigma', p, \tau', d - 1) * \delta(q, \sigma, p, \tau, d) = 0. \tag{3.24}$$

4. Para toda $(q, \sigma, \tau), (q', \sigma', \tau') \in Q \times \Sigma^2$:

$$\Sigma_{p \in Q} \delta(q', \sigma', p, \tau', -1) * \delta(q, \sigma, p, \tau, 1) = 0. \tag{3.25}$$

A condição 1 assegura a preservação da norma em cada passo de computação. As outras condições estão relacionadas com a reversibilidade da computação.

Os elementos da matriz do operador de evolução U de uma MTQ estão relacionados com a função δ por (cf. [66]):

$$\begin{aligned} \langle q', T', \xi' | U | q, T, \xi \rangle = \Big[& \delta_{\xi'}^{\xi+1} \delta\left(q, T(\xi), q', T'(\xi), 1\right) + \\ & \delta_{\xi'}^{\xi} \delta\left(q, T(\xi), q', T'(\xi), 0\right) + \\ & \delta_{\xi'}^{\xi-1} \delta\left(q, T(\xi), q', T'(\xi), -1\right) \Big] \prod_{m \neq \xi} \delta_{\mathbf{T'(m)}}^{\mathbf{T(m)}}. \end{aligned} \tag{3.26}$$

onde δ_j^i é a função delta de Kronecker (definida por $\delta_j^i = 1$ se $i = j$ e $\delta_j^i = 0$ se $i \neq j$). Com a equação (3.26) pode-se verificar se o operador U cumpre com as condições de finitude 1-3 mencionadas antes desta subseção. Além disso, pode-se definir δ por meio de U, e U por meio de δ.

Em [66, 67] demonstra-se que o operador U é unitário se e somente se δ cumpre as condições 1-4.

As MTQs podem ser definidas por meio da função de transição local δ sem fazer referência ao operador de evolução U como noção primitiva.

3.2.2 Protocolo de parada

Deutsch em [36] define o protocolo de parada de uma MTQ $\mathcal{Q}$ acrescentando um qubit de parada $\hat{n}_0$ a $\mathcal{Q}$. No começo da computação é atribuído o valor $|0\rangle$ a $\hat{n}_0$, toda $\mathcal{Q}$ válida atribui o estado $|1\rangle$ a $\hat{n}_0$ quando a computação termina e não modifica $\hat{n}_0$ nos passos anteriores. Deutsch afirma que $\hat{n}_0$ pode ser medido em cada passo sem afetar a computação. O resultado da computação é medido quando a medição de $\hat{n}_0$ dá o auto-valor correspondente ao estado $|1\rangle$.

Myers em [64] mostra que, no protocolo de parada definido por Deutsch, se a computação começar numa superposição de estados (em vez de num estado base), o qubit de parada $\hat{n}_0$ pode-se emaranhar com o estado da computação. Neste caso, a medição de $\hat{n}_0$ pode afetar a computação.

Ozawa em [66] propõe outro protocolo de parada, onde $\hat{n}_0$ (*indicador de parada*) não é um qubit independente, senão o observável correspondente ao projetor do estado final de $\mathcal{Q}$, isto é:

$$\hat{n}_0 = |q_f\rangle\langle q_f|. \tag{3.27}$$

Assumindo que existe um aparelho de medição para $\hat{n}_0$ satisfazendo o postulado de projeção (ver postulado 3), o protocolo de parada esta dado pelas regras:

1. $\hat{n}_0$ é medido instantaneamente depois de cada passo de computação. Tal medição é uma medição precisa do observável $\hat{n}_0$, que satisfaz o postulado de projeção.

2. Desde que o valor $|1\rangle$ é atribuído a $\hat{n}_0$ a MTQ não muda nem $\hat{n}_0$ nem o resultado da computação.

3. Logo depois de obtido o valor 1 na medição de $\hat{n}_0$, realiza-se a medição da fita para se obter o resultado da computação.

Ozawa demonstra que o protocolo de parada não afeta o resultado da computação, o que é equivalente a dizer que o monitoramento de $\hat{n}_0$ não afeta a distribuição de probabilidade de saída.

3.3 Circuitos quânticos

O modelo de *circuitos quânticos* é uma generalização do modelo de circuitos lógicos clássicos (ou circuitos booleanos), onde as portas lógicas são substituídas por operadores unitários, as entradas e saídas são substituídas por n-qubits e às conexões entre as portas se acrescentam algumas restrições, devidas aos princípios da mecânica quântica. A seguir, apresenta-se a definição formal de *porta quântica* (def. 3.1), mostram-se alguns exemplos e apresenta-se a definição formal de circuito quântico (def. 3.2), explicando as restrições sobre as conexões e tecendo alguns comentários sobre a medição. Na Seção 3.3.1 explica-se o conceito de *paralelismo quântico* no contexto dos CQs. Na Seção 3.3.2 apresentam-se os chamados *problema de Deutsch* e problema de Deutsch-Jozsa, cujas soluções são exemplos de circuitos quânticos (ou *algoritmos quânticos*) nos quais se tira proveito do paralelismo quântico. Por último (Seção 3.3.3), mencionam-se alguns resultados de *universalidade* de alguns conjuntos de portas quânticas.

Definição 3.1 (Porta quântica). *Uma* porta quântica *com n entradas e n saídas é um operador unitário* U: $H_{2^n} \rightarrow H_{2^n}$, *e é representado por uma matriz unitária de dimensão* 2^n *(cf. [51, p. 81]).*

A linearidade[16] dos operadores quânticos permite especificar as transformações das portas quânticas especificando somente as transformações sobre os estados base. Assim, as matrizes de Pauli (eq. 3.6) são portas quânticas que operam sobre um único qubit, realizando as transformações:

$$\begin{aligned} \sigma_x \equiv X: |0\rangle &\mapsto |1\rangle \\ |1\rangle &\mapsto |0\rangle \end{aligned} \tag{3.28}$$

$$\begin{aligned} \sigma_y \equiv Y: |0\rangle &\mapsto i|1\rangle \\ |1\rangle &\mapsto -i|0\rangle \end{aligned} \tag{3.29}$$

$$\begin{aligned} \sigma_z \equiv Z: |0\rangle &\mapsto |0\rangle \\ |1\rangle &\mapsto -|1\rangle \end{aligned} \tag{3.30}$$

O operador σ_x é também chamado de *porta de negação*.

[16] Ver Apêndice B. Definição B.9.

Outra porta quântica, que opera sobre um qubit é a chamada de *porta de Hadamard*:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{3.31}$$

as transformações realizadas por esta porta são:

$$\begin{aligned} H{:}|0\rangle &\mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle &\mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned} \tag{3.32}$$

Conjuntos parametrizados de portas quânticas podem ser definidas por meio de matrizes parametrizadas, por exemplo:

$$R_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & sen\,\frac{\theta}{2} \\ -sen\,\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, R_z(\theta) = \begin{bmatrix} e^{i\theta} & 0 \\ 0 & e^{-i\theta} \end{bmatrix}. \tag{3.33}$$

Estas portas vão ser úteis quando for oportuno se referir à universalidade (Seção 3.3.3).

De forma similar ao caso dos circuitos clássicos, as portas quânticas são representadas gráficamente por meio de uma caixa assinalada com o nome da porta. As entradas e saídas são representadas por linhas à esquerda e direita da porta, respectivamente. Para uma porta U que opera sobre n qubits a representação gráfica é:

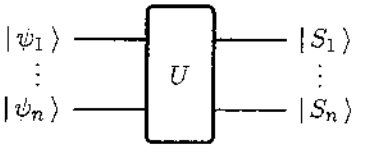

Figura 3.3: Representação gráfica de uma porta quântica

Um tipo especial de portas quânticas são as chamadas de *portas controladas*; nestas portas tem-se um qubit de controle que determina se a operação sobre os outros qubits é ou não realizada. Um exemplo destas portas é a chamada *porta de negação controlada*:

$$X_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.34}$$

As transformações realizadas por esta porta são:

$$\begin{aligned} X_c{:}|\,00\,\rangle &\mapsto |\,00\,\rangle \\ |\,01\,\rangle &\mapsto |\,01\,\rangle \\ |\,10\,\rangle &\mapsto |\,11\,\rangle \\ |\,11\,\rangle &\mapsto |\,10\,\rangle. \end{aligned} \tag{3.35}$$

Note que o segundo qubit é negado somente se o primeiro qubit é $|\,1\,\rangle$. Se o primeiro qubit é $|\,0\,\rangle$ não se realiza nenhuma operação. A representação gráfica da porta X_c é:

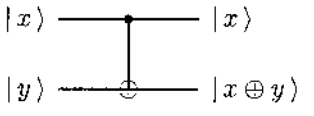

Figura 3.4: Representação gráfica da porta X_c

Em geral, a representação gráfica das portas U-controladas é:

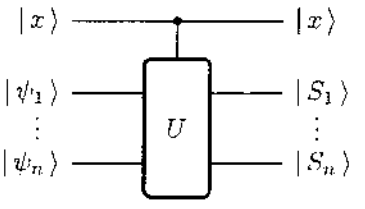

Figura 3.5: Representação gráfica de uma porta U-controlada

Definição 3.2 (Circuito quântico). *Um* circuito quântico *é uma coleção (finita) de portas quânticas conectadas aciclicamente. O* tamanho *e a* profundidade *do circuito fazem referência ao número de nós e à profundidade do grafo de conexão subjacente (cf. [51, p. 83]).*

Devido ao fato de que não existem realmente cabos quânticos, as portas se conectam compartilhando o qubit físico, por meio de uma interação de campo ou por outro meio físico (cf. [51, p. 83, nota de rodapé]).

Diferentemente dos circuitos lógicos clássicos, onde as operações de *fa-*

nin[17] e *fanout*[18] são permitidas, nos circuitos quânticos estas operações não são permitidas. A primeira por ser uma operação não reversível, e a segunda pelo *Teorema de não clonação* (cf. [73, p. 312]):

Teorema 3.1. *Não existe uma transformação unitária que copie ou clone qubits arbitrários, isto é, não existe transformação unitária U tal que $U(|a0\rangle) = |aa\rangle$ para todo $|a\rangle$.*

Demonstração. Suponha que existe uma transformação unitária U tal que $U(|a0\rangle) = |aa\rangle$ para todo $|a\rangle$. Sejam $|a\rangle$ e $|b\rangle$ dois estados ortogonais (podem ser tomados como base para o espaço dos qubits). Pela definição de U, tem-se que $U(|a0\rangle) = |aa\rangle$ e $U(|b0\rangle) = |bb\rangle$. Considere $|c\rangle = \frac{1}{\sqrt{2}}(|a\rangle + |b\rangle)$, por linearidade:

$$\begin{aligned} U(|c0\rangle) = & \frac{1}{\sqrt{2}}(U|a\rangle + U|b\rangle) \\ & \frac{1}{\sqrt{2}}(|aa\rangle + |bb\rangle). \end{aligned} \tag{3.36}$$

Mas, pela definição de U:

$$U(|c0\rangle) = |cc\rangle = \frac{1}{2}(|aa\rangle + |ab\rangle + |ba\rangle + |bb\rangle), \tag{3.37}$$

O qual contradiz o resultado obtido anteriormente. □

Medições em passos intermédios são às vezes realizadas nos circuitos quânticos, e os resultados são usados para controlar outras portas. Contudo, estas medições podem ser substituídas por portas quânticas controladas, postergando-se a medição para o final do circuito e obtendo-se o mesmo resultado (cf. [24, p. 186] e [51, p. 89]).

3.3.1 Paralelismo quântico

Seja $f\colon \{0,1\}^n \to \{0,1\}^m$ uma função qualquer e seja U_f a porta quântica tal que $U_f\colon |x,y\rangle \to |x, y \oplus f(x)\rangle$ (Figura 3.6).[19] O registro $|x\rangle$ é chamado de

[17] União de duas conexões em uma só, contendo a disjunção do conteúdo das duas conexões.

[18] Divisão de uma conexão em várias, cada uma contendo uma cópia do conteúdo da conexão inicial.

[19] Para toda função $f\colon \{0,1\}^{n+m} \to \{0,1\}^m$ a função $f_0\colon \{0,1\}^n \to \{0,1\}^{n+m}$ tal que $f_0(x,0^m) = (x, f(x))$ é reversível, portanto, pode ser construído U_f (cf. [71, Cap. 6]).

registro de entrada e o registro $|\, y \,\rangle$ é chamado de *registro de saída*. Tomando-se $|\, y \,\rangle = |\, 0 \,\rangle$ o registro de saída depois de aplicar U_f é precisamente $f(x)$.

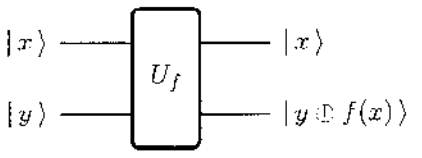

Figura 3.6: Porta lógica para uma função f, onde $|\, x \,\rangle$ e $|\, y \,\rangle$ são registros de n e m qubits, respectivamente

Tendo-se como entrada $|\, x \,\rangle$ uma superposição de todos os estados base, devido à linearidade de U_f, com uma única aplicação da porta U_f obtém-se como saída uma superposição das entradas e dos respectivos resultados de $f(x)$. Ou seja, f é avaliada simultâneamente sobre 2^n valores. A isso é ao que se chama de *paralelismo quântico* para o caso dos circuitos quânticos.

Mais formalmente, para o estado superposto:[20]

$$|\, x \,\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\, i \,\rangle, \tag{3.38}$$

tem-se que (lembrando que $|\, x, y \,\rangle$ representa o produto tensorial $|\, x \,\rangle \otimes |\, y \,\rangle$, cf. Definição B.25):

$$\begin{aligned} U_f(|\, x, y \,\rangle) &= U_f\left(\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\, i, y \,\rangle\right) \\ &= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} U_f(|\, i, y \,\rangle) \\ &= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\, i, y \oplus f(i) \,\rangle. \end{aligned} \tag{3.39}$$

Note que n qubits permitem trabalhar simultâneamente sobre 2^n estados, e portanto obtém-se um incremento exponencial no paralelismo com um

[20] O estado superposto da equação 3.38 pode se obter aplicando a porta H (eq. 3.31) individualmente sobre n qubits $|\, 0 \,\rangle$, segundo a equação:

$$H|\, 0 \,\rangle \otimes H|\, 0 \,\rangle \otimes \ldots \otimes H|\, 0 \,\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |\, i \,\rangle.$$

incremento linear no número de qubits. Lamentavelmente, de acordo com as leis da mecânica quântica, ao completar a medição somente uma das superposições é obtida, e as demais são inexoravelmente perdidas.

Para se obter algoritmos quânticos eficientes, normalmente calcula-se alguma função f tirando proveito do paralelismo quântico e (antes de realizar a medição) amplificam-se os valores $f(x)$ de interesse para a solução do problema através da aplicação de transformações unitárias adequadas, ou usa-se a superposição de todos os valores de $f(x)$ para determinar alguma propriedade comum de f. De certa forma, a chave da programação de computadores quânticos consiste na tarefa de se descobrir tais transformações unitárias (isto é, de se definir portas quânticas) convenientes.

Como exemplo, considere o circuito apresentado na seguinte figura (ver [73, p. 317]):

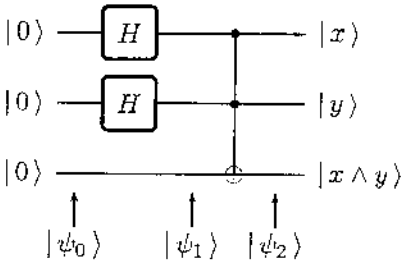

Figura 3.7: Exemplo de paralelismo quântico

onde é usada a porta T de Toffoli (ou negação duplamente controlada):

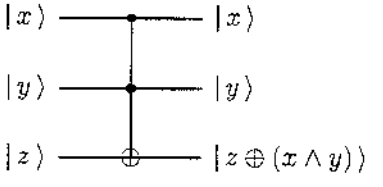

Figura 3.8: Porta de Toffoli

Note que se $|z\rangle = |0\rangle$ então $|z \oplus (x \wedge y)\rangle = |x \wedge y\rangle$.
Os estados $|\psi_i\rangle$ que aparecem em baixo do circuito (Figura 3.7) servem

para descrever os passos da computação, assim: a entrada do circuito é:

$$|\psi_0\rangle = |000\rangle. \tag{3.40}$$

Depois de aplicar as primeiras duas portas de Hadamard obtém-se a superposição de todos os possíveis valores de $|x\rangle$ e $|y\rangle$, junto com o valor $|0\rangle$ para $|z\rangle$, isto é:

$$\begin{aligned} |\psi_1\rangle &= H|0\rangle \otimes H|0\rangle \otimes |0\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle). \end{aligned} \tag{3.41}$$

Depois de aplicar T a tal superposição obtém-se:

$$\begin{aligned} |\psi_2\rangle &= T(\frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle)) \\ &= \frac{1}{2}(T(|000\rangle) + T(|010\rangle) + T(|100\rangle) + T(|110\rangle)) \\ &= \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |111\rangle). \end{aligned} \tag{3.42}$$

A superposição obtida pode ser vista como a tabela de verdade da conjunção. Ao fazer a medição (em qualquer ordem) obtém-se uma das linhas da tabela da conjunção, perdendo as outras irreversivelmente.

3.3.2 Problema de Deutsch e problema de Deutsch-Jozsa

Seja f: $\{0,1\} \to \{0,1\}$ uma função qualquer. Se diz que f é *constante* se $f(0) = f(1)$ ou é *balanceada* se $f(0) \neq f(1)$. O chamado de *problema de Deutsch* consiste em determinar, com uma única avaliação de f, se f é constante ou balanceada. Classicamente é claro que se necessita avaliar a função f duas vezes (para o valor 0 e para o valor 1) para determinar se f é constante ou balanceada. Quânticamente, aproveitando-se do paralelismo quântico, podem-se obter $f(0)$ e $f(1)$ simultâneamente, mediante uma única aplicação de U_f, e aproveitar a superposição dos resultados para determinar se f é constante ou balanceada.

A solução inicial do problema de Deutsch era do tipo probabilístico (cf. [36]). A solução determinística é devida a Cleve, Ekert, Macchiavello e Mosca (cf. [27]). A solução determinística é dada pelo circuito quântico seguinte:

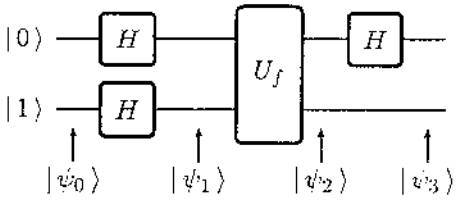

Figura 3.9: Circuito quântico para resolver o problema de Deutsch

A entrada do circuito é:

$$|\psi_0\rangle = |01\rangle. \tag{3.43}$$

Depois de se aplicar as primeiras duas portas de Hadamard, obtém-se o estado superposto:

$$\begin{aligned} |\psi_1\rangle &= H|0\rangle \otimes H|1\rangle \\ &= \left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right] \otimes \left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] \\ &= \frac{1}{2}\left[|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle)\right]. \end{aligned} \tag{3.44}$$

Aplicando-se a porta U_f ao estado $|\psi_1\rangle$, a função f é aplicada simultaneamente sobre cada um dos elementos do estado superposto, obtendo-se:

$$\begin{aligned} |\psi_2\rangle &= U_f\left(\frac{1}{2}\left[|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle)\right]\right) \\ &= \frac{1}{2}\left[|0\rangle(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle) + |1\rangle(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)\right] \\ &= \frac{1}{2}\left[(-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)\right]. \end{aligned} \tag{3.45}$$

Ou seja:

$$|\psi_2\rangle = \begin{cases} \pm\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right] \otimes \left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] & se\ f(0) = f(1). \\ \pm\left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] \otimes \left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] & se\ f(0) \neq f(1). \end{cases} \tag{3.46}$$

Aplicando-se a porta de Hadamard final, obtém-se:

$$|\psi_3\rangle = \begin{cases} \pm|0\rangle\left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] & se\ f(0) = f(1), \\ \pm|1\rangle\left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] & se\ f(0) \neq f(1). \end{cases} \tag{3.47}$$

Ao fazer uma medição sobre o primeiro qubit de $|\psi_3\rangle$ obtém-se 0 (com probabilidade 1) se $f(0) = f(1)$ (isto é, conclui-se que f é constante), e obtém-se 1 (com probabilidade 1) se $f(0) \neq f(1)$ (isto é, conclui-se que f é balanceada). Note que o algoritmo é determinístico, e efetua-se somente uma aplicação de U_f.

A generalização do problema de Deutsch para funções $f\colon \{0,1\}^n \to \{0,1\}$, onde f é balanceada se $f(x) = 0$ para a metade dos valores $x \in \{0,1\}^n$ e $f(x) = 1$ para a outra metade dos valores x, e onde parte-se da premissa de que f é balanceada ou constante, é o chamado de *problema de Deutsch-Jozsa* (ver [39]). A solução deste problema é uma generalização natural do algoritmo anterior. O circuito quântico para resolver o problema de Deutsch-Jozsa é:

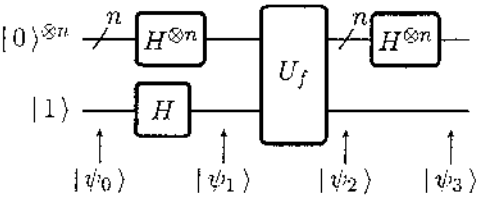

Figura 3.10: Circuito quântico para resolver o problema de Deutsch-Jozsa

$^{\otimes n}$ representa a aplicação do produto tensorial n vezes. A entrada do circuito é:

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle. \tag{3.48}$$

Depois de aplicar as primeiras portas de Hadamard obtém-se o estado superposto:

$$|\psi_1\rangle = \left[\sum_{x\in\{0,1\}^n} \frac{|x\rangle}{\sqrt{2^n}}\right] \otimes \left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right]. \tag{3.49}$$

Aplicando-se a porta U_f ao estado $|\psi_1\rangle$, a função f é aplicada simultâneamente sobre cada um dos elementos do estado superposto, obtendo-se:

$$|\psi_2\rangle = \left[\sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)}|x\rangle}{\sqrt{2^n}}\right] \otimes \left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right]. \tag{3.50}$$

Aplicando agora as portas de Hadamard do final do circuito obtém-se:

$$|\psi_3\rangle = \left[\sum_{z\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} \frac{(-1)^{x\cdot z + f(x)}|z\rangle}{2^n}\right] \otimes \left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right]. \tag{3.51}$$

onde $x \cdot z$ representa o produto interno (bit a bit) de x e z, módulo 2, isto é:

$$x \cdot z = (x_1 \wedge z_1) \ominus (x_2 \wedge z_2) \oplus \ldots \oplus (x_n \wedge z_n), \tag{3.52}$$

onde x_i representa o i-ésimo bit de x e $\ominus$ representa a suma binária. Se f for constante, em (3.51) a amplitude de probabilidade para o elemento $|0\rangle^{\otimes n}$ da superposição será 1 ou -1 (dependendo do valor constante de f), a amplitude de probabilidade dos outros elementos da base serão 0, portanto, se ao se fazer a medição se obtém o valor 0 para os n qubits, conclui-se que f é constante. Se f for balanceada, cancelam-se os valores positivos e negativos que contribuem para a amplitude de probabilidade do elemento $|0\rangle^{\otimes n}$, dando amplitude de probabilidade 0 para tal elemento; portanto, se ao se fazer a medição dos n qubits se obtém pelo menos um valor diferente de 0 conclui-se que f é balanceada.

O circuito quântico acima apresentado resolve o problema de Deutsch-Jozsa de maneira determinística, efetuando uma única aplicação de U_f, enquanto para o caso clássico é necessário (no pior caso) $2^{n-1}+1$ execuções de f para assegurar que f é constante ou balanceada (deve-se calcular f com diferentes entradas até encontrar dois valores diferentes ou até calcular a metade dos valores mais um). Como a complexidade de um algoritmo é medida pela complexidade do pior caso, a solução clássica determinística para determinar se f é balanceada ou não tem complexidade exponencial, enquanto o algoritmo quântico para solucionar o mesmo problema tem complexidade polinomial.

3.3.3 Universalidade

De maneira similar à existência de *conjuntos de portas clássicas universais*, os quais são conjuntos de portas clássicas tais que qualquer função clássica pode ser calculada por um circuito clássico contendo somente portas desse conjunto, existem também *conjuntos de portas quânticas universais*, os quais são conjuntos de portas quânticas tais que qualquer transformação unitária pode ser aproximada com precisão arbitrária por um circuito quântico contendo somente portas desse conjunto (cf. [24, p. 188]). Note que no caso dos circuitos quânticos exige-se somente uma "aproximação" com precisão arbitrária, pelo fato de o conjunto de transformações ser um conjunto contínuo.

Existem múltiplos conjuntos de portas quânticas universais, entre eles:

- A porta X_c (ou negação controlada, eq. 3.34) junto com todas as portas de um único qubit. Este conjunto permite calcular com exatidão

qualquer transformação unitária (ver [24, Seç. 4.5.2]). Realmente, só se necessita das portas de um único qubit da forma $R_y(\theta)$ e $R_z(\theta)$ (eq. 3.33) (ver [3, lema 4.1]). Note que este conjunto de portas é contínuo.

- As portas de Hadamard (eq. 3.31), fase (denotada por S), X_c (eq. 3.34) e $\pi/8$ (denotada por T), onde:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \tag{3.53}$$

  Este conjunto permite aproximar com precisão arbitrária qualquer transformação unitária (ver [24, Seç. 4.5.3]).

A questão da universalidade é importante quando se pretende estudar formalmente a questão da simulação do modelo de circuitos quânticos através de outros modelos de computação. No presente trabalho somente conseguem-se simular alguns algoritmos quânticos (construídos usando o modelo de CQs) através do modelo de MTPs e mostram-se algumas restrições do modelo de MTPs na simulação de algoritmos quânticos (ver Capítulo 4). Um dos problemas que se pretende estudar em pesquisa futura é investigar a proposta dos *circuitos paraconsistente* a partir do enfoque do *cálculo de polinômios* introduzida em [19]; a partir da concepção de circuitos paraconsistentes pretende-se simular o modelo de circuitos quânticos. Essa questão será retomada no capítulo de considerações finais.

Capítulo 4

Interpretação quântica da computação paraconsistente

Como descrito na introdução, superficialmente pode parecer estranho tentar relacionar os modelos de computação quântica e o modelo de MTPs. Porém, algumas interpretações da mecânica quântica, em particular a interpretação de Copenhagen e a interpretação de múltiplos mundos, permitem vislumbrar relações muito frutíferas entre estes modelos de computação. No presente capítulo, assumindo a interpretação da operação dos computadores quânticos sugerida por David Deutsch em [36], mostra-se como, pelo menos em alguns casos, o paralelismo quântico pode ser simulado através do que pode ser chamado de "paralelismo paraconsistente". Na interpretação do paralelismo quântico Deutsch lança mão da interpretação de múltiplos mundos da mecânica quântica, assumindo a coexistencia dos estados superpostos no processo de computação. A idéia na simulação do paralelismo quântico é simular a coexistencia dos estados superpostos atavés da multiplicidade (ou existência simultânea) de estados, posições e símbolos na configuração das MTPs. Estas idéias serão tratadas na Seção 4.1, construindo MTPs que simulam os circuitos quânticos que resolvem os chamados "problema de Deutsch" e "problema de Deutsch-Jozsa" (ver Seção 3.3.2). Estes são problemas bastante simples, mas para os quais não existe solução clássica determinística dentro das condições de eficiência exigidas.

Lembrando a definição do modelo de MTQs (Capítulo 3, Seção 3.2) e interpretando a superposição de configurações de tal modelo como a existência simultânea de múltiplos símbolos em diferentes posições da fita, múltiplos estados e múltiplas posições da máquina, o modelo de MTPs pode ser pensado como um modelo simplificado de MTQs, onde as probabilidades são

eliminadas e as diferentes configurações se misturam. Deste modo, algumas computações em MTQs podem ser simuladas através de MTPs de maneira mais ou menos natural. Contudo, como mostrado na Seção 4.2, através do modelo de MTPs definido não é possível simular, de maneira adequada, a superposição de estados e em particular os estado emaranhados, conceitos chave nos modelos de computação quântica. Portanto, propõem-se algumas conjeturas sobre como pode ser modificado o modelo de MTPs para que se possa obter uma simulação mais adequada de tais conceitos.

4.1 Uma solução do problema de Deutsch e do problema de Deutsch-Jozsa usando MTPs

No circuito quântico usado para resolver o problema de Deutsch (seção 3.3.2) inicialmente obtém-se uma superposição de estados usando portas de Hadamard. Em seguida, sobre tal superposição é aplicada a porta U_f, aproveitando-se do paralelismo quântico para se calcular a função f para os valores 0 e 1 simultâneamente, de tal maneira que, aplicando de novo a porta de Hadamard sobre o primeiro qubit da saída, obtém-se o qubit $|0\rangle$ se f for constante, ou o qubit $|1\rangle$ se f for balanceada. Uma forma similar deste procedimento pode ser simulado a partir do conceito de MTPs, como descrito a seguir:

Seja $\mathcal{M}$ uma MTP com as instruções:

$$i_1 = q_1\ 1\ 0\ q_2, \quad i_2 = q_1\ 1\ 1\ q_2, \quad i_3 = q_2\ 0\ f(0)\ q_3, \quad i_4 = q_2\ 1\ f(1)\ q_3,$$
$$i_5 = q_3\ 0^\circ\ 0\ q_4, \quad i_6 = q_3\ 1^\circ\ 0\ q_4, \quad i_7 = q_3\ 1^\bullet\ 1\ q_4.$$

Para a seqüência de entrada $n = 1$, no instante de tempo $t = 0$ a máquina $\mathcal{M}$ executa simultâneamente as instruções i_1 e i_2, escrevendo os símbolos 0 e 1 na posição 0 da fita, e passando ao estado q_2. Isto é análogo à superposição de estados obtida inicialmente na solução por meio de circuitos quânticos. Em tal configuração (instante $t = 1$) $\mathcal{M}$ executa simultâneamente as instruções i_3 e i_4, dando como resultado a computação simultânea de $f(0)$ e de $f(1)$, de maneira análoga ao calculado pela porta U_f na solução por meio de circuitos quânticos. Dependendo de se f for constante ou balanceada, no instante $t = 2$ a posição 0 da fita vai conter um único símbolo (que pode ser 0 ou 1) ou conter ambos os símbolos (0 e 1), respectivamente. Além disso, $\mathcal{M}$ vai passar ao estado q_3. Em tal configuração (instante $t = 2$) $\mathcal{M}$ executará uma das instruções i_5 ou i_6 se o símbolo que está lendo for único, ou a instrução i_7 se estiver lendo múltiplos símbolos. Portanto, $\mathcal{M}$ pára (no

instante $t = 4$) no estado q_4 e apresenta saída 0 se f for constante, ou no estado q_4 apresentando saída 1 se f for balanceada. A computação de $\mathcal{M}$ para o caso particular em que $f(0) = f(1) = 1$ é representada na Figura 4.1.

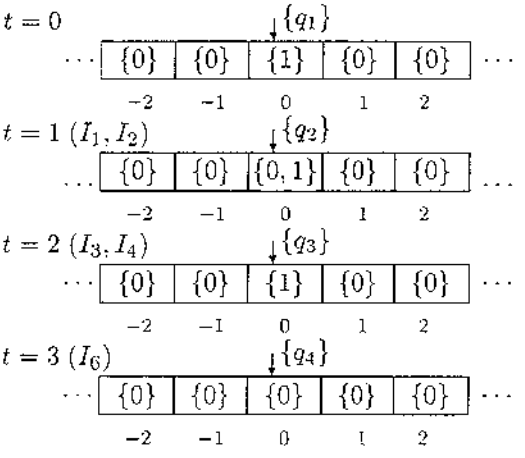

Figura 4.1: Computação do problema de Deutsch numa MTP para $f(0) = f(1) = 1$

A generalização da MTP $\mathcal{M}$ para solucionar o problema de Deutsch-Jozsa obtém-se alterando a seqüência de entrada '1' por uma seqüência de n símbolos '1'. Além disso, substituem-se as instruções i_1 e i_2, usadas para simular a geração da superposição de estados, pelas instruções:

$$i_1 = q_1\ 1\ 0\ q_2, \qquad i_2 = q_1\ 1\ 1\ q_2, \qquad i_3 = q_1\ 1\ R\ q_1,$$
$$i_4 = q_1\ 0\ L\ q_3, \qquad i_5 = q_3\ 1\ L\ q_3, \qquad i_6 = q_3\ 0^\circ\ R\ q_4.$$

Note que as instruções i_1, i_2 e i_3 são as mesmas do Exemplo 2.2 com $s_0 = 0$ e $s_1 = 1$; a Figura 2.3 mostra como vai-se simulando a superposição de estados. Essa computação é realizada em n passos, coincidindo com o número de portas quânticas utilizadas para criar a superposição dos n qubits na solução do problema de Deutsch-Jozsa com CQs (seção 3.3.2).[1] As instruções i_4, i_5 e i_6 voltam ao começo da seqüência de símbolos "superpostos" para começar a computação da função f. Em lugar das instruções i_3 e i_4 da MTP que resolve o problema de Deutsch, na MTP $\mathcal{M}$ para solucionar o problema de Deutsch-Jozsa supõe-se que $\mathcal{M}$ calcula f com as instruções i_7 a i_n, e que

[1] A complexidade de um algoritmo quântico, utilizando CQs, é medida pelo número de portas utilizadas.

tais instruções apresentam o resultado na casa seguinte ao fim da seqüência de entrada. Como na seqüência de entrada encontram-se estados "superpostos", todas as possíveis computações de f são realizadas simultâneamente, dando como resultado um único símbolo (que pode ser 0 ou 1) se f for constante, ou resultados simultâneos 0 e 1 se f for balanceada. As instruções i_5, i_6 e i_7 da MTP que resolve o problema de Deutsch, usadas para determinar se f é constante ou balanceada, de acordo com os resultados da computação simultânea de f, continuam iguais na MTP $\mathcal{M}$ que soluciona o problema de Deutsch-Jozsa, com a diferença que agora vão ser as instruções i_{n+1}, i_{n+2} e i_{n+3} respectivamente (deve-se ter em conta que o estado q_2 deve ser trocado pelo estado em que terminou a computação de f e que o estado q_3 deve ser trocado por um estado que não tenha sido usado). Tomando-se como resultado da computação a seqüência de símbolos não vazios (considerando 0 como o símbolo vazio) que tem início na posição em que a máquina pára (tal como definido por exemplo em [42] ou [21]), o resultado da computação vai ser 0 se f for constante, ou 1 se f for balanceada.

Para ilustrar a solução proposta no parágrafo anterior para solucionar o problema de Deutsch-Jozsa por meio de MTPs, seja $f\colon \{0,1\}^2 \to \{0,1\}$ tal que $f(x,y) = x \oplus y$, onde $\oplus$ representa a suma binária. As instruções da MTP $\mathcal{M}$ para calcular f são:

$$i_7 = q_4\ 0\ R\ q_5, \quad i_8 = q_4\ 1\ R\ q_6, \quad i_9 = q_5\ 0\ R\ q_7, \quad i_{10} = q_5\ 1\ R\ q_8,$$
$$i_{11} = q_6\ 0\ R\ q_9, \quad i_{12} = q_6\ 1\ R\ q_{10}, \quad i_{13} = q_7\ 0\ 0\ q_{11}, \quad i_{14} = q_8\ 0\ 1\ q_{11},$$
$$i_{15} = q_9\ 0\ 1\ q_{11}, \quad i_{16} = q_{10}\ 0\ 0\ q_{11}.$$

As instruções i_5, i_6 e i_7 da MTP que resolvem o problema de Deutsch serão então as instruções i_{17}, i_{18} e i_{19} de $\mathcal{M}$, respectivamente. A computação de $\mathcal{M}$ para a seqüência de entrada indicada $n = 1\ 1$, é descrita pela seqüência de descrições instantâneas $\alpha_0, \alpha_1, \ldots, \alpha_{10}$ onde:

$$\alpha_0 = \{q_1\}\{1\}\{1\};$$
$$\alpha_1 = \{q_2\}\{0,1\}\{q_1\}\{1\},\ (i_1, i_2, i_3);$$
$$\alpha_2 = \{0,1\}\{q_2\}\{0,1\}\{q_1\}\{0\},\ (i_1, i_2, i_3);$$
$$\alpha_3 = \{0,1\}\{q_3\}\{0,1\}\{0\},\ (i_4);$$
$$\alpha_4 = \{q_3\}\{0,1\}\{0,1\}\{0\},\ (i_5);$$
$$\alpha_5 = \{q_3\}\{0\}\{0,1\}\{0,1\}\{0\},\ (i_5);$$
$$\alpha_6 = \{0\}\{q_4\}\{0,1\}\{0,1\}\{0\},\ (i_6);$$
$$\alpha_7 = \{0\}\{0,1\}\{q_5, q_6\}\{0,1\}\{0\},\ (i_7, i_8);$$

$\alpha_8 = \{0\}\{0,1\}\{0,1\}\{q_7, q_8, q_9, q_{10}\}\{0\}$. $(i_9, i_{10}, i_{11}, i_{12})$;
$\alpha_9 = \{0\}\{0,1\}\{0,1\}\{q_{11}\}\{0,1\}$. $(i_{13}, i_{14}, i_{15}, i_{16})$;
$\alpha_{10} = \{0\}\{0,1\}\{0,1\}\{q_{12}\}\{1\}$. (i_{19}).

4.2 Restrições das MTPs na simulação de algoritmos quânticos

Mesmo que para alguns casos o paralelismo quântico possa ser simulado por meio das MTPs, como mostrado na seção anterior, o modelo de MTPs definido não permite uma simulação adequada do conceito quântico de superposição de estados, e em particular do conceito de estado emaranhado, conceitos estes, como mencionado, que são essenciais na construção de algoritmos quânticos eficientes. Para facilitar a explicação deste ponto vamos nos restringir a um sistema quântico de dois qubits (um 2-qubit). É conveniente lembrar, a partir da equação (3.11), que o estado geral de um 2-qubit é da forma $|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$, onde $\alpha_0, \ldots, \alpha_3$ são números complexos e $|xy\rangle$ representa o produto tensorial $|x\rangle \otimes |y\rangle$. O conceito de superposição de estados afirma que um sistema quântico pode se encontrar em qualquer combinação linear de seus estados base, o que pode ser interpretado fisicamente como a coexistência de tais estados (note que, para que este fenômeno faça sentido, é conveniente apelar para interpretações inflacionárias tais como as do tipo "múltiplos mundos" (cf. [82])). Para um 2-qubit a superposição de estados está representada pela equação envolvendo $|\psi\rangle$ acima. Como já foi dito no capítulo 3, os fatores α_i, da equação acima para $|\psi\rangle$, são chamados de amplitudes de probabilidade. O estado $|\psi\rangle$ pode ser interpretado então como a coexistência dos estados com amplitude de probabilidade diferente de 0. As amplitudes de probabilidade podem ser pensadas como "pesos" que se dão a cada um dos estados coexistentes. Ao contrário do modelo para computação quântica onde as amplitudes e probabilidade têm sentido, o modelo de MTPs não leva tais pesos em conta; eliminando-se menção a tais pesos, a coexistência dos estados nas MTPs deve ser vista como a existência simultânea equalitária desses múltiplos estados. É essa a intuição usada na simulação dos algoritmos para solucionar os problemas de Deutsch e de Deutsch-Jozsa por meio de MTPs (seção anterior).

Considerando-se uma MTP $\mathcal{M} = \langle Q, \Sigma, M, I\rangle$, com $Q = \{q_1, q_2\}$ e $\Sigma = \{s_0, s_1\}$, para a situação particular em que $\mathcal{M}$, no instante de tempo t e na posição x, se encontra simultâneamente nos estados q_1 e q_2 e com os símbolos

s_0 e s_1 sobre essa posição da fita (como representado na Figura 4.2), pode-se pensar que esta situação está simulando uma superposição $|\psi\rangle$ de dois qubits tal que $\alpha_i \neq 0$ para todo $0 \leq i \leq 3$. O primeiro qubit representa o estado da máquina no sentido de Turing (o estado quântico $|0\rangle$ representa o estado q_1 da máquina e o estado quântico $|1\rangle$ representa o estado q_2 da máquina) enquanto o segundo qubit representa o símbolo nessa posição da fita (o estado quântico $|0\rangle$ representa o símbolo s_0 e o estado quântico $|1\rangle$ representa o estado s_1). Para a situação em que $\mathcal{M}$ se encontra no estado

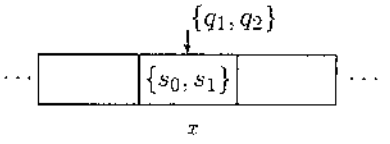

Figura 4.2: Superposição de estados e símbolos numa MTP

q_1 somente (isto é, não apresenta multiplicade de estados), mas está lendo simultaneamente os símbolos s_0 e s_1, as superposições $|\psi\rangle$ correspondentes serão aquelas onde $\alpha_2 = \alpha_3 = 0$, $\alpha_0 \neq 0$ e $\alpha_1 \neq 0$. As superposições $|\psi\rangle$ com $\alpha_0 = \alpha_1 = 0$, $\alpha_2 \neq 0$ e $\alpha_3 \neq 0$ vão corresponder à situação em que $\mathcal{M}$ se encontra somente no estado q_2 mas lendo simultaneamente os símbolos s_0 e s_1. Assim, pode-se continuar relacionando alguns estados $|\psi\rangle$ a situações da MTP $\mathcal{M}$, mas nem para todo estado $|\psi\rangle$ existe uma situação de $\mathcal{M}$ que a corresponda de maneira adequada a $|\psi\rangle$. Por exemplo, para o estado emaranhado $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ não existe uma configuração de $\mathcal{M}$ que simule $|\psi\rangle$ de maneira adequada. Para que uma configuração de $\mathcal{M}$ possa adequadamente simular a situação de dependência entre os dois qubits em estado emaranhado $|\psi\rangle$, $\mathcal{M}$ deve estar munida de uma relação entre o estado q_1 e o símbolo s_0, e entre o estado q_2 e o símbolo s_1, e somente essas relações devem se verificar entre esses estados e símbolos (ver Figura 4.3). Isso não é possível ocorrer com o modelo de MTPs definido, pois em tal modelo os resultados da execução de diferentes instruções se misturam indiscriminadamente, tal como explicado na seção 2.2.3 (segundo paragrafo). Contudo, o modelo de MTPs "não adjuntivo", proposto também na Seção 2.2.3 (segundo parágrafo) se coloca como um outro modelo possível de MTPS, que pode ser definido partindo-se de uma lógica paraconsistente não adjuntiva em lugar de $\mathbf{LFI1}^*$. Este parece ser um modelo mais adequado para simular estados emaranhados, mas o estudo de suas potencialidades e características ainda permanece uma questão a ser investigada (ver também

o capítulo de considerações finais).

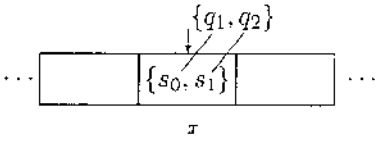

Figura 4.3: Estados e símbolos "emaranhados" numa MTP

Considerações finais

Na elaboração de uma dissertação, tese, ou trabalho de pesquisa (como em muitos outros labores), tem-se muitas vezes a sensação de que nunca se vai conseguir chegar a uma versão final, a um produto totalmente terminado e apresentável. De tal sensação não se conseguiu, nem se pretendeu, escapar no presente trabalho. Por isso, neste capítulo, em lugar de apresentar conclusões definitivas a respeito do trabalho realizado (atividade essa de bom grado deixada ao leitor), preferiu-se apresentar perguntas, questões abertas e idéias de trabalhos futuros que surgiram na realização desta dissertação e que, por questões de tempo ou simplesmente por não se ter ainda uma resposta satisfatória, não foram melhor desenvolvidos.

Tentando alcançar a pretensão de simular algoritmos quânticos por meio de "algoritmos paraconsistentes", simulando as características essenciais da computação quântica de modo a preservar a eficiência dos algoritmos quânticos, mostrou-se na Seção 4.2 que o modelo de MTPs aqui definido não permite uma simulação adequada do conceito quântico de superposição de estados, e em particular do conceito de estado emaranhado, conceito essencial na construção de algoritmos quânticos eficientes. O que não faz do modelo de MTPs aqui desenvolvido um mau modelo, ou um modelo desinteressante: só faz dele um modelo não adequado para a plenitude de tal pretensão. Contudo, como também foi descrito na Seção 4.2, é possível se definir um novo modelo de MTPs, seguindo o mesmo procedimento de axiomatização, substituição da lógica subjacente e definição do modelo, mas usando uma lógica paraconsistente não-adjuntiva em lugar de **LFI1***. Neste novo modelo é de se esperar que os resultados da execução simultânea de diferentes instruções permaneçam separados, devido à não-adjuntividade da lógica de base, abrindo assim a possibilidade de simular o entrelaçamento dos estados emaranhados. Esta é uma primeira idéia meritória de ser desenvolvida em trabalhos futuros.

Também na simulação de estados emaranhados, no papel que estes desempenham nos algoritmos quânticos, podem ser úteis conceituações de

semânticas formais que são bastante adequadas a lógicas não-clássicas em geral: a *semântica de traduções possíveis* e a *semântica de mundos possíveis*. A semântica de traduções possíveis, introduzida por Walter Carnielli em [17] e re-elaborada em [18] (ver também [35]), permite definir procedimentos de decisão para algumas lógicas não vero-funcionais (em particular para lógicas paraconsistentes), traduzindo as fórmulas da lógica original em fórmulas de lógicas vero-funcionais, e estabelecendo a validade das fórmulas da lógica original através da verificação da validade das fórmulas traduzidas. Por outro lado, a semântica de mundos possíveis, introduzida por Saul Kripke em [58] e [59], permite estabelecer a validade de uma fórmula a partir de interpretações clássicas das fórmulas em diferentes "mundos", e das relações de acessibilidade entre esses mundos. Por meio da semântica de traduções possíveis, ou da semântica de mundos possíveis, existe a possibilidade de se interpretar o múltiplo conteúdo de uma fita numa máquina de Turing paraconsistente (assim como os múltiplos estados e posições) por meio de traduções ou mundos possíveis adequados, considerando-os como uma coleção de fitas com conteúdo único (a fita com conteúdo múltiplo pode ser vista como se estiver num estado de superposição), abrindo assim outra possibilidade de se pensar na simulação de estados emaranhados. Além do mais, do ponto de vista estritamente lógico, o fato de que tanto a semântica de mundos possíveis quanto a semântica de traduções possíveis sejam candidatos a interpretar múltiplos conteúdos de uma fita numa máquina de Turing paraconsistente faz pensar que ambos os enfoques semânticos sejam na verdade casos especiais de um arcabouço semântico mais geral, e que seria então apropriado para a interpretação de fenômenos quânticos.

Por outro lado, usando a idéia de *cálculo de polinômios* sugerida por Walter Carnielli em [19], idéia esta que permite definir procedimentos de prova e de decisão para todas as lógicas multivalentes finitárias e mesmo para certas lógicas não vero-funcionais, é possível se definir um modelo de *circuitos paraconsistente*. O procedimento de decisão, obtido através do cálculo de polinômios, consiste em traduzir fórmulas a polinômios e reduzir demonstrações ao cálculo de zeros dos polinômios. Assim, podem-se associar polinômios (com variáveis adequadas) a *portas paraconsistentes*. As saídas das portas estarão então determinadas pelas soluções dos polinômios associados, e a conexão de portas estará determinada por composição de polinômios. De maneira similar ao que acontece no caso da computação quântica, é possível que o modelo de circuitos paraconsistentes resulte ser mais simples para a construção de algoritmos, e permita estabelecer uma relação mais estreita entre computação quântica e computação paraconsistente.

Uma outra questão que merece ser mencionada diz respeito às relações entre o problema da parada de uma máquina de Turing paraconsistente e a indecidibilidade da lógica (paraconsistente) de primeira ordem sobre a qual a máquina é definida. O resultado que discutimos na Seção 1.2 do Capítulo 1 sobre a equivalência entre a indecidibilidade da lógica de primeira ordem e o problema da parada em máquinas de Turing clássicas pode muito naturalmente ser estendido para a lógica $\mathbf{LFI1}^*$ e para as MTPs, e possivelmente para outras lógicas e outros modelos de computação. Resultados desse tipo, muito mais do que uma questão matemática, seriam relevantes em nossa proposta de abordar a computabilidade como dependente da lógica.

Saindo um pouco do escopo da presente dissertação e entrando mais no campo da computação quântica propriamente, é importante ressaltar que, embora se tenham construído alguns algoritmos quânticos significativamente mais rápidos do que seus correspondentes clássicos, ainda não são claras as verdadeiras potencialidades da computação quântica (como reconhecido pelo próprio Peter Shor em [77]). Portanto, é de vital importância esclarecer o papel que desempenham as características reconhecidas como essenciais da computação quântica, tais como a *superposição de estados*, o *paralelismo quântico*, os *estados emaranhados* e a *interferência quântica* na construção de algoritmos quânticos eficientes. Nesse sentido, uma boa fundamentação lógica das características mencionadas poderia permitir formular uma teoria a respeito da essência dos algoritmos quânticos.

São vários os trabalhos já disponíveis que perseguem a tentativa de fundamentar as características da computação quântica através da investigação da lógica subjacente a tais modelos (ver por exemplo [41, 72, 32, 22, 52, 53, 5, 6, 7]).

Maria Luisa Dalla Chiara, Roberto Giuntini e Roberto Leporini, definindo operadores lógicos a partir de portas quânticas, definem em [32] novas lógicas quânticas as quais chamam de *lógicas computacionais quânticas* (LCQs) Nessas lógicas é importante ressaltar a presença do novo operador lógico $\sqrt{\neg}$ (raiz quadrada da negação), sugerido por David Deutsch, Artur Ekert e Rossella Lupacchini em [38]. O comportamento do operador $\sqrt{\neg}$ está associado à operação realizada pela porta quântica $\sqrt{NOT}$, onde $\sqrt{NOT}(\sqrt{NOT}(|\psi\rangle)) = NOT(|\psi\rangle)$ para todo qubit $|\psi\rangle$. Diferentemente das chamadas lógicas quânticas ortodoxas, baseadas nas idéias de Garret Birkhoff e John von Neumann em [12] (onde as sentenças são interpretadas como subespaços fechados do espaço de Hilbert associado ao sistema físico em consideração), nas LCQs as sentenças são interpretadas como quantidades de informação, um tipo abstrato de objetos descrito no marco da teoria da informação quântica. Por meio das LCQs pode-se fundamentar, ainda

que parcialmente (para alguns casos específicos), as noções de superposição quântica, a interferência quântica e os estados emaranhados (cf. [32]).

Sendo os modelos de computação quântica baseados nos princípios da mecânica quântica, e existindo lógicas quânticas (ortodoxas), como a *ortológica* e a *lógica quântica orto-modular* (ver [33]), bastante estudadas e que possuem características muito interessantes, uma pergunta natural é por que não se tomar uma destas lógicas para tentar construir um fundamento lógico das características da computação quântica. De fato, um trabalho neste sentido já foi proposto por J.P. Rawling e S.A. Selesnick em [72], os quais usam a orto-lógica e a teoria de modelos associada e fundamentam, também parcialmente, as noções de paralelismo quântico e de interferência quântica.

Além das lógicas quânticas ortodoxas, existem várias outras lógicas que tentam dar um fundamento lógico aos fenômenos da mecânica quântica, entre elas a *lógica quântica exógena* [62] e a *lógica linear* [49, 48]. A idéia principal da lógica quântica exógena é considerar superposições de modelos clássicos (deixando-os intactos) como os modelos para a lógica quântica; tal abordagem está relacionada à semântica de traduções possíveis. Por outro lado, em [50] Jean-Yves Girard apresenta uma interpretação quântica de uma parte da lógica linear (a qual chama de *perfeita*) por meio de espaços quânticos coerentes. Na implicação linear a premissa é destruída quando usada, e este aspecto "perfeito" (como denominado por Girard) coincide com uma característica essencial dos fenômenos quânticos, onde uma medição pode alterar o estado atual do sistema [50, p. 4]. A este respeito, cabe aqui ainda observar que um estudo a respeito da semântica de quantais e do uso mais abstrato de C^*-algebras (na direção de [28]) em computação quântica ainda está para ser explorado. A lógica quântica exógena e a lógica linear, em princípio, também podem ser usadas para oferecer um fundamento lógico para as características da computação quântica.

Outra possibilidade interessante de trabalho a ser desenvolvido consiste então em estudar as lógicas anteriormente mencionadas, e em procurar encontrar uma melhor fundamentação lógica para as características da computação quântica através de alguma lógica adequada (possivelmente, uma já existente, ou uma proposta com fins específicos).

No caso do presente trabalho, acreditamos que ainda que as MTPs (pelo menos no particular modelo apresentado) não dêem conta completamente de todas as nuances dos fenômenos quânticos e não possam, portanto, expressar todas essas sutilezas da programação de algortimos quânticos, tais máquinas baseadas em lógicas paraconsistentes expressam um importante e novo paradigma de computação que permite manifestar características re-

levantes da computação quântica. Enfoques dessa natureza são certamente influentes nas questões filosóficas que envolvem os fenômenos quânticos, a interpretação da mecânica quântica e ainda mais diretamente a interpretação dos modelos de computação quânticos.

Apêndice A

Regras de dedução

Gentzen, no seu sistema de dedução natural (ver [47]), propõe 8 regras de dedução básicas para o cálculo proposicional clássico. Para cada conector ($\neg, \wedge, \vee, \rightarrow$) define-se uma regra de "introdução" e uma regra de "eliminação": estas regras apresentam-se nas Tabelas A.1 e A.2.

As 8 regras básicas são suficientes para realizar qualquer dedução no cálculo proposicional clássico. Porém, para simplificar as demonstrações, normalmente são usadas regras derivadas. Algumas regras derivadas apresentam-se na tabela A.3.

Para o cálculo de predicados de primeira ordem adicionam-se regras que permitem introduzir e eliminar quantificadores. Estas regras apresentam-se na tabela A.4.

Regras de introdução	Regras de eliminação
Implicação ($\rightarrow$)	
II: Teorema da dedução	EI: Modus ponens
$\begin{array}{l}\boxed{\begin{array}{l}A\\ \vdots\\ B\end{array}}\\ \hline A \rightarrow B\end{array}$	$\begin{array}{l}A \rightarrow B\\ A\\ \hline B\end{array}$
Conjunção ($\wedge$)	
IC: Adjunção	EC: Simplificação
$\begin{array}{l}A\\ B\\ \hline A \wedge B\end{array}$	$\dfrac{A \wedge B}{A}$ $\qquad$ $\dfrac{A \wedge B}{B}$
Disjunção ($\vee$)	
ID: Adição	ED: Prova por casos
$\dfrac{A}{A \vee B}$ $\qquad$ $\dfrac{B}{A \vee B}$	$\begin{array}{l}A \vee B\\ \boxed{\begin{array}{l}A\\ \vdots\\ C\end{array}}\\ \boxed{\begin{array}{l}B\\ \vdots\\ C\end{array}}\\ \hline C\end{array}$

Tabela A.1: Regras básicas para o cálculo proposicional clássico

Regras de introdução	Regras de eliminação
Negação ($\neg$)	
IN: Redução ao absurdo	EN: Dupla negação
$\begin{array}{l}\left[\begin{array}{l}A\\ \vdots\\ B \wedge \neg B\end{array}\right]\\ \hline \neg A\end{array}$	$\dfrac{\neg\neg A}{A}$

Tabela A.2: Regras básicas para o cálculo proposicional clássico (cont.)

Silogismo hipotético	Silogismo disjuntivo
$\dfrac{A \to B \quad B \to C}{A \to C}$	$\dfrac{A \vee B \quad \neg A}{B}$
Conmutatividade da conjunção e da disjunção	
$\dfrac{A \wedge B}{B \wedge A}$	$\dfrac{A \vee B}{B \vee A}$
Contraposição	Modus tollens
$\dfrac{A \to B}{\neg B \to \neg A}$	$\dfrac{A \to B \quad \neg B}{\neg A}$

Tabela A.3: Algumas regras derivadas para o cálculo proposicional clássico

Regras de introdução	Regras de eliminação
Generalização universal	Instanciação universal
$\dfrac{Pa}{\forall x Px}$ Condição: 'a' não deve ocorrer em nenhum suposto prévio não cancelado.	$\dfrac{\forall x Px}{Pa}$
Generalização existencial	Instanciação existencial
$\dfrac{Pa}{\exists x Px}$	$\dfrac{\exists x Px \qquad \begin{array}{|l} Pa \\ \vdots \\ A \end{array}}{A}$ Condição: 'a' não deve ocorrer em $\exists x Px$, nem em A, nem em nenhum suposto prévio não cancelado.

Tabela A.4: Regras para o cálculo de predicados clássico

Apêndice B

Elementos de álgebra linear

No presente apêndice apresentam-se alguns elementos básicos de álgebra linear, com a intenção de tornar o presente trabalho auto-contido e de evitar ao leitor não iniciado a necessidade de consultar um livro de álgebra linear para o entendimento do capitulo 3.

Definição B.1 (Corpo). *Um* corpo *é um conjunto C que contem pelo menos dois elementos, munido de duas operações binárias: adição (denotada por $+$) e multiplicação (denotada por $\cdot$), tal que as operações satisfazem às propriedades:*

1. *Associatividade da adição: para todo $a, b, c \in C$ tem-se que $(a+b)+c = a+(b+c)$.*

2. *Existência de identidade aditiva: existe um elemento em C, denotado por 0, tal que $a+0 = 0+a = a$ para todo $a \in C$.*

3. *Existência de inverso aditivo: para todo elemento $a \in C$ existe um elemento em C, denotado por $-a$, tal que $a+(-a) = (-a)+a = 0$.*

4. *Comutatividade da adição: para todo $a, b \in C$ tem-se que $a+b = b+a$.*

5. *Associatividade da multiplicação: para todo $a, b, c \in C$ tem-se que $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.*

6. *Existência de identidade multiplicativa: existe um elemento em C, denotado por 1, tal que $a \cdot 1 = 1 \cdot a = a$ para todo $a \in C$.*

7. *Existência de inverso multiplicativo: para todo elemento $a \in C$, se $a \neq 0$ então existe um elemento em C, denotado por a^{-1}, tal que $a \cdot a^{-1} = a^{-1} \cdot a = 1$.*

8. *Comutatividade da multiplicação: para todo* $a, b \in C$ *tem-se que* $a \cdot b = b \cdot a$.

9. *Distributividade: para todo* $a, b, c \in C$ *tem-se que* $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$ *e que* $c \cdot (a + b) = (c \cdot a) + (c \cdot b)$.

Definição B.2 (Espaço vetorial). *Seja* C *um corpo, cujos elementos serão chamados de "escalares". Um* espaço vetorial *sobre* C *é um conjunto não vazio* V*, cujos elementos serão chamados de "vetores", munido de duas operações: adição (denotada por +) e multiplicação por um escalar (denotada por justaposição). A adição atribui a cada par de vetores* $(\mathbf{u}, \mathbf{v}) \in V \times V$ *um vetor* $\mathbf{u} + \mathbf{v}$ *em* V*, enquanto a multiplicação por um escalar atribui a cada par* $(a, \mathbf{u}) \in C \times V$ *um vetor* $a\mathbf{v}$ *em* V*. As operações devem satisfazer às propriedades:*

1. *Associatividade da adição: para todo* $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ *tem-se que* $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$.

2. *Comutatividade da adição: para todo* $\mathbf{u}, \mathbf{v} \in V$ *tem-se que* $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$.

3. *Existência de identidade aditiva: existe um elemento em* V*, denotado por* $\mathbf{0}$*, tal que* $\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$ *para todo* $\mathbf{u} \in V$.

4. *Existência de inverso aditivo: para todo elemento* $\mathbf{u} \in V$ *existe um elemento em* V*, denotado por* $-\mathbf{u}$*, tal que* $\mathbf{u} + (-\mathbf{u}) = (-\mathbf{u}) + \mathbf{u} = \mathbf{0}$.

5. *Propriedades da multiplicação por um escalar: para todos os escalares* $a, b \in C$ *e para todo vetor* $\mathbf{u}, \mathbf{v} \in V$ *tem-se que:*

$$\begin{aligned} a(\mathbf{u} + \mathbf{v}) &= a\mathbf{u} + a\mathbf{v}, \\ (a + b)\mathbf{u} &= a\mathbf{u} + b\mathbf{u}, \\ (ab)\mathbf{u} &= a(b\mathbf{u}), \\ 1\mathbf{u} &= \mathbf{u}. \end{aligned}$$

Definição B.3 (Subespaço). *Um* subespaço *de um espaço vetorial* V *é um subconjunto* S *de* V *que satisfaz as condições de espaço vetorial com respeito às operações obtidas restringindo-se as operações de* V *a* S.

Definição B.4 (Combinação linear). *Seja* V *um espaço vetorial sobre o corpo* C*; qualquer expressão da forma:*

$$a_1\mathbf{v_1} + \ldots + a_n\mathbf{v_n},$$

onde $a_i \in C$ *e* $\mathbf{v_i} \in V$ *para todo* i*, é chamada uma* combinação linear *dos vetores* $\mathbf{v_1}, \ldots, \mathbf{v_n}$.

Definição B.5 (Independência linear). *Seja* V *um espaço vetorial sobre o corpo* C*. Um subconjunto não vazio de vetores* S *de* V *é dito* linearmente independente *se para todo* $\mathbf{v_1}, \ldots, \mathbf{v_n}$ *em* S*, tem-se que:*

$$a_1\mathbf{v_1} + \ldots + a_n\mathbf{v_n} = \mathbf{0} \text{ implica que } a_1 = \ldots = a_n = 0,$$

com $a_1, \ldots, a_n \in C$*. Se um subconjunto de vetores* S *de* V *não é linearmente independente então é* linearmente dependente.

Definição B.6 (Subespaço gerado). *Seja* V *um espaço vetorial sobre o corpo* C*. O* subespaço gerado *por um subconjunto de vetores* S *de* V*, denotado por* $span(S)$*, é o conjunto de todas as combinações lineares de vetores de* S*. Isto é:*

$$span(S) = \{a_1\mathbf{v_1} + \ldots + a_n\mathbf{v_n} \mid a_i \in C, v_i \in S\}.$$

Definição B.7 (Base). *Qualquer conjunto de vetores* S *de* V *que seja linearmente independente e que gere* V *(isto é:* $span(S) = V$*) é chamado uma* base *de* V.

Pode-se demonstrar que se V é um espaço vetorial, todas as bases de V têm a mesma cardinalidade. Isso permite definir a noção de dimensão de um espaço vetorial.

Definição B.8 (Dimensão). *A cardinalidade das bases de* V *é chamada de* dimensão *do espaço.*

A partir de agora, e para simplificar as definições, trataremos apenas com espaços vetoriais de dimensão finita. Assim, "espaço vetorial" significará, daqui em diante, "espaço vetorial de dimensão finita". Pode ser provado que, se V é um espaço vetorial de dimensão n sobre um corpo C, então V é isomorfo a C^n (o espaço vetorial de todos os vetores sobre C de comprimento n).

Definição B.9 (Operador linear). *Sejam* V *e* W *espaços vetoriais sobre o mesmo corpo* C*. Um* operador linear *é qualquer função* A*:* $V \rightarrow W$ *tal que:*

$$A\left(\sum_{i=1}^{n} a_i\mathbf{u_i}\right) = \sum_{i=1}^{n} a_i A(\mathbf{u_i}),$$

para todo $n \geq 1$*, todos os escalares* $a_i \in C$ *e todos os vetores* $\mathbf{u_i} \in V$.

Definição B.10 (Representação matricial). *Sejam V e W espaços vetoriais sobre o corpo C, e seja A: $V \to W$ um operador linear entre esses espaços. Suponha que $\mathbf{v_1}, \ldots, \mathbf{v_m}$ é uma base para V e que $\mathbf{w_1}, \ldots, \mathbf{w_n}$ é uma base para W. Então, para cada $j = 1, \ldots, m$, existem números $A_{1j} \in C$ até $A_{nj} \in C$ tais que:*

$$A(\mathbf{v_j}) = \sum_{i=1}^{n} A_{ij}\mathbf{w_i}.$$

A matriz cujas entradas são os valores A_{ij} é chamada de representação matricial *do operador A. A representação matricial do operador A é completamente equivalente ao operador A.*

No caso particular em que $W = C$ então a representação matricial de um operador linear A: $V \to C$ é dada por um vetor-linha de $C^{1\times n}$.

Definição B.11 (Transposta conjugada de uma matriz). *A* transposta conjugada de uma matriz, *cujos elementos são números complexos, é obtida trocando linhas por colunas e calculando a conjugada de cada elemento da matriz.*[1] *Formalmente, se $A = (A_{ij})$ então a transposta conjugada de A é a matriz $A^\dagger = (\overline{A_{ji}})$.*

Definição B.12 (Produto interno). *Seja V um espaço vetorial sobre o corpo C, onde $C = \mathbb{R}$ ou $C = \mathbb{C}$. Um* produto interno *em V é uma função $\langle \cdot, \cdot \rangle$: $V \times V \to C$ com as seguintes propriedades:*

1. *Para todo $\mathbf{v} \in V$ tem-se que $\langle \mathbf{v}, \mathbf{v} \rangle \geq 0$ e que $\langle \mathbf{v}, \mathbf{v} \rangle = 0$ se e somente se $\mathbf{v} = \mathbf{0}$.*

2. *Para todo $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ e para todo $a, b \in C$ tem-se que $\langle r\mathbf{u} + s\mathbf{v}, \mathbf{w} \rangle = r\langle \mathbf{u}, \mathbf{w} \rangle + s\langle \mathbf{v}, \mathbf{w} \rangle$.*

3. *Para todo $\mathbf{u}, \mathbf{v} \in V$, se $C = \mathbb{R}$ então $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$ ou se $C = \mathbb{C}$ então $\langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}$.*

Definição B.13 (Espaço de Hilbert complexo de dimensão finita). *Um* espaço de Hilbert complexo de dimensão finita *é um espaço vetorial sobre $\mathbb{C}$ com produto interno.*

Para o caso de espaços de Hilbert de dimensão infinita existem algumas restrições técnicas adicionais, que não necessitam ser apresentadas aqui.

[1] Todo número complexo α pode ser escrito na forma $\alpha = a + bi$, e sua conjugada $\overline{\alpha}$ é dado por $\overline{\alpha} = a - bi$.

Dado que um espaço de Hilbert complexo de dimensão finita V é, em particular, um espaço vetorial de dimensão finita sobre $\mathbb{C}$, então V é isomorfo a $\mathbb{C}^n$, para algum $n \geq 1$. Neste texto somente trataremos de espaços de Hilbert complexos de dimensão finita. Assim, a partir de agora usaremos "espaço de Hilbert" para nos referir a "espaços de Hilbert complexos de dimensão finita".

Exemplo B.1 (Produto interno usual num espaço de Hilbert). *Seja V um espaço de Hilbert e sejam $\mathbf{v}, \mathbf{w}$ vetores de V. a função $\langle \cdot, \cdot \rangle \colon V \times V \to \mathbb{C}$ tal que $\langle \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{v} \,|\, \mathbf{w} \rangle$, onde $\langle \mathbf{v} \,|\, \mathbf{w} \rangle$ indica o produto matricial da transposta conjugada do vetor-coluna $\mathbf{v}$ (Definição B.11) pelo vetor-coluna $\mathbf{w}$, é um produto interno em V.*
Observação: dado que V é isomorfo a $\mathbb{C}^n$ (para algum $n \geq 1$) então podemos supor que $\mathbf{v}$ e $\mathbf{w}$ são da forma $\mathbf{v} = (v_1, \ldots, v_n)$ e $\mathbf{w} = (w_1, \ldots, w_n)$ (para números complexos v_i e w_i). Logo, $\langle \mathbf{v} \,|\, \mathbf{w} \rangle = \sum_{i=1}^{n} \overline{v_i} w_i$.

Definição B.14 (Vetor dual). *Se V é um espaço de Hilbert e $\mathbf{v} \in V$ então o* vetor dual *de $\mathbf{v}$ é o operador linear $\hat{\mathbf{v}} \colon V \to \mathbb{C}$ tal que $\hat{\mathbf{v}}(\mathbf{w}) = \langle \mathbf{v}, \mathbf{w} \rangle$ para todo $\mathbf{w} \in V$.*

Observe que, se V tem dimensão n e $\mathbf{v} \in V$ então a representação matricial de $\hat{\mathbf{v}}$ é o vetor-coluna de $\mathbb{C}^{n \times 1}$ que corresponde à transposta conjugada do vetor-linha $\mathbf{v}$.

Definição B.15 (Notação de Dirac). *Na notação de Dirac:*

1. *Os vetores de um espaço de Hilbert V são denotados por $|\cdot\rangle$, e são chamados de* kets.

2. *O produto interno de $|\mathbf{v}\rangle$ com $|\mathbf{w}\rangle$ é denotado por $\langle \mathbf{v} \,|\, \mathbf{w} \rangle$.*

3. *Os vetores duais dos kets são denotados por $\langle \cdot |$, e são chamados de* bras. *Assim, o dual do vetor $|\mathbf{v}\rangle$ é denotado por $\langle \mathbf{v}|$, e $\langle \mathbf{v}|(|\mathbf{w}\rangle) = \langle \mathbf{v} \,|\, \mathbf{w} \rangle$ para todo $|\mathbf{w}\rangle \in V$.*

4. *Dados um ket $|\mathbf{v}\rangle$ e um bra $\langle \mathbf{w}|$, o* produto externo *de $|\mathbf{v}\rangle$ com $\langle \mathbf{w}|$ é o operador linear $|\mathbf{v}\rangle\langle \mathbf{w}| \colon V \to V$ tal que $|\mathbf{v}\rangle\langle \mathbf{w}|(|\mathbf{u}\rangle) = \langle \mathbf{w} \,|\, \mathbf{u} \rangle |\mathbf{v}\rangle$ para todo vetor $|\mathbf{u}\rangle$ de V. Observe que $\langle \mathbf{w} \,|\, \mathbf{u} \rangle |\mathbf{v}\rangle$ denota o produto do escalar $\langle \mathbf{w} \,|\, \mathbf{u} \rangle$ pelo ket $|\mathbf{v}\rangle$.*

Definição B.16 (Ortogonalidade). *Seja V um espaço vetorial com produto interno. Dois vetores $\mathbf{u}, \mathbf{v} \in V$ são* ortogonais *se $\langle \mathbf{u}, \mathbf{v} \rangle = 0$.*

Definição B.17 (Norma). *Seja V um espaço vetorial com produto interno e $\mathbf{v} \in V$ um vetor. A* norma *do vetor $\mathbf{v}$ está definida por:*

$$\|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}.$$

Definição B.18 (Vetor unitário). *Seja V um espaço vetorial com produto interno e $\mathbf{v} \in V$ um vetor, se diz que $\mathbf{v}$ é um* vetor unitário *se $\|\mathbf{v}\| = 1$.*

Definição B.19 (Base orto-normal). *Seja V um espaço vetorial com produto interno. Uma* base orto-normal *para V é uma base onde todos os elementos são unitários e onde todo par de elementos são ortogonais.*

Definição B.20 (Auto-vetor, auto-valor e auto-espaço). *Seja A um operador linear sobre um espaço vetorial V (A: $V \to V$). Um* auto-vetor *de A é um vetor diferente de $\mathbf{0}$ tal que $A\mathbf{v} = v\mathbf{v}$, onde v é qualquer escalar. Neste caso v é chamado o* auto-valor *correspondente ao auto-vetor $\mathbf{v}$. O espaço gerado pelos auto-vetores de A é chamado de* auto-espaço *de A.*

Definição B.21 (Representação diagonal). *Seja V um espaço de Hilbert. A representação diagonal de um operador linear A: $V \to V$ é uma representação da forma:*

$$A = \sum_i \lambda_i |i\rangle\langle i|.$$

onde os vetores $|i\rangle$ forman um conjunto orto-normal de auto-vetores de A. com auto-valores correspondentes λ_i. Observe que $|i\rangle\langle i|$: $V \to V$ é um operador linear (ver Definição B.15) para todo i.

Diz-se que um operador A é diagonalizável *se tem representação diagonal.*

Definição B.22 (Operador adjunto). *Seja A: $V \to V$ um operador linear sobre um espaço de Hilbert V. Então existe um único operador linear A^*: $V \to V$ tal que:*

$$\langle A(\mathbf{u}), \mathbf{v} \rangle = \langle \mathbf{u}, A^*(\mathbf{v}) \rangle,$$

para todo $\mathbf{u}, \mathbf{v} \in V$. O operador A^* é chamado operador adjunto *de A.*

Se (A_{ij}) é a representação matricial de A (ver Definição B.10) então a representação matricial de A^* é dada pela matriz transposta conjugada $(A_{ij})^\dagger$ (ver Definição B.11).

Definição B.23 (Operador auto-adjunto ou Hermitiano). *Um operador linear A: $V \to V$ num espaço de Hilbert V é chamado* auto-adjunto *ou* Hermitiano *se seu operador adjunto é ele mesmo, isto é, se $A^* = A$.*

Definição B.24 (Operador unitário). *Um operador linear A: $V \to V$ num espaço de Hilbert V é* unitário *se $A \circ A^* = I$, onde "$\circ$" representa a composição de operadores e "I" representa a operador identidade sobre o espaço de Hilbert V.*

Definição B.25 (Produto tensorial de duas matrizes). *O* produto tensorial de duas matrizes *é usualmente chamado de* produto de Kronecker. *Para as matrizes* u e v *o produto tensorial é denotado por* $\mathbf{u} \otimes \mathbf{v}$ *e é dado por:*

$$\mathbf{u} \otimes \mathbf{v} = \begin{bmatrix} u_{11}\mathbf{v} & u_{12}\mathbf{v} & \cdots \\ u_{21}\mathbf{v} & u_{22}\mathbf{v} & \\ \vdots & & \ddots \end{bmatrix} = \begin{bmatrix} u_{11}v_{11} & u_{11}v_{12} & \cdots & u_{12}v_{11} & u_{12}v_{12} & \cdots \\ u_{11}v_{21} & u_{11}v_{22} & & u_{12}v_{21} & u_{12}v_{22} & \\ \vdots & & \ddots & & & \\ u_{21}v_{11} & u_{21}v_{12} & & & & \\ u_{21}v_{21} & u_{21}v_{22} & & & & \\ \vdots & & & & & \end{bmatrix}$$

Definição B.26 (Produto tensorial de espaços vetoriais). *O* produto tensorial de espaços vetoriais *é um conceito útil que permite compor espaços vetoriais de maneira a definir espaços vetoriais de maior dimensão. Suponha que V e W sejam espaços vetoriais de dimensões m e n, respectivamente. Então o produto tensorial de V e W (denotado por $V \otimes W$) é um espaço vetorial de dimensão $m \times n$. Os elementos de $V \otimes W$ são combinações lineares do produto tensorial de elementos de V e de W. Quando os elementos de V e de W são representados matricialmente (como vetores-coluna), o produto tensorial destes elementos está dado pela definição anterior.*

Referências Bibliográficas

[1] Agudelo, Juan C.: *Máquinas de Turing paraconsistentes: algunas posibles definiciones y consecuencias*. Trabalho de Graduação - Especialização em Lógica e Filosofía. Universidad EAFIT, 2003. Disponível em: http://sigma.eafit.edu.co:90/~asicard/archivos/mtps.ps.gz.

[2] Agudelo, Juan C. e Andrés Sicard: *Máquinas de Turing paraconsistentes: una posible definición*. Matemáticas: Enseñanza Universitaria, XII(2):37–51, 2004. Disponível em: http://revistaerm.univalle.edu.co/Enlaces/volXII2.html.

[3] Barenco, Adriano, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John Smolin, e Harald Weinfurter: *Elementary gates for quantum computation*. Physical Review A, 52(5):3457–3467, 1995.

[4] Batens, Diderik: *Dynamic dialectical logics*. In Priest, G., R. Routley, e J. Norman (editores): *Paraconsistent Logic: essays on the inconsistent*, páginas 187–217. Munich: Philosophia Verlag, 1989.

[5] Battilotti, Giulia: *Basic logic and quantum computing: logical judgements by an insider observer*. Disponível em: http://arxiv.org/abs/quant-ph/0407057, 2004.

[6] Battilotti, Giulia e Paola Zizzi: *The internal logic of Bell's states*. Disponível em: http://arxiv.org/abs/quant-ph/0412199, 2004.

[7] Battilotti, Giulia e Paola Zizzi: *Logical interpretation of reversible measurement in quantum computing*. Disponível em: http://arxiv.org/abs/quant-ph/0408068, 2005.

[8] Benioff, Paul: *The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represen-*

ted by Turing machines. Journal of Statistical Physics, 22:563–591. 1980.

[9] Bennett, Charles H.: *Logical reversibility of computation.* IBM Journal of Research and Development, 17:525–532, 1973.

[10] Bernstein, Ethan e Umesh Vazirani: *Quantum complexity theory.* Proc. of the 1993 ACM Symposium on Theory of Computing, páginas 11–20, 1993.

[11] Bernstein, Ethan e Umesh Vazirani: *Quantum complexity theory.* SIAM Journal on Computing, 26(5):1411–1473, 1997.

[12] Birkhoff, Garret e John Von Neumann: *The Logic of Quantum Mechanics.* Annals of Mathematics, 37(4):823–843, October 1936.

[13] Boolos, George e Richard Jeffrey: *Computability and logic.* Cambridge University Press, 3a. edição, 1989.

[14] Braunstein, Samuel L.: *Error Correction for Continuous Quantum Variables.* Physical Review Letters, 80(18):4084–4087, 1998.

[15] Bub, Jeffrey: *Interpreting the Quantum World.* Cambridge University Press, 1997.

[16] Bub, Jeffrey: *Why the quantum?* Studies in History and Philosophy of Modern Physics, 35B:241–266, 2004. Versão preliminar disponível em: `http://arxiv.org/quant-ph/0402149`.

[17] Carnielli, Walter A.: *Many-valued logics and plausible reasoning.* In *Proceedings of the Twentieth International Symposium on Multiple-Valued Logic*, páginas 328–335, Charlotte, NC, USA, 1990. IEEE Computer Society.

[18] Carnielli, Walter A.: *Possible-translations semantics for paraconsistent logics.* In Batens, D., C. Mortensen, G. Priest, e J. P. Van Bendegem (editores): *Frontiers of Paraconsistent Logic: Proceedings of the I World Congress on Paraconsistency*, Logic and Computation Series, páginas 149–163. Baldock: Research Studies Press, King's College Publications, 2000.

[19] Carnielli, Walter A.: *Polynomial Ring Calculus for Many-valued Logics.* In Werner, B. (editor): *Proceedings of the 35th International Symposium on Multiple-Valued Logic*, páginas 20–25. IEEE Computer Society.

2005. Versão preliminar disponível em *CLE e-Prints* vol 5, n. 3, 2005: http://www.cle.unicamp.br/e-prints/vol_5,n_3,2005.html.

[20] Carnielli, Walter A., João M. de Almeida, e Sandra de Amo: *Formal inconsistency and evolutionary databases.* Logic and logical philosophy, páginas 115–152, 2000.

[21] Carnielli, Walter A. e Richard L. Epstein: *Computabilidade, funções computáveis, lógica e os fundamentos da Matemática.* São Paulo: Editora UNESP, 2005. ISBN 85-7139-650-7.

[22] Cattaneo, Gianpero, Maria Luisa Dalla Chiara, Roberto Giuntini, e Roberto Leporini: *An unsharp logic from quantum computation.* Disponível em: http://arxiv.org/abs/quant-ph/0201013, 2002.

[23] Chibeni, Silvio S.: *Aspectos da Descrição Física da Realidade*, volume 21 de *Coleção CLE.* CLE/UNICAMP, 1997.

[24] Chuang, Isaac L. e Michael A. Nielsen: *Quantum Computation and Quantum Information.* Cambridge: Cambridge University Press, 2000.

[25] Church, Alonso: *A note on the Entscheidungsproblem.* Journal of Symbolic Logic, 1(1):40–41, 1936. pp. 101-102.

[26] Church, Alonso: *An unsolvable problem of elementary number theory.* American Journal of Mathematics, 58:345–363, 1936.

[27] Cleve, Richard, Artur Ekert, Chiara Macchiavello, e Michele Mosca: *Quantum algorithms revisited.* Proceedings of the Royal Society of London. Series A, 454:339–354, 1998.

[28] Coniglio, Marcelo E. e Francisco Miraglia: *Non-Commutative Topology and Quantales.* Studia Logica, 65(2):223–236, 2000.

[29] Copeland, Jack: *Hipercomputation.* Minds and machines, 12:461–502, 2002.

[30] da Costa, Newton C. A. e Décio Krause: *Remarks on the aplications of paraconsistent logic to physics.* Versão preliminar disponível em: http://philsci-archive.pitt.edu/archive/00001566.

[31] da Costa, Newton C. A. e Décio Krause: *Complementarity and paraconsistency.* In Rahman, S., J. Symons, D. M. Gabbay, e J. P. Bendegen (editores): *Logic, Epistemology, and the Unity of Science*, volume 1 de *Logic, Epistemology, and the Unity of Science.* Springer, 2004.

[32] Dalla Chiara, Maria Luisa, Roberto Giuntini, e Roberto Leporini: *Computational Logics. A Survey.* In Hendricks, Vincent F. e Jacek Malinowski (editores): *Trends in Logic: 50 Years of Studia Logica*, páginas 229–271. Kluwer Academic Publishers, 2003. Versão preliminar disponível em: `http://arxiv.org/abs/quant-ph/0305029`.

[33] Dalla Chiara, María Luisa e Roberto Giuntini: *Quantum Logics.* In *Handbook of Philosophical Logic*, volume 6, páginas 129–228. Netherlands: Kluwer Academics Publishers, 2002.

[34] Davis, Martin: *Computability and unsolvability.* New York: Dover Publications, Inc., 1982.

[35] de Almeida, João M.: *Semânticas de traduções possíveis.* Tese de Mestrado, Universidade Estadual de Campinas, Instituto de Filosofia e Ciências Humanas, 1999.

[36] Deutsch, David: *Quantum theory, the Church-Turing principle and the universal quantum computer.* Proceedings of the Royal Society of London. Series A, 400:97–117, 1985.

[37] Deutsch, David: *Quantum computational networks.* Proceedings of the Royal Society of London. Series A, 425:73–90, 1989.

[38] Deutsch, David, Artur Ekert, e Rossella Lupacchini: *Machines, logic and quantum physics.* Bulletin of Symbolic Logic, 6(3):265–283, Sep. 2000. Versão preliminar disponível em: `http://arxiv.org/abs/math.LO/9911150`.

[39] Deutsch, David e Richard Jozsa: *Rapid solution of problems by quantum computation.* Proceedings of the Royal Society of London. Series A, 439:553–558, 1992.

[40] D'Ottaviano, Itala M. L. e Newton C. A. da Costa: *Sur un problème de Jaskowski.* Comptes Rendus de l'Académie des Sciences de Paris, 270:1349–1953, 1970.

[41] Dunn, Michael, Tobias J. Hagge, Lawrence S. Moss, e Zhenghan Wang: *Quantum logic as motivated by quantum computing.* Journal of Symbolic Logic, 70(2):353–359, 2005.

[42] Epstein, Richard L. e Walter A. Carnielli: *Computability: computable functions, logic, and the foundations of mathematics.* Belmont, CA: Wadsworth/Thomson Learning, 2a edição, 2000.

[43] Everett, Hugh: *Relative state formulation of quantum mechanics.* Reviews of Modern Physics, 29, 1957. Tesis de doutorado, Princenton University: reimpreso em *The Many-Worlds Interpretation of Quantum Mechanics.* B. DeWitt e N. Graham, eds., Princenton University Press, 1973; The theory of the universal wave function, p. 1–140.

[44] Feynman, Richard P.: *Simulating Physics with Computers.* International Journal of Theoretical Physics, 21:467–488, 1982.

[45] Freire, Olival Jr.: *Popper, probabilidade e mecânica quântica.* Episteme, páginas 103–127, 2004.

[46] Garola, Claudio: *Semantic Realism: A New Philosophy for Quantum Physics.* International Journal of Theoretical Physics, 38(12):3241–3252, 1999.

[47] Garrido, Manuel: *Lógica simbólica.* Madrid: Editorial Tecnos, 3a edição, 1997.

[48] Girard, Jean Yves: *Linear logic, its syntax and semantics.* In Girard, Lafont, e Regnier (editores): *Advances in Linear Logic*, páginas 1–42. Cambridge University Press, 195. Disponível em: `http://iml.univ-mrs.fr/~girard/Synsem.pdf.gz`.

[49] Girard, Jean Yves: *Linear Logic.* Theoretical Computer Science, 50:1–102, 1987.

[50] Girard, Jean Yves: *Between logic and quantic: a tract.* Disponível em: `http://iml.univ-mrs.fr/~girard/LLcup.pdf.gz`, 2004.

[51] Gruska, Jozef: *Quantum computing.* Cambridge: McGraw-Hill International (UK) Limited, 1999.

[52] Gudder, Stan: *Quantum Computational Logic.* International Journal of Theoretical Physics, 42(1):39–47, January 2003.

[53] Gudder, Stan: *Computational Logic on Fock Space.* International Journal of Theoretical Physics, 43(6):1409–1422, June 2004.

[54] Hilbert, David, Lothar Nordheim, e John von Neumann: *Über die Grundlagen der Quantenmechanik.* Mathematische Annalen, 98:1–30, 1927.

[55] Jammer, Max: *The Philosophy of Quantum Mechanics.* John Wiley & Sons, inc., 1974.

[56] Jaśkowski, Stanisław: *Rachunek zdań dla systemów dedukcyjnych sprzecznych*. Studia Societatis Scientiarum Torunensis. Sectio A, I(5):57–77, 1948. Traduzido para o inglês como 'A propositional calculus for inconsistent deductive systems' em *Logic and Logic Philosophy*, 7:35-56, 1999, Proceedings of the Stanisław Jaśkowski's Memorial Symposium. Toruń, Polonia, Julhio de 1998.

[57] Kleene, Stephen C.: *General recursive functions of natural numbers*. Mathematische Annalen, 112:727–742, 1936.

[58] Kripke, Saul: *A completeness theorem in modal logic*. Journal of Symbolic Logic, 24:1–14, 1959.

[59] Kripke, Saul: *Semantical analysis of modal logic. I. Normal modal propositional calculi*. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, 9:67–96, 1963.

[60] Lloyd, Seth e Samuel L. Braunstein: *Quantum Computation over Continuous Variables*. Physical Review Letters, 82(8):1784–1787, 1999.

[61] Manin, Yuri: *Computable and Uncomputable (in Russian)*. Sovetskoye Radio, Moscow, 1980.

[62] Mateus, Paulo e Amílcar Sernadas: *Exogenous Quantum Logic*. In *Proc. of CombLog'2004, Workshop on Combination of Logics: Theory and Applications*, páginas 142–149. CLC, Department of Mathematics, IST, Lisbon, Portugal, 2004. Disponível em: `http://wslc.math.ist.utl.pt/ftp/pub/SernadasA/04-MS-fiblog24s.pdf`.

[63] Moreno, Luis: *Factorización cuántica de números enteros: una introspectiva al algoritmo de Shor*. Monografía Ingeniería de Sistemas, Universidad EAFIT, 2004. Disponível em: `http://sigma.eafit.edu.co:90/~asicard/archivos/MonografiaLuis2004.tar.gz`.

[64] Myers, John M.: *Can a Universal Quantum Computer Be Fully Quantum?* Physical Review Letters, 78(9):1823–1824, 1997.

[65] Odifreddi, Piergiorgio: *Classical recursion theory. The theory of functions and sets of natural numbers*. Studies in logic and the foundations of mathematics, volume 125. Amsterdan: North-Holland, 1989.

[66] Ozawa, Masanao: *Quantum Turing machines: local transitions, preparation, measurement and halting problem*. Disponível em: `http://arxiv.org/abs/quant-ph/9811069`, 1998.

[67] Ozawa, Masanao e Haramichi Nishimura: *Local transition function of quantum Turing machines.* Disponível em: `http://arxiv.org/abs/quant-ph/9811069`, 1999.

[68] Penrose, Roger: *The emperor's new mind: concerning computers, minds and the laws of physics.* Oxford: Oxford University Press, 1989.

[69] Popper, Karl R.: *Quantum Theory and the Schism in Physics.* Hutchinson, London, 1982.

[70] Post, Emil: *Finite Combinatory Process-Formulation 1.* The Journal of Simbolic Logic, 1(3):103–105, 1936.

[71] Preskill, John: *Quantum Computation.* Notas do curso de computação quântica ministrado pelo professor John Preskill, disponível em `http://www.theory.caltech.edu/people/preskill/ph229/`.

[72] Rawling, J. Piers e Stephen A. Selesnick: *Orthologic and quantum logic: models and computational elements.* Journal of the ACM, 47(4):721–751, 2000, ISSN 0004-5411.

[73] Rieffel, Eleanor e Wolfgang Polak: *An Introduction to Quantum Computing for Non-Physicists.* ACM Computing Surveys, 32(3):300–335, 2000.

[74] Schütte, Kurt: *Beweistheorie.* Springer Verlag, 1960.

[75] Shor, Peter W.: *Algorithms for Quantum Computation: Discrete Log and Factoring.* In *Proc. 35th Symposium on Foundations of Computer Science*, páginas 124–134. IEEE Computer Society Press, 1994.

[76] Shor, Peter W.: *Polinomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.* SIAM Journal on Computing, 26(5):1484–1509, 1997.

[77] Shor, Peter W.: *Introduction to Quantum Algorithms.* In Samuel J. Lomonaco, Jr. (editor): *Proceedings of Symposia in Applied Mathematics (AMS PSAPM/58)*, páginas 143–159. American Mathematical Society, 2002.

[78] Sylvan, Richard e Jack Copeland: *Computability is logic-relative.* In Priest, Graham e Dominic Hyde (editores): *Sociative logics and their applications: essays by the late Richard Sylvan*, páginas 189–199. London: Ashgate Publishing Company, 2000.

[79] Tarski, Alfred, Andrzej Mostowski, e Raphael M. Robinson: *Undecidable Theories*. North-Holland, 1953.

[80] Tegmark, Max: *The Interpretation of Quantum Mechanics: Many Worlds or Many Words?* Fortschritte der Physik, 46:855–862, 1998. Versão preliminar disponível em: http://http://arxiv.org/abs/quant-ph/9709032.

[81] Turing, Alan M.: *On computable numbers, with an application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society, páginas 230–265, 1936. A correction, ibid, vol 43. 1936-1937. págs. 544 - 546.

[82] Vaidman, Lev: *Many-Worlds Interpretation of Quantum Mechanics*. In Zalta, Edward N. (editor): *The Stanford Encyclopedia of Philosophy*. 2002. http://plato.stanford.edu/archives/sum2002/entries/qm-manyworlds/.

[83] Väänänen, Jouko: *A short course on finite model theory*. Notes based on lectures first gave at the *Summer School of Logic, Language and Information* in Lisbon in August 1993. Disponível em: http://www.math.helsinki.fi/logic/people/jouko.vaananen/shortcourse.pdf.

[84] Vélez, Mario e Andrés Sicard: *Sobre un modelo de computación cuántica sobre variables continuas*. Versão preliminar disponível em: http://sigma.eafit.edu.co:90/~asicard/archivos/ccc.ps.gz, 2000.

[85] Yao, Andrew C.: *Quantum circuit complexity*. In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, páginas 352–360. IEEE Computer Society Press, Los Alamitos, CA, 1993.