unesp
Universidade Estadual Paulista
Faculdade de Filosofia e Ciências
Campus de Marília

Luciana Pavani de Paula Bueno

Estudo da Memória nos Modelos Mecânicos da Inteligência

Marília
2000

cm 1 2 3 4 5 6 7 unesp 11 12 13 14 15 16 17 18

LUCIANA PAVANI DE PAULA BUENO

ESTUDO DA MEMÓRIA NOS MODELOS MECÂNICOS DA INTELIGÊNCIA

t. 1.102

Fac. de Filosofia e Ciências
BIBLIOTECA
Campus de Marília

Dissertação apresentada à Faculdade de Filosofia e Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de Marília, para a obtenção do título de Mestre em Filosofia (Área de Concentração: Ciência Cognitiva e Filosofia da Mente).

Orientador: Prof. Dr. Cosme Damião Bastos Massi

1410001102

Marília
2000

unesp

autor B928e
1.102

Nº cham 154
B928e

Nº Tombo 1.102
Aquisição D
Preço 5,00
Data 10/08/2001
Proced. Doação

Bueno, Luciana Pavani de Paula
B928e Estudo da memória nos modelos mecânicos da inteligência / Luciana Pavani de Paula Bueno. – 2000.
187 f. ; 30 cm.

Dissertação (mestrado) – Universidade Estadual Paulista. Faculdade de Filosofia e Ciências de Marília, 2000.
Bibliografia: f. 184-186.

1. Memória. 2. Modelos mecânicos. 3. Ciência Cognitiva. I. Autor. II. Título.

CDD 153.12

LUCIANA PAVANI DE PAULA BUENO

ESTUDO DA MEMÓRIA NOS MODELOS MECÂNICOS DA INTELIGÊNCIA

COMISSÃO JULGADORA

DISSERTAÇÃO PARA OBTENÇÃO
DO GRAU DE MESTRE

Presidente e Orientador: Prof. Dr. Cosme Damião Bastos Massi
Departamento de Filosofia – FFC – UNESP

2º Examinador: Profª. Drª. Plácida Leopoldina Amorim da Costa Santos
Departamento de Biblioteconomia – FFC – UNESP

3º Examinador: Profª. Drª. Maria Eunice Guilici Gonzales
Departamento de Filosofia – FFC – UNESP

Marília, 12 de setembro de 2000

unesp

DADOS CURRICULARES

LUCIANA PAVANI DE PAULA BUENO

NASCIMENTO 13.09.1974 – São José do Rio Preto/SP

FILIAÇÃO João Manoel de Paula Bueno
Guiomar Pavani de Paula Bueno

1993/1996 Curso de Graduação em Ciências da Computação
Instituto de Biociências, Letras e Ciências Exatas
UNESP - Campus de São José do Rio Preto

1997/2000 Curso de Pós-Graduação em Filosofia, nível de Mestrado
Área de Concentração: Ciência Cognitiva e Filosofia da Mente
Faculdade de Filosofia e Ciências
UNESP - Campus de Marília

Aos meus pais

Que sempre me incentivaram e apoiaram em todas as circunstâncias. Pelos valores que me foram passados, inclusive o grande valor do estudo.

AGRADECIMENTOS

- ao Prof. Dr. Cosme Damião Bastos Massi, pelo auxílio na escolha do tema a ser desenvolvido, pela atenção, pelas sugestões pertinentes e pela orientação deste trabalho;

- à Profª. Drª. Maria Eunice Quilici Gonzales, pelo acompanhamento feito por meio das diversas leituras e das preciosas críticas dispensadas a esta dissertação. Pelo incentivo que me foi dado durante todo o curso, por me ter feito perceber a importância e a beleza da Ciência Cognitiva, e por seu tratamento amigo;

- a todo o Corpo Docente do Programa de Pós-Graduação, pelos conhecimentos que me passaram e pelo apoio recebido;

- às professoras Mariana Claudia Broens e Maria Cândida Soares Del Masso, pela ajuda nas leituras deste trabalho;

- aos colegas e amigos de curso, pela convivência enriquecedora, pelos estudos em conjunto e pelos momentos de diversão – vocês são muito especiais!

- à UNESP e à CAPES, pelo apoio institucional e pelas bolsas de estudo concedidas;

- aos meus pais, os grandes incentivadores e auxiliadores na obtenção de minhas maiores conquistas.

CONTEÚDO

Resumo **1**

Introdução **2**

1 Modelos Mecânicos da Inteligência **8**
1.1 A elaboração de modelos como instrumentos explicativos da Ciência Cognitiva 9
1.2 As idéias de Turing sobre modelos mecânicos inteligentes 15
1.3 O olhar crítico dos filósofos (a respeito das idéias de Turing) 22
1.4 A memória humana como referencial para os modelos mecânicos 31

2 Máquinas Abstratas **37**
2.1 Máquina de estado finito 38
2.2 Máquina de Turing 60
2.3 A importância da memória da máquina para a resolução de problemas 91

3 Computadores **104**
3.1 Computadores: conceito e origem 105
3.2 Organização dos Computadores: categorização e componentes básicos 112
3.3 A memória dos computadores 121

4 Redes Neurais Artificiais **143**
4.1 Princípios Fundamentais das Redes Neurais Artificiais 144
4.2 Caracterização das Redes Neurais Artificiais 153
4.3 A Rede de Hopfield 162

Notas Finais **175**

Bibliografia **184**

Abstract **187**

BUENO, L. P. P. *Estudo da memória nos modelos mecânicos da inteligência*. Marília, 2000. 187p. Dissertação (Mestrado em Filosofia – Área de Concentração: Ciência Cognitiva e Filosofia da Mente) – Faculdade de Filosofia e Ciências, Campus de Marília, Universidade Estadual Paulista "Júlio de Mesquita Filho".

RESUMO

Embora a criação de modelos mecânicos seja a estratégia usada pela Ciência Cognitiva para simular (e assim explicar) o comportamento inteligente, a aceitação dessa estratégia não constitui unanimidade entre os cognitivistas, uma vez que há opiniões divergentes sobre as possibilidades de as máquinas serem apropriadas para essa simulação. Nesse contexto, é nossa intenção verificar qual a relevância da memória das máquinas para que alguns processos mentais possam ser adequadamente simulados em modelos mecânicos. A fim de levantar subsídios para tal esclarecimento, desenvolvemos um estudo das máquinas envolvidas nas linhas de pesquisa representacionistas da Ciência Cognitiva e, mais especificamente, analisamos o conceito de memória nessas máquinas. Por fim, com base nesse estudo, expomos nosso ponto de vista a respeito da possível ampliação da eficiência dos modelos inteligentes quando diante de uma capacidade adequada de memória.

Palavras-chave: Memória; modelos mecânicos; Ciência Cognitiva.

INTRODUÇÃO

Este trabalho tem como objetivo levantar subsídios para o esclarecimento da hipótese de que a memória da máquina é relevante para que os processos mentais sejam apropriadamente simulados por algum modelo mecânico.

Relacionando a inteligência humana com o computador, existe uma história que remonta, pelo menos, ao artigo clássico de Turing, de 1950, *Computadores e Inteligência*.

Turing, um dos pioneiros da computação, deixa claro, em suas pesquisas, convicções favoráveis a respeito das máquinas, sugerindo que, resolvido o problema da falta de memória, elas seriam capazes de exibir comportamento inteligente semelhante ao humano. Suas idéias incitaram, em muitos pesquisadores, a tentativa de uma explicação da inteligência humana através de modelos mecânicos.

A partir de então, há pelo menos 50 anos, o modelo mecânico da mente tem sido enfatizado nos estudos de inteligência artificial. No entanto, durante a última década, também se tornou aparente que tanto os modelos computacionais tradicionais quanto os modelos conexionistas apresentam limitações em suas tentativas de explicar a mente.

cm 1 2 3 4 5 6 7 unesp 12 13 14 15 16 17 18

Com relação ao mundo natural, sabe-se que, por muitos anos, a memória tem sido objeto de intensa e produtiva pesquisa, sempre na busca de entender como ela é organizada e como esta envolve as várias funções do cérebro. Também é grande a dificuldade em se saber o que é inteligência. Apesar de não serem esses dois assuntos pesquisados em toda sua amplitude, é tomada como certa a relação entre inteligência e memória. Schank e Birnbaum (1996, p.13) afirmam que "a inteligência é uma questão de aprendizado, de adquirir memória ou conhecimento base de uma extensão suficiente, e de desenvolver os mecanismos de recuperação necessários para usá-lo". De fato, a idéia de memória prodigiosa costuma estar sempre associada a algum fator positivo e eficiente ligado à inteligência.

Sabendo-se que a inteligência natural pressupõe memória, e que a extraordinária memória do homem é parte de sua complexidade, estaria Turing correto ao sugerir que a eficiência das máquinas como modelos apropriados de processos cognitivos se reduziria à sua capacidade de memória? Ou, ainda, seria necessário levar em conta, além da questão tecnológica, fatores que podem ser considerados um tanto misteriosos como consciência, emoção, intuição?

Analisando-se essas questões na área da Ciência Cognitiva, pode-se perceber concepções inadequadas sobre computação e sobre certos requisitos para uma explicação científica a respeito das máquinas, o que faz surgir um sem número de problemas de ordem teórica, tais como simplificações, falsas analogias e até erros conceituais. Por exemplo, sabe-se que certos pesquisadores, em particular alguns filósofos, não têm conhecimento em profundidade sobre

unesp

memória em máquinas. Em suas pesquisas, eles costumam referir-se à memória no sentido de senso comum, enquanto Turing tem como referencial as máquinas.

Sendo assim, a proposta deste trabalho é a apresentação de algumas máquinas às quais há grande referência dentro da Ciência Cognitiva, bem como a especificação de suas memórias, a fim de se obter suporte, no contexto das máquinas, para que, futuramente, seja possível a verificação de algumas hipóteses levantadas em torno desse assunto, tal como a hipótese de Turing de que as máquinas podem ser modelos apropriados de processos cognitivos ao adquirirem suficiente capacidade de memória.

O desenvolvimento deste trabalho está dividido em quatro capítulos.

No Capítulo 1, Seção 1, enfatiza-se a importância dos modelos mecânicos como instrumentos explicativos da Ciência Cognitiva. Na Seção 2, é apresentado o ponto de vista de Turing de que as máquinas, desde que dotadas de memória suficiente, constituem bons modelos dos processos mentais inteligentes. As idéias de Turing recebem críticas de vários filósofos, e assim sendo, na Seção 3, é apresentada a descrença de Searle e Dreyfus quanto às possibilidades dos computadores efetivarem as hipóteses de Turing. Por fim, na Seção 4, enfatiza-se a memória humana como parâmetro de comparação para o estudo da memória nos modelos mecânicos.

Depois de comentada a questão da modelagem como método investigativo da Ciência Cognitiva, são apresentadas, em seqüência, duas máquinas abstratas que se distinguem entre si, em razão de sua capacidade de

memória. Ao analisar duas máquinas que, basicamente, se diferenciam pela capacidade de memória, poderá ser notado como, de fato, a memória influencia na capacidade da máquina de resolver problemas. Também, é analisada a própria noção de memória nessas máquinas, com o propósito de maior esclarecimento.

Dessa forma, o Capítulo 2 trata de duas máquinas abstratas. Na Seção 1, é apresentada a máquina de estados finitos, conhecida como a máquina fundamental da teoria dos autômatos, a qual é destacada por possuir memória finita. Na Seção 2, é apresentada a máquina de Turing, a qual abrange todas as capacidades da máquina de estados finitos, sendo porém, de natureza ilimitada. A Seção 3 desse capítulo enfatiza a importância da memória na capacidade de uma máquina resolver problemas, incluindo o fato de ser ela (a memória) a responsável pela distinção de dois tipos de máquinas: os autômatos finitos, como é a máquina de estados finitos, e os autômatos não finitos, como é a máquina de Turing.

No Capítulo 3, após a caracterização de duas máquinas abstratas, é feita a apresentação de uma máquina real, o computador, o qual possibilita a verificação ou ilustração de uma série de aspectos do funcionamento da mente. Essa máquina é considerada pelo fato de alguns pesquisadores acreditarem que aspectos cognitivos humanos possam ser explicados através de modelos computacionais. Para sua apresentação, é inicialmente considerada, na Seção 1, uma definição do termo computador que se mostra relevante para a Ciência Cognitiva, e também é feita a caracterização histórica dessa máquina. Em seguida, na Seção 2, é explicitada a categorização dos computadores, juntamente com seus componentes básicos, a fim de que se tenha uma boa compreensão de

unesp

termos essenciais e de seu modo de funcionamento. Por fim, na Seção 3, é feita uma análise da memória dos computadores. A noção de memória de máquina, particularmente da memória do computador, é a noção de memória que Turing possuía. Sendo assim, esta é melhor apresentada para que o referencial de Turing fique claramente explicitado.

No Capítulo 4, Seção 1, é feita uma apresentação dos princípios fundamentais e das características das Redes Neurais Artificiais, pelo fato de seus pesquisadores acreditarem que, por possuírem uma estrutura com mais plausibilidade biológica, talvez elas possam contribuir para uma simulação adequada de certos aspectos cognitivos. Além disso, são mostradas as principais diferenças entre os mecanismos simbólicos, usados pelos computadores, e os mecanismos físicos, usados pelas Redes Neurais Artificiais. Esta seção inclui, ainda, um breve histórico dessas redes. Na Seção 2, são comentados os componentes básicos das Redes Neurais Artificiais, bem como os seus mecanismos de aprendizagem, treino e memória. Por último, na Seção 3, é apresentada a ilustração de um modelo de rede neural neurofisiologicamente motivado na noção de memória associativa, a Rede de Hopfield. Também, são apresentadas as características da memória associativa, de forma que se possa perceber como a organização da memória, no modelo de Hopfield, parece ser próxima à memória humana.

Finalizando, nas Notas Finais, será feito um balanço do estudo da memória, tal como caracterizado nos capítulos anteriores, a fim de se verificar

unesp

em que sentido ela auxilia no entendimento das máquinas como instrumentos apropriados para simulação do comportamento inteligente.

Capítulo 1

MODELOS MECÂNICOS DA INTELIGÊNCIA

Este capítulo trata da estratégia que a Ciência Cognitiva utiliza na tentativa de explicar alguns processos cognitivos: o uso de modelos mecânicos. A abordagem foi organizada em quatro seções, sendo que, na Seção 1, enfatiza-se a importância dos modelos mecânicos como instrumentos explicativos da Ciência Cognitiva, já que, nessa ciência, procura-se explicar a atividade cognitiva criando-se modelos que a imitem. No entanto, será que as máquinas podem ser consideradas instrumentos apropriados para a simulação do comportamento inteligente? Na Seção 2, é visto que Turing parece acreditar que sim, que as máquinas constituem bons modelos do comportamento inteligente, desde que elas possam ser dotadas de suficiente memória. As idéias de Turing recebem críticas de vários filósofos, de modo que, na Seção 3, é comentada a postura de Searle e Dreyfus, contrária às convicções de Turing apresentadas na seção anterior. Por fim, na Seção 4, o objetivo é a inclusão de um referencial para a memória das máquinas, e esse referencial é a memória humana, uma vez que esta pode ser considerada o melhor sistema, conhecido, de memória em condições de ser simulada.

unesp

1.1 A elaboração de modelos como instrumentos explicativos da Ciência Cognitiva

O interesse pela natureza do conhecimento, segundo Gardner (1995), remonta aos antigos gregos. As idéias e teorias polêmicas existentes desde então " (...) continuaram a repercutir através da tradição intelectual do Ocidente" (Gardner, 1995, p.18). As discussões sobre o conhecimento foram fundamentadas, na Idade Média, principalmente na versão dada por Aristóteles, e, durante os períodos do Renascimento e do Iluminismo, nas idéias de pensadores como Descartes, Locke e Kant.

Sendo assim, "ao final do século XIX, tinha ocorrido uma proliferação de novas ciências e de especialidades filosóficas, várias delas pretendendo discutir a natureza da mente humana" (Gardner, 1995, p.18).

Continuando na investigação acerca do conhecimento, Gonzales *et al.* (1997, p.vii) afirmam que

> No século XX, sob o domínio da mentalidade científica e tecnológica, os esforços de *conhecimento do conhecimento* têm se voltado para o projeto de uma **ciência cognitiva**, que possa complementar, e eventualmente corrigir, através do uso do método experimental e da construção de artefatos tecnologicamente sofisticados, as teorias sobre a natureza e funcionamento de sistemas cognitivos. Coloca-se deste modo um novo projeto para filósofos e cientistas, que procuram unir esforços, na elaboração de uma abordagem inovadora dos problemas filosóficos tradicionais sobre o conhecimento.

O novo projeto a que se referem Gonzales *et al.*, que buscava fundar um novo campo do conhecimento, foi oficialmente reconhecido por volta de 1956, mas a utilização formal do termo Ciência Cognitiva se deu no início da década de 1970.

unesp

Gardner (1995, p.19) define Ciência Cognitiva " (...) como um esforço contemporâneo, com fundamentação empírica, para responder a questões epistemológicas de longa data – principalmente aquelas relativas à natureza do conhecimento, seus componentes, suas origens, seu desenvolvimento e seu emprego".

Apesar da tentativa de explicação da natureza do conhecimento se remontar aos gregos, a Ciência Cognitiva é aceita como uma ciência *radicalmente nova* (Gardner, 1995). Além de 'contar' com uma cooperação interdisciplinar de pesquisadores de várias áreas, "a Ciência Cognitiva diferencia-se das demais ciências cognitivas (por exemplo, da neurociência cognitiva) pela importância que atribui ao computador, concebido como instrumento apropriado para simular a atividade cognitiva" (Gonzales *et al.*, 1997, p.vii-viii).

Desse modo, ao se estudar Ciência Cognitiva atualmente, os cientistas cognitivos têm contado com a contribuição de métodos empíricos, os quais empregam a técnica de modelagem computacional, a fim de testar suas teorias e suas hipóteses, recorrendo, para isso, a várias ferramentas da matemática, da física e da computação.

Segundo Dupuy (1996), essas ferramentas com que os cientistas cognitivos estão equipados apenas proporcionaram uma evolução para as ciências cognitivas, pois tudo começou antes da existência do computador. Nesse sentido, ele afirma que:

> (...) essa metáfora [a metáfora do computador] não era absolutamente necessária e, ... as coisas na realidade começaram antes de existir o computador – ou mais precisamente, quando ele existia como objeto

> material técnico, mas ainda não se dispunha de uma teoria funcionalista desse objeto. Essa teoria que se nos tornou tão familiar, pela qual distinguimos o 'programa' (*software*) do 'material' (*hardware*) é um produto da revolução conceitual que assinala o advento das ciências cognitivas, e não sua origem. (Dupuy, 1996, p.26-7)

Embora a fundamentação empírica das hipóteses cognitivas não seja o marco inicial das ciências cognitivas, ela foi a responsável pelo tratamento científico das atividades cognitivas, dado que a investigação filosófica, e meramente especulativa, feita até o momento, passa a ser apoiada na experimentação (Soares, 1993).

Contudo, como afirma Dupuy (1996, p.22), " (...) há um fazer na atividade científica muito mais universal do que o que encontramos na experimentação (...) ", ele está se referindo à fabricação de modelos. Enquanto a filosofia e a história da ciência se preocupam com a natureza e os fundamentos do conhecimento científico, a construção de objetos sob a forma de modelos é uma atividade própria da ciência e lhe é essencial.

Assim, neste momento, torna-se importante destacar a distinção entre ciências cognitivas e Ciência Cognitiva. Sabendo-se que, dentre as várias ciências cognitivas encontra-se a Ciência Cognitiva, esta última tem como principal característica a utilização de métodos empíricos em modelos, e se concretiza através do pressuposto de que as máquinas representam bons modelos do funcionamento da mente humana.

De fato, a fabricação de modelos desempenha um importante papel na Ciência Cognitiva, pois nessa ciência explica-se a atividade cognitiva criando-se modelos. Além disso, a fabricação de modelos mecânicos possibilita

unesp

uma maneira mecanicista de tratar seu objeto de estudo, a mente. Portanto, não se pode tirar o modelo mecânico da Ciência Cognitiva.

Segundo Dupuy, na ciência, o termo modelo tem sentido inverso do que lhe é atribuído na língua corrente. Em seu uso comum, o significado de modelo é "ser exemplo para"; entretanto "o modelo científico é, inicialmente, uma imitação. Ele tem com a realidade o mesmo tipo de relação que um 'modelo reduzido' mantém com o objeto de que ele é a reprodução mais facilmente manipulável" (Dupuy, 1996, p.22).

Na tentativa de se elaborar uma definição de modelo não limitada à investigação científica, depara-se com uma grande dificuldade, visto que ele é a própria dinâmica da criação científica. Por isso, a solução de Dupuy (1996, p.23) é identificar o modelo " (...) como uma forma abstrata que vem encarnar-se ou realizar-se nos fenômenos".

Pelo fato de o modelo abstrair da realidade o sistema de relações funcionais e colocar em segundo plano as características dos elementos dos fenômenos, tais como o número, a identidade e a natureza dos elementos, há a possibilidade de se criar vários modelos diferentes para a imitação de um único fenômeno, obtendo-se resultados equivalentes.

Dado que o modelo científico é uma imitação que o homem faz da natureza, o cientista, com o domínio dessa atividade, passa a possuir sobre os modelos alguns controles que não seriam possíveis sobre os fenômenos reais, como por exemplo, o controle explicativo e preditivo. Esses controles só são possíveis graças ao poder da ferramenta matemática, pois esta permite a

exploração das propriedades do modelo com eficiência, rapidez e elegância, e isto a experimentação feita sobre o próprio fenômeno nunca alcançará. Igualmente possível com os modelos é o controle do poder criador da analogia, em que domínios diferentes, representáveis num mesmo modelo, permitem experiências interessantes entre eles.

Além disso, pode-se destacar um outro aspecto importante dos modelos na Ciência Cognitiva, o qual se expressa na afirmação de Dupuy de que só se conhece modelando. Segundo ele, "conhecer é produzir um modelo do fenômeno e efetuar sobre ele manipulações ordenadas. Todo conhecimento é reprodução, representação, repetição, simulação" (Dupuy, 1996, p.27).

Importante também, na tentativa de explicações das atividades cognitivas, é o *papel* da simulação científica. Conforme Dupuy (1996, p.38), " (...) a simulação é uma forma particular de modelização que consiste em reproduzir, no mais das vezes no computador, o funcionamento de um sistema; trata-se, pois, de uma atividade de conhecimento".

Sendo a mente, algo tão controverso, o objeto de estudo da Ciência Cognitiva, entende-se que a técnica de construção de modelos que simulem as hipóteses acerca de seu funcionamento possa auxiliar na elucidação de aspectos relevantes para essa ciência.

Dadas as propriedades dos modelos, o modelo cognitivo pretende ser a explicação de algumas atividades cognitivas através de processos mecânicos. As explicações da natureza e da estrutura dos processos mentais pretendem ser, dessa forma, realizadas através da modelagem.

unesp

A fundamentação teórica do modelo cognitivo está baseada na idéia de que a cognição é essencialmente mecânica, caracterizando-se por formas de organização mecânica. Sendo assim, uma pressuposição utilizada na Ciência Cognitiva é a de que os seus modelos constituem uma forma de organização que incorpora hipóteses mecânicas sobre o funcionamento da mente. De acordo com a definição do senso comum, *mecânico* pode ser definido como "relativo a processos em que se podem determinar uma série de fases subordinadas e dependentes umas das outras; em que cada momento é determinado por condições antecedentes invariáveis" (Ferreira, 1975, p.902). Neste trabalho, a noção do termo *mecânico* é usada no sentido comum.

Na utilização de modelos cognitivos como método de análise de processos cognitivos, são dois os paradigmas desenvolvidos: os modelos cognitivos simbólicos e os modelos cognitivos conexionistas.

Os modelos cognitivos *simbólicos* incorporam hipóteses sobre a cognição humana através de um instrumento experimental importante: o computador. Esses modelos podem ser entendidos como um conjunto de instruções mecânicas, ou seja, um algoritmo.

Por outro lado, os modelos cognitivos *conexionistas* são definidos por Caudill & Butler (1992, p.3), como sendo

> (...) um sistema de processamento de informação que é **não algorítmico**, não digital, e intensivamente paralelo. Não é um computador no sentido em que este é pensado atualmente, nem é programado como um computador. Em vez disso, ele consiste em um número de processadores muito simples e altamente interconectados, chamados neurônios, os quais são análogos às células neurais biológicas, ou neurônios, do cérebro.

unesp

Os modelos conexionistas são desenvolvidos devido à hipótese de os mecanismos da cognição humana consistirem na interação de um grande número de neurônios (unidades de processamento). Sendo assim, alguns conexionistas, através dos modelos cognitivos, procuram entender a cognição humana pela construção de redes de unidades de processamento.

Como pôde ser visto até o momento, dadas as características e propriedades dos modelos, estes constituem o grande empreendimento da Ciência Cognitiva. Considerando que essa ciência procura explicar aspectos da mente ligados à inteligência através de modelos, a seguir, será feita a descrição de alguns dos empreendimentos elaborados nessa área de estudos.

1.2 As idéias de Turing sobre modelos mecânicos inteligentes

Conforme visto na seção anterior, a Ciência Cognitiva desenvolve as hipóteses acerca de sistemas cognitivos utilizando-se da construção de modelos que simulem essas hipóteses.

Desse modo, entende-se que é com o intuito de modelar aspectos do conhecimento humano que o cientista britânico Alan Turing, em artigo publicado em 1950, propôs a questão "Podem as máquinas pensar?".

Para analisar se uma máquina podia ou não pensar sem se preocupar demasiadamente com definições do significado de "máquina" e "pensar", Turing elaborou um teste simples, o qual ele chamou de Jogo da Imitação.

unesp

Resumidamente, esse jogo tem a participação de três elementos com objetivos distintos:

ELEMENTO	OBJETIVO	SÍMBOLO
Interrogador	Determinar qual é o homem e qual é a mulher	C
Homem	Tentar induzir o interrogador a uma identificação errada	A
Mulher	Ajudar o interrogador dando respostas verdadeiras	B

Durante o jogo, o interrogador, permanecendo em local separado dos outros dois elementos, deve formular perguntas, e estes (tanto o homem quanto a mulher), de acordo com seus objetivos, devem dar as respostas.

Assim, a partir desse jogo, pode-se substituir a pergunta original "Podem as máquinas pensar?" pelas questões:

> 'O que acontecerá quando uma máquina ocupar o lugar de A nesse jogo?' Será que o interrogador decidirá erroneamente com a mesma freqüência, quando o jogo é jogado dessa forma, do que quando o fazia ao tempo em que o jogo era jogado entre um homem e uma mulher? (Turing, 1950, p.51)

Na segunda fase do jogo ocorre uma mudança nos elementos participantes. Estes passam a ser: um interrogador, uma máquina e um ser humano. A questão que surge nesta fase é: será a máquina capaz de confundir o interrogador tanto quanto o homem é capaz? Conforme Turing, caso a máquina fosse confundida com o ser humano, ela poderia ser considerada *pensante*.

Esse trabalho de Turing, questionando se as máquinas podem ou não pensar, influenciou pesquisadores a concentrarem seus esforços na tentativa de explicar alguns aspectos do comportamento inteligente humano usando modelos mecânicos.

unesp

Diante da tentativa de modelizar aspectos do comportamento humano inteligente, deve-se questionar a adequação dos modelos ao estudo da mente: seriam os modelos mecânicos bons modelos de mente?

Turing desenvolveu seu artigo de 1950 procurando fornecer subsídios para uma resposta positiva à sua pergunta sobre a possibilidade de construção de máquinas pensantes. No entanto, ao se referir à máquina, Turing (1950, p.53) esclareceu que não estava interessado em todo tipo de técnica de engenharia, e colocou uma restrição ao tipo de máquina implicada no jogo:

> (...) o atual interesse por 'máquinas pensantes' foi despertado por um tipo particular de máquina, geralmente chamado de 'computador eletrônico' ou 'computador digital'.

Sobre essa restrição, Turing fez 2 considerações:

- Tal restrição não é drástica, o que envolve uma explicação da natureza e das propriedades dos computadores digitais (explicações contidas no capítulo 3 deste trabalho);
- A identificação de processos informacionais dos computadores digitais com os processos de pensamento humano só será insatisfatória se os computadores digitais se mostrarem incapazes de boa demonstração no jogo da imitação.

Embora já existissem computadores digitais funcionando (o que pode ser verificado no histórico apresentado no capítulo 3 desta dissertação), para Turing, o importante era saber se havia possibilidade de existir um computador capaz de ter bom desempenho no jogo da imitação, e não se todos os computadores digitais teriam bom desempenho nesse jogo.

unesp

Segundo Turing, os computadores digitais podem ser classificados como máquinas de estado discreto, as quais se movimentam por saltos de um estado bem definido para outro. Tais máquinas podem ser definidas por tabelas, desde que tenham um número finito de estados possíveis. Dado o estado inicial da máquina e os sinais de entrada, pode-se predizer todos os seus estados futuros. Além disso, ele diz que os computadores digitais possuem a propriedade especial de imitar qualquer máquina de estado discreto e, por esse motivo, são considerados máquinas universais.

Com essa caracterização de *computador digital*, Turing tentou conduzir a aceitação da hipótese de que os computadores digitais podem de fato imitar, de modo muito próximo, as ações de um computador humano. Para ele, em tal situação de estudo, "o computador humano deve seguir regras fixas, não tendo autoridade para se desviar delas em nenhum detalhe" (Turing, 1950, p.54), podendo-se supor que essas regras sejam fornecidas por um livro, e este pode ser alterado sempre que se confie um novo trabalho a seu operador. Turing ainda esclareceu que o "livro de regras" usado pelo computador humano é uma ficção, dado que os seres humanos se lembram do que precisam fazer, de acordo com suas necessidades. Já no computador digital, esse "livro de regras" é substituído por uma tabela real de instruções, que é a programação da máquina, necessária para que ela possa realizar alguma operação.

Nessa interpretação, "'programar uma máquina que realize a operação A' significa colocar a tabela de instruções apropriada dentro da máquina, de tal forma que esta realize A" (Turing, 1950, p.56). E assim,

unesp

> Se alguém quiser construir uma máquina que imite o comportamento do computador humano em alguma operação complexa, terá de perguntar-lhe como ela é feita e então traduzir a resposta na forma de uma tabela de instruções. A construção de tabelas de instruções é comumente descrita como 'programação'. (Turing, 1950, p.56)

Através das considerações até aqui apresentadas, ficam claras as convicções favoráveis de Turing a respeito da possibilidade de se criar máquinas inteligentes.

No entanto, a polêmica existente em torno da questão proposta por Turing sobre a natureza mecânica da inteligência gerou uma série de objeções e argumentos contrários às suas convicções. Turing considerou algumas opiniões opostas às dele, em seu artigo de 1950, e aproveitou para refutá-las, argumentando ser possível uma máquina pensar desde que ela possua suficiente capacidade de memória. Sendo assim, ele rebateu as objeções apresentadas apontando a falta de memória[1] como o motivo das máquinas atuais ainda não conseguirem realizar tal habilidade.

Para Turing (1950, p.67), "muitas limitações advêm da capacidade de memória assaz reduzida dessas máquinas". Ele reformulou, então, sua questão "Podem as máquinas pensar?" para uma questão, segundo ele equivalente, na qual introduziu as condições que acreditou serem necessárias para uma resposta positiva à sua questão original:

---

1 Ao apontar a falta de memória como a dificuldade encontrada para que seja possível a uma máquina pensar, fica clara a idéia de que Turing não está se referindo a modelos abstratos, os quais não 'sofrem' necessariamente problemas de memória, mas, sim, se refere a modelos empíricos reais, para os quais a quantidade limitada de memória é uma preocupação concreta. De fato, a memória a que Turing se refere é a memória dos computadores.

unesp

> Fixemos nossa atenção num computador digital particular C. É verdade que modificando-se este computador para obter uma memória adequada, aumentando-lhe convenientemente a velocidade de ação, e provendo-o de um programa apropriado, C pode ser preparado para desempenhar satisfatoriamente o papel de A no jogo da imitação, sendo o papel B desempenhado por um homem? (Turing, 1950, p.60)

Reformulada a questão, Turing (1950, p.61), então, respondeu-a, deixando claras suas convicções:

> Acredito que, dentro de cerca de 50 anos, será possível programar computadores, com uma capacidade de memória de cerca de 10^9, para fazê-los jogar o jogo da imitação tão bem que um interrogador médio não terá mais de 70% de probabilidade de chegar à identificação correta, após 5 minutos de interrogatório. A pergunta original "Podem as máquinas pensar?" é, a meu ver, insignificante demais para merecer discussão.

Quando respondeu às objeções acerca de suas convicções, Turing já possuía conhecimento sobre a máquina que estava tratando – o computador digital. Sabia que a tabela de instruções dentro da máquina, ou seja, seu programa, fica na memória, e que a memória dos computadores digitais é finita.

Tomando-se uma das objeções às idéias de Turing, a de Lady Lovelace, por exemplo, nela encontra-se que:

> (...) 'a Máquina Analítica não tem nenhuma pretensão de *criar* o que quer que seja. Pode fazer tudo quanto saibamos ordenar-lhe que faça (o grifo é de Lady Lovelace).' Esse enunciado é referido por Hartree (1949), que acrescenta: 'Isso não implica que não seja possível construir equipamento eletrônico que 'pense por si mesmo', ou no qual, em termos biológicos, alguém possa estabelecer um reflexo condicionado, que serviria de base para a 'aprendizagem'. A questão, muito estimulante, de se isso é ou não possível em princípio, foi sugerida por algum desses recentes desenvolvimentos. Mas não parece que as máquinas construídas ou projetadas naquela época tivessem essa propriedade. (Turing, 1950, p.70)

unesp

Turing declarou-se de acordo com Hartree, não pelo fato de ele concordar que as máquinas em questão não fossem criativas, mas sim porque Lady Lovelace não era satisfatoriamente convincente a ponto de fazer crer que as máquinas tivessem essa propriedade de criar. Turing acreditava ser possível que, de alguma forma, as máquinas implicadas no Jogo da Imitação pudessem ser criativas.

Assim, apoiado em sua convicção de que uma máquina de estado discreto possui a capacidade de criar, Turing (1950, p.70-1) rebateu as idéias de Lady Lovelace dizendo que:

> A Máquina Analítica era um computador digital universal, de forma que, caso sua capacidade de memória e sua rapidez fossem adequadas ela poderia, por meio de uma programação conveniente, ser levada a imitar a máquina em questão. Este argumento provavelmente não ocorreu à Condessa ou a Babbage. De qualquer modo, eles não tinham a obrigação de alegar tudo quanto pudesse ser alegado.

Nessa citação Turing tenta mostrar que, sob determinadas condições (capacidade de memória suficiente, bem como rapidez e programação suficiente), a máquina pode desenvolver sua capacidade de criar.

No parecer de Turing sobre a objeção aqui apresentada, pode-se também observar sua opinião sobre o fato de uma máquina não possuir alguns comportamentos atribuídos ao ser humano. Ainda, segundo ele, "a crítica de que uma máquina não pode ter muita diversidade de comportamento é simplesmente uma maneira de dizer que ela não pode ter muita capacidade de memória" (Turing, 1950, p.70).

Além de deixar claro que o maior entrave à possibilidade de uma ´máquina pensante´ deve-se à sua limitada capacidade de memória, Turing

(1950, p.76-7) ainda fez uma previsão da capacidade de memória necessária para que tal limitação deixe de ser problema para o Jogo da Imitação:

> As estimativas da capacidade de memória do cérebro variam de 10^{10} a 10^{15} dígitos binários (bits). Inclino-me para valores mais baixos e acredito que apenas uma pequena fração dessa capacidade seja usada para os tipos superiores de pensamento. A maior parte dela é provavelmente usada para a retenção de impressões visuais. Eu ficaria surpreso se mais que 10^{9} bits fossem necessários para atuação satisfatória no Jogo da Imitação, pelo menos contra um homem cego. .... Uma capacidade de memória de 10^{7} bits seria uma possibilidade muito prática, mesmo pelas técnicas atuais.

Portanto, para Turing, conforme se ressaltou várias vezes nesta dissertação, a dificuldade encontrada nos modelos mecânicos da inteligência é a falta de memória das máquinas. Sendo assim, resolvendo-se o problema da capacidade de memória, as máquinas poderiam ser bons modelos explicativos do comportamento inteligente.

1.3 O olhar crítico dos filósofos (a respeito das idéias de Turing)

Como uma conseqüência à resposta afirmativa sugerida por Turing à questão "Podem as máquinas pensar?", a tese mecanicista da mente é fortalecida, levando pesquisadores da Inteligência Artificial (IA) a sugerirem que aspectos inteligentes da mente humana podem ser entendidos através da analogia com um programa de computador.

No entanto, ainda que Turing insista em sua idéia de que os modelos mecânicos, em particular o computador digital, possa(m) ser bom(s) modelo(s) para explicar o pensamento inteligente, há vários críticos que questionam se tal propriedade da mente é ou não similar àquela do computador.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

Existem pesquisadores céticos que dizem que os modelos relevantes para explicar determinadas características da mente não são do tipo de um computador, e que, inclusive, elas não podem ser simuladas naquilo que é relevante para um computador.

Entre os críticos do projeto da IA encontra-se John Searle. Este dá o nome de IA forte à teoria computacional da mente na qual esta é considerada um programa de computador. De acordo com Searle (1997, p.285), "a idéia básica do modelo da mente baseado no computador é que a mente é o programa, e o cérebro, o *hardware* de um sistema computacional".

Na definição de Searle (1998, p.37),

> Um computador é (...) um equipamento que manipula símbolos formais. Estes, por sua vez, são freqüentemente descritos como 0's e 1's, ainda que qualquer símbolo antigo sirva tão bem quanto. O autor da concepção moderna de computação, Alan Turing, aborda este ponto afirmando que um computador pode ser visto como uma máquina que contém um cabeçote que lê uma fita. Na fita estão impressos 0's e 1's. A máquina pode realizar exatamente quatro operações. ... Ela realiza estas operações de acordo com um conjunto de regras do tipo 'sob a condição C realize ato A'. Estas regras são chamadas de *programa*. Computadores modernos funcionam codificando informação codificada em impulsos elétricos e processando tais informações de acordo com as regras do programa.

Searle (1997, p.37) considera como característica principal desse mecanismo computacional o fato de ele ser "totalmente definido em termos da manipulação de símbolos". Neste sentido, a computação é apenas "um conjunto sintático de operações", de modo que "os únicos atributos dos símbolos que importam para a implementação do programa são os formais ou os sintáticos".

O que Searle pretende esclarecer com o que foi dito é que os programas do computador são estruturas sintáticas, consistindo em regras para se

manipular símbolos, e que o programa implementado consiste nas manipulações sintáticas.

No entanto, ele afirma que "a mente consiste em algo mais do que a mera manipulação de símbolos formais. A mente tem conteúdos" (Searle, 1998, p.37). Neste ponto, Searle quer deixar claro o fato de a mente ter uma capacidade semântica; mais do que símbolos formais não interpretados, ela possui conteúdo semântico.

Sendo as características do programa do computador distintas das apresentadas pela mente, Searle (1998, p.38-9) conclui:

> A mente não poderia ser apenas um programa de computador, já que os símbolos formais do programa de computador , tomados isoladamente, não são suficientes para garantir a presença do conteúdo semântico que ocorre na mente. ... A sintaxe não é a mesma coisa que a semântica, nem é, por si só, suficiente para garantir um conteúdo semântico. Conseqüentemente, programas não são mentes.

Antes que uma máquina, por exemplo o computador, seja tomada como instrumento explicativo de características da mente na Ciência Cognitiva, espera-se encontrar em que lugar, nas manipulações de símbolos formais, se encaixam os fenômenos mentais, muitos deles conscientes e ocorrendo no cérebro.

Searle (1998, p.43-4) continua sua refutação à IA forte dizendo:

> (...) a IA forte, que se orgulha de seu 'materialismo' e da visão do cérebro como uma máquina, não chega nem perto de ser suficientemente materialista. O cérebro é, de fato, uma máquina, uma máquina orgânica; e seus processos, como as descargas neuronais, são processos orgânicos mecânicos. Mas a computação não é um processo automatizado como as descargas neuronais ou a combustão interna. Pelo contrário, a computação é um processo matemático abstrato que existe apenas em relação a observadores e intérpretes conscientes. Observadores como nós tem encontrado meios de implementar a computação como máquinas elétricas à base de silício, mas isso não torna a computação algo elétrico ou químico.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

Apesar de rejeitar a IA forte, Searle diz aceitar a versão da teoria computacional da mente a qual ele designou de IA fraca. Nessa versão " (...) o computador é uma ferramenta útil para fazer simulações da mente, da mesma forma como é útil nas simulações de quase qualquer coisa que descrevamos com precisão, tal como as condições meteorológicas ou o fluxo de dinheiro na economia" (Searle, 1998, p. 36-7).

Com base na tese de Church de que um sistema que pode ter uma caracterização precisa pode ser simulado em um computador digital, Searle acredita que seja possível haver alguma descrição do cérebro tal que, de acordo com ela, se possa fazer simulações computacionais das operações do cérebro. Ou seja, satisfeita a condição da tese de Church, tanto o cérebro quanto qualquer outro sistema bem descrito, como, por exemplo, o comportamento do mercado de valores de Nova York, pode ser simulado em um computador digital.

Embora acredite na possibilidade de simulação computacional de alguns processos cerebrais, a idéia da simulação traz a Searle uma certa inquietação: o que as simulações realmente explicam sobre os sistemas simulados? Essa inquietude é ilustrada por ele através do exemplo de um programa de edição de textos de um computador digital:

> (...) o programa de edição de texto simula uma máquina de escrever melhor do que qualquer programa de IA que conheço simula o cérebro. Porém, nenhuma pessoa sensata vai pensar: 'Finalmente compreendemos como as máquinas de escrever funcionam; são implementações de programas de edição de texto'. Simplesmente não é o caso, em geral, que simulações computacionais proporcionem explanações causais dos fenômenos simulados. ... Não supomos, em geral, que as simulações computacionais de processos cerebrais nos dêem alguma explanação no lugar de ou além de explicações neurobiológicas sobre como o cérebro efetivamente funciona. E em

> geral não usamos 'X é uma simulação computacional de Y' para designar uma relação simétrica. Isto é, não supomos que porque o computador simula uma máquina de escrever, então a máquina de escrever simula um computador. Não supomos que, porque um programa meteorológico simula um furacão, logo a explanação causal do comportamento do furacão é fornecida pelo programa. Então por que deveríamos abrir uma exceção para esses postulados em que processos cerebrais desconhecidos estão envolvidos? Há motivos satisfatórios quaisquer para abrir a exceção? Searle (1997, p.311-2)

Retomando a idéia de Dupuy (1996) a respeito da simulação, foi visto que ela é uma forma particular de modelização. Assim, ao questionar a eficácia das simulações na explicação dos processos mentais, Searle mostra também sua inquietação em relação aos métodos da Ciência Cognitiva. De acordo com a Seção 1 deste capítulo, os modelos constituem o grande empreendimento dessa ciência, já que nela procura-se explicar a atividade cognitiva criando-se modelos. Segundo Searle, sendo a simulação ineficiente para a explicação do sistema, a Ciência Cognitiva, com seu método de pesquisa atual, não conseguirá alcançar seu objetivo.

Quanto à questão "Podem as máquinas pensar?", para Searle (1998, p.40),

> Os cérebros humanos às vezes computam. Eles somam 2+2 e obtêm 4, por exemplo. Dessa forma, levando em conta uma certa definição de computador, os cérebros são computadores porque computam. Logo, alguns computadores podem pensar – o seu cérebro e o meu, por exemplo.

Searle (1998, p.40) continua sua afirmação dizendo que "o cérebro é uma máquina, uma máquina biológica, e pode pensar. Portanto, ao menos algumas máquinas podem pensar..."

Neste ponto, pode-se notar, entre Turing e Searle, uma diferença na interpretação do tipo de máquina envolvida na questão "Podem as máquinas

pensar?". Embora os dois sugiram uma resposta afirmativa à questão, o tipo de máquina considerada por Turing não é aquele considerado por Searle.

Turing, em seus argumentos para responder à questão citada, utiliza-se do computador digital real, concreto. Como foi visto, sua idéia é de que os computadores digitais concretos poderiam ser planejados para realizar quaisquer operações que pudessem ser feitas por um computador humano. Sendo assim, se o computador humano pode pensar, por que o computador digital não pode?

Já Searle, ao apresentar as razões que o levam a acreditar que as máquinas podem pensar, utiliza-se da definição de que os cérebros são máquinas computadoras porque computam. Portanto, para ele, o tipo de máquina envolvido é uma máquina biológica, o cérebro.

Diante dessas considerações, pode-se assim questionar: o cérebro é um computador digital?

Considerando a interpretação de Searle, essa pergunta não seria satisfatoriamente respondida ao se dizer que sim, ou seja, afirmar que o cérebro é um computador digital, da mesma forma que o coração, o sistema solar, o estado de Kansas são também computadores digitais. Segundo Searle (1997, p.297), é muito simples dizer apenas "sim, cérebros são computadores digitais porque tudo é um computador digital". Sob este ponto de vista, qualquer sistema viraria máquina. Assim, não existiria fenômeno físico que não fosse possível computar, dependendo do sentido que se desse a ele. A idéia defendida por Searle é de que a

unesp

mente, de fato, computa. Contudo, o computador, pelo fato de computar, não é a mente.

Para que a questão acima seja melhor esclarecida, existem outras que necessitam ser primeiramente respondidas, tais como: o que é exatamente um computador digital? Porém, sabe-se que há pouca concordância teórica entre os profissionais no que diz respeito a algumas questões fundamentais sobre a natureza das máquinas utilizadas nos modelos, o que, por vezes, provoca uma precipitação na resposta a certos questionamentos. Muitas controvérsias seriam evitadas se houvesse maior consenso em torno das definições dessas máquinas. Essas dificuldades reforçam o objetivo deste trabalho, que inclui o levantamento de subsídios, no contexto das máquinas, para que certas discussões possam ser futuramente retomadas.

Outro crítico que pode ser destacado por sua postura contrária à inteligência artificial é Hubert Dreyfus. Em seus escritos, ele enfatiza as diferenças fundamentais entre seres humanos e computadores. Segundo ele, a inteligência artificial nunca poderá explicar satisfatoriamente a natureza e a complexidade da inteligência humana. De forma radical, Dreyfus acredita que as técnicas de programação utilizadas na simulação demonstram pouca ou nenhuma semelhança com os processos mentais humanos.

Em seu livro de 1972, *What computers can't do:* a critique of artificial reason, Dreyfus declara que o comportamento humano não pode ser descrito através de regras lógicas formais. Ele ressalta que o computador jamais poderá assemelhar-se ao homem devido ao comportamento indeterminado deste

unesp

último, ou seja, ele entende que a vida é o que os próprios humanos fazem dela, e não algo definido *a priori*. A principal dificuldade apontada por Dreyfus está no fato de, em suas convicções, o pensamento humano depender da biologia do corpo humano. Dessa forma, aspectos biológicos, sensoriais e sociais do humano são relevantes ao se tratar da inteligência, e, por isso, as máquinas não são bons modelos do comportamento humano.

Em 1992, após o advento dos modelos conexionistas, Dreyfus, através de seu livro *What computers **still** can't do:* a critique of artificial reason, lança novos argumentos contra a inteligência artificial, nos quais, novamente, enfatiza que os seres humanos são seres únicos no que diz respeito à inteligência. Segundo ele, os conexionistas, ao proporem uma estrutura mais próxima do biológico, pretendem criar um tipo de corpo. No entanto, essa criação ainda não corresponde à necessidade almejada, pois não abarca a complexidade do corpo humano biológico imerso no contexto social.

Para ele, o corpo tem um papel importante no comportamento inteligente, pois a habilidade humana para comportar-se inteligentemente é vista como uma *atividade corporal* que se desenvolve para satisfazer as necessidades do corpo. Conforme essas necessidades, fatos específicos são organizados, tornando-se tanto acessíveis, quanto relevantes e significantes ao homem.

Por outro lado, a inteligência artificial procura abstrair esses fatos da maneira como são organizados, e usá-los, de modo já produzido, para simular o comportamento inteligente. Dreyfus questiona se existe algum modelo conexionista, entre os propostos até hoje, que possa materializar procedimentos

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

com um grau de abstração apropriado. Outro problema encontrado é que, fatos específicos, fora do contexto, tornam-se simplesmente uma imensa quantidade de dados difícil de ser 'trabalhada'. Segundo Dreyfus (1992, p. 281-2)

> Não existem outras técnicas de processamento de dados, até o presente, além da acumulação de fatos, e, uma vez que as suposições filosóficas tradicionais sustentando o trabalho na inteligência artificial têm sido colocadas em questão, não há razão para supor que técnicas digitais de armazenamento e recuperação de dados serão sempre suficientemente poderosas para enfrentar o montante de dados gerados quando tentamos tornar claro nosso conhecimento do mundo. Desde que os dados sobre o mundo podem ser infinitos e a formalização do nosso modo-de-vida pode ser impossível, seria mais racional supor que técnicas de armazenamento digital nunca possam chegar a tal tarefa.

Uma das dificuldades apontadas por Dreyfus em sua análise é a de que necessidades humanas indeterminadas pretendem ser simuladas em uma máquina, cujo único modo de existência é uma série de estados determinados. Além disso, dado que os seres humanos também produzem fatos, tem-se que os próprios fatos podem ser mudados pelas revoluções conceituais, individuais e culturais.

Não existindo fatos fixos e imutáveis, o problema tecnológico suposto por Turing acerca da dificuldade no armazenamento de dados, ou seja, a falta de memória das máquinas, deixa de ser primordial, uma vez que não seria esse o elemento essencial para o sucesso da modelagem da inteligência pelas máquinas. Tal sucesso, na visão de seus críticos, dependeria da estrutura biológica e social dos organismos imersos em uma coletividade.

unesp

1.4 A memória humana como referencial para os modelos mecânicos

Ao propor um estudo da memória cuja finalidade é verificar o limite de sua relevância na simulação dos processos mentais em modelos mecânicos, tem-se como idéia primordial a de fazê-lo a partir do que se conhece a respeito da memória humana.

A razão está em que, *a priori*, a memória humana é o único tipo de memória que se tem disponível no cotidiano e, por ser considerada bem sucedida, é aquela que se encontra em condições de ser simulada. Além disso, é razoável supor que, seja o que for um sistema de memória de uma máquina, ele é algo parecido com o sistema de memória humana, pois, se assim não fosse, dificilmente o termo *memória* seria utilizado para designar o local de armazenamento de dados na máquina. . Inclusive, a maioria das propostas de memória em máquinas é dada através de comparação, pelo menos implícita, com a memória humana.

À primeira vista, o sistema de memória humana pode ser considerado um sistema capaz de armazenar e recuperar informações de acordo com suas necessidades (Caudill & Butler, 1990) e, assim, pode parecer um sistema simples, o que não é verdade. São inúmeras as propriedades desse sistema e algumas delas ainda precisam ser conhecidas. Não é simples a compreensão de como milhares de conexões são aptas a captar, armazenar e recuperar dados, enquanto outros dados são simplesmente perdidos ou esquecidos.

unesp

Embora, atualmente, ainda se prossigam as tentativas de entendimento da memória humana, e sejam inúmeros os dados a se conhecer sobre ela, funções como a de retenção e recuperação da informação na memória são propriedades reconhecidas como sendo de grande relevância no sistema de memória humana.

Além dessas, existem outras funções que podem ser admitidas como importantes para a memória e que *oferecem* dificuldades quando consideradas na elaboração dos modelos mecânicos. Como exemplo, pode-se citar o esquecimento e o aspecto sócio-cultural.

O **esquecimento** constitui uma questão básica relacionada à memória, pois, embora muito do desempenho humano para executar uma tarefa dependa de sua capacidade de acumular informação, é esquecendo que se pode evitar que a memória humana acabe por se tornar algo tão inútil quanto um "depósito de lixo". O neurofisiologista Luria (1988), em um de seus trabalhos mais famosos, escreveu sobre um homem com uma habilidade incomum: nunca esquecia as informações guardadas em sua memória. Ele era capaz de se lembrar de tudo, literalmente. Luria acompanhou esse homem por mais de 20 anos e sua pesquisa resultou no livro *The Mind of a Mnemonist*: a little book about a vast memory. Nesse livro ele relata os fatos que ocorriam com ´S.´, nome dado pelo autor ao paciente que não conseguia deixar de reter, na memória, tudo o que via e ouvia.

Com uma memória tão prodigiosa, poder-se-ia esperar que 'S.' tivesse facilidade para solucionar problemas, no entanto, o não esquecimento

unesp

trazia-lhe uma série de obstáculos e causava-lhe problemas até mesmo com coisas simples. Por exemplo, ele não conseguia entender como uma só palavra podia ter mais de um sentido. Também era incapaz de fazer abstrações, pois como lembrava de cada coisa individualmente com a mesma nitidez, não conseguia lidar com agrupamentos em categorias.

Para um leigo que casualmente observasse 'S.', este pareceria possuidor de uma memória sem fim. No entanto, Luria admitiu que a memória dele não devia ser infinita. Ele não conseguiu, entretanto, apresentar uma quantificação definida para essa memória.

Fica ainda a questão: Por que nem sempre as informações armazenadas podem ser recuperadas? Apesar de ser grande a falta de controle que o homem tem sobre sua memória, quanto ao que ele deseja que seja lembrado ou esquecido, são várias as hipóteses que relacionam memória e esquecimento. Em Soares *et al.* (1997) podem ser encontradas algumas das mais importantes:

- *Hipótese da debilitação*: considera os engramas, que são traços hipotéticos da memória no cérebro, como sendo a base física da memória no Sistema Nervoso Central. O esquecimento vem pelo fato de os engramas serem extintos com o passar do tempo. Não é considerada uma boa hipótese, já que os fatos não a comprovam – pessoas idosas podem se lembrar muito bem de fatos antigos e se esquecer daqueles mais recentes.
- *Hipótese do desuso*: considera que as informações presentes na memória são perdidas quando não são utilizadas. Também não é uma hipótese comprovada

pelos fatos, já que lembranças antigas podem ser recuperadas mesmo quando não são utilizadas constantemente.

- *Hipótese da interferência*: considera que as informações adquiridas mais recentemente podem interferir naquelas anteriormente armazenadas e vice-versa. Esta hipótese parece mais plausível que as duas anteriores, pelo fato de a memória humana ter um caráter dinâmico. Sendo assim, quando novas idéias sobre um fato são aprendidas, elas poderão modificar o que foi aprendido anteriormente a respeito dele.
- *Hipótese de Atkinson e Shiffir*: considera que as informações obedecem a um fluxo não automático (isto é, um fluxo que depende de fatores como atenção e motivação do sujeito) entre os elementos que compõem o sistema da memória. Primeiramente, as informações chegam ao ser humano através de estímulos sensoriais. Uma parte dessas informações é selecionada e armazenada na memória de curto prazo. Se forem repetidas sem interferências, serão mantidas aí por um tempo, sendo que algumas dessas informações são passadas para a memória de longo prazo; enquanto, outras, são perdidas. Esta hipótese pode ser considerada a explicação mais completa da ligação memória e esquecimento, pois envolve os três elementos básicos que compõem o sistema de memória humana (a memória sensorial, a de curto e a de longo prazo), englobando também seu caráter dinâmico.

Quanto ao **aspecto sóciocultural**, Vygotsky (1994) aponta-o como um dos fatores importantes para a memória, justificando a importância da interação com o ambiente – ponto central na compreensão da relação entre

memória e aprendizagem. A ênfase na relevância dos fatores sócioculturais como elemento estruturador do universo cognitivo humano constitui, de acordo com Del Masso (2000, p.86), a grande contribuição para os estudos da memória nos processos de aprendizagem. A autora argumenta, ainda, que:

> Idealmente, a parte física da memória operativa deveria estar em harmonia com os fatores sociais, históricos, culturais, entre outros, quase que num processo de retroalimentação (estabelecendo uma via de mão dupla essencialmente dinâmica entre o indivíduo e o ambiente.

O que Vygotsky pretende deixar claro é que o desenvolvimento das funções cognitivas dependem, necessariamente, do meio externo social, uma vez que essas funções podem ser influenciadas e modificadas pela interação com esse meio. Fica, portanto, explícita, a importância significativa atribuída ao ambiente para o estudo da memória.

Tendo em vista que o objetivo deste trabalho é o estudo da memória nos modelos mecânicos, e não o estudo da memória humana, esta seção foi incluída pelo fato de a Ciência Cognitiva elaborar a maioria dos modelos de processos cognitivos com base no padrão humano. Não foram e nem serão aqui especificados detalhes a respeito dos elementos que compõem o sistema de memória humana. No entanto, conforme já foi dito, o ser humano será, em determinadas circunstâncias deste trabalho, tomado como parâmetro de comparação para o estudo dos modelos mecânicos. Assim, serão encontrados breves paralelos entre a memória das máquinas e a memória humana, sempre com o intuito de esta última servir como *pano de fundo*, ou seja, um referencial para nossa investigação.

Conforme visto neste capítulo, Turing insiste na hipótese de que a inteligência é uma capacidade de memória. Mas, qual é o conceito de memória a que Turing se refere? Foi mostrado, na Seção 2, que Turing parece estar se referindo à noção da memória de máquina, a única que lhe era familiar. Será que essa noção é apropriada para entender a inteligência humana? A fim de levantar subsídios para verificar em que sentido a memória das máquinas auxilia na meta cognitivista de utilizá-las como modelos apropriados de inteligência humana, são comentadas, nos próximos capítulos, noções de máquinas consideradas como instrumentos explicativos de aspectos cognitivos humanos. Inicialmente, o estudo se concentra em duas máquinas abstratas que se distinguem por suas capacidades de memória, a fim de que se possa esclarecer a questão da influência da memória na capacidade mecânica de resolução de problemas.

unesp

Capítulo 2

MÁQUINAS ABSTRATAS

Neste capítulo são apresentadas duas máquinas, as quais permitem que os computadores sejam pensados de forma abstrata, sem que haja a necessidade de se fazer referência a qualquer tipo específico de máquina. Sendo assim, são aqui estudadas duas célebres máquinas abstratas, as quais representam as mais importantes descrições mecanicistas envolvendo computação simbólica. Para cada uma delas é dada a definição, modo de funcionamento, características, capacidade e limitações. Além disso, é enfatizada a noção de memória nessas máquinas a fim de esclarecer tal noção. Como as máquinas estudadas neste capítulo distinguem-se, basicamente, pela quantidade de memória disponível em cada uma, a comparação entre elas permitirá que se verifique a relevância da memória da máquina na sua capacidade fundamental de resolver problemas. Na Seção 1, é comentada a máquina de estados finitos, conhecida como a máquina fundamental da teoria dos autômatos, a qual é destacada por possuir memória finita. Na Seção 2, o mesmo é feito com a máquina de Turing, a qual abrange todas as capacidades da máquina de estados finitos, sendo, porém, de natureza ilimitada. Por fim, na Seção 3, enfatiza-se a importância da memória na capacidade de uma máquina resolver problemas, procurando mostrar que, de fato, a memória constitui elemento decisivo nas capacidades de uma máquina.

unesp

2.1 Máquina de estado finito

2.1.1 Apresentação geral da MEF

Máquina de estado finito (MEF), também chamada autômato finito (AF), corresponde ao conceito: " (...) máquinas que agem em passos 'discretos' claramente separados de um para outro de um número finito de configurações ou estados" (Minsky, 1967, p.11). Dessa forma, uma MEF responde a eventos 'saltando' de um estado para outro de acordo com um conjunto formal de regras, as quais são específicas para cada problema a ser solucionado pela máquina.

A idéia de máquina de estado finito vem de um ramo interdisciplinar da Matemática chamado Teoria das Linguagens Formais, tendo suas raízes também na Ciência da Computação e na Lingüística. Esses dispositivos matemáticos são baseados em modelos de comportamento de circuitos elétricos seqüenciais e são de uso e aplicações em muitas áreas, podendo ser pensados como um modelo mais simples de algoritmo ou mesmo de um computador.

Tal tipo de máquina, altamente idealizada, é surpreendente em virtude de um de seus aspectos. Este é mencionado por Minsky (1967, p.11), quando se refere à máquina dizendo que "de certa maneira, pode ser usada para aproximar qualquer coisa que possa ser feita por outro sistema físico finito". O fato de ser de natureza finita e limitada não elimina sua importância, pois, mesmo não sendo tão poderosa, ela é interessante por explicar a criação de tipos de

máquinas mais complicadas, já que é muito mais fácil descrever completamente, sem ambigüidades, a estrutura e o comportamento de uma MEF do que descrever um modelo mais realístico de sistema mecânico. Mesmo a restrição de ter estados finitos não significa algo tão severo, uma vez que muitas dessas máquinas podem ser construídas com um número necessário e suficiente de estados (Minsky, 1967).

Uma MEF, de maneira simplificada e informalmente, consiste em alguns componentes básicos (Drobot, 1989):

- Um *conjunto de estados*. Os estados são as possíveis configurações da máquina, ou seja, as situações que a máquina pode assumir. Por exemplo, em uma máquina que é uma roda, cujas posições se alternam de 120° em 120°, as possíveis configurações que ela pode assumir são 120°, 240° e 360°, cada uma representando um estado. Nesse caso, a máquina terá um conjunto com três estados possíveis. No entanto, em um dado momento a máquina estará em um *único* estado.
- Um *conjunto de eventos que a máquina reconhece*. Os eventos costumam representar algum tipo de entrada externa, embora a MEF também possa gerar eventos internos.
- Uma *tabela*. Esta é responsável pelo mapeamento de cada par (evento, estado) numa determinada ação da máquina.

Ao se tratar de MEF, deve-se ter claramente que, para qualquer máquina que seja desse tipo, os símbolos que ela reconhece, o número de estados

que possui e as possíveis transições nela feitas são todos fixos e finitos. Mesmo assim, um número infinito de entradas pode ser testado pela máquina. Isso representa um dos aspectos da versatilidade e do poder de uma MEF, bem como um dos paradoxos associados à infinitude.

2.1.2 Funcionamento da MEF

Dada uma noção geral da MEF, neste item será apresentado um conceito matemático dessa máquina.

A MEF consiste em um conjunto finito de estados e em um dispositivo de leitura de entradas, o qual lê, da esquerda para direita, símbolo por símbolo de uma cadeia de entrada.

Uma máquina de estado finito M começa em um determinado *estado inicial*, lendo o símbolo mais à esquerda da cadeia de entrada. Num dado momento, a máquina vai estar em um *estado* Q, lendo um símbolo de *entrada* x e, então passará para um próximo estado. O que determina o próximo estado da máquina é a regra $\delta(Q,x)$, chamada *função transição*, e ela depende tanto do estado atual Q quanto da entrada atual x, sendo totalmente independente de informações anteriores. Esse processo será sempre repetido com a próxima entrada da direita, considerando-se o processo encerrado depois de lido todos os símbolos da cadeia de entrada ou, caso a máquina encontre uma regra $\delta(Q,x)$ indefinida.

Recebida uma entrada e estando em um determinado estado, a função saída é que dirá qual a resposta da máquina para tal situação. As possíveis respostas da máquina a eventos formam um conjunto de saída.

A MEF pode ser utilizada para reconhecer algumas cadeias de entrada e rejeitar outras. Para isso, ela possui estados chamados *finais*, de modo que, quando executa toda a seqüência de símbolos na cadeia e termina em um desses estados finais, diz-se que a cadeia foi reconhecida ou aceita; caso contrário, diz-se que a cadeia executada foi rejeitada e a máquina M não a aceita.

2.1.3 *Definição Formal*

Conforme Drobot (1989, p.33), uma máquina de estado finito determinística[2] M é constituída por 5 elementos:

> 1. Um conjunto finito, não vazio $\boldsymbol{Q} = \{Q_0, Q_1, ..., Q_n\}$. Os elementos de $\boldsymbol{Q}$ são chamados **estados**.
> 2. Um conjunto finito, não vazio $\Sigma = \{a_1, a_2, ..., a_k\}$ de símbolos permitidos nas cadeias de entrada. O conjunto Σ é chamado o **alfabeto** de entrada de M.
> 3. Um estado designado $Q_0 \in \boldsymbol{Q}$, chamado o **estado inicial**.
> 4. Uma **função de transição de estado** $\delta(Q,a)$ definida para todos os estados $Q \in \boldsymbol{Q}$ e todos $a \in \Sigma$. Para cada $Q \in \boldsymbol{Q}$ e cada $a \in \Sigma$, o valor de $\delta(Q,a)$ é um outro estado de $\boldsymbol{Q}$. Assim: $\boldsymbol{Q} \times \Sigma \Rightarrow \boldsymbol{Q}$.
> 5. Uma coleção não vazia Φ de estados a partir de $\boldsymbol{Q}$. Os elementos de Φ são chamados **estados finais** ou *accepting states*.

A máquina de estado finito M será então uma quíntupla:

$$M = \{Q, \Sigma, Q_0, \delta, \Phi\}$$

---

2 As máquinas de estado finito são classificadas em MEF determinística e MEF não determinística. Para a MEF determinística, o valor de cada função de transição $\delta(Q,x)$ é unicamente definido, não permitindo qualquer tipo de escolha para o par (estado, símbolo de entrada) de um certo momento. Já em uma MEF não-determinística, tem-se a extensão da definição de uma MEF determinística para permitir várias possibilidades para uma mesma função de transição.

Essa definição formal dada por Drobot é muito usada em um tipo de MEF classificada como *reconhecedora*, a qual tem como objetivo reconhecer cadeias de entrada como sendo ou não de uma linguagem. Uma **linguagem** L sobre um alfabeto Σ, por definição, é a coleção de cadeias finitas de elementos de Σ.

De forma simples, em uma MEF 'reconhecedora', dada uma entrada, esta será aceita (reconhecida) pela máquina se ela terminar a leitura em um estado final. Caso contrário, a entrada será rejeitada. O conjunto de todas as cadeias de entrada aceitas por essa MEF é o que forma a sua linguagem. Pelo fato de esse tipo de MEF ter como objetivo apenas reconhecer cadeias de entrada, a MEF 'reconhecedora' não fornece saída.

Conforme von Eckardt (1996), as máquinas de estado finito com entrada e que não produzem saída são classificadas como *acceptors*. Nesse tipo de MEF deve ser incluída a MEF 'reconhecedora'. Existem MEF que produzem saídas sem a necessidade de receber uma entrada, classificadas como *generators*. Além desses dois tipos, algumas MEF, dependendo da tarefa a ser realizada pela máquina, podem responder a cada sinal de entrada, selecionando para cada determinada entrada um sinal de saída; ou seja, a partir da entrada, produzir a saída em função de seus estados internos. Essas MEF são conhecidas como *finite transducer* ou *finite state automata with output* e, fazendo-se a comparação com a definição formal dada por Drobot, no caso delas, dois novos objetos devem ser especificados:

- Um conjunto finito, não vazio $P = \{b_1,...,b_m\}$ de símbolos permitidos como resposta a um determinado símbolo de entrada. Este conjunto é chamado **alfabeto de saída** de *M*.

- Uma **função de saída** $\varphi(Q,a)$ definida para todos os estados $Q \in \boldsymbol{Q}$ e todos $a \in \Sigma$. Para cada $Q \in \boldsymbol{Q}$ e cada $a \in \Sigma$, o valor de $\varphi(Q,a)$ é um símbolo do alfabeto de saída de *M*.

Além disso, para essas MEF, os estados finais não são definidos, sendo tais máquinas descritas pela sêxtupla:

$$M = \{Q, \Sigma, P, \delta, \varphi, Q_0\}$$

2.1.4 Representações da MEF

Existem maneiras diferentes de se representar uma máquina de estado finito, devendo-se ter sempre em mente que a MEF é uma representação matemática abstrata e não um dispositivo físico qualquer.

Para o conhecimento de algumas dessas representações, serão descritas duas MEF. A primeira é uma máquina 'reconhecedora'. Dada uma cadeia de entrada, a máquina verificará se essa cadeia pertence ou não à linguagem reconhecida por ela. As representações dessa máquina são baseadas na MEF descrita por Drobot (1989, p.30).

A maneira mais simples de se representar uma MEF, de forma auto-explicativa, é através de uma ilustração chamada **Diagrama de transição de estados** ou apenas **Diagrama de estados**.

A Figura 1 representa uma MEF pelo diagrama de transição de estados. No diagrama, os círculos representam os estados internos e as flechas representam as transições de um estado para outro, de acordo com o símbolo de entrada que ocorre em sua cauda.

A representação da Figura 1 descreve que o alfabeto de entrada da MEF é {a,b,c}, o conjunto de estados é $\{Q_0,Q_1,Q_2\}$, o estado inicial é Q_0 e o estado final Q_2. A seta apontando para Q_0 indica que ele é o estado inicial. Deve ser notado que, em algumas situações, a máquina permanece no mesmo estado, como, por exemplo, na regra $\delta(Q_0,a)=Q_0$. Se a máquina está no estado Q_0 e o símbolo de entrada é 'a', a máquina permanece no estado Q_0 e lê o próximo símbolo.

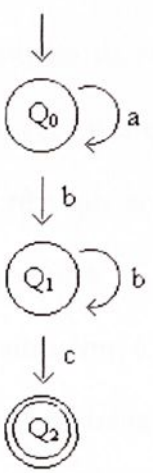


Figura 1 – Representação de uma MEF pelo Diagrama de transição de estados

A máquina descrita acima pode ainda ser representada de maneiras deferentes e uma delas pode ser vista na Figura 2. Independente da

opção de representação, a MEF é a mesma, ou seja, executa as mesmas ações e reconhece a mesma linguagem.

Na Figura 2, tem-se a representação da MEF por meio de um **Tabela de transição de estado**.

		Entradas		
		a	**b**	**c**
estados	Q_0	Q_0	Q_1	
	Q_1		Q_1	Q_2
	Q_2			

Figura 2: Tabela de transição de estados

Como não poderia ser diferente, a representação da Figura 2 descreve que o alfabeto de entrada da MEF é {a,b,c}, o conjunto de estados é $\{Q_0,Q_1,Q_2\}$, o estado inicial é Q_0, e o estado final Q_2. Nessa representação, o estado inicial e o estado final têm que ser dados explicitamente, já que na tabela não existe nenhum indicador deles. A leitura da tabela deve ser feita pela intersecção da coluna de entradas com a linha de estados. Assim, se a máquina está no estado Q_0 e o símbolo de entrada é ´a´, verifica-se o próximo estado na intersecção da linha do estado Q_0 com a coluna de entrada ´a´. Nesse caso, a máquina permanece no estado Q_0 e passa a ler o próximo símbolo de entrada. Deve ser lembrado que a máquina não aceitará a cadeia de entrada recebida caso ela se encontre na situação de estar em um par (estado, entrada) indefinido, como

por exemplo $\delta(Q_1,a)$; ou, também, se ela terminar de ler a cadeia de entrada sem estar em um estado final.

Analisando-se a MEF descrita tem-se que a linguagem aceita por essa máquina é algo do tipo {bc, abc, aabc, bbbbc, aaaaabc, etc...}, ou seja, essa máquina é utilizada para reconhecer cadeias que comecem com *n* ´a´s e continuem com *m* ´b´s , terminando com um ´c´; onde *n* é maior ou igual a zero e *m* é maior que zero.

Como a máquina acima é do tipo que reconhece uma linguagem, a 'resposta' para a cadeia de entrada recebida é: esta cadeia faz ou não parte de uma linguagem.

A segunda MEF a ser descrita possui o conjunto de estados $\{Q_0,Q_1\}$, sendo Q_0 o estado inicial, o alfabeto de entrada $\Sigma = \{0,1\}$ e o conjunto de respostas {par, ímpar}. Dada uma certa cadeia de entrada, essa máquina verifica se a quantidade de vezes que o símbolo ´1´ aparece na cadeia é par ou ímpar. As representações dessa máquina são baseadas na MEF descrita por von Eckardt (1996, p.110).

A Figura 3, abaixo, descreve a máquina especificada através de duas tabelas: uma para determinar o estado interno da máquina (representando a função de transição de estado), e outra para determinar a resposta da máquina a estímulos de entrada (representando a função de saída). A Figura 4 descreve a última máquina através do diagrama de transição.

	Q_0	Q_1
0	Q_0	Q_1
1	Q_1	Q_0

Função de Transição de Estados

	Q_0	Q_1
0	par	ímpar
1	ímpar	par

Função de Respostas

Figura 3: Tabelas de uma MEF

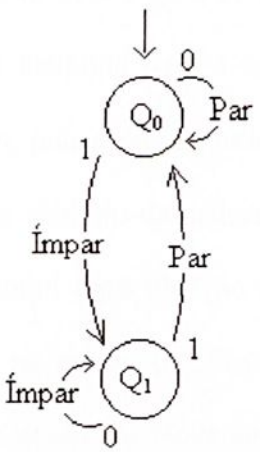


Figura 4: Diagrama de transição de estados

2.1.5 Exemplos de MEF

Utilizando a primeira MEF já descrita, a 'reconhecedora', suponha que a entrada para a máquina seja σ = aaabbc. A máquina executará a seguinte seqüência de movimentos:

$$Q_0 \xrightarrow{a} Q_0 \xrightarrow{a} Q_0 \xrightarrow{a} Q_0 \xrightarrow{b} Q_1 \xrightarrow{b} Q_1 \xrightarrow{c} Q_2$$

Aqui, $Q_0 \xrightarrow{b} Q_1$ significa que a máquina mudou do estado Q_0 para Q_1 quando leu a entrada 'b', o que é equivalente a $\delta(Q_0,b) = Q_1$.

Dada a entrada original, a máquina executou a seqüência de movimentos como descrito acima, começando no estado inicial Q_0 e com o símbolo ´a´. Depois de ler o símbolo em branco, a máquina parou no estado Q_2. Como este é um estado final, tem-se que a máquina aceitou σ.

Observando-se a Figura 1, e acompanhando toda a seqüência de movimentos feitos pela máquina para a entrada σ, pode-se verificar que a máquina inicia no estado Q_0, pois a seta apontando para esse estado indica ser ele o estado inicial. O primeiro símbolo da cadeia de entrada é ´a´, de forma que, seguindo a seta que tem como especificação o estado Q_0 e o símbolo ´a´, a máquina deve ser mantida no estado Q_0. O próximo símbolo a ser lido é ´a´, mantendo-se a máquina no estado Q_0. Novamente o símbolo à direita é ´a´, e o estado continua sendo Q_0. Nesse ponto, o próximo símbolo é ´b´. Seguindo-se as setas da representação, para o estado Q_0 e o símbolo ´b´, o estado posterior é Q_1. Aí, então, o estado é Q_1, o símbolo lido é ´b´, e a máquina é mantida no estado Q_1. O último símbolo a ser lido é o ´c´, com a máquina no estado Q_1. Após esse movimento, o estado passa a ser Q_2, mas como a cadeia termina, a máquina pára. Deve-se, então, verificar qual o estado da máquina no momento. Como ela está no estado Q_2, que é um estado final, conclui-se que a cadeia σ é aceita pela máquina.

Utilizando a segunda MEF também já descrita, a 'verificadora' do número de símbolos ´1´, suponha que a entrada para a máquina seja σ = 10110. Nesse caso, a máquina executará a seguinte seqüência de movimentos:

$1 \rightarrow \boxed{Q_0} \rightarrow$ Ímpar, e o estado interno da máquina será Q_1

$0 \rightarrow \boxed{Q_1} \rightarrow$ Ímpar, e o estado interno da máquina será Q_1

$1 \rightarrow \boxed{Q_1} \rightarrow$ Par, e o estado interno da máquina será Q_0

$1 \rightarrow \boxed{Q_0} \rightarrow$ Ímpar, e o estado interno da máquina será Q_1

$0 \rightarrow \boxed{Q_1} \rightarrow$ **Ímpar**, e o estado interno da máquina será Q_1

Como pode ser observado, sempre que o símbolo ´1´ aparece, tanto o estado interno quanto a resposta da máquina são alterados. No acompanhamento dos movimentos da máquina feitos para a cadeia recebida, a entrada final foi ímpar pelo fato de haver, nessa cadeia, um total de três símbolos ´1´.

2.1.6 Considerações sobre a caixa-preta da MEF

A MEF, do ponto de vista do usuário, pode ser considerada uma caixa fechada com canais de entrada e de saída. Isso se deve ao fato de o interior da máquina não precisar ser conhecido pelo usuário. Este pode interessar-se apenas pelas propriedades de entrada e saída da máquina.

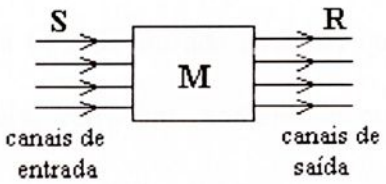


Figura 5 – Máquina caixa-preta

Para a análise da máquina, primeiramente como uma caixa fechada, deve-se observar estes elementos:

- *Canais* – um canal pode ser caracterizado por um conjunto de estados. Dessa forma, em um dado momento, cada canal estará em um dos números finitos de estados possíveis.[3]
- *Sinais* – os sinais que passam pelos canais da máquina, segundo sua natureza física, podem ser pulsos elétricos, impulsos eletroquímicos, ou impulsos mecânicos.

Na Figura 5, o canal de entrada chama-se S e o canal de saída R. Pelo canal S podem passar os sinais s_1, ..., s_n; enquanto o canal R possibilita a passagem dos sinais r_1, ..., r_m.

Na MEF, considera-se a entrada como vinda de algum ambiente E, e, de alguma maneira, a saída selecionada pela máquina M, agirá no ambiente.

3 O fato de o número de estados possíveis de um canal ser finito é um dos postulados chave da Teoria dos Autômatos. (Minsky, 1967, p.13)

▪ *Funcionamento*: todo o comportamento da máquina é determinado por eventos (geralmente algum tipo de entrada externa) que ocorrem em momentos discretos de tempo, ou seja, entre um momento e outro nada acontece. Esses momentos são indicados por uma variável *t* que representa o tempo através de números inteiros ($t = 0,1,2,3,....$). Como *M* opera em instantes discretos *t*, em $t = 0$ a máquina está em uma condição inicial conhecida. Em cada momento subseqüente *t*, *M* recebe, seqüencialmente, um símbolo de entrada $S(t)$ como estímulo em seu canal de entrada, e transmite um sinal de saída $R(t)$ ao seu canal de saída. Os símbolos de $S(t)$ pertencem ao alfabeto de entrada Σ de *M* ($S(t) \in \Sigma$, $t = 1,2,3,....$), enquanto $R(t)$ pertence ao alfabeto da máquina.

Para se *descrever a caixa-preta da máquina M*, isto é, verificar seu funcionamento interno e, conseqüentemente, entender como ela trabalha, existe a necessidade de se explicar seu comportamento. A explicação desse comportamento consiste em esclarecer o que a máquina faz para selecionar um determinado estado de saída *R(t)* a partir de um dado estado de entrada *S(t)*, ou seja, como a saída depende da entrada. Minsky (1967), diz que, em geral, isso dependerá da história do sistema.

A *história da máquina* no tempo *t*, $H(t)$, consiste no que segue: é uma listagem de todas as informações sobre a máquina desde o tempo em que foi criada até o tempo *t*, ou seja, desde o primeiro evento colocado em operação, com toda a seqüência de estados e entradas recebidos após esse evento, até o tempo *t*.

Sendo assim, caso a máquina recebesse o sinal de entrada s_i no tempo t, a escolha do sinal r_j em $t+1$ [4] dependeria:

- de qual sinal s_i ocorreu em t
- da história da máquina em t

Essa relação pode ser expressa formalmente através desta função F:

$$R(t+1) = F(H(t), S(t))$$

2.1.7 A memória da MEF e sua história

Como já foi dito, para a determinação da resposta da máquina a um estímulo de entrada é necessário reconstruir sua história atual. Acrescenta-se que, para uma dada máquina, pode existir uma variedade de histórias possíveis em um tempo t. Então, deve-se notar que alguns eventos de um passado remoto deverão contribuir para as respostas a serem dadas. Portanto, as máquinas devem ter uma *memória* desses eventos remotos e "se todo evento antigo deixou um 'rastro' separado, independente, a máquina necessita ter capacidade infinita, em algum sentido, para armazená-la" (Minsky, 1967, p.15).

4 A escolha do sinal de saída no momento t não depende do sinal de entrada do mesmo instante t, não se admitindo sinais que atravessem uma unidade instantaneamente. O sinal de saída em t dependerá do sinal de entrada no tempo anterior $t-1$.

unesp

Quanto ao armazenamento, a MEF, formada por um conjunto finito de partes discretas, não se enquadra no tipo de máquina com capacidade infinita, apresentando algumas limitações referentes à quantidade de informações que ela consegue armazenar. Máquinas desse tipo não têm a capacidade de armazenar a listagem completa de histórias, se estas forem muito longas. Sendo assim, se a memória é limitada e a máquina não pode se lembrar de tudo o que aconteceu com ela, não conseguirá distinguir, entre todas, as suas possíveis histórias.

Sobre essa limitação há um postulado chave na teoria dos autômatos finitos que diz: "tais máquinas podem distinguir, por seu comportamento presente e futuro, somente algum número finito de possíveis histórias. Estas serão chamadas **estados internos** da máquina, *Q(t)*" (Minsky, 1967, p.16).

Portanto, a máquina de estado finito, cuja característica é possuir um número finito de estados internos, é reconhecida pela propriedade de não ter capacidade infinita e não poder armazenar um registro completo de eventos para histórias arbitrariamente longas. Uma MEF tem memória somente para um número finito de resultados, que são considerados os estados da máquina. Dessa forma, este tipo de máquina não é apto a executar instruções que não possuam limite no tamanho da ação a ser efetivada.

De acordo com a definição de estado interno acima expressa, a relação entre a entrada e a saída da máquina pode ser dada em termos mais diretos, e isso a simplifica, já que não é necessária a dependência de toda a história da máquina até o momento. Utilizando-se *Q(t)*, pode-se usar o termo MEF como sendo uma 'caixa-preta' que, dada uma entrada, ela gera uma saída em função de seus estados internos. Nesse caso, a resposta da máquina a um estímulo de entrada é:

$$R(t+1) = F(Q(t), S(t))$$

É necessário, também, para uma descrição completa da máquina, conhecer qual o seu estado interno no tempo $t+1$. Essa informação é obtida conhecendo-se a entrada no tempo t e o estado interno em t. Isso pode ser expresso através da relação G da forma:

$$Q(t+1) = G (Q(t), S(t))$$

Através das relações F e G pode-se observar que, para se obter a resposta da máquina e seu estado subsequente, é necessário conhecer apenas o seu estado e a sua entrada atual.

Um aspecto importante a ser destacado no estudo da MEF é que existe uma relação simples entre a estrutura da máquina e o seu comportamento. Os passos dados pela máquina podem ser calculados quando se tem:

- descrição: as relações F e G;
- condição inicial: o estado inicial;
- cadeia de entrada.

A fim de exemplificar as considerações feitas neste item, será apresentada aqui a *máquina de memória*. Esta máquina, descrita por Minsky (1967, p.20), possui dois estados internos $\{q_0,q_1\}$, aceita no canal de entrada dois tipos de sinais $\{s_0,s_1\}$, e produz dois sinais de resposta $\{r_0,r_1\}$. A máquina de memória, sendo uma MEF, fica completamente descrita ao se conhecer suas funções F e G, as quais são dadas na Figura 6 abaixo:

G	q_0	q_1
s_0	q_0	q_0
s_1	q_1	q_1

Função de Transição de Estados

F	q_0	q_1
s_0	r_0	r_1
s_1	r_0	r_1

Função de Respostas

Figura 6: Tabelas da Máquina de Memória

O objetivo dessa máquina é determinar qual o sinal recebido no tempo *t-1*, estando a máquina em um dado estado no tempo *t*. Em outras palavras, o estado interno no tempo *t* dirá se o sinal recebido em *t-1* foi ´s_0´ ou ´s_1´. Devido a esse atraso de uma unidade de tempo ocorrido entre a entrada e a resposta, considera-se que essa máquina tem uma habilidade rudimentar de 'lembrar' o sinal ocorrido em *t-1*. Considerando, ainda, a definição de estados internos, essa

máquina possui um tipo 'frágil' de memória, limitada em sua capacidade, visto que ela pode distinguir apenas duas classes de histórias possíveis, q_0 e q_1.

2.1.8 Restrições resultantes da memória limitada da MEF

Embora a MEF seja capaz de operar em cadeias longas de símbolos de entrada e também de produzir cadeias longas de saída, sua falta de memória, logicamente, traz conseqüências quanto ao tipo de tarefas que podem ser feitas por máquinas desse tipo. Algumas dessas conseqüências são:

- *Linguagens reconhecidas pela MEF*

Em itens anteriores foi declarado que a MEF pode 'reconhecer' linguagens. No entanto, a falta de memória dessa máquina influencia o fato de existirem linguagens que não podem ser reconhecidas por uma MEF. Por exemplo, ao tentar construir uma MEF que aceite apenas cadeias de entrada com o mesmo número de 'a' e 'b', onde $\Sigma = \{a,b\}$, será possível perceber que a linguagem $\{a^n b^n \mid n = 0,1,2,....\}$ não é reconhecida por nenhuma MEF. Para que isso seja possível é necessário adicionar memória à máquina, ou um número infinito de estados.

Claro que poderiam ser definidas MEFs para exemplos específicos de cadeias como 'ab' ou 'aabb' ou 'aaabbb'. No entanto, combinar essas máquinas para se criar uma única máquina que reconheça {ab, aabb, aaabbb,...} irá requerer estados distintos que marquem quantos 'a's foram

reconhecidos. Isso é necessário porque um estado em uma MEF é apenas um nome qualquer, não carrega nenhuma informação adicional de como a computação alcançou esse estado. Dessa forma, se houver dois caminhos diferentes pelos quais a computação possa alcançar um certo estado Q_1 e a próxima computação depender de qual desses caminhos Q_1 veio, então Q_1 precisará ser quebrado em dois estados separados. Nesse caso, cada estado terá que representar algo do tipo "reconheceu um ´a´", "reconheceu dois ´a´s", "reconheceu três ´a´s", e assim por diante. Se o número de ´a´s a serem reconhecidos não tiver um limite, dado que o número de estados de uma MEF é finito, a máquina não conseguirá reconhecer um número n arbitrariamente longo.

Além disso, pelo fato de os estados serem apenas nomes e não carregarem nenhuma informação sobre como eles foram alcançados durante a computação, o formalismo da MEF não pode expressar a regularidade de n ´a´s seguidos de n ´b´s.

- *Problema da Multiplicação*

Suponha-se que se queira uma máquina capaz de adicionar números binários. Dado que, para executar essa computação é necessário um número finito de símbolos reconhecidos, um conjunto finito de instruções e um conjunto finito de estados de memória, tal computação pode ser executada pela MEF.

Definindo-se uma MEF com as características:

- um conjunto de símbolos {0,1};
- um conjunto de instruções com as regras para adição binária: 0+0= 0; 0+1=1; 1+1=0 e vai 1 (*carry*); 1+1+1=1 e vai um (*carry*);
- um conjunto de dois estados: estado com *carry*, estado sem *carry*;

então, ela estará apta a executar a soma.

Para a operação de adição, a quantidade de informação a ser armazenada não é grande. O armazenamento de um dígito em cada 'casa' (unidade, dezena, centena, etc ...), feito através do estado com *carry*, já é suficiente para os cálculos; não constitui, portanto, problema.

Contudo, a limitação da MEF está relacionada com o "Problema da Multiplicação".

Em uma multiplicação, se o tamanho dos números envolvidos na operação for grande, a dificuldade será decorrente do fato de tornar-se necessário guardar uma quantidade muito grande de informações durante o processo da multiplicação. Isso porque, até se chegar à resposta final, as somas parciais ocorridas devem ser armazenadas e, para números arbitrariamente longos, as somas parciais são grandes. Esse problema pode ser comprovado pelo teorema: "Nenhuma máquina de estado finito fixa pode multiplicar números decimais ou binários arbitrariamente longos". A prova desse teorema é encontrada em Minsky (1967, p.27).

A possibilidade de multiplicar números arbitrariamente longos foge ao alcance de qualquer MEF devido à capacidade finita de memória dessa máquina. Pelo fato de não existir um limite de tamanho nas somas parciais, a operação de multiplicação exige um 'poder computacional' maior do que o disponível a uma MEF.

Segundo Johnson-Laird (1988, p.40), "O conceito de 'poder computacional' refere-se ao que uma computação pode fazer mais do que sua velocidade ou eficiência". Costuma-se pensar que todo o poder de uma máquina deve-se a novos e potentes tipos de instruções. No entanto, "o que é necessário é uma memória ilimitada para os resultados intermediários obtidos durante a computação. Memória é poder" (Johnson-Laird, 1988, p.41). Seguindo esta idéia, a obtenção de uma máquina mais potente não é alcançada somente com instruções originais, mas, essencialmente, através de uma memória com maior capacidade de armazenamento .

Sendo uma memória mais potente algo essencial para computações mais complexas, pode-se, então, considerar a MEF um tipo de máquina que deixa a desejar nesse sentido.

unesp

2.2 Máquina de Turing

2.2.1 Apresentação geral da MT

A máquina de estado finito apresentada e comentada anteriormente, embora tenha a sua importância, é de natureza limitada.

A partir deste item será estudado outro tipo de dispositivo, chamado máquina de Turing (MT), o qual abrange todas as capacidades da MEF e é mais poderoso do que esta.

Curiosamente, a máquina de Turing parece ser o dispositivo automático mais poderoso possível, já que é, em sua essência, a definição matemática precisa do conceito de um algoritmo ou de um procedimento efetivo. Conforme Teixeira (1998, p.23), "a máquina de Turing constitui a melhor formalização da noção de algoritmo que se tem notícia na história da Matemática".

Informalmente, um *algoritmo*[5] é um procedimento clerical (preciso, determinado) que pode ser aplicado a qualquer entrada e que, eventualmente, produzirá, para cada entrada, uma saída correspondente.

---

5 Uma definição mais aprimorada de algoritmo, descrita por Rogers (1947, p.2), é dada de acordo com estes termos:

* 1. um algoritmo é dado como sendo um conjunto de instruções de tamanho finito;
* 2. há um agente computacional, geralmente humano, que pode reagir às instruções e realizar as computações;
* 3. há facilidades para fazer, armazenar e recuperar passos numa computação;
* 4. seja P um conjunto de instruções como em *1 e L um agente computacional como em *2. Então L reage a P de modo que, para qualquer entrada, a computação seja realizada de forma discreta, sem o uso de métodos contínuos ou criações análogas;
* 5. L reage a P de modo que a computação seja feita deterministicamente, sem recorrer a métodos randômicos ou a artifícios.

Como, na sua essência, a máquina de Turing pode ser entendida dessa maneira, é possível afirmar que qualquer definição precisa de algoritmo é equivalente ao conceito de máquina de Turing.

A classe de máquinas abstratas conhecidas como Máquinas de Turing foi definida em agosto de 1936 no *paper On Computable Numbers, with an Application to the Entscheidungsproblem*, cujo autor é Alan Mathison Turing, em resposta a questões levantadas na época, semelhantes às citadas a seguir (Dupuy, 1996).

Em 1931, Kurt Gödel estabeleceu o teorema da incompletude: "para todo sistema formal, no sentido da lógica (...), suficientemente rico para conter a aritmética, temos a propriedade: ou o sistema é inconsistente (leva a teoremas contraditórios), ou existe pelo menos uma proposição verdadeira que não é demonstrável nesse sistema" (Dupuy, 1996, p.28-9). Assim, "permanecendo no ´interior´ do sistema, ´jamais´ estaremos certos sobre o valor de verdade da sentença, a qual é, portanto, ´indecidível´" (Dupuy, 1996, p.29). Essa questão foi considerada uma atualização do problema de lógica proposto por David Hilbert, em 1928, o da ´decisão´ (*Entscheidungsproblem*): "dada uma fórmula qualquer do cálculo de predicados, existe um procedimento matemático, geral, efetivo, que permita determinar se essa fórmula é demonstrável ou não?" (Dupuy, 1996, p.28)

unesp

Seguindo o trabalho de Gödel, Turing mostrou que alguns problemas não são computáveis [6] e, através de seu *paper* de agosto de 1936, desenvolveu a teoria de uma máquina capaz de resolver quase todos os problemas dados a ela, chegando à construção teórica da máquina de Turing.

Com esses estudos, Turing contribuiu não apenas com a formulação do conceito de um computador de propósito geral (máquina de computar que pode ser programada para resolver qualquer problema que se pode solucionar), mas também com a elaboração da Teoria Matemática da Computação, em que ficou demonstrada a existência de problemas sem solução algorítmica. A conclusão resultante é que um problema terá solução algorítmica se existir uma máquina de Turing para representá-lo. Desses estudos surgiu também a Teoria da Computabilidade, que abrange o conjunto de estudos destinados a encontrar formas de descrever e representar algoritmos.

2.2.2 Idealização da MT

A máquina idealizada por Turing pode ser imaginada como uma fita de papel infinita e dividida em quadrados (células) ao longo de seu comprimento, acompanhada por uma cartolina que tem uma janela onde é visível apenas uma célula por vez.

Dessa forma, a janela pode andar para a direita ou para a esquerda na fita e, ao andar, a máquina 'lê' um símbolo na fita. De acordo com

---

6 Quando um algoritmo é usado para calcular valores de uma função numérica, tal função é descrita como função efetivamente calculável, ou função algoritmicamente computável, ou função efetivamente computável ou simplesmente, função computável. Assim, **função computável** é a função cujos valores podem ser calculados de maneira efetiva.

Portanto, tem-se que, enquanto um algoritmo é o procedimento para computar uma função, uma função computável por algoritmo é o resultado do procedimento.

esse símbolo e o estado da máquina, ela pode apagar o que leu, escrever por cima ou deixar como está.

Para tomar decisão sobre o próximo símbolo lido, a máquina, antes, muda de estado interno. A mudança de um estado para outro é feita seguindo um conjunto de regras.

O comportamento da máquina fica totalmente determinado pelas informações que ela lê, e esta, para ser capaz de resolver problemas, precisa ser programada, ou seja, seguir um conjunto de regras.

2.2.3 Partes da MT

A máquina de Turing consiste em:

- uma *fita* dividida em um número infinito contável de células (espaços divididos em quadrados). Um exemplo de fita pode ser visto na Figura 7.

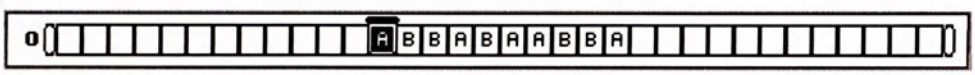


Figura 7 - Uma fita usada na representação de uma máquina de Turing

- uma *cabeça de leitura e escrita*, dispositivo que, em certo instante, aponta para uma das células da fita e a examina.
- um número finito de *estados*, que são as possíveis situações que a máquina pode assumir.

2.2.4 Regras da MT

Para realizar alguma tarefa a máquina vai percorrer a fita.

Na **fita** coloca-se um símbolo por célula, e, para cada máquina, são permitidos aparecer na célula da fita somente os símbolos que lhe são definidos. Esse conjunto de símbolos permitidos forma o chamado *alfabeto da fita*: Σ. Embora tenha tamanho infinito, em um determinado instante, a fita possui somente algumas células com símbolo, enquanto outras ficam em branco. O número de células não brancas pode mudar durante a execução de uma tarefa pela máquina.

A **cabeça de leitura/escrita** da máquina, num dado momento, vai estar apontando para uma única célula da fita e poderá:

- deixar o símbolo desta célula inalterado;
- apagar o símbolo atual e imprimir outro;
- imprimir um símbolo se for uma célula em branco,

e poderá se movimentar:

- uma célula para a direita;
- uma célula para a esquerda;
- permanecer na célula atual.

Também, em cada instante, a máquina vai estar num certo **estado** e poderá:

- permanecer no mesmo estado;
- passar para outro estado.

Dois estados serão destacados na máquina: o *estado inicial* (estado a partir do qual a máquina começa a funcionar) e o *estado final* (estado de parada da máquina).

A máquina pára quando:

- entra no estado de parada, estado em que a máquina pára sua operação;
- a cabeça de leitura/escrita da máquina aponta para a célula mais à esquerda da fita e a instrução pede que a cabeça movimente uma célula para a esquerda;
- não existe uma instrução dada à máquina para o estado atual com o símbolo apontado na fita.

Cada máquina de Turing é especificada por uma seqüência finita de instruções, chamada programa, que contém as regras para o seu funcionamento e que determina o seu comportamento. São essas instruções que definem, baseadas no símbolo e no estado atual da máquina, o que a cabeça de leitura/escrita irá escrever na fita, como irá se mover e qual será seu próximo estado.

2.2.5 Funcionamento da MT

A máquina de Turing opera em passos discretos.

A princípio a máquina T está em algum estado inicial específico, e a fita contém alguma seqüência de símbolos entre os permitidos. Esse conteúdo inicial da fita é chamado entrada. Na maioria dos casos é definido

que o primeiro símbolo que serve como entrada é aquele armazenado na célula mais à esquerda da fita, estando a cabeça de leitura/escrita posicionada ali [7].

A máquina começa a funcionar a partir do estado inicial e vai caminhando através dos estados enquanto opera na fita. Basicamente ela usa a seqüência de instruções, os estados e o conteúdo da fita para determinar como os símbolos de entrada devem ser processados.

Num dado momento a máquina está num estado Q qualquer e sua cabeça de leitura/escrita aponta para uma célula da fita contendo um símbolo $x \in \Sigma$. O par (Q, x) é chamado de *configuração* (ou descrição instantânea) da máquina e, o possível comportamento dessa máquina é determinado por essa configuração.

Com base nessas duas informações, (Q e x), a máquina executa três operações:

- vai para um estado;
- coloca um símbolo na fita;
- move a cabeça de leitura/escrita.

Essas três ações, quando executadas juntas, são chamadas *movimento*. A máquina executa cada movimento seguindo sua seqüência finita de instruções.

Esses procedimentos vão se repetindo enquanto a máquina funciona. Ela opera indefinidamente, até que caia numa situação de parada.

---

7 Isto não é realmente necessário. Caso defina-se que a máquina comece a trabalhar em outra parte da fita, basta que se incluam instruções para esta poder encontrar o primeiro símbolo não branco da entrada.

2.2.6 *Definição Formal*

Conforme Drobot (1989), uma máquina de Turing T é consistuída por quatro elementos:

1. Um conjunto finito Q de **estados**. Em Q existe o **estado de parada** (ou *halting state*) , geralmente denotado pelo símbolo ***H***.
2. Um conjunto finito de símbolos Σ chamado de **alfabeto da fita**. Um desses símbolos pode ser '**#**', chamado símbolo em branco [8].
3. Um estado designado $\mathbf{Q_0} \in Q$ chamado de **estado inicial** da máquina.
4. Uma **função transição** $\delta(Q,x)$, que forma um conjunto de regras ou instruções para o funcionamento da máquina.

A função $\delta(Q,x)$ é *determinada* pelo par (Q,x), em que $Q \in Q$ e $x \in \Sigma$. O *valor* de $\delta(Q,x)$ é uma tripla (Q',x',N), em que $Q' \in Q$; $x' \in \Sigma$; e N = L, R ou S. A *interpretação* da função transição $\delta(Q,x) = (Q',x',N)$ é esta: se a máquina está num estado Q e sua cabeça de leitura/escrita aponta para o símbolo x, então a máquina passa para o estado Q', troca o símbolo x por x' e move sua cabeça:

- uma célula para direita se N = R;
- uma célula para esquerda se N = L;

ou, permanece no mesmo lugar se N = S.

A máquina de Turing T será então a quádrupla:

$$T = \{Q, \Sigma, Q_0, \delta\}$$

---

8 Por definição, tanto o número de símbolos impressos na fita quanto o número de estados são finitos. A razão para isso é que, ao permitir uma infinidade se símbolos e estados, poderiam existir inúmeros desses componentes diferindo em uma computação simples e/ou em uma extensão pequena de fita. Além disso, se os símbolos ou os estados forem infinitos, o número de instruções da máquina também deverá ser. Com isso, a resposta de uma computação poderia não ser dada, considerando-se a dificuldade em se encontrar a regra para a configuração atual. O efeito da restrição do número de símbolos e estados é algo que não afeta seriamente a computação, já que é possível conseguir alternativas para esse tipo de limitação.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

A máquina de Turing pode ser usada como 'reconhecedora' de linguagens. Deve ser lembrado que, dada uma linguagem L, uma máquina de Turing T reconhece L se for capaz de decidir se determinada cadeia σ pertence ou não a L. Se, para a cadeia σ, essa máquina parar em um estado 'aceitável', então σ é válida e pertence a L; caso contrário, $\sigma \notin L$.

Novamente, assim como se deu com a MEF, a definição formal dada por Drobot é a mais comumente usada para as MT 'reconhecedoras'.

Pode ser lembrado que, segundo Minsky (1967), uma MEF capaz de responder a cada sinal de entrada é caracterizada por: um alfabeto de símbolos de entrada, um alfabeto de símbolos de saída, um conjunto de estados internos e um par de funções Q e R que descrevem a relação entre entrada e estado interno, subseqüentemente descrevendo o comportamento da máquina.

Como a máquina de Turing é uma MEF mais poderosa, devido à capacidade de armazenamento proporcionada pela fita, na MT, essa caracterização da MEF terá alguma modificação para que se possa descrever sua associação com a fita externa. Tais mudanças podem ser observadas tanto na descrição feita por Drobot como na feita por Minsky, as quais estão sendo apresentadas neste trabalho [9].

9 Segundo Minsky, 1967, p. 118, descrições formais da MT podem ainda ser encontradas em Turing (1936), Post (1943), Kleen (1953) e Davis (1958). Em cada uma dessas descrições existem diferenças técnicas em suas formulações. Contudo, tais diferenças são insignificantes ou sem importância.

unesp

Em uma máquina de Turing, o conjunto de símbolos de entrada continua com a função de especificar os símbolos permitidos nas cadeias de entrada, com a diferença de que estes, agora, devem estar inscritos na fita. Num dado instante t (observe que tanto a MEF como a MT operam em instantes discretos), a entrada para a máquina T será o símbolo examinado pela cabeça de leitura/escrita em t.

Tomando-se as considerações feitas para a MEF a respeito das relações F, G, R e Q, pode-se dizer que a mudança de estados na MT continuará sendo determinada pela função G.

A principal diferença entre as caracterizações da MEF e da MT se dá na saída da máquina T, quando então, a saída possui duas funções:

- Imprimir um símbolo na célula que está sendo examinada no momento. O símbolo a ser impresso é um elemento do mesmo conjunto de símbolos de entrada Σ;

- Determinar a posição da cabeça de leitura/escrita da máquina T: se ela se moverá para esquerda, para a direita, ou permanecerá na célula atual.

Sendo assim, para a descrição completa da máquina de Turing, deve-se conhecer três funções:

$$Q(t+1) = G(Q(t), S(t))$$

$$R(t+1) = F(Q(t), S(t))$$

$$D(t+1) = \mathrm{D}(Q(t), S(t))$$

A nova função incluída, D, é usada para determinar o modo como a máquina T se moverá.

Usando-se as funções acima para se determinar cada movimento, ou cada ciclo de operação da máquina, tem-se que: dado o estado atual q_i e o símbolo lido s_j, a máquina T:

- vai para um estado $G(q_i, s_j)$;
- imprime o novo símbolo $F(q_i, s_j)$ na célula da fita onde está sua cabeça;
- move a cabeça de leitura/escrita de acordo com $D(q_i, s_j)$.

Seguindo essa descrição, as partes de estado-finito da MT podem ser descritas por um conjunto de quíntuplas da forma:

$$(q_i, s_j, G(q_i, s_j), F(q_i, s_j), D(q_i, s_j))$$

2.2.7 Representação da MT

As representações usadas para descrever a MT são análogas às usadas para descrever a MEF. A máquina de Turing também pode ser representada por diagramas de transição de estados ou tabelas de transição de estados, sendo tais representações correspondentes.

unesp

Para uma primeira representação, será construída uma máquina de Turing que irá reconhecer um tipo de linguagem. Essa representação é baseada na MT descrita por Drobot (1989, p.151).

A linguagem reconhecida por essa máquina é:

$$L = \{ 1^n \mid n = 1, 3, 5, ... \}$$

A Figura 8 representa a referida máquina de Turing pelo diagrama de transição de estados. Nessa figura pode-se verificar que o alfabeto dessa MT é {$, 1, S, N, #}, o conjunto de estados {Q_0, Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, H} e o estado inicial Q_0. As funções de transição podem ser claramente percebidas através desse diagrama. Nos diagramas da MT, os círculos representam os estados internos, e as flechas representam as transições de um estado para outro, de acordo com o que ocorre em sua cauda. A simbologia presente na cauda de cada flecha deve ser interpretada da forma: o símbolo à esquerda da barra ´/´ é o símbolo atual, o qual está sendo lido pela cabeça de leitura/escrita no momento. Os dois símbolos entre parênteses à direita de ´/´ informam o símbolo de saída e qual o sentido de movimento da máquina, respectivamente. Sendo assim, se a máquina está no estado Q_0, tem sua flecha apontada para o estado Q_1 e em sua cauda $ / ($,R); isso significa que, ao ler o símbolo ´$´, estando em Q_0, a máquina irá para o estado Q_1, imprimirá o símbolo ´$´ na fita e moverá a cabeça de leitura/escrita para a direita.

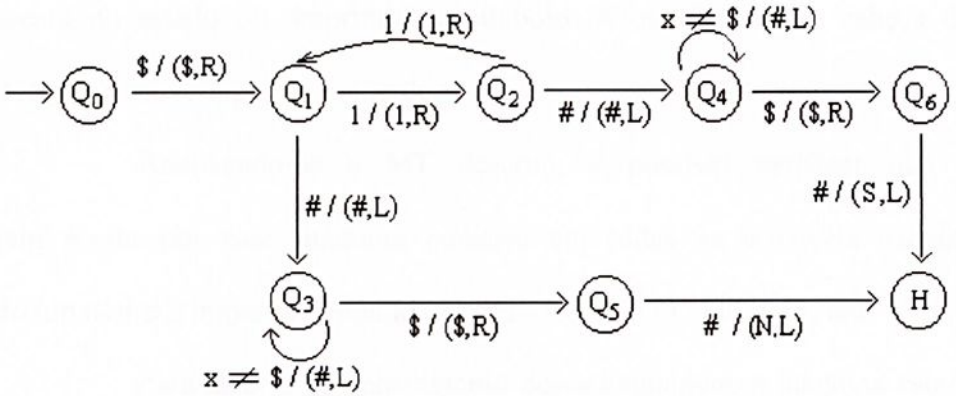


Figura 8 - Representação da MT pelo diagrama de transição de estados (Drobot, 1989)

Na Figura 9 a mesma máquina pode ser vista em uma outra representação correspondente, a tabela de transição de estados.

		SÍMBOLOS DA FITA				
		1	**$**	**#**	**S**	**N**
ESTADOS	Q_0		$(Q_1,\$,R)$			
	Q_1	$(Q_2,1,R)$		$(Q_3,\#,L)$		
	Q_2	$(Q_1,1,R)$		$(Q_4,\#,L)$		
	Q_3	$(Q_3,\#,L)$	$(Q_5,\$,R)$	$(Q_3,\#,L)$		
	Q_4	$(Q_4,\#,L)$	$(Q_6,\$,R)$	$(Q_4,\#,L)$		
	Q_5			(H,N,L)		
	Q_6			(H,S,L)		
	H					

Figura 9 – Representação da MT pela tabela de transição de estados

Nesta segunda representação, o movimento da máquina, quando esta estiver no estado Q_3 e lendo o símbolo ´1´, está descrito na intersecção da linha e coluna correspondentes a seu comportamento atual: $(Q_3,\#,L)$, ou seja,

permanecerá no estado Q_3, imprimirá o símbolo ´#´ na fita, e sua cabeça de leitura/escrita se moverá para a esquerda.

Analisando-se a MT descrita, é possível verificar que a linguagem aceita por essa máquina consiste em todas as possíveis cadeias contendo um número ímpar do símbolo ´1´: {1, 111, 11111, 1111111, etc...}.

Para fazer o reconhecimento dessa linguagem, a máquina estará movendo sua cabeça de leitura/escrita para direita e alternando-se entre o estado Q_1 (marcando um número par de símbolos ´1´) e Q_2 (marcando um número ímpar de símbolos ´1´), de acordo com a quantidade de símbolos ´1´ encontrados. Quando a máquina encontrar o símbolo ´#´ à direita da fita, ela passará a se movimentar para a esquerda, substituindo todos os símbolos da fita por ´#´, até que o símbolo mais à esquerda, ´$´, seja encontrado. Neste momento, a máquina escreve ´S´ (significando "sim", quando a cadeia de entrada possuir um número ímpar do símbolo ´1´) ou ´N´ (significando "não", quando a cadeia de entrada possuir uma quantidade par do símbolo ´1´).

A máquina de Turing 'construída' acima tem o papel de 'reconhecer' determinados tipos de linguagens. No entanto, uma MT pode executar uma série de outras tarefas com as informações que ficam armazenadas em suas fitas. Por exemplo, pode-se 'construir' uma máquina que execute o papel de 'calculadora', sendo capaz de somar, subtrair, multiplicar,

dividir, calcular logaritmos; outra, capaz de comparar entradas ou, ainda, executar outras funções ditas complexas [10].

Com a finalidade de se fazer uma comparação, mais uma máquina será apresentada a seguir, com as representações e notações baseadas na MT descrita por Minsky (1967, p.121).

Essa MT é conhecida como 'verificadora' de parênteses, a qual é capaz de determinar se uma seqüência de parênteses é bem formada.

Parênteses são sinais de pontuação, os quais são supostos existirem pareados para permitir intercalações em frases ou períodos. Por serem formados em par, pode-se distinguir, em cada caso, quais são bem formados, como é o caso dos exemplos (( )) (( )), ( ) ( ), ( ) (( )( )), e quais não são bem formados, conforme os exemplos (( ),) (, ( ) ( )), que não completam seus pares. Essa tarefa não pode ser feita por uma MEF, visto que esta não possui a capacidade de percorrer ambos os sentidos da fita e, assim, examinar células pelas quais já havia passado. Já a MT consegue o controle da seqüência de parênteses porque pode percorrer sua fita em ambas as direções e, portanto, pode encontrar uma resposta para essa situação.

---

10 Alguns autores classificam as MTs em *acceptors* e *transducers*. *Acceptors* seriam aquelas máquinas que produzem somente uma saída binária (aceita/rejeita), permitindo à máquina aceitar ou rejeitar suas entradas. Nesse tipo de MT estariam as máquinas "reconhecedoras" de linguagens. As MTs classificadas como *transducers* seriam as máquinas capazes de produzir resultados (saídas) mais complicados, os quais seriam deixados na fita quando a computação termina.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

Os símbolos da fita da maquina T verificadora são: { (,), A, X, BF, NBF }. O conjunto de estados é $\{Q_0, Q_1, Q_2, H\}$, sendo Q_0 o estado inicial.

A Figura 10 descreve essa máquina através de três tabelas, contendo as quíntuplas que fazem parte das regras da máquina, enquanto a Figura 11 descreve a mesma máquina usando a tabela de transição de estados.

Q	S	Q´	S´	D
Q_0	)	Q_1	X	L
Q_0	(	Q_0	(	R
Q_0	A	Q_2	A	L
Q_0	X	Q_0	X	R

Q_0

Q	S	Q´	S´	D
Q_1	)	Q_1	)	L
Q_1	(	Q_0	X	R
Q_1	A	H	NBF	-
Q_1	X	Q_1	X	L

Q_1

Q	S	Q´	S´	D
Q_2	)			
Q_2	(	H	NBF	-
Q_2	A	H	BF	-
Q_2	X	Q_2	X	L

Q_2

Figura 10: Quíntuplas para a MT verificadora de parênteses

Na figura acima, **Q** e **S** são os estados e as entradas atuais. **Q´**, **S´** e **D** são os próximos estados, as entradas e a direção do movimento, respectivamente. Essas transições também podem ser dadas na forma $(Q_0,)$, Q_1, X, L).

		SÍMBOLOS DA FITA					
		)	(	A	X	BF	BNF
ESTADOS	Q_0	$(Q_1$,X,L)	$(Q_0$,(,R)	$(Q_2$,A,L)	$(Q_0$,X,R)		
	Q_1	$(Q_1$,),L)	$(Q_0$,X,R)	(H,NBF,-)	$(Q_1$,X,L)		
	Q_2		(H,NBF,-)	(H,BF,-)	$(Q_2$,X,L)		
	H						

Figura 11 – Representação da MT pela tabela de transição de estados

unesp

A Figura 12 descreve a MT através de um diagrama de transição. A notação desse diagrama é utilizada por Minsky (1967, p.123), e pode ser usada somente em circunstâncias em que cada estado está associado a uma direção fixa. Para essa notação, cada flecha no diagrama está representando alguma quíntupla (q_i, s_j, q_{ij}, s_{ij}, d_{ij}). Então q_i é o estado descrito abaixo de cada círculo na cauda da flecha, s_j é o símbolo na cauda da flecha, s_{ij} está escrito no meio da flecha, q_{ij} é o estado na cabeça da seta e d_{ij} é o símbolo escrito dentro do círculo para q_{ij}. As quíntuplas mais comuns, da forma (q_i, s_j, q_i, s_j, d_{ij}), costumam ser omitidas.

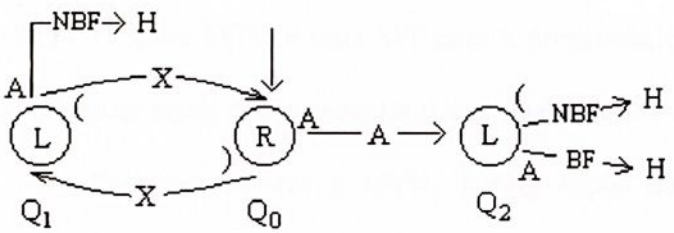


Figura 12 – Representação da MT pelo diagrama de transição de estados (Minsky, 1967)

Para essa máquina verificar se a seqüência de parênteses é ou não bem formada, ela se moverá para a direita à procura de um símbolo ´)´. Ao encontrá-lo, a máquina passará a se mover para a esquerda, à procura de um símbolo ´(´. Encontrando-o, a máquina os substituirá por um ´X´, até que chegue ao indicador de final de cadeia, nessa especificação, representado por um ´A´. Nesse ponto, se ainda houver algum símbolo, ´)´ ou `(´, sem seu par, a máquina imprimirá na fita o resultado ´NBF´, pois a cadeia de parênteses não é bem

formada. Caso contrário, a máquina imprimirá o resultado ´BF´, pois a cadeia de parênteses é uma seqüência bem formada.

As máquinas de Turing representadas acima computam, cada uma, um único tipo de tarefa. Isso porque, a partir de um conjunto fixo de regras, uma MT é capaz de realizar apenas um tipo de tarefa, como por exemplo, somar dois números no sistema unitário.

Isso significa que seria necessário uma máquina diferente para cada tipo de tarefa que se quisesse realizar. Para dar uma solução a esse inconveniente, Turing imaginou uma única máquina, chamada *máquina de Turing universal* (**MTU**). Uma MTU é uma MT com a propriedade de ser apta a ler a descrição de qualquer outra MT e executar o que qualquer outra MT tenha feito.

Para caracterizar a MTU, Turing supôs que o programa que determina a tarefa a ser executada pela máquina pudesse ser considerado como entrada e também pudesse ser escrito na fita. Assim, tanto a entrada quanto o programa a ser executado pela máquina poderiam ser escritos juntos na fita.

Com a informação apropriada em sua fita (nesse caso as células da fita armazenando também a descrição de operações diferentes), a MT poderia emular as ações de máquinas diferentes. Com a MTU tornou-se possível a leitura de qualquer conjunto de regras contido na fita e, assim, emular qualquer máquina cujo comportamento possa ser descrito simbolicamente.

2.2.8 *Exemplos de MT*

Considere-se a primeira máquina de Turing $T = \{Q, \Sigma, Q_0, \delta\}$ descrita no item anterior. Suponha-se que a cadeia de entrada para tal máquina seja $\sigma = 11111$. Nesse caso, será executada a seguinte seqüência de movimentos:

$$[Q_0, \underline{\$}11111] \rightarrow [Q_1, \$\underline{1}1111] \rightarrow [Q_2, \$1\underline{1}111] \rightarrow [Q_1, \$11\underline{1}11]$$

$$\rightarrow [Q_2, \$111\underline{1}1] \rightarrow [Q_1, \$1111\underline{1}] \rightarrow [Q_2, \$11111\underline{\#}] \rightarrow [Q_4, \$1111\underline{1}]$$

$$\rightarrow [Q_4, \$111\underline{1}] \rightarrow [Q_4, \$11\underline{1}] \rightarrow [Q_4, \$1\underline{1}] \rightarrow [Q_4, \$\underline{1}] \rightarrow [Q_4, \underline{\$}]$$

$$\rightarrow [Q_6, \$\underline{\#}] \rightarrow [H, \underline{\$}S]$$

No momento em que a máquina pára, ou seja, quando entra no estado de parada H, o símbolo ´S´ já foi escrito na fita, significando que a entrada dada, uma seqüência de cinco símbolos ´1´, pertence à linguagem L reconhecida pela máquina de Turing descrita.

Os movimentos da máquina são dados em uma notação desenvolvida para descrever configurações da MT. Nessa notação, podem ser observados dois componentes. O primeiro componente identifica o estado atual da máquina. O segundo, possui um símbolo sublinhado, que fica sob a cabeça de leitura/escrita, e identifica o símbolo que está sendo lido, no momento, pela máquina. Os demais símbolos descrevem o conteúdo da fita à esquerda e à direita da cabeça da máquina. Quando o símbolo sublinhado é o primeiro ou o último da cadeia de símbolos, significa que o conteúdo da direita ou da esquerda da cabeça de leitura/escrita são símbolos brancos.

A máquina vai de uma configuração para outra através de um movimento. Assim, quando a MT acima está na configuração $[Q_2, \$1\underline{1}111]$, com a aplicação de $\delta(Q_2,1)$, ela vai para a configuração $[Q_1, \$11\underline{1}11]$. Esse movimento é denotado por: $[Q_2, \$1\underline{1}111] \rightarrow [Q_1, \$11\underline{1}11]$.

Para exemplificar o funcionamento da segunda MT, também descrita no item anterior, será usada a cadeia de entrada σ = (( ) (( )). Tanto nesse exemplo como no anterior, deve-se observar a existência de dois símbolos especiais: ´$´ e ´A´. Esses dois símbolos são somente sinais de pontuação para indicar quando a entrada começa e/ou termina. São usados apenas por conveniência – para deixar claro o símbolo mais à esquerda e o mais à direita da entrada – podendo ser dispensados, em alguns casos, com o custo de se construir uma máquina mais complicada. A seqüência de movimentos para σ será:

$[Q_0,\ A\,\underline{(}\,(\,)\,(\,(\,)\,)]\ \rightarrow\ [Q_0,\ A\,(\,\underline{(}\,)\,(\,(\,)\,)]\ \rightarrow\ [Q_0,\ A\,(\,(\,\underline{)}\,(\,(\,)\,)]$

$\rightarrow\ [Q_1,\ A\,(\,\underline{(}\,X\,(\,(\,)\,)]\ \rightarrow\ [Q_0,\ A\,(\,X\,\underline{X}\,(\,(\,)\,)]\ \rightarrow\ [Q_0,\ A\,(\,X\,X\,\underline{(}\,(\,)\,)\,]$

$\rightarrow\ [Q_0,\ A\,(\,X\,X\,(\,\underline{(}\,)\,)]\ \rightarrow\ [Q_0,\ A\,(\,X\,X\,(\,(\,\underline{)}\,)]\ \rightarrow\ [Q_1,\ A\ (\,X\,X\,(\,\underline{(}\,X\,)]$

$\rightarrow\ [Q_0,\ A\,(\,X\,X\,(\,X\,\underline{X}\,)]\ \rightarrow\ [Q_0,\ A\,(\,X\,X\,(\,X\,X\,\underline{)}\,]$

$\rightarrow\ [Q_1,\ A\,(\,(\,X\,X\,(\,X\,\underline{X}\,X)]\ \rightarrow\ [Q_1,\ A\,(\,(\,X\,X\,(\,\underline{X}\,X\,X\,]$

$\rightarrow\ [Q_1,\ A\,(\,(\,X\,X\,\underline{(}\,X\,X\,X]\ \rightarrow\ [Q_0,\ A\,(\,(\,X\,X\,X\,\underline{X}\,X\ X]$

$\rightarrow\ [Q_0,\ A\,(\,(\,X\,X\,X\,X\,\underline{X}\,X]\ \rightarrow\ [Q_0,\ A\,(\,(\,X\,X\,X\,X\,X\,\underline{X}\,]$

$\rightarrow\ [Q_0,\ A\,(\,(\,X\,X\,X\,X\,X\,X\,\underline{A}]\rightarrow\ [Q_2,\ A\,(\,(\,X\,X\,X\,X\,X\,\underline{X}\,A]$

$\rightarrow\ [Q_2,\ A\,(\,(\,X\,X\,X\,X\,\underline{X}\,X]\rightarrow\ [Q_2,\ A\,(\,(\ X\,X\,X\,\underline{X}\,X\,X]$

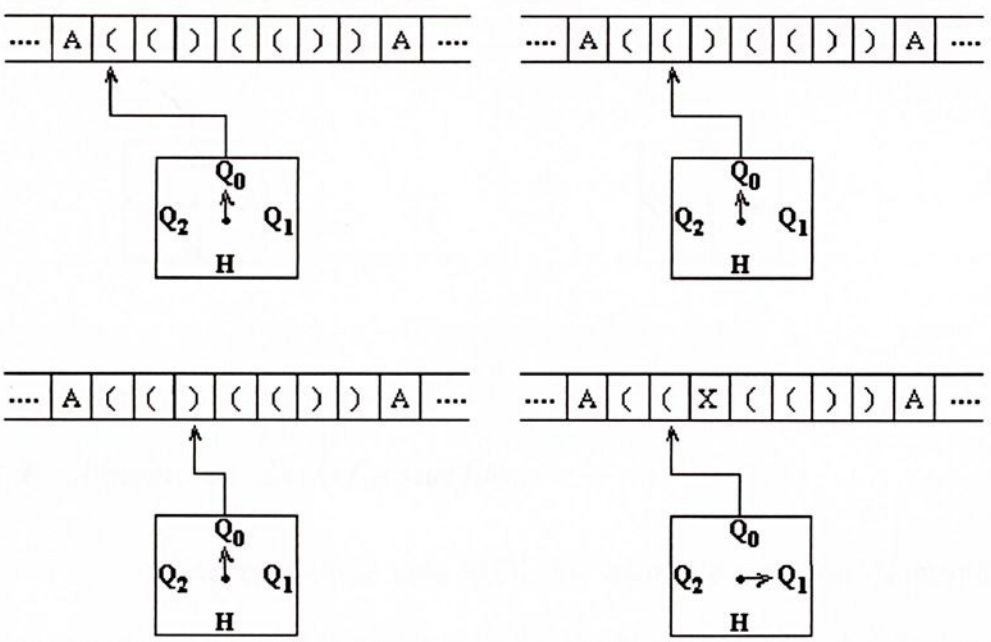
cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

$\rightarrow$ [Q_2, A ((X X $\underline{X}$ X X X] $\rightarrow$ [Q_2, A ((X $\underline{X}$ X X X X]

$\rightarrow$ [Q_2, A (($\underline{X}$ X X X X X] $\rightarrow$ [Q_2, A ($\underline{(}$ X X X X X X]

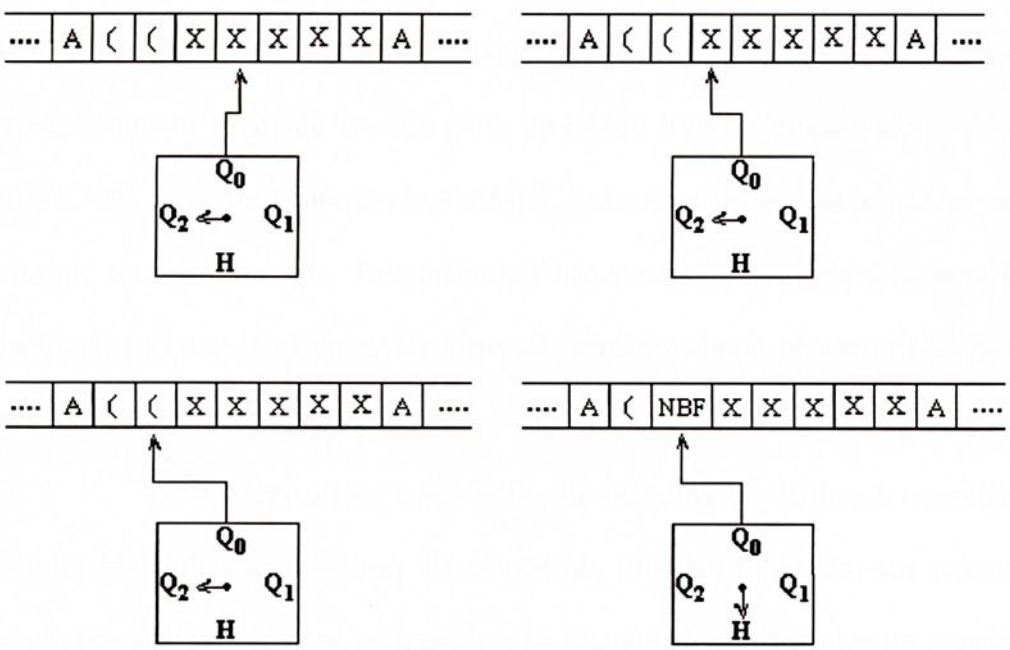
$\rightarrow$ [H, A ($\underline{\text{NBF}}$ X X X X X X]

Nesse exemplo a máquina alcançou o estado de parada H e imprimiu o símbolo 'NBF', o que significa que a cadeia de entrada dada não é bem formada. Os movimentos dessa máquina para σ = (( ) (( )) podem, ainda, ser observados na seguinte notação:



Depois de quatro movimentos essa máquina estará na configuração do último desenho acima. Como, para a entrada dada, o número de movimentos é grande, não serão mostrados todos os passos para a checagem da

seqüência de parênteses. Pulando alguns movimentos, os quatro últimos serão os seguintes:



2.2.9 A memória da MT e sua fita

Ao especificar uma MEF, foi visto que esse tipo de máquina obtém seus dados de entrada valendo-se de algum ambiente *E*. A máquina de Turing, no entanto, é associada a um tipo especial de ambiente: a sua fita. Assim pode-se dizer que uma MT é uma MEF com a fita funcionando como ambiente.

A fita da MT, ligada à máquina através de sua cabeça de leitura/escrita (a qual aponta, em certo momento da operação, para alguma célula da fita), pode armazenar e, depois, recuperar seqüências de símbolos.

Portanto, a MT possui uma memória ou armazenamento externo. O acesso da MT a essa memória exterior significa um acréscimo ao armazenamento existente em sua parte de estado finito. Comparando-se a MEF com a MT, deve ser lembrado que a MEF, por não possuir esse armazenamento externo, torna-se limitada. Tais máquinas não possuem memória suficiente para distinguir todas as suas possíveis histórias, somente algum número finito destas: seu número finito de estados internos.

Além disso, como a fita da máquina de Turing é considerada infinita em ambas as direções, diz-se que ela não tem limite em sua extensão, sendo possível usá-la para armazenar uma quantidade arbitrariamente grande de informação. Assim, a memória da MT tem uma capacidade teórica infinita [11]. Também, sua habilidade em mover-se para a esquerda ou para a direita é considerada uma forma de prover a máquina de um tipo de 'memória', visto que ela pode examinar símbolos já analisados anteriormente e mudá-los, se necessário.

---

11 Turing concedeu à máquina uma fita infinita em comprimento, interminável em ambas as direções. Na realidade, a fita não precisa ser infinita, basta que tenha o comprimento necessário para a tarefa. Existe até a restrição de que, quando a MT é iniciada, apenas algumas das células de sua fita possuem os símbolos de entrada, estando as demais células em branco. Com essa restrição, pode-se pensar na fita como sendo algo finito em qualquer tempo particular, mas com o adendo de que, chegando-se ao fim dessa porção finita, novas células poderão ser anexadas à fita, fazendo a máquina pedir mais fita à medida que necessite. Dessa forma ela não terá problemas com armazenamento.

unesp

Na MT, a mesma fita pode ser usada como meio de abastecer a máquina de qualquer dado inicial (a entrada, que é a cadeia de símbolos contidos na fita antes da máquina começar sua tarefa), de armazenar os resultados parciais ocorridos durante a computação, e de apresentar o resultado final da computação (a saída, que é a cadeia de símbolos que permanecem na fita quando a máquina pára). Com um espaço de armazenamento externo ilimitado ao qual pode recorrer, a máquina também é capaz de produzir saídas de tamanho ilimitado.

Por causa desses fatores, a MT é considerada máquina infinita, ou seja, essa máquina, por meio de sua fita, possui uma capacidade de memória potencialmente infinita.

Deve ser observado que cada movimento da máquina, tal como escrever um símbolo na fita, mover a posição da cabeça de leitura/escrita, exige um certo tempo, e a MT é ineficiente com relação ao tempo usado para sua passagem sobre blocos longos de fita. Confirmando o exposto, Minsky (1967, p.128) afirma que "ninguém propôs seriamente o uso de uma estrutura da máquina de Turing para qualquer computação prática". No entanto, desprezando-se a lentidão dessa máquina abstrata, tem-se que ela não faz uso ineficiente da capacidade de memória da fita.

Ainda é relevante observar que, por causa da fita da MT, essa máquina forma um sistema fechado, no qual pode ser feita a descrição completa de sua computação. Isso não é possível na MEF porque a descrição de uma computação não é completamente incluída na descrição da máquina, uma vez que seus dados de entrada vêm de algum ambiente qualquer.

Na verdade, existem várias maneiras equivalentes de se imaginar uma memória com acesso ilimitado (Johnson-Laird, 1988). No entanto, Turing teve a idéia considerada mais simples: a de uma fita infinita dividida em células.

Embora o propósito de Turing não fosse o de inventar um computador, mas sim, de descrever problemas logicamente possíveis de serem solucionados, sua máquina hipotética acabou antecipando algumas características que o computador deveria ter. A fita infinita da MT pode ser vista como uma memória de propósito geral, pela qual a máquina está apta a ler, escrever e apagar; ações comuns às da memória RAM de um computador moderno.

2.2.10 Resultados obtidos com a memória ilimitada da MT

Na última seção referente à MEF, Seção 2.1.8, foram comentadas limitações referentes às tarefas a serem executadas por essa máquina, as quais ocorrem devido à quantidade restrita de informações que podem ser nela armazenadas.

Antes de comentar como a MT se 'comporta' com a memória ilimitada, é interessante analisar, informalmente, as semelhanças e diferenças existentes entre essas duas máquinas.

Algumas características e o comportamento da MT a qualificam como uma MEF. Entre as características compartilhadas por elas podem ser destacadas:

- conjunto fixo finito de símbolos de entrada e saída;
- conjunto finito de instruções ou regras;
- conjunto de instruções específicas de como deve ocorrer a transição de estados;
- a computação procede-se em uma seqüência de passos discretos;
- em um dado momento no tempo, a máquina está em apenas um, de um número finito de estados internos.

A máquina de Turing, por sua vez, possui algumas características que lhe são específicas, tais como:

- armazenamento externo ilimitado (fita);
- cabeça de leitura/escrita, que liga a máquina à fita, e que se pode mover pela fita tanto para a esquerda quanto para a direita;
- os símbolos do alfabeto podem ser escritos ou lidos a partir da fita.

Analisando-se tais características, pode-se observar que, no que diz respeito à descrição formal, os dois tipos de máquina apresentados até agora são similares, pois, em dado momento, podem estar em apenas um de um número finito de estados e baseiam-se na configuração atual para executar uma instrução, entre outras características. Essa similaridade na descrição é também refletida na sua capacidade fundamental: uma função computável por uma MEF é computável por uma MT. Isso conduz a concluir-se que a principal diferença entre as máquinas está relacionada com a existência de uma fita infinita na MT.

Feitas essas considerações, pode-se passar para o objetivo principal deste item, que é mostrar como a MT pode desempenhar computações além da capacidade da MEF, ou seja, mostrar que, quando uma MEF é equipada

com uma fita ilimitada, o resultado é um dispositivo computacional mais poderoso.

Voltando-se à MEF, ao se comentar as linguagens reconhecidas por esse tipo de máquina, foi visto que sua falta de memória influencia no fato de existirem linguagens que não podem ser reconhecidas por ela. Como exemplo foi dada a linguagem $L = \{a^n b^n \mid n = 0, 1, 2, ...\}$.

A seguir, será aqui descrita e exemplificada uma MT capaz de reconhecer essa linguagem.

Seja a máquina de Turing T representada pela Figura 13. Por esta figura verifica-se que o alfabeto da máquina é {a, b, #}, o conjunto de estados $\{Q_0, Q_1, Q_2, Q_3\}$, sendo as funções de transição claramente apresentadas na própria figura. O estado inicial é Q_0 e o estado final Q_1.

		Símbolos da Fita		
		a	**b**	**#**
ESTADOS	Q_0	$\delta(Q_0,a,L)$	$\delta(Q_0,b,L)$	$\delta(Q_1,\#,R)$
	Q_1	$\delta(Q_2,\#,R)$	$\delta(Q_3,\#,S)$	
	Q_2	$\delta(Q_2,a,R)$	$\delta(Q_2,b,R)$	$\delta(Q_3,\#,L)$
	Q_3		$\delta(Q_0,\#,L)$	

Figura 13 – Máquina de Turing reconhecedora de $L = \{a^n b^n \mid n = 0,1,2,...\}$

Seja a entrada para esta máquina σ = aabb. A fita inicial e os movimentos da máquina T para σ são:

....	#	#	a	a	b	b	#	#	....

$[Q_0, \underline{a}abb] \rightarrow [Q_0, \underline{\#}aabb] \rightarrow [Q_1, \underline{a}abb] \rightarrow [Q_2, \underline{\#}abb]$

$\rightarrow [Q_2, a\underline{b}b] \rightarrow [Q_2, ab\underline{b}] \rightarrow [Q_2, abb\underline{\#}] \rightarrow [Q_3, ab\underline{b}]$

$\rightarrow [Q_0, a\underline{b}\#] \rightarrow [Q_0, \#\underline{a}b] \rightarrow [Q_0, \underline{\#}ab] \rightarrow [Q_1, \#\underline{a}b]$

$\rightarrow [Q_2, \underline{b}] \rightarrow [Q_2, b\underline{\#}] \rightarrow [Q_3, \underline{b}] \rightarrow [Q_0, \underline{\#}] \rightarrow [Q_1, \underline{\#}]$

Neste ponto a máquina irá parar, pois não existe uma instrução para o par (Q_1,#). Como a máquina parou no estado Q_1, que é o estado final, a entrada σ é aceita pela máquina, ou seja, pertence à linguagem L.

Através desse exemplo pode-se verificar que existe uma MT capaz de reconhecer a linguagem L, fato que não ocorre com a MEF.

Para reconhecer essa linguagem, cada vez que a máquina T faz a leitura de um símbolo ´a´ ela troca-o por um símbolo em branco e passa a revistar uma outra seção, mais à direita da fita, à procura de um símbolo ´b´. Ao encontrá-lo, substitui-o também por um símbolo em branco e volta para a parte esquerda da fita à procura de outro símbolo ´a´.

Descrevendo isso com relação aos estados, tem-se que o estado Q_0 é usado para encontrar o início da entrada, fazendo que a cabeça de leitura/escrita se mova para a esquerda até encontrar um símbolo branco. Ao

encontrá-lo, o estado atual passa a ser Q_1, o qual é usado para apagar o símbolo ´a´ mais à esquerda da fita, no momento. Apagado o ´a´, o novo estado passa a ser Q_2. Com Q_2, a cabeça de leitura/escrita move-se para a direita até alcançar o final da entrada, marcado por um símbolo branco à direita. Neste momento, o estado atual é Q_3, que é usado para apagar o símbolo ´b´ mais à direita da fita. Após apagá-lo, o próximo estado passa a ser Q_0, e o ciclo recomeça até que encontre uma situação de parada. Se a máquina parar no estado final Q_1, significa que a entrada pertence à linguagem L; caso contrário, não pertence.

A habilidade da máquina para realizar essa tarefa apóia-se no fato de ela ter procedimentos para voltar a examinar células pelas quais já passou anteriormente, e, assim, manter um controle da seqüência e do número de ´a´s e ´b´s que existem na entrada. Além disso, ela pode fazer isso para cadeias infinitamente longas, pelo fato de sua fita ser infinita.

Como pôde ser mostrado, há funções que podem ser computadas na MT, mas não em uma MEF. A razão disso, deve-se, em grande parte, ao fato de as MEFs não terem memória infinita. Adicionando-se memória, a máquina passa a poder armazenar resultados intermediários ilimitados, o que a torna capaz de encontrar respostas antes impossíveis.

Até o momento destacou-se grandemente a MT por sua habilidade de armazenar quantidades ilimitadas de informação. De fato, a possibilidade de uma memória potencialmente infinita é algo fundamental para que a máquina tenha condições de computar algumas tarefas. No entanto, será que é a finitude de uma máquina o que sempre limitará seu uso? Ou ainda, a MT,

com sua memória ilimitada, é capaz de computar qualquer função, ou realizar qualquer tarefa?

Para auxiliar na resposta de tais questões, serão comentadas situações que a MT não consegue resolver; sem, contudo, entrar em maiores detalhes a respeito dessas situações. Existem muitos problemas que podem ser expostos, e não solucionados por uma MT. Geralmente, o motivo de uma resposta negativa está:

- nas limitações práticas do tempo de operação da máquina;
- na complexidade conceitual de suas estruturas ou programas.

Pode ser corretamente argumentado que o número de MT diferentes é contável (Drobot, 1989, p.170, Teorema 5.3). No entanto, o número de funções existentes é incontável. Assim, algumas dessas funções não são computadas por uma MT, ou seja, existem problemas que não são solucionáveis por esse tipo de máquina, apesar de sua memória infinita.

Por exemplo, pode-se verificar que a maioria dos números é incomputável por uma MT. Segundo Teixeira (1998), um número (inteiro ou real) é computável se existir uma MT capaz de imprimir todos os seus dígitos, um após outro. Como existem números reais de dígitos infinitos, por exemplo π = 3.14159265....., para esses números a máquina não irá parar nunca. O problema, entretanto, é que, de acordo com suas regras fundamentais, o programa da MT deve parar sempre, já que o número de passos de um algoritmo deve ser finito. Chega-se, então, à conclusão que, como a quantidade de números reais é maior que a de números inteiros, a maioria dos números é incomputável.

Um outro problema que a máquina de Turing não consegue resolver é o *Halting problem*, o problema da parada (Drobot, 1989, p.171, Teorema 5.4). Turing demonstrou que não existe qualquer máquina de Turing $T = \{Q,\Sigma,Q_0,\delta\}$, para a qual, dada uma entrada qualquer σ, seja possível decidir se sua computação é finita.

Um fator que deve ser notado é que a adição de quaisquer aspectos extras a uma MT não altera seu poder computacional. Pode-se tornar a máquina não-determinística, dar-lhe fita infinita em apenas uma ou ambas as direções, adicionar-lhe fitas extras, ou uma fita bi-dimensional, entre outros. Nenhuma alteração a tornará apta a computar algo além daquilo que pode ser computado na máquina original, padrão. Essas modificações " (...) podem afetar **como** alguma coisa é feita – o tipo de programa que é usado, sua velocidade e eficiência – mas não **o que** pode ser feito" (Johnson-Laird, 1988, p.49).

A respeito dos limites das máquinas, Dupuy (1996) destaca dois aspectos descritos por Mosconi, sobre as máquinas de Turing. Um é o fato de possuírem estruturas sumárias e poderem fazer muitas coisas; o outro, por serem máquinas ditas infinitas e não poderem fazer tudo.

Esse ponto de vista sintetiza o valor, a importância e até os limites do *poder computacional* da máquina de Turing.

2.3 A importância da memória da máquina para a resolução de problemas

Analisados os objetos e propriedades da MEF e da MT, será examinado, nesta seção, o que pode ou não ser finito numa máquina. A importância disso é que a diferença entre a possibilidade de ser finito ou infinito contribui para que se chegue a uma conclusão sobre que fatores são importantes em uma máquina para definir sua capacidade fundamental de resolver problemas.

Iniciando-se pela análise dos dados de entrada de uma máquina, pode-se questionar se há um tamanho limitado para a entrada dos dados.

Considerando-se que tanto a MEF quanto a MT seguem um algoritmo para poder computar qualquer função, entende-se que, de acordo com a caracterização informal de algoritmo, a entrada não é relevante. Desde que o número de entradas seja finito, não há limite no tamanho da entrada, podendo esse tamanho ser infinito.

A resposta negativa à questão do tamanho limitado para a entrada dos dados é dada, sem problemas, quando se trata de um algoritmo ideal, sem levar em consideração a máquina. No entanto, considerando-se uma máquina com memória limitada, a resposta para a questão em discussão passa a ser afirmativa, pois é possível que, dada uma entrada arbitrariamente longa, esta requeira uma quantidade arbitrariamente longa de espaço de armazenamento, que exceda seu limite. Em casos como este, deve existir um limite no tamanho da entrada. Essa situação pode ser facilmente exemplificada observando-se a MEF 'verificadora' do número de símbolos '1', apresentada na primeira seção deste

capítulo. Nessa MEF, dada uma entrada com os símbolos ´0´ e ´1´, a resposta da máquina é se quantidade de símbolos ´1´, na entrada, é par ou ímpar. Como foi visto, a máquina não teve problema algum para executar a entrada σ=10110, pois tal entrada não exigiu que a máquina tivesse algo além de suas propriedades. No entanto, supondo-se que, para essa mesma máquina, seja fornecida uma entrada σ arbitrariamente longa, com alguns milhares de símbolos ´0´ e ´1´ (como por exemplo σ=1110001011000110111110000111.....), tal entrada, embora esteja de acordo com o alfabeto da máquina, não poderá ser executada pela MEF, pois exigirá da máquina uma memória de armazenamento que ela não contém. Em situações como esta, a entrada interfere na capacidade de resolução de problema da MEF. Por não ter memória suficiente para entradas arbitrariamente longas, a máquina não é capaz de executar todo tipo de entrada que lhe é fornecida [12]. Esta limitação devida à memória pode também ocorrer com a saída dos dados.

Dado que as máquinas analisadas resolvem problemas segundo um conjunto de instruções, que é o algoritmo, coloca-se a questão: o número de instruções deve ter um limite?

Novamente pode-se entender que, não considerando as limitações práticas, a teoria dos algoritmos não define um número máximo de instruções possíveis. O algoritmo é finito, podendo ser arbitrariamente grande,

12 Problemas desse tipo poderiam ser amenizados ou mesmo eliminados se fosse possível à máquina ler apenas os dados de entrada necessários ao corrente estado da computação. Computados esses dados, eles seriam apagados e apenas os próximos dados necessários à nova computação seriam lidos. Nesse sistema, entradas longas também poderiam ser computadas. Apesar de ser uma solução possível, ela não é desejável, já que exigiria o acréscimo de um operador externo, o qual não faria parte da máquina, para controlar os dados de entrada. A ineficiência dessa "solução" vem da própria descaracterização de autômato que se tem com o acréscimo do operador externo.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

sem um tamanho máximo fixado. No entanto, como anteriormente, em uma máquina finita, o tamanho arbitrariamente grande do algoritmo pode exceder o limite de memória disponível.

Considerando-se que tanto o algoritmo da MEF quanto da MT baseiam-se nos possíveis símbolos a serem usados na fita (o alfabeto da fita) e em seus estados internos, pergunta-se: existe um limite para esses dois objetos?

Diferentemente do tamanho da entrada, da saída e do conjunto de instruções que, a princípio, desconsiderando-se as limitações práticas, não exige um limite finito fixo, o alfabeto da máquina e o seu número de estados devem ser finitos. A finitude necessária desses dois objetos da máquina deve-se ao fato de a infinitude de qualquer um deles implicar em um conjunto de instruções também infinito, já que eles estão presentes nas instruções. No entanto, um algoritmo é, por definição, um conjunto de instruções de tamanho finito, e essa característica não pode ser violada.

Outro fator questionável é que, em muitos casos, pode ser desejável que uma computação sempre termine. Porém, uma computação necessariamente precisa terminar?

Utilizando-se da lógica, a autora desta dissertação demonstrará, a seguir, que não se pode obrigar uma computação a terminar sempre. Apesar de ser isso desejável em algumas situações, não é uma verdade absoluta que toda computação sempre pára, ou seja, chega a seu término.

Dentro da lógica tem-se que toda teoria, para que possa ser procedente, exige consistência e completude. Se uma teoria é um conjunto de

afirmações, ela é *consistente* se não existirem duas afirmações contraditórias, como por exemplo A e ¬A (A e não A), ou seja, se não existir algo do tipo A é teorema da teoria, e ¬A também é teorema da teoria. Como o princípio de uma teoria é separar as verdades das mentiras, não se pode aceitar uma teoria inconsistente, pois ela se torna uma teoria trivial: tudo se torna válido dentro dessa teoria, não se podendo separar as verdades das falsidades. Além disso, uma teoria deve também ser *completa*, o que significa que uma teoria deve 'dar conta' de todas as suas verdades, ou seja, não se pode deixar de fora afirmativas sobre essa teoria. Por exemplo, uma afirmativa sobre a teoria dos números naturais deve ser ou verdadeira ou falsa, nunca ficar sem uma resposta.

Trazendo esse conceito lógico de teoria para a Teoria dos Algoritmos, pode-se concluir que, criando-se uma Teoria dos Algoritmos apenas com as funções totais (funções definidas para todos os elementos do domínio), haverá algoritmos sem *loop* (conjunto de instruções executado repetidamente até que alguma condição seja satisfeita), com um número finito de passos, e que sempre darão resposta. No entanto, uma teoria apenas com funções totais não será uma teoria completa, constituindo-se uma teoria logicamente incompleta já de partida. Este resultado é assim demonstrado:

Seja $\phi_1\ \phi_2\ \phi_3\ \phi_4\ \phi_5$ ..... a lista de todas as funções totais computáveis. É possível construir uma função total computável da forma $\mathbf{f}(n) = \phi_n(n) + 1$, a qual é diferente de qualquer uma das funções da lista.

Sabe-se que $\phi_n(n)$ é, por definição, computável e que, *somar um* também é uma operação computável. Sendo o resultado da soma de uma

função computável com outra função computável, uma função computável, deduz-se que $\mathbf{f}(n) = \phi_n(n) + 1$ é uma função computável que não pertence à lista.

A partir da $\mathbf{f}(n)$ dada, pode-se afirmar que existe função total computável que não seja nenhum ϕ_k, $\forall k$; ou seja, a conclusão é que nos ϕ_k, $\forall k$ não estão todas as funções totais e computáveis, já que $\mathbf{f}(n)$ não pertence aos ϕ_k, $\forall k$.

Ao conseguir uma função total computável que não seja teorema pertencente à teoria, conclui-se que a Teoria dos Algoritmos apenas com as funções totais constitui uma teoria de início incompleta, fato que viola a lógica. Perdendo o controle lógico, perde-se o controle dos erros de uma teoria, não se podendo admitir tal situação como 'ponto de partida'.

Para que a Teoria dos Algoritmos possa ser considerada uma teoria completa devem ser incluídas as funções parciais (funções que não precisam estar definidas para todos os elementos do domínio). De fato, o que se pode afirmar é que a teoria das funções totais é, de início, incompleta. Contudo, até o momento não foi encontrado um exemplo de incompletude ao serem usadas funções parciais. Desse modo, é necessário trabalhar com algoritmos usando-se as funções parciais.

Uma das características das funções parciais é que elas nem sempre dão uma resposta. Sendo assim, o *loop*, como se enquadra nessa situação, é uma função parcial, e, portanto, é uma estrutura fundamental do ponto de vista lógico. Por esse motivo, não por uma necessidade computacional, o *loop* não pode ser eliminado da Teoria dos Algoritmos. É possível que, enumerados todos os

programas sem *loop*, seja dado um procedimento efetivo (algoritmo) que só possa ser resolvido usando-se um *loop*.

Resumindo-se a demonstração feita, tem-se que, em razão da necessidade lógica das funções parciais, não se pode obrigar um algoritmo a parar sempre. Um programa deve ser capaz de computar tudo o que é computável, e a teoria das funções totais não 'dá conta' do domínio das funções computáveis. Portanto, a necessidade de parar não é algorítmica, dado que o conceito de computável não exige que o algoritmo pare.

É pertinente que se analise, a seguir, a idéia de memória na MEF e na MT, pois esse conceito é o que constitui a principal diferença entre elas. Isso é possível a partir da questão: pode haver um limite para a memória de uma máquina?

Ao executar uma instrução, é necessário que o seu resultado seja guardado. A *memória de execução*, responsável por guardar a instrução a ser executada, embora não possa ter um tamanho definido *a priori*, é finita. No entanto, a memória onde o resultado de uma instrução fica armazenado, a *memória de armazenamento de dados*, não pode ser limitada, uma vez que isso prejudica a capacidade da máquina de resolver problemas.

Para exemplificar a importância da memória nas máquinas de computar, será retomado aqui o "Problema da Multiplicação". Como visto no estudo da MEF, existe MEF capaz de adicionar números, mas não existe qualquer máquina desse tipo capaz de fazer a multiplicação. Por que a MEF é capaz de somar e não de multiplicar?

unesp

Em uma adição, o armazenamento da condição 'foi +1' (*carry*), ou 'não foi +1', em cada 'casa', é suficiente para que se execute a operação; não havendo, portanto, problema com o espaço de armazenamento.

Em uma multiplicação, embora a entrada, a saída e o programa, ou seja, a quantidade de instruções que a calculem seja finita, é possível que seja necessário guardar uma quantidade muito grande de informações durante o processo da multiplicação, o que 'exige' uma memória sem limites.

Isso ocorre porque a multiplicação é uma soma de parcelas. Em uma multiplicação tradicional de dois números quaisquer x (multiplicador) e y (multiplicando), o produto de x por y é obtido multiplicando-se x por cada dígito de y. O resultado de cada multiplicação de um dígito de y por x constitui uma das parcelas, as quais serão arranjadas de acordo com as regras da multiplicação, e depois, somadas, fornecerão o resultado (produto). Portanto, nessa operação, o número de parcelas é equivalente ao número de dígitos do multiplicando. Caso a entrada a ser tomada como multiplicando seja arbitrariamente longa, terá que haver um grande espaço de armazenagem de dados para que os passos da multiplicação (as somas parciais) sejam armazenadas. Ou seja, na multiplicação é possível perceber que a operação cresce de acordo com a entrada. Conforme o tamanho da entrada, quantidade maior de espaço para trabalho é requerida, e isso pode ser imaginado observando-se a Figura 14.

unesp

```
      75326
   x 193764182     ⇒  o que ocorreria se esta entrada
-----------------     tivesse um bilhão de dígitos?
    parcelas
       ...
       ...
       ...
-----------------
    resultado
```

Figura 14 – Exemplo de multiplicação

A fim de confirmar a memória como um elemento decisivo para que algumas computações possam ser realizadas, será apresentado um exemplo que parece violar o problema da multiplicação na MEF.

Seja *M* uma MEF capaz de multiplicar dois números no sistema unário, no qual 0=0, 1=1, 2=11, 3=111 e assim por diante.

Para que a multiplicação possa ser efetuada nesse autômato finito *M*, tanto os dados de entrada quanto os dados de saída não são memorizados. Ao contrário disso, o operador, que equivale ao ambiente externo que envia e recebe dados da máquina, registra, passo a passo, esses dados fora da máquina.

Sendo assim, para a multiplicação de dois números quaisquer nessa máquina, o operador precisa pegar cada símbolo da entrada que servirá como multiplicando e apresentá-lo a cada símbolo da entrada que será o multiplicador. O resultado da multiplicação desses dois símbolos será armazenado pelo próprio operador.

unesp

Para que haja sincronia entre o recebimento de cada símbolo das entradas e o acesso à saída, a máquina deverá produzir um sinal no momento certo de o operador apresentar e receber os símbolos.

A representação de um passo da computação dessa máquina pode ser vista na Figura 15.

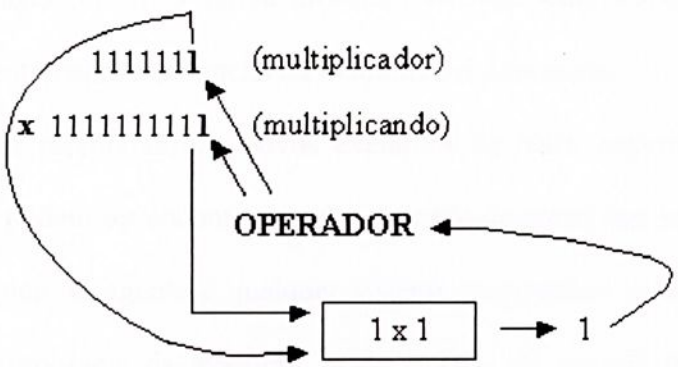


Figura 15 – Um passo da computação da MEF multiplicadora

Considerando-se a idéia levantada no exemplo, é possível que um autômato finito realize uma multiplicação tão longa quanto necessária, fato que violará a idéia até agora defendida, da impossibilidade dessa operação ser realizada por tal tipo de máquina.

Entretanto, algumas características da máquina exemplificada podem ser questionadas, o que será usado para que se entenda que esse exemplo não serve para infringir o "problema da multiplicação". Por que o exemplo dado não viola o problema?

Isso não ocorre simplesmente pelo fato de essa máquina não estar de acordo com as funções reconhecidas como pertencentes a uma máquina. Ao criar a máquina finita capaz de computar a multiplicação, o que ocorreu foi um desvio de funções, já que a memória potencialmente infinita passou a ser característica necessária ao operador, que é algo externo à máquina, quando se deveria esperar que a própria máquina fosse capaz de 'dar conta' de todas as partes da computação. Com a tarefa atribuída erroneamente ao operador da máquina, conseqüentemente o potencial da máquina foi diminuído.

Da mesma forma, novos exemplos de MEF capazes de fazer uma multiplicação podem ser encontrados, dependendo do *papel* que se atribua ao agente [13] da máquina. O agente é qualquer sistema responsável pela realização das instruções do programa da máquina, e ele é parte da própria máquina. O agente da MEF e mesmo da MT é um agente *simples*, com ações apenas mecânicas, sendo que qualquer outro tipo de tarefa (não mecânico) que lhe seja dado, não respeitará o seu verdadeiro *papel*. Se o agente não for *simples*, ele poderá executar tudo; adquirirá capacidade de decidir, inclusive, o indecidível mecanicamente. Pode-se dizer que, nessa situação, tudo passa a ser computável.

Sendo assim, ao se pensar em uma máquina, deve ser considerado o *verdadeiro papel* do agente e do operador, uma vez que uma atribuição equivocada de tarefas a qualquer um deles, pode fazer que o problema da memória *falsamente* desapareça. Portanto, pode-se novamente afirmar que o problema da multiplicação permanece tal como primeiramente anunciado, não existindo, então, autômato finito capaz de multiplicar.

---

13 Ao tratar do **agente**, neste trabalho, não há o intuito de entrar na clássica discussão de consciência, vontade, alma, entre outros. Não se trata do agente possuidor de intenções do sujeito humano, mas sim de um agente mecânico, entendido apenas como um mecanismo, uma força propulsora. Trata-se de um ser passivo, sem vontade própria e sem um sujeito.

T. 502

A partir da análise feita acima, é possível concluir que existem processos que podem ser precisamente descritos, assim como a operação de multiplicação, e que, no entanto, não podem ser realizados por uma máquina limitada.

Existem problemas que podem ser resolvidos apenas por máquinas com memória ilimitada, como pôde ser visto nos exemplos das seções anteriores, já que a memória da fita de Turing realmente escapa às limitações da MEF.

Apesar disso, Minsky (1967, p.114) questiona o fato de se estudar e caracterizar as máquinas com capacidade ilimitada. Segundo ele, essas máquinas não existem em um universo temporal finito; não sendo possível, portanto, a construção de tal máquina.

O próprio Minsky (1967, p.114) responde a seu questionamento acima, dizendo que:

> (...) nós tomamos a posição de que esta extensão é efetivamente necessária para se obter qualquer compreensão realmente prática dos computadores reais. É uma discussão valorosa para que se possa ver como a teoria da máquina infinita seria mais realista do que a teoria finita, para propósitos práticos.

No entanto, utilizando-se da lógica, é possível concluir que essa extensão aos autômatos infinitos é uma obrigação que vai além da *obrigação de valor* argumentada por Minsky (1967). Será demonstrado abaixo que essa é uma *obrigação lógica*.

Atendo-se apenas aos autômatos finitos, tal como a MEF, podem facilmente ser encontradas exceções que violam a Tese de Turing, a qual

unesp

diz que *todo* procedimento calculável mecanicamente é calculável por uma máquina de Turing. Se a máquina de Turing tivesse que ser definida como um autômato finito, a Tese de Turing seria violada pelo fato de essa máquina não poder executar todos os algoritmos possíveis.

Um exemplo de exceção é o conhecido Problema da Multiplicação. Tudo o que é necessário para a condição de algoritmo está presente na multiplicação, e, no entanto, a multiplicação não pode ser feita nos autômatos finitos. Isso significa que, ao prender-se aos autômatos finitos, tem-se, de início, uma teoria incompleta, que não 'dá conta' de seu objeto.

Se o objetivo da teoria sobre os procedimentos efetivos (ou algoritmos) é 'dar conta' de todos os procedimentos efetivos, não é possível parar no domínio dos autômatos finitos, já que estes permitem que sejam encontradas exceções. Por outro lado, até hoje, não se encontrou nenhum exemplo de violação da Tese de Turing ao se tratar com a máquina de Turing infinita, ou seja, incluindo-se o infinito na máquina, a completude da teoria sobre procedimento efetivo não foi violada. Portanto, a obrigação de que o infinito seja incluído na teoria é dada por uma necessidade lógica.

Sendo assim, deixando-se de lado as soluções práticas que podem ser encontradas para as limitações atribuídas às máquinas finitas, deve-se destacar a importância da memória em uma máquina. De acordo com o estudo feito, pode-se perceber que existem tarefas que podem ser feitas com a MT e não podem ser feitas com a MEF, sendo a memória importante para definir a capacidade fundamental de uma máquina resolver problemas. Sua importância

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

para o 'poder computacional' é tal que a memória é a responsável pela divisão das máquinas, criando a distinção entre dois tipos delas: os *autômatos finitos*, como a MEF, e os *autômatos não finitos*, como a MT.

Por meio de questionamentos envolvendo componentes das máquinas abstratas, foi feita a análise de quais fatores são indispensáveis para a capacidade fundamental de uma máquina *seguidora* de regras resolver problemas. A seguir, serão apresentados os computadores digitais, que são, em essência, máquinas de Turing. O próximo capítulo procurará, mais especificamente, esclarecer a noção de memória na máquina utilizada nos modelos cognitivos simbólicos, ou seja, no computador.

unesp

Capítulo 3

COMPUTADORES

Após o estudo de duas máquinas abstratas, a MEF e a MT, é relevante que se faça a descrição de uma máquina em que, tal como essas duas, as regras abstratas e os algoritmos *governem* o sistema, mas que essa máquina seja uma ferramenta real, concreta. Assim, neste capítulo, é iniciado o estudo das máquinas concretas utilizadas na modelagem das capacidades cognitivas humanas. O computador, amplamente utilizado nos modelos cognitivos simbólicos, possibilita que questões do interesse da Ciência Cognitiva sejam testadas em uma máquina, trazendo o *status* de científico para essa ciência. Para o estudo dos computadores, inicialmente, na Seção 1, é apresentada uma definição do termo computador que demonstra ser relevante para a Ciência Cognitiva, e é feita a caracterização histórica dessa máquina. Em seguida, na Seção 2, é explicitada a categorização dos computadores juntamente com seus componentes básicos, a fim de que se tenha uma boa compreensão de termos essenciais e do modo de funcionamento. Por fim, na Seção 3, é feita uma análise da memória dos computadores. A noção de memória de máquina, particularmente da memória do computador, é a noção de memória que Turing possuía. Dessa forma, esta é aqui melhor apresentada, para que o referencial de Turing fique claramente explicitado.

3.1 Computadores: conceito e origem

3.1.1 O conceito de computador

Ao fazer uma retrospectiva a fim de se chegar ao conceito de computador, foi verificado que, segundo Johnson-Laird (1988), a MT, formulada em 1936, precede a idade do computador por quase uma década. Ele ainda acrescenta que a máquina de Turing universal, capaz de simular qualquer máquina de Turing, é o ascendente abstrato do computador digital moderno.

Após a definição teórica de máquinas feita por Turing, através da MT, é possível deduzir que ficava faltando uma máquina real, concreta, que pudesse incorporar, materializar, essa definição teórica. Na comparação entre a máquina teórica e a concreta, Johnson-Laird (1988, p.51) acrescenta que "qualquer coisa que pode ser computada por um computador digital pode ser computada por uma máquina de Turing". Portanto, é a partir da MT que se pode chegar ao *conceito* desenvolvido de *computador*, essa máquina concreta que tem sido amplamente utilizada para as modelagens dos modelos cognitivos simbólicos.

Na Ciência da Computação, a definição que freqüentemente se encontra para o termo *computador* assemelha-se a esta: computador é uma máquina composta de elementos físicos do tipo eletrônico capaz de executar uma grande variedade de trabalhos, ou seja, capaz de receber, armazenar, manipular e produzir dados ou informações com alta velocidade e precisão, de acordo com um conjunto de instruções detalhadas, escritas de forma adequada (programas).

No entanto, segundo von Eckardt (1996), na definição de computador comumente usada na Ciência da Computação, os termos ´informações´ e ´dados´ são utilizados de maneira ambígüa, muitas vezes como sinônimos, referindo-se a esses termos tanto como sendo o conteúdo daquilo que é recebido, armazenado e manipulado, quanto como sendo o condutor desse conteúdo.

Na Ciência Cognitiva, em que a distinção entre os possíveis e diferentes significados desses termos (informações e dados) é relevante, uma melhor definição para computador seria:

> Um computador é um dispositivo capaz de receber, armazenar, manipular e produzir informação automaticamente em razão de receber, armazenar, manipular e produzir representações dessa informação. Esse processamento da informação ocorre de acordo com um conjunto finito de regras que são efetivos e que estão, de alguma maneira, na própria máquina. (von Eckardt, 1996, p. 114)

É possível perceber aí uma distinção de significados, em que *dado* é a representação de uma informação que chega para ser processada ou que surge como resposta de um processamento. Ele não envolve o conteúdo (interpretação da representação), que é um dos componentes da informação. Sendo assim, pode-se dizer que a informação possui dois componentes: a forma e o conteúdo. Os computadores, por processarem os dados sem que haja qualquer entendimento acerca do que eles representam, são capazes de lidar apenas com a forma. O conteúdo, ou seja, o significado dos dados quando encontrados em um contexto, é dado pelo usuário.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

3.1.2 Histórico dos computadores

Antes de se examinar os principais componentes de um computador moderno e saber como ele armazena e organiza seus dados, será apresentado um breve histórico sobre essa máquina, o que permitirá perceber quão antiga é sua história e quão rápida foi a sua evolução nas últimas décadas.

Iniciando-se o histórico desde tempos mais primitivos, sabe-se que há muito o homem vem tentando livrar-se de trabalhos manuais e repetitivos, entre os quais estão as operações de cálculo. Conforme Alcalde *et al.* (1991, p.7), a palavra ´cálculo´ é originada do termo latino *calculus* e era usada, há milhares de anos, "para denominar pequenas pedras que eram usadas para contar, deslizando-se por sulcos cavados no chão". A partir desse elemento de cálculo, surgiu o primeiro dispositivo manual de cálculo, chamado **ábaco** (± 2500 a. C.).

Nos séculos XVI e XVII, novos dispositivos mecânicos foram inventados para auxiliar os cálculos. No final do século XVI, John Napier inventou os **Bastões de Napier**, colunas de tabelas com números que podiam multiplicar e dividir de forma automática. Em 1642, Blaise Pascal inventou a **Máquina de Pascal**, capaz de fazer somas e subtrações.

Passando para o século XIX, uma criação de Charles Babbage, idealizada desde 1830, se destacou: a **Máquina Analítica**, a qual podia ser programada, introduzindo-se, assim, o conceito de programação. De acordo com Andler (1988, p.31), com essa máquina

> Babbage aprendeu que o essencial [para a universalização de uma máquina] reside não nas operações de base, mas no seu encadeamento sistemático, na sua *organização*. (...) A chave desta organização é a bifurcação condicional – a máquina é capaz de escolher, em tal momento,

> entre um de dois procedimentos dados, em função do valor de um resultado calculado *pela própria máquina* em uma etapa anterior. Babbage tinha compreendido que dotando sua máquina de "se... então" ("*se* C, *então* fazer A; *senão* fazer B"), conferia-lhe a universalidade.

Por volta de 1880, Herman Hollerith propôs dados perfurados em cartões e automaticamente lidos e tabulados em uma máquina especialmente projetada: a **Máquina de Hollerith**. Esta foi de grande valia para a realização do censo nos Estados Unidos.

Contudo, foi no século XX, principalmente na década da Segunda Guerra Mundial (1930-1940), que houve um grande aumento no desenvolvimento dos computadores, uma vez que a guerra intensificou a necessidade de cálculos científicos como, por exemplo, tabelas balísticas (Andler, 1988).

Em 1936, os estudos de Turing e sua construção teórica da **Máquina de Turing** deram início à Teoria Matemática da Computação, da qual se obtém a definição formal de algoritmo e pela qual se demonstrou que nem todos os problemas possuem solução algorítmica. Desses estudos surgiu também a Teoria da Computabilidade, na qual se procuram formas de descrever e representar algoritmos (Alcalde *et al.*, 1991).

Em 1940 surgiu o **Computador Bell a Relé**, que se destacou por ser um dispositivo eletromecânico que funcionava como chaves e podia realizar operações mecânicas. Foi projetado para cálculos científicos, mas não podia ser programado.

Entre 1937 e 1944, em um esforço conjunto entre Howard H. Aiken, da Universidade de Harvard e a IBM (*International Business Machines*

unesp

Corporation), foi criada a Calculadora Automática de Seqüência Controlada, ou **Mark I**, o primeiro computador eletromecânico, que funcionava baseado em relés e engrenagens. Já se usavam cartões e fitas perfuradas como entrada.

O primeiro computador totalmente eletrônico, o **ENIAC** (*Electronic Numerical Integrator and Calculator*), popularmente chamado de "cérebro eletrônico", foi desenvolvido entre 1943-1946, por Mauchly e Eckert, da Universidade da Pensilvânia. Esse computador usava válvulas no lugar de relés, e era mil vezes mais rápido que o Mark I. As instruções que formavam cada um de seus programas eram incluídas em unidades separadas do ENIAC, e eram conectados para se executar a computação. Uma limitação existente era que se gastavam horas fazendo um programa que era feito por modificações nos circuitos, e podia ser executado em segundos.

Em 1945, John von Neumann demonstrou que o computador poderia ter uma estrutura simples e fixa, capaz de executar qualquer tipo de computação, dado um programa na forma adequada, sem a necessidade de se modificar sua parte física.

A idéia de von Neumann, que contribuiu para que computadores e programas se tornassem mais rápidos, flexíveis e eficientes, tem como ponto principal a técnica de se armazenar programas. Nesta técnica, as instruções dos programas e os dados devem ser armazenados juntos, na mesma unidade de memória, e, dessa forma, as instruções também poderiam ser modificadas, tais como os dados. Assim, o conceito de von Neumann pode ser expresso da forma: *programa* + *dados*, com ambos 'residindo' na memória da

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

máquina. Segundo Dupuy (1996, p.77), essa idéia formulada por von Neumann é que " (...) elaborou os conceitos da próxima geração de computadores". Em seus conceitos, a memória tornou-se um local onde as partes de uma computação ficavam armazenadas, eram trabalhadas aos poucos e, finalmente, formavam o resultado da tarefa, sendo esta a base da maioria dos processadores atuais.

À medida que eram notadas as vantagens desse novo conceito, ele tornou-se padrão (Alcalde *et al.*, 1991) e, de 1946 a 1952, com o **EDVAC** (*Electronic Discrete Variable Automatic Calculator*), desenvolveu-se o projeto do primeiro computador com instruções armazenadas. A idéia para o EDVAC era ter seu programa armazenado internamente, dentro do próprio computador. Isto foi possível pelo fato de o EDVAC possuir mais memória interna do que qualquer outro dispositivo de computar da época. Sua memória era produzida por linhas de retardo acústico de mercúrio, o que permitia esse aumento em sua capacidade de armazenar informações. Outra inovação do EDVAC foi o uso do sistema numérico binário, no qual se trabalha apenas com os dígitos 0 e 1.

Entre 1947-1949, foi construído um outro computador, o **EDSAC** (*Electronic Delay Storage Automatic Calculator*). Esta máquina, além dos conceitos já incluídos ao EDVAC, fazia uso de um registrador, que é um dispositivo capaz de armazenar informação, geralmente em pequena quantidade, apenas para as informações que estavam sendo manipuladas no momento.

O grupo de computadores envolvendo o EDVAC, o EDSAC e o **UNIVAC** (Computador Automático Universal, 1951-1952, o primeiro dispositivo de estado sólido desenvolvido, no qual as válvulas foram substituídas

por discos de cristal, tendo sua memória feita com núcleos de ferrite), inaugurou o uso da memória RAM (*Random Access Memory*), principal tipo de memória dos computadores convencionais, e que será estudada em um item à frente. Esse grupo de máquina inclui, também, os primeiros computadores comercialmente disponíveis.

A partir de 1950, a IBM, utilizando-se dessas inovações, passou a lançar vários modelos de computadores (Norton, 1996): o modelo 701, em 1953, usado na Guerra da Coréia; os modelos 702 e 705, em 1955; o modelo 704, um computador científico construído em 1956. Em 1959, com o modelo 7090, um novo componente, muito mais significativo que os outros usados até então, foi inaugurado: o transistor.

O passo seguinte foi dado em 1971, quando surgiu o *microprocessador*, a marca de uma nova geração na tecnologia dos computadores. Com o microprocessador, toda a CPU (unidade central de processamento) do computador passou a ser implementada em um único circuito integrado. Em 1974, a Intel lançou o microprocessador 8080 e, em 1975, a revista *Popular Electronics* apresentou o computador **Altair**, que usava esse microprocessador e tinha uma memória de 256 bytes (mais ou menos o tamanho de um parágrafo).

A partir de 1975 os avanços foram significativos e não pararam de acontecer. Em 1977 surgiu o computador **Apple II**. Este computador tinha 4 Kb de memória, um televisor conectado à máquina e unidades de disco para armazenar os programas feitos nele.

Em 1981 a IBM lançou o **IBM *Personal Computer***. A configuração original padrão do PC usava o microprocessador 8088 da Intel, tinha 16 Kb de memória, teclado, e uma porta para a conexão de um gravador para armazenamento de programas e dados. Sem monitor específico, o computador era ligado a um aparelho de TV. Seu modelo topo de linha incluía 64 Kb de memória, duas unidades de disco flexível, um monitor monocromático e uma impressora matricial.

Após o PC, foram muitas as melhorias feitas: maior velocidade nos microprocessadores, melhor gerenciamento de memória, desenvolvimento de capacidade multitarefa, entre outras, até chegar aos supercomputadores, hoje à disposição do homem.

3.2 Organização dos Computadores: categorização e componentes básicos

3.2.1 Categorização dos computadores

Apesar de a Ciência da Computação não ter desenvolvido uma taxonomia sistemática para tipos de computadores, existem categorias usadas para classificá-los, as quais, inclusive, têm se tornado padrão na informática. Costumam ser consideradas as categorias: geração, classe e modo de processamento.

Quanto à **geração**, resumidamente, os computadores da primeira geração, no início da década de 50, baseavam-se em válvulas. Os computadores da segunda geração, no início da década de 60, usavam transistores. Os da

terceira geração, no final da década de 60, usavam circuitos integrados. A quarta geração, na década de 70, foi marcada pela integração em larga escala (LSI), que permitiu milhares de circuitos serem colocados em um único chip. Para a quinta geração espera-se uma integração em escala muito grande (VLSI), com abordagens sofisticadas, como o uso da inteligência artificial, e um processamento verdadeiramente distribuído.

Relacionado com a **classe** dos computadores, é interessante saber que, a princípio, só havia o termo computador. Depois que tipos diferentes de computadores de propósito geral apareceram, variando de acordo com o tamanho físico e com os recursos, tais como potencialidade de cálculo, capacidade de armazenamento interno e número de periféricos disponíveis, surgiram termos específicos para distingui-los, os quais são colocados aqui em ordem crescente de potência: palmtop, handheld, notebook/laptop, microcomputador, estação de trabalho, minicomputador, mainframe e supercomputador.

De acordo com o **modo de processamento**, há duas classificações. Na primeira, os computadores classificam-se como analógicos ou digitais dependendo da forma como eles manipulam seus sinais elétricos.

Um evento é de tipo *analógico*, ou contínuo, se os valores das suas variáveis não apresentam descontinuidades, e, dessa forma, pode-se pensar nos valores que os números reais podem assumir. Já um evento é de tipo *digital*, ou discreto, se os valores de suas variáveis assumem apenas determinados valores. Neste caso, pode-se pensar nos valores dos números naturais.

Considerando o que se entende por analógico e digital, um computador é *analógico* se manipular sinais elétricos analógicos, ou seja, se "representar seus valores sob forma de sinais que variam continuamente, e que podem assumir uma quantidade infinita de valores dentro de uma faixa limitada, a qualquer instante" (Dicionário de Informática, 1998, p.129). Esses sinais apresentam alterações graduadas ao longo do tempo, usando medidas físicas do tipo contínuo (por exemplo, voltagem, rotação angular, ...) para executar seus cálculos. Ainda, segundo von Eckardt (1996, p.18), esse tipo de computador "representa a informação por meio de magnitudes continuamente variadas, de forma que qualquer mudança no valor dessa magnitude é computacionalmente significante" .

Um computador é *digital* se manipular sinais digitais, descontínuos, ou seja, se representar seus valores através de estados discretos (distintos e separados), tais como os dígitos binários zero e um, correspondendo a dois níveis de voltagem, desligado e ligado. Dessa forma, este computador executa suas operações com informações representadas na forma digital, pelo uso do código binário. As diversas combinações de zeros e uns são usadas para representar todos os tipos de informações – números, letras, símbolos gráficos e instruções de programa. Além disso, os estados discretos são continuamente alterados, de modo a permitir a movimentação, a transformação e o armazenamento das informações. Atualmente, 95% dos computadores são digitais e programados através de linguagens de programação.

unesp

Embora os sinais analógicos, de operação contínua, possam captar melhor as sutilezas do mundo real, nem mesmo os aparelhos eletrônicos mais modernos conseguem reproduzi-los perfeitamente. Também, em circuitos longos, os sinais analógicos acabam perdendo sua intensidade e ficam sujeitos a interferências ou ruídos. Como o sistema digital tem o sinal inicialmente regenerado no código binário, esse sinal pode ser recuperado posteriormente sem interferências, permitindo uma transmissão mais clara.

Algumas analogias têm sido feitas entre o computador digital e o cérebro humano. Por exemplo, em Gardner (1995, p.279), é encontrada a seguinte afirmação de Lashley:

> O cérebro tem sido comparado a um computador digital porque o neurônio, como um comutador ou uma válvula, ou completa ou não completa um circuito. Mas neste ponto a semelhança termina. O comutador do computador digital tem um efeito constante, e seu efeito é grande em proporção ao *output* total da máquina. O efeito produzido pelo neurônio varia com a sua recuperação da fase refratária e com seu estado metabólico. O número de neurônios envolvidos em qualquer ação chega a milhões, de forma que a influência de qualquer um deles é desprezível... Qualquer célula do sistema pode ser dispensada... O cérebro é uma máquina analógica, não digital.

Na segunda classificação estão os computadores *seriais* ou *paralelos*, dependendo da característica de seus processos. Processo é um programa em execução. Um computador é classificado como *serial* se for capaz de efetuar apenas uma operação por vez, e é classificado como *paralelo* se for capaz de efetuar mais de uma operação ao mesmo tempo. Na verdade, um computador só poderá executar mais de uma operação ao mesmo tempo se possuir dois ou mais processadores funcionando simultaneamente; caso contrário, o que pode acontecer é o computador conseguir alternar de uma operação para

outra numa velocidade grande, o que pode dar a ilusão de uma execução paralela. Quando se trata de um número de processos potencialmente ativos, com apenas um sendo executado em cada instante, esse processamento é denominado concorrente. A maioria dos computadores atualmente comercializados é processado em série, possuindo um único processador, e executando seus programas passo a passo.

Estabelecendo-se uma breve comparação com a mente humana, é possível perceber que, diferente da maioria dos computadores digitais convencionais, o cérebro não é um dispositivo com um único processador, mas possui um vasto componente físico paralelo para armazenar as informações. Embora o movimento das experiências conscientes pareçam ser seriais, muitos processos ocorrem em paralelo, sendo a mente capaz de fazer muitas operações de cada vez. Portanto, segundo Thagard (1998) é mais correto comparar a mente a um computador de processador paralelo.

3.2.2 Componentes básicos dos computadores

O sistema de computador moderno é extremamente complexo, composto por processadores, memórias, terminais, interfaces de rede e um conjunto de dispositivos de entrada e saída (Torres, 1998). Sem a intenção de analisar todo o sistema, o que, além de muito extenso, implicaria no desvio do tema do trabalho, serão descritos aqui somente os componentes principais do computador.

Pelo fato de a maioria dos computadores modernos terem alguma relação com a *arquitetura* (conjunto de instruções, organização da memória, estrutura de entrada/saída e estrutura de barramento) projetada por von Neumann, na qual se tem programas e dados armazenados na mesma memória, a arquitetura *padrão* dos computadores baseia-se nela.

Os componentes básicos da arquitetura de von Neumann (Caudill & Butler, 1990; Norton, 1996) são:

- *Dispositivos de entrada e saída* (E/S): elementos que tratam da comunicação do computador com o seu exterior: usuários, outras máquinas ou outros dispositivos.

Os dispositivos de entrada são responsáveis pela entrada de dados no computador. Além de receberem as informações do exterior da máquina, têm também a função de torná-las inteligíveis para ela. São exemplos desses dispositivos o teclado, o mouse e o scanner.

Os dispositivos de saída são responsáveis pelo retorno dos dados já processados pelo computador. Como exemplo pode-se citar o monitor comum e a impressora.

- *Memória*: dispositivo encarregado de armazenar programas, dados e resultados necessários para o sistema realizar seus trabalhos. A memória de um computador possui capacidade finita. A organização e os tipos de memória do computador serão detalhados na próxima seção.
- *Unidade Lógico-aritmética* (**ULA**): componente encarregado de realizar os cálculos lógicos e aritméticos da máquina.

- *Unidade de Controle* (**UC**): componente encarregado de gerenciar as demais unidades do computador. Opera com a memória, leva dados para a ULA e realiza o controle das operações de E/S.

Juntas, a UC e a ULA constituem a *Unidade Central de Processamento* (CPU), componente do computador responsável por coordenar, controlar e realizar as operações da máquina. A CPU conta também com um conjunto de registradores, pequenas memórias temporárias usadas para propósitos gerais na execução de instruções pelo computador. O termo CPU é usado para referir-se ao *processador* do computador.

O processador e a memória, por serem os responsáveis por efetuar a transformação de informações brutas (classificando-as, somando-as,...) em informações úteis, são considerados os componentes mais importantes do *processamento* no computador.

A engenharia dos componentes básicos do computador, baseada na arquitetura de von Neumann, é apresentada na Figura 16. Nesta figura, o fluxo da informação pode ser verificado seguindo-se a orientação das setas.

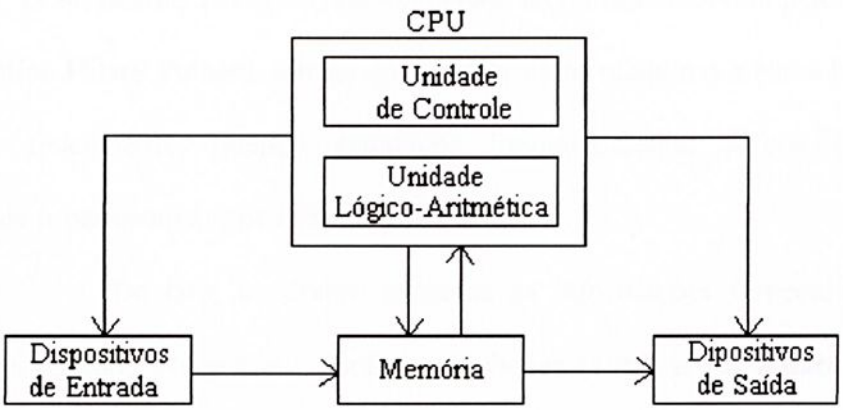


Figura 16 - Engenharia dos componentes básicos do computador

O computador recebe os dados de entrada através dos dispositivos de entrada. Esses dados ficam armazenados na memória, juntamente com o programa a ser executado. Um programa consiste em seqüências de instruções que comandam como deve ser o processamento. A CPU, responsável por todas as ações da máquina, controlando e sincronizando todo o sistema, é quem busca as instruções do programa na memória e as executa, manipulando as entradas de acordo com essas instruções. Terminado o programa, os dispositivos de saída 'recolhem' da memória o resultado do processamento das entradas, que são as saídas, e as apresenta, podendo estas serem visualizadas.

Em Gardner (1995) é encontrada uma declaração de que as atividades do computador parecem, em alguns aspectos, com os processos cognitivos, pelo fato de o computador receber informação, manipular, armazenar e buscar dados na memória, classificar entradas e reconhecer padrões, entre outros aspectos.

Dentre as diversas teorias que tentam caracterizar a mente, tais como o materialismo, o dualismo, o behaviorismo, o naturalismo biológico (Thagard, 1998; Searle, 1997), o *funcionalismo*, teoria desenvolvida pelo filósofo e matemático Hilary Putnam, afirma que, apesar de as máquinas e seres humanos diferirem fisicamente, quando estudados funcionalmente, ambos possuem sistemas de processamento de informação.

De fato, o cérebro processa as informações fornecidas pelos órgãos dos sentidos. No entanto, conforme Kohonen (1987, p.5), "a **natureza** do processamento da informação no cérebro e nos computadores é completamente

diferente". Nem mesmo os computadores convencionais de última geração, baseados na lógica de programação convencional, "ainda não contêm princípios de processamento de informação do mesmo tipo que os encontrados no cérebro" (Kohonen, 1987, p.5).

Com relação à memória, em um computador convencional, como pôde ser visto na Figura 16, ela é um componente à parte, individual. Para que a memória possa ser acessada e manipulada, existe um tráfego contínuo de operações entre ela e os outros componentes da máquina. No cérebro humano, ainda não se sabe em qual de suas partes se encontram os traços de memória, mas, para Kohonen (1987, p.5), "certamente, não existe nenhuma área específica que seja dedicada à memória".

Mesmo que contenha todos os componentes fundamentais analisados, "um computador sem seu software nada mais é do que um pedaço inútil de metal. No entanto, quando equipado com o software adequado, ele é capaz de armazenar, processar e recuperar informação". (Tanenbaum, 1995, p.3)

Essa frase é colocada aqui com a finalidade de deixar claro que, para o computador realizar qualquer atividade produtiva, existe a necessidade da integração entre sua parte física, conhecida como *hardware* e composta por todos os elementos materiais, e sua parte lógica, conhecida como *software*, a qual dota o equipamento físico de programas e dados, capacitando-o a realizar várias atividades.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

O *software*, por ser algo lógico, precisa ser registrado de alguma forma em um suporte físico, e somente a partir daí é capaz de direcionar o trabalho do computador.

Em um computador, o suporte físico do software é a memória. É deste componente básico que este trabalho se propõe a tratar mais detalhadamente. Por isso, a seguir serão apresentados alguns aspectos essenciais para o estudo da memória.

3.3 A memória dos computadores

3.3.1 A organização da memória

"Memória é crucial em computação", disse Johnson-Laird, em 1988 (p.144). De fato, o computador não poderia funcionar sem esse componente básico, e a quantidade de memória disponível em um computador pode influenciar grandemente a potência da máquina. Já há vários anos von Neumann compreendera que uma alta velocidade de processamento seria de pouca utilidade sem que fosse acompanhada de capacidade de memória suficiente. Maior quantidade de memória resulta em potencial para se fazer certas computações, para se executar programas maiores e mais poderosos e para se manipular maior quantidade de informações. De qualquer forma, todo programa útil precisa, primeiramente, de um lugar para armazenar suas instruções, dependendo também desse *local* para guardar seus dados.

Quando os computadores começaram a desenvolver-se, um dos principais problemas encontrados por seus projetistas era o de armazenar dados (Norton, 1996). Na época, um dos dispositivos disponíveis era o relé, que é, essencialmente, um interruptor que podia ser ativado quando lhe era aplicada certa voltagem e desativado quando lhe era removida essa voltagem.

Para construírem um mecanismo capaz de armazenar dados, os projetistas utilizaram-se da tecnologia disponível, e os dados passaram a ser representados em dois estados fundamentais, opostos e distintos, tal como ligado ou desligado, aberto ou fechado, sim ou não.

Para essa representação dos dados foi convencionado que a chave desligada seria representada pelo símbolo 0, e a chave ligada pelo símbolo 1.

Com uma chave ligada e outra desligada, apenas dois números podem ser representados, o zero e o um. No entanto, combinando-se chaves ligadas e desligadas, é possível representar números maiores. Por exemplo, com quatro relés pode-se fazer dezesseis combinações de chaves ligada/desligada, o que permite a representação de dezesseis quantidades distintas.

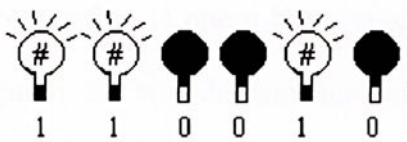
Como, na lógica binária, existem dois dígitos binários, a substituição de cada chave desligada por zero e a chave ligada por um, torna possível contarem-se as representações de quantidades distintas usando-se o sistema binário.

O sistema binário é um sistema de numeração posicional – a posição do dígito no número indica sua contribuição – com base dois (Alcalde *et*

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

al., 1991). Portanto, um número binário é montado através de uma cadeia de símbolos de zeros e uns, os dois símbolos possíveis, sendo sua quantidade determinada de acordo com o símbolo e sua posição dentro da cadeia.

Genericamente, um número XYZTQV na base n equivale a $X * n^5 + Y * n^4 + Z * n^3 + T * n^2 + Q * n^1 + V * n^0$. Especificamente, um número binário pode ser visto na Figura 17.



$= 1*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 0*2^0$

Figura 17 – Exemplo de um número no sistema binário

Atualmente, com a evolução tecnológica, os dispositivos usados na CPU dos computadores são muito mais sofisticados, contudo possuem princípio semelhante ao dos relés (Norton, 1996). E, como é muito mais simples construírem-se máquinas capazes de distinguir duas condições opostas do que qualquer outro número de condições, as máquinas modernas ainda armazenam seus dados por esses dois estados fundamentais (ligado/desligado), sendo o sistema de numeração binário o núcleo da computação digital.

Em virtude dessa representação dos dados pelo sistema binário, todo o conteúdo da memória do computador, em seu nível mais baixo, é representado por números, inclusive letras, sinais de pontuação, símbolos e as próprias instruções do computador. Por usar números binários, estes números são representados com apenas dois símbolos, o zero e o um. Cada um destes símbolos é chamado bit. O termo bit vem da contração de dígito binário, em inglês *binary*

digit, e é definido, no processamento e armazenamento de dados, como a menor unidade de informação tratada pelo computador.

Um bit é representado fisicamente por um elemento específico, como por exemplo, um pulso isolado enviado através de um circuito ou um pequeno ponto num disco magnético, capaz de conter um 0 ou um 1.

Para que todos os outros símbolos usados na escrita possam ser representados, já que o bit representa apenas o zero e o um, deve-se formar um conjunto de bits de tamanho suficiente para a representação, uma vez que inclusive as instruções de um programa são seqüências de bits.

Existem vários conjuntos de dígitos binários e cada conjunto recebe um nome específico.

O conjunto de oito bits é chamado byte – abreviação *de binary term*. Um byte, usualmente, representa a quantidade de memória necessária para se armazenar um único caractere – uma letra, um algarismo, um sinal de pontuação ou outro símbolo qualquer. Um dos sistemas mais importantes desenvolvidos para se representar caracteres usando oito bits, e o mais comum, é o ASCII (*American Standard Code for Information Interchange*), criado pela organização ANSI (*American National Standards Intitute*). Por ter oito bits, o sistema ASCII é um esquema de codificação que permite a representação de 256 caracteres (2^8), sendo capaz de representar caracteres alfabéticos maiúsculos e minúsculos, algarismos, sinais de pontuação, caracteres de controle e outros símbolos. A combinação de oito bits proporciona um número adequado de símbolos; uma combinação com mais de 8 bits proporcionaria muitos símbolos

extras, o que acarretaria em desperdício de espaço. Um exemplo de símbolo da tabela ASCII é a letra **A**:

$$\mathbf{A} = (\mathbf{01000001})_2 = (\mathbf{65})_{10}$$

A letra **A**, na memória do computador é, então, representada como a cadeia dos oito dígitos binários 01000001.

Um byte representa uma quantidade muito limitada de informação. Enquanto este trabalho está sendo digitado no computador, por exemplo, cada caractere de seu texto ocupa um byte de memória. Como pode ser observado, somente até esta página, vários caracteres tiveram que ser armazenados. Portanto, para maiores representações, são necessários vários bytes. O byte é a unidade comumente usada para se "medir" o tamanho da memória em um computador. O número de bytes que um dispositivo de memória pode armazenar corresponde à sua capacidade de armazenamento ou capacidade de memória. Um conjunto de 1024 bytes é chamado kilobyte, e abreviado como K, KB ou Kbyte. O número 1024 vem da idéia de se ter um conjunto de mil bytes já que o prefixo kilo significa 10^3 – mil unidades (Norton, 1996). Como o computador é um sistema binário, sua memória utiliza o multiplicador binário mais próximo para escrever essa quantidade. Assim, o kilobyte vem de 2^{10} = 1024. A capacidade de armazenamento da memória é freqüentemente expressa em unidades teóricas de informação, sendo que os conjuntos de bytes considerados padrão de medidas de quantidade de memória são:

Unidade	Abreviação	Tamanho em bytes	Equivalência
Byte	B	2^0	8 bits
Kilobyte	KB	2^{10}	1024 bytes
Megabyte	MB	2^{20}	1024 kilobytes
Gigabyte	GB	2^{30}	1024 megabytes
Terabyte	TB	2^{40}	1024 gigabytes
Petabyte	PB	2^{50}	1024 terabytes
Exabyte	EB	2^{60}	1024 petabytes
Zettabyte	ZB	2^{70}	1024 exabytes
Yottabyte	YB	2^{80}	1024 zettabyttes

Como foi visto, tudo no nível mais baixo de organização do computador é representado por bits, e parece que ele os 'entende'. Apesar de não ser capaz de responder a toda a linguagem natural humana, ele também parece compreender e responder a uma grande quantidade de expressões próximas dessa linguagem.

Realmente, através de algumas palavras-chave, às vezes até em linguagem natural, fornecidas ao computador, ele pode, com o auxílio de *softwares* especiais que "transformam" essas palavras-chave em zeros e uns, e vice-versa, produzir certas informações que são entendidas pelas pessoas. No entanto, "os computadores não entendem nada. Tudo o que fazem é reconhecer dois estados distintos, produzidos pela eletricidade, pela polaridade magnética ou pela luz refletida. Em essência, eles sabem dizer se um interruptor está ligado ou desligado". (Norton, 1996, p.102)

Para o computador, os bytes são apenas representações de caracteres alfanuméricos, e seus algarismos – seqüências de bits – correspondem a minúsculos pedaços de silício dispostos em fila. Particularmente, a

representação da letra A é 01000001, sendo o segundo e o oitavo pedaços carregados, enquanto os outros pedaços não têm carga.

Conforme Norton (1996, p.102),

> Embora cada sinal minúsculo identificado dessa maneira não tenha significado, o computador contém tantos interruptores, chamados transistores, e opera em velocidades tão fenomenais que podemos usá-lo para que cada um desses sinais forme padrões complexos que sejam significativos para nós. Mas o computador faz tudo isso sem nada entender; as idéias, as visões e as conclusões vêm das pessoas.

Analisando a organização da memória de um computador digital verifica-se que, na tecnologia usada pelos computadores digitais, os dados são representados na memória por códigos binários. Pelo fato de esses códigos não apresentarem características 'inteligentes', será que, ao tentar compará-los a uma possível forma de representação da informação na memória humana, onde não se sabe ao certo como os sinais sensitivos são transformados antes de deixarem os traços na memória, eles seriam suficientes para explicar as representações internas?

Além disso, será mesmo possível criar circuitos elétricos que possam, além de modelar o funcionamento do cérebro, também ser suficientes para modelar a atividade mental, já que, por exemplo, não se acredita que os elementos da memória, tais como as sinapses, tomem um valor binário, mas sim que eles representem valores contínuos mais ou menos ligados ?

Em Gardner (1995) podem ser encontradas opiniões de críticos radicais, como Benny Shannon, que sustenta que a informação contextual afeta o processamento da informação sensorial, e também questiona se algum modelo que

trata os 'bits de informação' separadamente de seu significado e do contexto em que se apresentam, pode fazer justiça à cognição humana.

3.3.2 Principais tipos de memória

Ao se referir à memória de um computador, em relação à estrutura física, pode-se pensar em alguns chips (placas, geralmente de silício, gravadas com circuitos eletrônicos) ligados à placa-mãe (placa principal do computador, onde ficam conectados o processador e seus demais componentes), ou em um disco magnético protegido por um invólucro e com orifícios que possibilitam a leitura ou a gravação da informação, ou outros tipos. Isso leva à conclusão de que, em um computador, podem ser diferenciados vários tipos de memória ou meios de armazenamento de dados, coexistindo.

Por haver a coexistência de muitos tipos de memória, o próprio termo *memória* acabou tomando um significado bastante amplo, podendo ser usado para referir-se a diferentes partes do computador onde os dados podem ser trabalhados e armazenados. No entanto, ela é comumente usada para referir-se à memória principal do computador, a RAM, que é o 'local de trabalho' do computador, ou seja, onde precisam estar os dados em uso.

Segundo Gennari (1999, p.219), a memória pode ser classificada:

> (...) quanto à utilização (principal, cache, real, virtual), quanto à volaticidade (voláteis e não voláteis), quanto ao tipo de acesso (randômica, seqüencial, mista), quanto ao meio (placa, disco, fita), quanto à composição (magnética, de bolha, óptico-magnética, ...), quanto à localização (alta, média, baixa), quanto ao endereçamento (normal, estendida, expandida) e ainda quanto ao custo, ao tempo de

> acesso, à capacidade de armazenamento e outros dados menos importantes.

Devido à existência desses diferentes tipos de memória disponíveis em um computador, eles serão apresentados e comentados, a seguir, a fim de esclarecer as principais tecnologias usadas e suas funções.

- **RAM** (*Random Access Memory* – Memória de Acesso Aleatório)

Quando alguém vai descrever a configuração do seu computador e diz, por exemplo: "tenho um Pentium 233 com 64 MB de memória", normalmente essa quantidade se refere à RAM do computador. A RAM é necessária porque a CPU não possui uma área suficiente para armazenar grande quantidade de bytes, o que é importante para que possa armazenar programas inteiros e os dados manipulados pelos programas.

O computador usa a memória RAM para manter, temporariamente, as instruções e os dados necessários para ele poder executar uma tarefa. Como as instruções e os dados em uso precisam estar contidos na RAM para serem processados pela CPU e, se não estiverem, precisam ser transferidos para ela antes de serem usados, ela também é chamada memória principal ou memória primária. Fisicamente, essa memória consiste em chips alojados em placas de circuitos, ligadas à placa-mãe.

Ao contrário da memória humana, que pode armazenar informações indefinidamente, a memória principal do computador armazena uma quantidade limitada de informações e apenas enquanto o computador estiver

ligado. Os chips da RAM perdem seu conteúdo assim que o computador é desligado – é uma memória volátil. Esta característica permite a classificação dessa memória como sendo uma área de armazenamento de curto prazo incorporada ao *hardware* do computador. Também são assim classificados os dois próximos tipos de memória a serem analisados – a cache e o registrador.

Para melhor entendimento do 'local de trabalho' do computador (RAM), é relevante que seja mostrada sua estrutura. A RAM é subdividida em pequenas unidades chamadas *palavras* (Norton, 1996; Alcalde *et al.*, 1991). Cada 'palavra' possui um endereço único, chamado **posição de memória**, e é capaz de manter uma cadeia de bits de tamanho fixo (geralmente de 8, 16 ou 32 bits). O tamanho da 'palavra', expresso em *bits*, é uma característica importante do computador, pois determina, entre outros, o tamanho dos dados que podem ser armazenados. Uma 'palavra' é o volume de dados que pode ser gerenciado pelo processador em uma operação, ou seja, o conjunto de caracteres tratados como uma unidade para ele. A memória principal pode então ser pensada como uma longa lista de bits, em que cada conjunto deles é associado a uma posição para que ele possa ser localizado. O processo desenvolvido pela CPU ao se executar um programa pode então, de acordo com von Eckardt (1996), ser melhor especificado na forma: leia a instrução que se encontra na memória, determine a posição de memória dos dados requeridos na instrução, execute a instrução, e escreva o resultado na posição de memória desejada.

A representação da memória subdividida pode ser vista na Figura 18.

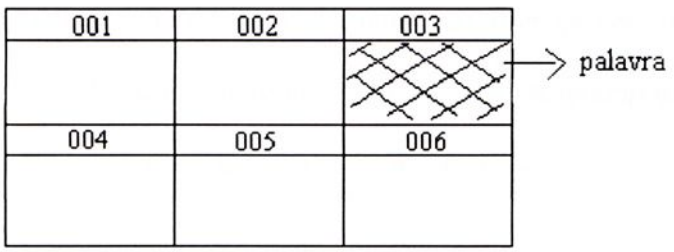


Figura 18 – Representação da subdivisão da memória em *palavras*

Cada retângulo da Figura 18 representa uma 'palavra'. As posições de memória, na maioria dos computadores, são numeradas da esquerda para direita, no sistema hexadecimal, começando em zero e indo em seqüência até o último conjunto de bits de memória disponível.

Na RAM, como em qualquer outro tipo de memória do computador, podem ser armazenados apenas bits, e o conteúdo de cada posição de memória consiste em um conjunto de bits. Assim, considerando-se uma dada posição de memória de 8 bits, por exemplo, a 003, um possível conteúdo para ela, a letra maiúscula L na tabela ASCII, pode ser visto na Figura 19.

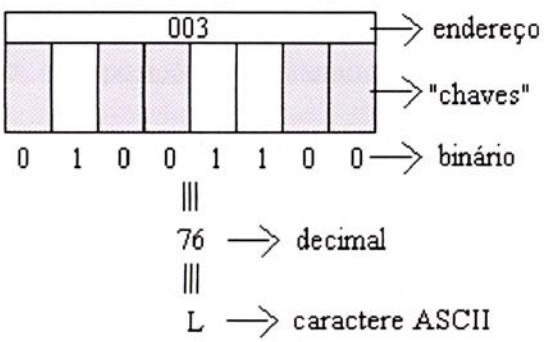


Figura 19 – Representação de uma posição de memória

O processador pode executar duas ações fundamentais com a memória principal: 'ler' e 'escrever' bits em uma dada posição de memória. Na

ação de leitura em uma posição de memória, nenhuma alteração em seu conteúdo é efetuada. Já uma ação de escrita envolve a destruição do conteúdo da posição de memória e a gravação de um novo conteúdo na mesma posição.

- **Memória Cache**

A cache é uma área de memória de alta velocidade, desenvolvida especialmente para acelerar a intermediação entre o processador e a memória RAM. Sendo assim, a CPU pode acessar instruções e dados localizados na memória cache muito mais rapidamente que quando os acessa da memória principal.

O termo cache vem do francês, *cacher*, que significa ocultar, esconder. A idéia provavelmente surgiu do fato de a cache não ter um sentido transparente para o usuário – se o computador tiver memória principal ele funciona com ou sem cache; a alteração só será notada no desempenho da máquina.

A cache acelera o processamento em um computador porque armazena o conteúdo das posições de memória atualizadas da RAM e os endereços onde esses dados estão armazenados. Quando um computador possui memória cache, assim que a CPU tenta ler um endereço da memória, um circuito controlador, chamado controlador de cache, verifica, primeiramente, se o endereço solicitado está na cache. Caso não esteja, os dados são recuperados da memória primária, levados para a CPU, e copiados na cache. Na próxima vez que esses mesmos dados forem pedidos, eles já estarão na cache, o que acarretará em um acesso mais rápido.

Conforme Torres (1998, p.83), "memória cache é apenas um outro nome para a memória estática que está funcionando como intermediária na conversação do processador com a memória RAM".

- **Registradores**

Os registradores são conjuntos de bits de alta velocidade existentes dentro da CPU. Localizados na ALU, em geral os registradores têm espaço suficiente para armazenar apenas alguns bytes de informação. Sua principal função é o endereçamento de memória, ou seja, nos registradores é que ficam temporariamente armazenados os endereços dos dados prestes a serem processados, ou os resultados intermediários de algum processamento.

O tamanho dos registradores é o que determina a quantidade de dados com que o computador pode trabalhar num dado momento. Quanto maior o tamanho dos registradores, maior o endereço, maior a capacidade de memória e mais depressa o computador conseguirá processar os dados; sendo, assim, um componente importante para a velocidade global do equipamento.

- **ROM** (*Read Only Memory* – Memória Somente de Leitura)

Quando o computador é ligado, deve haver no equipamento um tipo de memória que o 'ensine' o que fazer inicialmente. Este é um dos principais motivos pelos quais o computador precisa da memória ROM. Por exercer essa função, a ROM é uma memória permanente, não-volátil, usada para armazenar instruções vitais para o computador, entre elas, aquela que o faz começar a funcionar. Deve ser lembrado que, ao ligá-lo, a RAM está vazia, sem qualquer instrução disponível. Quanto à estrutura física, ela consiste em um chip de

memória capaz de reter instruções e dados mesmo quando o computador é desligado.

A ROM é um tipo de memória cujo conteúdo, normalmente, pode ser lido, mas não alterado, pois é projetada para desenvolver funções específicas, as quais já vêm gravadas de fábrica em seu chip. Assim, sua característica de não poder ter os dados facilmente modificados funciona como uma medida de segurança contra erros acidentais ou mesmo contra algum ato malicioso.

- **Memória Secundária**

Como os dados que estiverem apenas na RAM, inevitavelmente serão perdidos assim que o computador for desligado, foram desenvolvidos sistemas de armazenamento de dados capazes de manter os dados para um futuro uso, criando-se, para isso, um meio de guardar os dados que não estão sendo processados no momento. Estes sistemas recebem o nome de meios de armazenamento ou, ainda, memória secundária e, devido à característica de manter os dados intactos indefinidamente, mesmo quando o computador for desligado, são classificados como um tipo de memória de longo prazo.

Sem entrar em detalhes sobre os tipos de memória secundária disponíveis atualmente, podem ser destacados, de forma bem resumida, alguns dos tipos mais conhecidos. Assim, a memória secundária envolve, entre outros: disquetes, discos rígidos, fitas magnéticas, CD.

Os três primeiros dispositivos citados (disquetes, discos rígidos, fitas magnéticas) usam a tecnologia magnética para ler e gravar dados. Para

armazenar os dados, suas superfícies são revestidas de um material de sensibilidade magnética que, em geral, são milhões de partículas de óxido de ferro, que reagem a um campo magnético, podendo a força de um campo magnético ser usada para representar dados. Neste caso, as modificações da polaridade magnética são utilizadas para codificar as informações na forma binária – enquanto uma polaridade é igual a 1, a polaridade oposta é igual a zero. Assim, se o bit 0 é representado pela magnetização positiva, o bit 1 será pela negativa, e vice-versa.

Como alternativa ao armazenamento magnético, foi desenvolvido o armazenamento óptico, representado aqui pelo CD-ROM. Os dados em um CD são armazenados por meio de raios laser, e a unidade leitora do CD é um mecanismo óptico que utiliza uma fonte de raios laser para percorrer o disco à procura de informação.

Apesar das diferentes tecnologias usadas nos tipos de memória, o fato é que, seja ela qual for, todas armazenam bits. O código desse armazenamento é criado considerando-se apenas dois estados físicos (0 ou 1), os quais podem ser implementados de forma diferente nos variados tipos de memória.

3.3.3 Hierarquia da memória

Quando a CPU precisa de algum dado, este pode ser encontrado na cache interna, na cache externa, na memória primária ou na memória secundária.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

As diferentes áreas de armazenamento de dados em um computador, do ponto de vista operacional, podem ser organizadas em níveis conhecidos como hierarquia de memória. Esses níveis informam a distância para a CPU, a velocidade de acesso e o tamanho, de forma que as memórias menores, mais rápidas e mais caras ficam no topo, enquanto memórias maiores e mais lentas ficam na base.

A Figura 20 mostra a hierarquia de memória em um computador de alto desempenho.

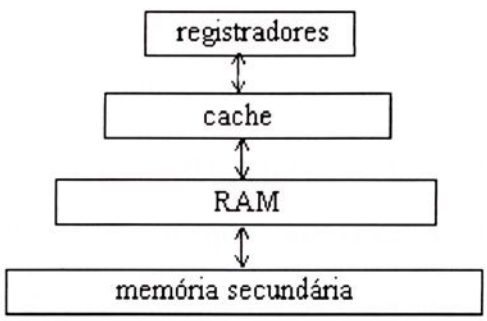


Figura 20 – Hierarquia da memória

Analisados os principais tipos de memória de um computador, os diferentes tipos de acesso e a hierarquia, notam-se, de acordo com Johnson-Laird (1988), algumas divergências entre a memória do computador e a memória humana.

O computador, através da RAM, permite um meio termo entre o lento armazenamento de longo prazo (qualquer tipo de memória secundária) e o rápido armazenamento de curto prazo (responsável pelos resultados intermediários da computação: os registradores).

A memória humana parece estar organizada em um sistema em que ambas as memórias, a de curto e a de longo prazo, são de fácil acesso (tal como a RAM), e são mantidas de forma relativamente permanente (tal como a memória secundária). O sistema de memória humana possui ainda uma memória sensorial, enquanto os computadores, em sua maior parte sem qualquer tipo de 'órgão dos sentidos' , não a tem. Para os futuros robôs, especialmente os andróides, equipados com visões, certamente haverá a necessidade de uma grande memória, na qual serão mantidas as várias representações visuais que serão computadas.

A idéia de Johnson-Laird (1988), de que as pessoas possuem a mesma facilidade para acessar a memória de longo prazo, não estabelece uma unanimidade com Sternberg, o que, inclusive, exemplifica a dificuldade encontrada para se explicar o complexo funcionamento da memória humana. Em Soares *et al.* (1997, p.19) encontra-se que:

> A partir de 1966, Sternberg fez um estudo sobre a busca de informações na memória a curto prazo, descobrindo a rapidez desta, constatando, assim que a busca na memória a longo prazo, era aparentemente demorada.

Portanto, segundo Sternberg, as velocidades nos acessos às memórias de curto e de longo prazo seriam equivalentes ao tempo de acesso às memórias do computador consideradas de curto e de longo prazo, mantendo-se a inexistência, na memória humana, de um meio termo que é representado, no computador, pela memória RAM.

unesp

3.3.4 Importância do sistema de memória

Conforme já enfatizado anteriormente, a memória é crucial para a operação de um computador. Para executar qualquer tarefa, o computador precisa de instruções que lhe digam o que fazer, e a memória é o meio físico que as mantém. Sem memória um computador não poderia ler programas ou reter dados. Além disso, a memória tem significante importância nos seguintes aspectos do sistema do computador:

- *Desempenho* – a quantidade de memória em um computador é um importante fator de contribuição para o desempenho do sistema. Trabalhar com memória insuficiente pode causar uma diminuição de 50%, ou mais, do potencial do computador, o que ocasiona limitações ao sistema. De acordo com Thagard (1998, p.132-3), "ao desenvolver um modelo computacional, precisamos preocupar-nos com a velocidade e a flexibilidade, bem como com o potencial computacional abstrato. Existem muitas formas de realizar as computações, mas para a Ciência Cognitiva necessitamos de técnicas computacionais que tenham a velocidade e flexibilidade necessárias à plausibilidade psicológica e aplicabilidade prática".
- *Segurança e estabilidade* – normalmente, a memória de um computador é segura e estável, pois, a menos que haja algum erro, um computador nunca *esquece*. Enquanto o computador estiver ligado, a RAM e as memórias secundárias, quando em perfeito estado, são um meio seguro de manter os dados. Se por um lado, em condições ideais, um computador não esquece, a ocorrência de um único bit ruim em seu *hardware*, uma pontinha de corrosão em uma das

unesp

tomadas, ou mesmo uma breve queda no suprimento de energia, são suficientes para que os dados sejam perdidos. Tratando-se da memória humana, embora esta tenha seus lapsos e fraquezas ocasionais, mesmo o ser humano não estando em condições ideais – pode estar cansado, de ressaca ou possuir algum dano cerebral – em geral, ele mantém suas atividades, ainda que em ritmo mais lento e de forma menos precisa. Além disso, a memória humana possui a habilidade de, a partir de apenas alguma informação, mesmo que parcial, poder lembrar-se de outras, o que contrasta com o fato de um único bit ruim colocar em risco toda a memória do computador.

- *Suporte ao software* – geralmente, quanto mais recente for um programa, mais memória ele irá requerer para que possa ser executado. Para que se tenha uma idéia de como a quantidade de memória necessária aumentou nos últimos anos, quando o IBM-PC foi lançado, em 1981, ele possuía 64 KB de RAM disponível, o que significou, na época, um grande avanço.

Ligado à quantidade de memória, há um aspecto importante sobre o qual se tratará a seguir: seu limite.

3.3.5 Limites do sistema de memória

Dado que todos os tipos de memória em um computador são finitos, com tamanhos determinados, geralmente medidos em bytes, e que a quantidade de memória disponível influencia na performance do sistema e determina a possibilidade de acesso a certos programas, podem ser feitos os seguintes questionamentos: qual é a quantidade de memória necessária para um computador? O que determina o limite para sua memória finita?

Na verdade, a quantidade de memória necessária para um computador executar tarefas é algo particular, pois essa quantidade varia de acordo com o tipo de trabalho que se deseja que ele execute, e de acordo com os tipos de software que serão usados na máquina.

No entanto, mesmo que se queira executar trabalhos de envergadura expressiva em um computador - o que exige grande quantidade de memória disponível - deve-se ter em mente que, todo computador tem uma quantidade máxima de memória que ele pode suportar. Os fatores que impõem o limite na quantidade de memória que pode ser usada em um computador são vários:

- *Limites de endereçamento* – todo processador tem um limite no número de bytes, ou seja, de endereços de memória que ele consegue acessar. Esse limite é dado pela amplitude de um sistema existente no computador chamado barramento de endereços (*address bus*), que consiste em um conjunto de fios usados para indicar o endereço de memória de onde, ou para onde, os dados são transportados em operações de leitura e escrita. A amplitude do barramento de endereços controla a endereçabilidade da memória, que corresponde à quantidade de memória que um processador pode usar. Quanto maior a quantidade de fios, mais bits poderão passar em um barramento, e mais memória poderá ser acessada pelo processador.
- *Limites de chipset* – todo computador tem em seu *hardware* um circuito lógico, chamado controlador de memória, responsável por gerar os sinais necessários para controlar a leitura e a escrita dos dados. Ele é, normalmente,

integrado dentro de um conjunto de chips – o *chipset*, que é uma coleção de circuitos integrados, projetados para trabalharem juntos na execução de determinadas tarefas. Como o controlador de memória é parte do sistema de *chipset*, cada *chipset* tem um limite de quantidade de memória que pode controlar.

- *Limites físicos* – os computadores modernos possuem a memória principal fisicamente desenvolvida em módulos, que são pequenas placas de circuitos impressos integrados, soldados juntamente. A placa-mãe com o *chipset* é que determina quantos módulos de memória podem ser usados. Como existe o limite de largura e de número de módulos possíveis, o espaço físico também impõe um limite na quantidade de memória.

As considerações feitas neste capítulo conduzem a uma concordância com a idéia de Kurzweil (1999), que quanto mais complexas forem as expectativas em torno das possibilidades de um computador digital, quantidade maior de memória será necessária. Por meio de dados estatísticos, ele mostra que, no começo do século XX, os computadores tinham seu desempenho aperfeiçoado em dobro aproximadamente a cada três anos; atualmente, porém, esta melhora se dá a mais ou menos cada doze meses. Com base nesse estudo, Kurzweil faz uma previsão para os computadores digitais: próximo ao ano 2020, os computadores alcançarão a capacidade de memória e a velocidade de computação do cérebro humano, o que permitirá, ao menos, diminuir a desigualdade de capacidades e desempenhos atualmente encontrados entre o cérebro humano e o computador mais avançado. Ele considera que, atualmente, os computadores sejam cerca de um milhão de vezes mais simples que o cérebro, dependendo do domínio a ser

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

considerado. A capacidade de memória do cérebro humano é de, aproximadamente, cem trilhões de sinapses, número que pode ser estimado como sendo um milhão de bilhões de bits de capacidade de memória nos circuitos eletrônicos. Kurzweil acredita que uma quantidade de silício equivalente a esse número de sinapses proporcionará uma velocidade de execução de tarefas perto de um bilhão de vezes mais rápida que o cérebro humano.

Tais idéias incitam ainda mais a polêmica em torno da hipótese de Turing sobre as máquinas. No entanto, elas são aqui consideradas a fim de sugerir algo mais concreto: que a tendência de se buscar memória cada vez mais eficiente e em maior quantidade continuará, seja para alcançar a diminuição da desigualdade de capacidades homem/máquina, seja para aumentar o volume de informação a ser armazenado.

Resta saber se, alcançando-se a capacidade de memória que se supõe ser satisfatória, ela corresponderá às expectativas em torno dos computadores vistos como instrumentos apropriados para simular a inteligência humana, conforme sustentam alguns pesquisadores.

Comentada a máquina utilizada nos modelos cognitivos simbólicos, a saber, o computador, será feito, a seguir, um estudo sobre outro tipo de máquina, que é a utilizada nos modelos cognitivos conexionistas, conhecida como Redes Neurais Artificiais.

CAPÍTULO 4

REDES NEURAIS ARTIFICIAIS

Neste capítulo é feito um estudo sobre as Redes Neurais Artificiais, as quais são propostas numa tentativa de trazer idéias sobre a possibilidade de a arquitetura dos sistemas conexionistas permitir que se construam sistemas verdadeiramente inteligentes e, em especial, se isto seria alcançado devido à memória desse sistema. Utilizadas nos modelos cognitivos conexionistas, as Redes Neurais Artificiais apresentam uma estrutura com mais plausibilidade biológica, sendo inspiradas no sistema nervoso real. Elas não constituem uma idéia abstrata: existem muitas delas já construídas e sendo amplamente aplicadas. Pelo fato de inovarem em relação aos computadores, fazendo possível suas aplicações sem que regras explícitas sejam dadas, talvez as Redes Neurais Artificiais possam contribuir para uma simulação mais adequada de alguns aspectos cognitivos. Assim, na Seção 1, é feita uma apresentação dos princípios fundamentais e das características das Redes Neurais Artificiais, incluindo também o que são, qual o objetivo e um breve histórico dessas redes. Na Seção 2, passam a ser mostrados os componentes básicos das Redes Neurais Artificiais, em que consiste a aprendizagem, o treino e a memória nessas redes. Por último, na Seção 3, é apresentada a ilustração de um tipo de rede neural neurofisiologicamente motivado em memória associativa: a Rede de Hopfield. Também, nesta seção, são apresentadas as características da memória associativa, de forma que se possa perceber como a organização da memória no modelo de Hopfield parece ser próxima à memória humana.

4.1 Princípios Fundamentais das Redes Neurais Artificiais

4.1.1 Por que Redes Neurais Artificiais ?

Apesar do crescimento exponencial verificado desde o início do século em termos de 'poder', os computadores convencionais, baseados na lógica digital, têm sido bem sucedidos em apenas algumas áreas.

De fato, em determinados domínios restritos tais como jogos de xadrez (quem não se lembra da vitória do famoso computador Deep Blue sob Gary Kasparov, em 1997?), compra e venda de ações, e certos diagnósticos médicos; os computadores parecem mesmo ter ultrapassado a inteligência humana. No entanto, os seres humanos permanecem superiores em tarefas que exijam flexibilidade e criatividade. Escrever o roteiro de um filme, responder com humor, descrever o ambiente em que se está, pintar um quadro, são exemplos de atividades que, pelo menos por enquanto, um computador convencional é incapaz de executar.

A Ciência da Computação contemporânea, conforme ressaltam Levine *et al.* (1992), é baseada em descrições formais. Nessas descrições, as entidades de interesse são representadas por símbolos, os quais não possuem, necessariamente, ligações com o conteúdo semântico que representam. Nesse caso, "'computação' é um processo formal [determinado por regras] em que operadores [seqüências de regras de transformações formais] são aplicados a *tokens* [símbolos] para produzir outros *tokens*" (Levine *et al.*, 1992, p.380).

A aplicação das regras formais, em uma computação simbólica, é mediada por um intérprete. O intérprete é o responsável pela seleção das regras a serem aplicadas a um conjunto de símbolos num dado momento.

Segundo Bittencourt (1996), existem duas principais linhas de pesquisa para a construção de *sistemas inteligentes*: a linha Conexionista e a linha Simbólica, a qual será, neste trabalho, referenciada como Inteligência Artificial (**IA**).

Incluída na forma de computação comentada acima, baseada na aplicação de regras formais, está a linha *Simbólica* ou, simplesmente, IA.

A linha **Simbólica** segue a tradição lógica. Para representar o conhecimento, ela utiliza a lógica de predicados, redes semânticas, roteiros, e a aprendizagem é baseada em explicações, descobertas, espaços de versões. A IA foi construída a partir de idéias filosóficas, científicas e tecnológicas herdadas da integração de outras ciências. A possibilidade da existência de uma mente mecânica é uma questão central para estudiosos desse campo e envolve um sonho muito antigo. Uma definição precisa e formal para IA envolveria questões filosóficas, numa tentativa necessária de definir o significado tanto de inteligência quanto de artificial, motivo pelo qual alguns pesquisadores se recusam a propor tal definição.

A linha **Conexionista** pretende modelar a inteligência humana simulando componentes do cérebro: os neurônios e suas interligações. Nesta linha, as regras não são usadas como uma estrutura explícita. Para representar o conhecimento, ela usa um grande número de conexões valoradas. Há 3 tipos de

treinamento para a aprendizagem nesta linha: supervisionado, por reforço e não supervisionado.

Essas duas grandes linhas de pesquisa constituem também as duas principais linhas representacionistas da Ciência Cognitiva: a IA e o Conexionismo. A distinção entre elas é assim exposta por Gonzales (1998, p.3):

> (...) [a] Inteligência Artificial (**IA**), focaliza as *regras abstratas e os algoritmos* no estudo dos elementos estruturadores das representações mentais, que por hipótese constituiriam os pilares do nosso universo cognitivo. (...) [o] Conexionismo, ou Redes Neurais Artificiais (**RNA**), enfatiza as *leis naturais e os mecanismos físicos* atuantes no processo de formação, organização e desenvolvimento de tais representações. Apesar de distintas, essas duas tendências possuem em comum pressupostos básicos fundamentais que direcionam os seus programas de pesquisa através da elaboração de modelos computacionais para a explicação da natureza e estrutura dos estados mentais.

Sendo assim, ambas as linhas de pesquisa comentadas tentam resolver questões relacionadas ao comportamento inteligente através dos modelos mecânicos, mas com enfoques diferentes e com diferentes pontos fortes e fracos.

Por algumas razões, tais como a falta de computadores suficientemente potentes para a complexidade do método, e o apontamento de algumas limitações das Redes Neurais, a linha Simbólica, inicialmente, teve a preferência dos pesquisadores.

Nos anos 80, alguns fatores como melhor conhecimento da estrutura do cérebro, melhores algoritmos, progressos teóricos, maior facilidade de simulação, tecnologia disponível para fabricar *hardware* especializado, computação paralela, e os sucessos obtidos em aplicações, conduziram a um renascimento do interesse e enfoque maior sobre o Conexionismo, atraindo vários

unesp

pesquisadores: engenheiros elétricos, cientistas da computação, filósofos, psicólogos e matemáticos, entre outros.

Dado que a MEF, a MT e o computador digital manipulam símbolos, seguem regras lógicas explícitas, e, segundo Kóvacs (1997, p.144), constituem, atualmente, " (...) a ampla área das técnicas essencialmente algorítmicas de IA", essas máquinas serão tomadas, neste trabalho, como exemplos para a primeira linha representacionista da Ciência Cognitiva. A seguir, será melhor analisada a máquina utilizada na segunda linha representacionista desta ciência – as Redes Neurais Artificiais.

Uma Rede Neural Artificial (**RNA**) consiste em um sistema de processamento de informações de inspiração biológica, composto por um grande número de unidades de processamento ligadas por uma rede de interconexão.

As Redes Neurais Artificiais [14] , também referidas por Sistemas Conexionistas, Processamento Paralelo Distribuído (PDP), Redes Adaptativas ou Computação Seletiva (Bittencourt,1996), têm por *objetivo* entender o funcionamento fisiológico e psicológico do sistema neural humano e investigar a possibilidade de simular comportamentos inteligentes através de modelos baseados na estrutura e funcionamento do cérebro.

4.1.2 Breve Histórico das Redes Neurais Artificiais

Sem o intuito de buscar um aprofundamento na origem dessa linha de pesquisa, é encontrado em Dupuy (1996) a afirmação de que o artigo pioneiro sobre RNA, intitulado *A Logical Calculus of the Ideas Immanent in*

14 As Redes Neurais Artificiais referenciadas neste trabalho são, mais rigorosamente, modelos simulados em RNA; e não são, de fato, as próprias RNA.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

Nervous Activity, foi publicado no ano de 1943, e assinado pelo neuropsiquiatra Warren McCullock e pelo matemático Walter Pitts. Nesse artigo, os autores "partem em busca dos mecanismos materiais e lógicos que encarnam a mente" (Dupuy, 1996, p.52). McCullock e Pitts mostraram que o comportamento do neurônio biológico pode ser modelado em termos matemáticos e foram os primeiros a propor um modelo do cérebro sob a forma de uma rede de neurônios idealizados. Suas conclusões foram de extrema importância para a futura implementação computacional do neurônio artificial, sendo as principais delas:

- a atividade do neurônio é um processo 'tudo ou nada', ou seja, um neurônio estará ativado, e transmitirá a saída para outros neurônios da rede, se sua saída ultrapassar um valor limite; caso contrário, estará em repouso;
- cada entrada de um neurônio tem um valor ou peso associado. A determinação da saída de um neurônio é feita pela soma ponderada das entradas com os ganhos arbitrários de entrada como fatores de ponderação. Sendo este um valor positivo, o neurônio tenderá a excitar-se; caso contrário, tenderá a inibir-se.

O neurônio formal de McCulloch e Pitts é algo simples, podendo ser entendido como o seguinte dispositivo binário: caso o resultado dos cálculos da saída do neurônio seja maior ou igual a um dado limiar, a saída do neurônio é um pulso; caso contrário, um não pulso. Kovács (1996, p.29) admite que a essência da proposta de McCulloch e Pitts foi a seguinte:

> A inteligência é equivalente ao cálculo de predicados que, por sua vez, pode ser implementado por funções booleanas. Por outro lado, o sistema nervoso é composto de redes de neurônios, que, com as devidas simplificações, tem a capacidade básica de implementar estas funções booleanas. Conclusão: a ligação entre inteligência e atividade nervosa fica estabelecida de forma científica.

Em 1949, Donald Hebb, no livro *The Organization of Behavior*, propôs um modo de fazer a atualização de pesos sinápticos (Rumelhart & McCleland, 1986, p.36) e esclareceu alguns pontos importantes:

- numa RNA, a informação é armazenada nos pesos;
- quando as unidades de processamento A e B são simultaneamente excitadas, o peso da conexão entre elas é aumentado;
- o peso da conexão entre duas unidades de processamento é simétrico (ou seja, o peso de uma conexão entre unidades A e B é igual ao peso da conexão de B para A);
- os pesos são alterados quando ocorre aprendizado.

A partir dessa época outros pesquisadores se destacaram, como Minsky (1954), Selfridge (1961), Block (1962), e contribuíram para se chegar a um modelo geral de RNA (Rumelhart & McCleland, 1986).

O primeiro modelo de RNA implementado foi o **perceptron**, construído por Frank Rosenblatt, em 1962. O perceptron é uma rede de múltiplos neurônios, os quais ficam dispostos em várias camadas. Os neurônios que recebem diretamente as entradas formam a 'camada de entrada'; os neurônios que ficam na camada final formam a 'camada de saída'; os neurônios que não formam nem a camada de entrada nem a de saída constituem as camadas internas, comumente referenciadas como 'camadas ocultas'. A grande novidade do perceptron foi ele possuir a propriedade de poder aprender a computar. Esta propriedade foi adicionada por Rosenblatt ao neurônio de McCulloch e Pitts (Rumelhart & McCleland, 1986). Apesar do impacto desse modelo, ele foi

fortemente criticado por Marvin Minsky e Seymour Pappert no livro *Perceptrons*, pela sua incapacidade de solucionar alguns problemas muito simples, como por exemplo, o problema do XOR. A função XOR é uma função lógica que aceita duas entradas binárias e gera apenas uma saída binária. Nessa função, duas entradas binárias irão gerar uma saída *1* se apenas *uma* delas for *1*; caso contrário, irão gerar uma saída *0*. O problema está no fato de ser impossível para o perceptron aprender a resolver um problema com a característica de ter que separar, de forma linear, duas saídas de categorias diferentes, como é o caso do *0* e do *1*. Isto se deve ao fato de, nesse modelo, a superficie de decisão linear não existir; ou seja, não há uma *única* linha que consiga separar as saídas *1* das saídas *0*. Rich *et al.* (1994, p.573) observam que a deficiência está "no modo como o perceptron representa o conhecimento".

A publicação de *Perceptrons*, em 1969, cujos autores defendem a hipótese de que os modelos apresentados não possuem sustentação matemática para terem credibilidade, levou a um desestímulo de estudo nesse campo da RNA, iniciando um período chamado de *anos negros* das Redes Neurais Artificiais – os anos 70. Contudo, apesar das dificuldades de pesquisa na área, ela não foi abandonada e alguns pesquisadores chegaram a novas concepções de RNA.

Assim, em 1972, Teuvo Kohonen definiu um novo modelo de RNA, conhecido como mapa auto-organizável de características (Rich *et al.*, 1994). No **modelo de Kohonen** não há apenas um neurônio responsável pela representação do padrão de entrada, mas sim um conjunto de neurônios que se

interagem. Seu trabalho também foi importante por ter introduzido o aprendizado não-supervisionado no estudo das RNA.

Uma nova evolução nos estudos das RNA ocorreu em 1982, com a contribuição do físico John Hopfield, criador do **modelo de Hopfield**, envolvendo memórias associativas.

Em 1986, a aprovação do modelo **Backpropagation** ou da retropropagação do erro, que é uma rede de múltiplas camadas, inventada e popularizada por Rummelhart, Hinton e Williams (Kovács, 1996), ampliou o potencial do perceptron, permitindo a superação das limitações do modelo primitivo e recuperando o prestígio das Redes Neurais Artificiais.

Desde então, vários modelos de RNA foram desenvolvidos e aperfeiçoados, tornando-se cada vez maior o número de suas aplicações práticas.

4.1.3 Neurônio Biológico / Neurônio Artificial

O neurônio biológico "foi identificado anatomicamente e descrito com notável detalhe pelo neurologista espanhol *Ramón y Cajal*, no século XIX" (Kovács, 1996, p.13). Um **neurônio biológico** (Figura 21) consiste em uma célula nervosa capaz de realizar uma forma de processamento de informação. De acordo com o resultado do trabalho de vários pesquisadores, "passou-se a entender o neurônio biológico como sendo, basicamente, o dispositivo computacional elementar do sistema nervoso, que possui entradas – muitas entradas – e uma saída" (Kovács, 1996, p.15). Neste tipo de neurônio, os dendritos são os responsáveis pela recepção dos sinais de entrada. O corpo da

célula – o centro dos processos metabólicos - processa os sinais e o axônio transmite a informação. Os pontos de conexão entre os neurônios que conectam os dendritos de um neurônio ao axônio de outro, são chamados sinapses. Nos neurônios do sistema nervoso central dos vertebrados, as entradas ocorrem através das sinapses, e são sinais conhecidos como impulsos nervosos. Os impulsos nervosos "constituem a informação que o neurônio processará, de alguma forma, para produzir como saída um impulso nervoso no seu axônio" (Kovács, 1996, p.15).

Cada neurônio é estimulado pelas sinapses, ou seja, por uma ou mais conexões vindas de outros neurônios. Dependendo do sinal produzido pela força das conexões, o estímulo da sinapse será de natureza inibitória ou excitatória. Uma conexão excitatória "contribui para a formação de um impulso nervoso no axônio de saída, enquanto que uma conexão inibitória age no sentido oposto" (Kovács, 1996, p.15).

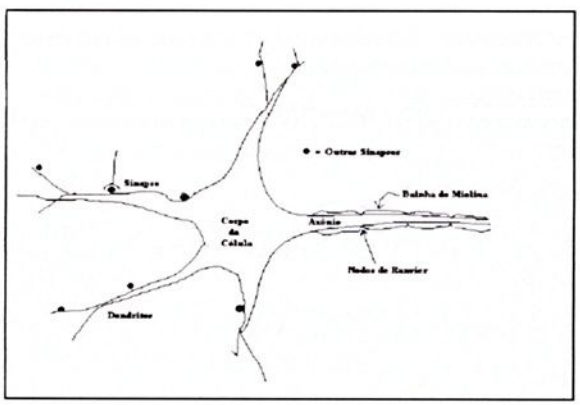


Figura 21 - Neurônio biológico

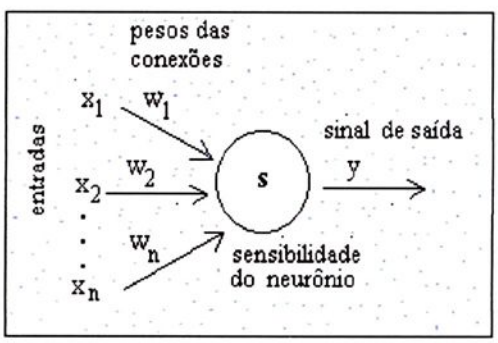


Figura 22 - Neurônio artificial

A estrutura de uma RNA é inspirada biologicamente e, dessa forma, procura ser organizada de acordo com a arquitetura neural biológica. Seus

componentes, os **neurônios artificiais** (Figura 22), referenciados neste trabalho como **neurodos** (Caudill & Butler, 1990), baseiam-se em células nervosas e nas conexões existentes entre elas. Apesar de os neurônios biológicos serem um tanto complicados e ainda não completamente entendidos, os estudiosos de modelos de Redes Neurais Artificiais identificaram o que eles fazem de mais significativo, a fim de que os neurodos desenvolvam as funções mais elementares de um neurônio biológico. Os neurodos foram assim descritos por Pinker (1998, p.110-1):

> Os neurônios, efetivamente, somam uma série de quantidades, comparam a soma com um limiar e indicam se o limiar foi excedido. Essa é uma descrição conceitual do que eles fazem; a descrição física correspondente é que um neurônio disparando é ativo em vários graus, e seu nível de atividade é influenciado pelos níveis de atividade dos axônios entrantes de outros neurônios ligados em sinapses aos dendritos do neurônio (estruturas de inputs). Uma sinapse possui uma força que vai de positiva (excitatória), passando por zero (nenhum efeito) a negativa (inibitória). O nível de ativação de cada axônio entrante é multiplicado pela força da sinapse. O neurônio soma esses níveis entrantes; se o total exceder um limiar, o neurônio se tornará mais ativo, enviando por sua vez um sinal para qualquer neurônio ligado a ele. Embora os neurônios estejam sempre disparando e os sinais entrantes meramente os façam dispara a uma velocidade detectavelmente maior ou menor, às vezes convém descrevê-los como estando desligados (velocidade de repouso) ou ligados (velocidade elevada).

4.2 Caracterização das Redes Neurais Artificiais

4.2.1 Componentes Básicos

São vários os modelos de RNA, sendo que, quase todos são estruturados a partir destes componentes: padrões, funções, conexões e unidades de processamento.

Os **padrões** podem ser entendidos como conectividades estabelecidas entre as unidades de processamento. O reconhecimento de padrões é uma das principais aplicações bem sucedidas das RNA.

As seguintes **funções** caracterizam os modelos matemáticos utilizados no treinamento e reconhecimento de padrões (Rumelhart & McCleland, 1986; Kóvacs, 1996):

- Funções de ativação: responsáveis por determinar o nível de excitação ou atividade de um neurodo para uma dada entrada.
- Funções de transferência de sinais entre neurônios ou funções de limiar (*threshold*): responsáveis pela forma e intensidade de alteração dos valores transmitidos entre os neurônios interconectados. Existem basicamente cinco funções de transferência: linear, de degrau, em rampa, sigmóide e gaussiana, cada uma com sua equação e aplicação particular.
- Funções de aprendizado: responsáveis por realizar a alteração dos valores dos pesos da rede, possibilitando o aprendizado de um determinado padrão.

Numa RNA, a informação flui através de **conexões,** em direção definida, e é captada pelas unidades de processamento. Cada conexão possui um peso que modula o sinal (ou valor) de saída de uma unidade de processamento, e esse sinal é passado para outra unidade. O peso da conexão é o responsável pela *memorização* do padrão e é ajustado durante o processo de aprendizagem. Dessa forma, uma conexão define:

- o fluxo da informação em uma direção;
- a modulação da informação que é passada para outra unidade.

As conexões são excitatórias se possuírem valor numérico positivo; inibitórias se possuírem valores negativos. Não haverá conexão se o peso for igual a zero.

Através das **unidades de processamento**, ou simplesmente neurodos, é realizado todo o processamento da informação. O resultado do processamento consiste no estado geral da rede após alcançar uma condição de equilíbrio (Rumelhart & McCleland, 1986; Caudill & Butler, 1990). Cada unidade capta a informação que lhe foi enviada por meio de um número de sinais de entrada e, com base nessa entrada, produz um único valor de saída. O sinal de saída de uma unidade – uma função das entradas e dos pesos – é enviado para outras unidades como sinais de entrada, através das conexões. Destaca-se que o 'poder' de uma RNA está no modo como suas unidades de processamento são conectadas. Conforme será mostrado, a memória (distribuída) de uma RNA reside, justamente, no processo de conexão das unidades.

4.2.2 Aprendizagem, Treino e Memória

Sabe-se que o sistema nervoso biológico possui a capacidade de aprender a executar uma função. Por analogia, os conexionistas conceberam maneiras de ensinar a RNA até que esta aprendesse a executar uma função desejada. Por isso, entre as características das arquiteturas conexionistas, destaca-se a habilidade para aprender. As Redes Neurais Artificiais *aprendem* a resolver problemas. Elas *não são programadas* para resolvê-los (Caudill *et al.*, 1992). Em

lugar de seguir um programa, esses sistemas são capazes de aprender um comportamento desejado.

A **aprendizagem**, em RNA, consiste em mecanismos específicos para alterar, de forma sistemática, os pesos das conexões entre os neurodos, a fim de melhorar o desempenho de resposta da rede. No entanto, para que uma RNA desenvolva determinada tarefa (tenha um aprendizado) é preciso treiná-la. Assim, tanto a aprendizagem quanto o treino são de fundamental importância para a RNA; sendo, porém, conforme ressaltam Caudill & Butler (1992), aprendizagem e treino, coisas *distintas*.

O **treino** em RNA, por sua vez, é o processo através do qual ocorre a aprendizagem. Durante o treino, os pesos das conexões entre os neurodos vão sendo alterados de modo que cada conjunto de entrada produza o conjunto de saída desejado. Uma rede pode ser treinada com objetivos diferentes.

De acordo com Rumelhart & McCleland (1986), existem dois paradigmas principais nos quais se ajustam os mecanismos de *aprendizagem*:

- *Aprendizagem associativa*: neste paradigma, aprende-se a produzir um padrão de ativação particular em um conjunto de neurodos sempre que um padrão particular diferente ocorra em outro conjunto de neurodos. Para essa aprendizagem, a rede é treinada com a finalidade de reconhecer alguns padrões que são mantidos fixos e poderão ser recordados no futuro.
- *Detecção de Regularidades*: neste paradigma, os padrões não são fornecidos à rede. Aprende-se a descobrir as regularidades inerentes a um conjunto de elementos dados, e criam-se padrões para classificá-lo de acordo com essas

regularidades. Dessa forma, pode-se dizer que este mecanismo de aprendizagem permite a auto-organização na rede, dado que ela cria seus próprios padrões, sem a ajuda de um tutor externo.

Os principais métodos usados para o *treinamento* da rede (Rumelhart & McCleland, 1986; Caudill & Butler, 1990) são:

- *Supervisionado*: neste treinamento, a saída desejada para a rede é pré-estabelecida, o que força a rede a ajustar os pesos a fim de representar essa saída corretamente. Para esse tipo de treino é fornecido à rede um padrão de entrada, o correspondente padrão de saída desejado e um mecanismo corretor de erros usado para alterar os pesos. Nesse treinamento, existe a figura de um professor que confere a resposta da rede para uma dada entrada e a compara com a resposta correta. Se a resposta da rede for a correta, a correção será nula; caso contrário, ela será feita de acordo com um valor proporcional ao produto da entrada e da resposta correta.

- *Por reforço*: neste treinamento, o padrão de saída não é fornecido à rede. Esta é treinada por meio de uma técnica de 'punição' e 'recompensa'. Primeiramente é fornecido à rede um exemplo de entrada. A seguir, a rede produz, aleatoriamente, um resultado desse exemplo. O resultado é avaliado, a rede ajusta seus pesos e o processo se repete. Uma avaliação de valor positivo indica bom desempenho e os pesos são reforçados, enquanto um valor negativo indica desempenho insatisfatório e os pesos diminuídos, a fim de indicar como está o comportamento da rede. Esta procura um conjunto de pesos que impeça avaliações negativas no futuro.

- *Não-Supervisionado* ou *Auto-organização*: este treinamento limita-se a fazer uma representação da distribuição de probabilidade dos padrões de entrada na rede. Vários padrões de entrada são fornecidos à rede, mas ela não recebe nenhuma realimentação sobre suas saídas, nem mesmo qualquer reforço. A rede tem apenas permissão para detectar regularidades e relações entre as partes das entradas. Nesse treinamento não existe a figura do professor e nenhuma instrução específica sobre como aprender uma função. Segundo Caudill & Butler (1990, p.121), "sistemas com esse tipo de treinamento são úteis em aplicações nas quais não se sabe a resposta 'correta' e nas quais o próprio sistema deve decidir quais características serão usadas para classificar os dados de entrada."

Assim, no primeiro paradigma para aprendizagem, aprendizagem associativa, utiliza-se o treinamento supervisionado e por reforço; já no paradigma detecção de regularidades, utiliza-se o treinamento não-supervisionado ou por auto-organização.

A maioria dos algoritmos de treinamento de RNA foi inspirada nas idéias do neurobiologista Donald Hebb (1949). Em seu livro *The Organization of Behavior*, de 1949, ele propõe que o aprendizado ocorre como resultado de modificações corretas das sinapses. Uma variante da Lei de Hebb, e que contém a sua idéia básica, é encontrada em Rumelhart & McClelland (1986, p.53):

> Se uma unidade de processamento, u_i recebe uma entrada de outra unidade de processamento, u_j e, se ambos estão altamente excitados, então o peso w_{ij}, de u_j para u_i, será fortalecido.

Portanto, de acordo com a *Lei de Hebb*, a intensidade da conexão entre duas unidades de processamento aumenta se as duas unidades são excitadas simultaneamente. Essa lei pode ser assim formalizada:

$$\Delta w_{ij} = \eta u_i u_j$$

onde:
Δw_{ij}: variação no peso da conexão entre as unidades i e j
η: constante de proporcionalidade representando a taxa de aprendizado
u_i e u_j: atividades das unidades

As conexões entre as unidades de processamento deixam traços na memória, os quais possuirão maior ou menor intensidade. A lei de Hebb descreve, assim, um mecanismo estruturador da memória distribuída.

No ser humano, aprendizagem, memória e outros processos mentais *nobres* podem ser tratados como diferentes manifestações do mesmo princípio geral que governa a mente (Johnson-Laird, 1988). A ligação memória e aprendizagem é fortalecida pelo fato de a memória ser um processo dinâmico e não algo estático – não se recupera nada extremamente igual ao que foi armazenado. Caudill & Butler (1990, p.118), ao se referirem a aprendizado, afirmam que "pelo fato de a memória, o aprendizado, e o processamento serem virtualmente inseparáveis da estrutura física na arquitetura neural, é inevitável uma similaridade do aprendizado e da memória no sistema biológico e no artificial".

De fato, não existiria aprendizagem sem memória (Caudill & Butler, 1990), visto que a memória é a estrutura que permite a fixação do

unesp

conhecimento; estão, portanto, memória e aprendizagem, estreitamente vinculadas.

De acordo com Caudill & Butler (1990, p.7), "as memórias são armazenadas ou representadas, em uma Rede Neural Artificial, na forma de pesos variáveis de interconexão entre os neurodos."

Em Kovács (1997, p.160), a afirmação é que:

> Os vetores de ponderação [com os pesos das conexões] w_i podem ser considerados como as *memórias* armazenadas na rede dos padrões elementares e é nesse sentido que o conteúdo da memória é armazenado na conectividade do sistema nervoso.

Através do exposto até aqui, pretende-se deixar claro que, no paradigma das Redes Neurais Artificiais, as características mais importante da memória estão associadas às conexões entre os neurodos.

Uma das hipóteses acerca da memória das RNA é que, nelas, cada traço de memória é distribuído sobre conexões diferentes, e cada conexão pode participar de muitos trações de memória diferentes. Segundo Rumelhart & McCleland (1986), a idéia básica da memória distribuída é simples: conceitos diferentes correspondem a padrões de atividades diferentes a respeito do mesmo grupo de neurodos da rede, de forma que um mesmo neurodo é compartilhado por vários padrões. Dessa forma, a memória distribuída é um sistema de memória que armazena informações por todo o sistema, e não apenas em um único endereço.

Por exemplo, ao representar um triângulo em uma RNA, não existirá um neurodo específico, localizado, que represente o triângulo. Ao invés disso, será uma área da memória, com alguns neurodos interligados, que estará representando o conceito de triângulo. A Figura 23 ilustra um simples esquema

unesp

distribuído do conceito de triângulo. Nessa figura, os círculos representam os neurodos, enquanto a conexão de três desses elementos representa genericamente um triângulo.

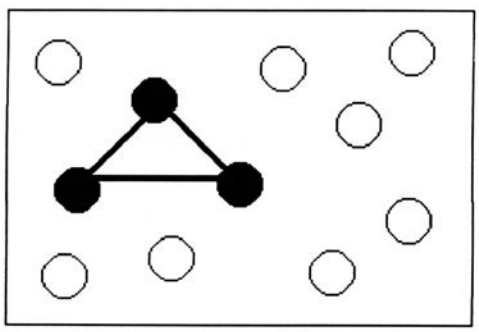


Figura 23 – Representação de um triângulo em uma RNA

O aprendizado e a memória de uma rede geralmente acontece como conseqüência da alteração das conexões já existentes. É, portanto, importante para uma RNA possuir um grande número de ligações, pois isso influencia na habilidade da memória de manter vários padrões memorizados separadamente, ou de ter seletividade suficiente para recuperação das representações armazenadas, gerando um aumento na fidelidade da recuperação. Inclusive, uma estimativa mais segura para a capacidade prática da memória depende do número de padrões independentes que podem ser armazenados nela, sem problemas para sua recuperação (Kohonen, 1987).

Além das características mencionadas, as RNA possuem outras habilidades destacáveis (Wasserman, 1989):

- *Generalização*: capacidade de reproduzir o padrão original mesmo através de ruído e distorção. Depois de treinada, a resposta da rede pode ser não sensitiva a

algumas variações no conjunto de entrada. Devido a essa propriedade, a rede poderá encontrar um conjunto de pesos que mapeie as entradas para as saídas correspondentes, ainda que não lhe sejam apresentadas todas as entradas e saídas possíveis; possibilitando, assim, uma certa flexibilidade do sistema.

- *Abstração*: capacidade de abstrair elementos de um conjunto de entrada. Algumas RNA possuem a habilidade de extrair um conjunto ideal de elementos a partir de um conjunto de entrada imperfeito – ou seja, a possibilidade de aprender a produzir algo novo.

4.3 A Rede de Hopfield

4.3.1 Apresentação da Rede de Hopfield

São vários os modelos de Redes Neurais Artificiais, os quais têm sido estudados e utilizados em diversas áreas do conhecimento. De acordo com Costa (1999, p.31), "dentre os diversos tipos de redes, as do tipo *Associative Memories* (memórias associativas) têm merecido destaque. Sua importância está relacionada à maneira com que o cérebro humano aprende e armazena informações."

Dada a importância das redes desse tipo e a tentativa de se buscar, nesta parte do trabalho, um modelo de RNA capaz de trazer algo relevante para o estudo da memória, será aqui discutido um dos modelos de memória associativa por Redes Neurais Artificiais mais importantes da última década: a Rede de Hopfield.

John Hopfield, nos anos 80, deu uma importante contribuição para a área de Redes Neurais Artificiais ao caracterizar um paradigma da neurocomputação utilizando essas redes como memórias associativas. Sua rede, tal como originalmente proposta, é um simples mecanismo neural artificial recursivo, totalmente conectado – ou seja, uma rede com um certo número de neurodos, sendo cada neurodo conectado com todos os demais, exceto com ele mesmo. Com essa forma de conexão, todos os neurodos estão interconectados e, conforme ressalta Pinker (1998, p.115), "a rede começa a mostrar semelhanças com os processos de pensamento humano de maneira que as redes de conexões esparsas não mostram".

Outra característica de destaque na rede de Hopfield é sua arquitetura, pois permite uma forma de operação diferente em relação aos demais modelos de rede. Pode ser observado na Figura 24 que não existe uma clara distinção entre os neurodos de entrada e os de saída, sendo por isso considerado um modelo auto-associativo com uma única camada.

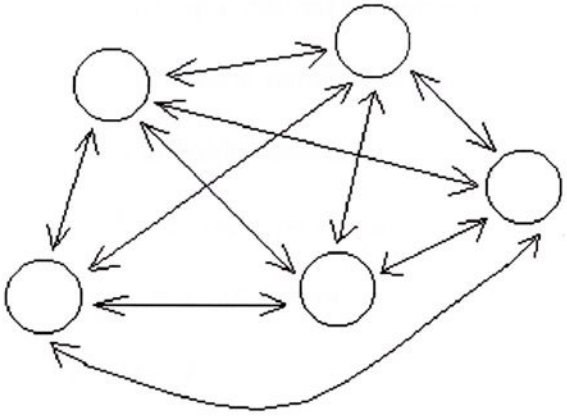


Figura 24: Uma visão da arquitetura da rede de Hopfield

Como nos demais modelos conexionistas, cada conexão entre os neurodos tem um peso. A matriz de pesos dessa rede é simétrica, pois os pesos da conexão de um neurodo a outro são os mesmos em ambas as direções. Além disso, cada neurodo tem ainda um limiar (*threshold*) que se comporta como uma função do tipo degrau, na versão discreta da Rede de Hopfield, e como uma função contínua, por exemplo a tangente hiperbólica, na versão contínua dessa rede.

Os neurodos da rede são apresentados como simplificações dos neurônios biológicos e podem existir em apenas um de dois estados possíveis: o *ativado* ou o *não-ativado*. Estes dois estados podem ser representados, respectivamente, de forma unipolar (0, 1) ou bipolar (-1, +1).

As *entradas* – caracterizadas por um conjunto de valores iniciais (0, 1) ou (-1, +1) – são aplicadas simultaneamente a todos os neurodos. A partir desse momento a rede é 'abandonada', passando a ocorrer um ciclo de sucessão de estados. A todo instante, os neurodos mudam seus estados de ativação, dependendo das entradas que estiverem recebendo dos outros neurodos. O ciclo estará ocorrendo até que a rede convirja para uma solução 'estável', ou seja, até quando os estados dos neurodos pararem de se modificar. A *saída* dessa rede é considerada como sendo o conjunto de valores de todos os neurodos no momento em que a rede atinge um estado 'estável' (Hopfield, 1982; Rumelhart & McCleland, 1986; Wasserman, 1989).

Basicamente, a operação que ocorre nesse sistema de rede é a seguinte: como cada neurodo é conectado a todos os outros, o valor de sua ativação afeta o valor dos demais. Sendo assim, em seu estado inicial, a rede apresenta uma quantidade de valores diferentes, um tentando afetar os outros. Nesse momento, enquanto um neurodo tenta ativar outros, existem neurodos tentando desativá-lo, o que gera uma instabilidade na rede. Durante a sucessão de estados, a rede tende a encontrar, entre todos os seus possíveis valores de ativação, a solução de melhor 'acomodação' desses valores. O estado final é o momento em que a quantidade de neurodos tentando ativar outros neurodos torna-se equivalente à quantidade daqueles que estão tentando desativá-los. Esse equilíbrio entre neurodos sendo ativados e desativados, é o que trará estabilidade à rede.

O grande atrativo da rede de Hopfield é que esse modelo pode armazenar e recuperar "recordações" ou padrões treinados de modo bastante semelhante ao do cérebro humano pois, em ambos, os padrões podem ser recuperados mesmo com a apresentação de informações parciais (Costa, 1999). Além disso, essa rede também pode generalizar, categorizar e realizar correção de erros.

Outra característica importante da rede de Hopfield é sua *recursividade*. As saídas são tomadas como novas entradas: aplica-se a entrada, e a rede produz uma primeira saída que é re-alimentada à rede como uma nova entrada; esta produz uma nova saída, e assim por diante. O processo se repete até que as saídas dos neurodos permaneçam constantes.

Hopfield foi também o responsável pela introdução do conceito de *função energia* ou *função de Lyapunov* nas Redes Neurais Artificiais, usado para determinar a estabilidade dos padrões armazenados na rede. Essa função é definida como:

$$E = -\left(\frac{1}{2}\right)\sum\sum_{i \neq j} w_{ij} o_i o_j$$

onde:
w_{ij}: representa o peso da conexão entre os neurônios i e j da rede
o_i: representa a saída do i-ésimo neurônio
o_j: representa a saída do j-ésimo neurônio

Conforme comenta Costa (1999, p.34),

> A propriedade central da função energia é que ela sempre decresce ou permanece constante quando o sistema evolui no tempo. Assim, para o caso de αi=0 (limiar de ativação do i-ésimo neurônio), poderemos associar a estabilidade dos padrões armazenados na rede com o mínimo da função energia.

Na rede de Hopfield, os padrões são simplesmente armazenados por diminuir sua quantidade de energia. Como armazenar padrões corresponde a fazer a função energia tão pequena quanto possível, para se armazenar um padrão é necessário, portanto, minimizar o valor da *função energia* para esse padrão particular, de forma que ele ocupe o ponto mínimo do local de energia.

Assim, desde que a operação da rede garanta que sua energia seja reduzida a cada atualização, a rede tenderá em direção aos estados que são mínimos locais da função energia. As posições desses mínimos são determinadas pelos pesos w_{ij}.

Escolhendo uma função energia da forma acima descrita, Hopfield estabeleceu as condições para que o processo de funcionamento da rede

seja convergente, ou seja, para que a dinâmica da rede convirja para seus mínimos locais. A dinâmica da rede de Hopfield é, portanto, esta: dada a informação ao modelo, este sofre um ciclo de sucessão de estados até que encontre um local de acomodação, que é o mínimo local.

Com relação à forma de armazenamento na rede de Hopfield, Costa (1999, p.34-5) afirma que o algoritmo de armazenamento e de reconhecimento de padrões pode ser resumido nos seguintes passos:

1. Armazene os padrões usando a equação

$$w_{ij} = \sum (2o_i^s - 1)(2o_j^s - 1) \qquad i \neq j$$

onde o^s é o conjunto de padrões a serem armazenados, s=1,2,....n; ou seja, S={o^1, o^2,..., o^n} é um dado conjunto de treinamento.

2. Enquanto a rede não convergir, faça:

   2.1 - Apresente para a rede o vetor a ser reconhecido

   2.2 - Para cada unidade do vetor aplique a equação

$$o_i(t_{k+1}) = \mu_s \left[\sum_j w_{ij} o_j(t_k) \right]$$

onde μ_s é a função degrau.

(Texto das páginas 34-5 apresentado com algumas modificações)

De acordo com a concepção conexionista, a capacidade de armazenamento de padrões determina a estrutura de memória. Tal capacidade de armazenamento, no caso da rede de Hopfield, é definida como o número máximo de padrões armazenados que podem ser recuperados corretamente. Esse número

máximo impõe um limite sobre o número de padrões que a rede pode armazenar. Segundo Costa (1999, p.34), Hopfield determinou experimentalmente que esse limite fosse dado aproximadamente por $P = 0.15 * n$, onde P é o número de padrões e n o número de neurodos da rede. Assim, pode-se afirmar que o número de padrões a serem efetivamente armazenados na rede é pequeno em comparação ao número de unidades, visto que, usualmente, o número desses padrões é da ordem de 15% do número de unidades. Isso faz a rede de Hopfield original ser relativamente ineficiente como um dispositivo de memória. No entanto, como já ocorre atualmente, pode-se propor muitas extensões e generalizações desse modelo.

4.3.2 A memória associativa

Hopfield foi o primeiro a formular, precisamente, o princípio físico de armazenagem de informação em uma rede dinamicamente estável. Sua pesquisa deu origem a um modelo neurofisiologicamente motivado em memórias associativas, também chamadas *memórias endereçadas pelo conteúdo* ou, ainda, *memórias com conteúdo endereçável*.

Em um computador convencional, como analisado anteriormente, os bits em si não têm significado. Sendo assim, aos bytes feitos com eles são dados endereços arbitrários, que nada têm a ver com seu conteúdo. As posições de memória são, então, acessadas por endereços e, para localizar um padrão armazenado em algum lugar da memória, esse padrão deve ser procurado com base em seu endereço.

A memória associativa consiste em um *método* de armazenamento inspirado no possível mecanismo de funcionamento da memória humana, no qual os itens de dados são acessados com base na análise de seu conteúdo e não com base em sua localização, ou em um endereço físico. O fato de lidar com o conteúdo, e não com endereços, faz da memória associativa uma memória mais flexível.

Na linguagem coloquial, usualmente o termo 'lembrar' consiste em associar algo com algum tipo de sugestão. Por exemplo, ao olhar a gravura de algum local visitado na infância, é comum 'lembrar-se' desse tempo; ou ainda, ao encontrar alguma palavra não nítida, talvez olhando por uma janela embaçada num dia de chuva, seja possível reconhecer essa palavra.

O ato de 'lembrar' pode ser descrito como: dada uma coleção de dados, esta é ordenada de alguma maneira, para, então, ser registrada na memória. A partir daí, esses dados podem ser pensados como padrões armazenados. Em um exemplo de reconhecimento de caracteres de uma palavra, um padrão armazenado é o conjunto de unidades de criação do caractere, ou seja, seus pixels, cuja disposição é determinada por uma versão arqueotipada do caractere. Quando qualquer parte do padrão é apresentada como sugestão, ocorre uma identificação do padrão (reconhecimento do padrão), e todo o padrão é recuperado ou associado à parte. Sendo assim, pode ser mencionado que, na memória associativa, um sistema de processamento de informação 'lembra' quando ela recupera informação armazenada anteriormente, em resposta a dados associados (Pinker,1998).

No caso da memória humana auto-associativa, um item é representado ligando-se as unidades que representam suas propriedades. Por exemplo, a representação de uma banana é feita ligando-se as unidades (neurônios) que contenham suas propriedades: amarela, consistente, mastigável, entre outras. Essas unidades, quando ligadas com pesos fortes, estabelecem padrões nos quais está registrada a informação sobre bananas.

Analogamente, nas Redes Neurais Artificiais, o reconhecimento de um item é feito da seguinte forma: como as unidades que representam as propriedades de um item são ligadas com pesos fortes, as unidades inicialmente ativadas reforçam uma às outras de modo que, após algumas rodadas de ativação, todas as unidades pertencentes ao item estejam na condição de ativada, indicando que o item foi reconhecido.

As conexões entre as unidades de um item permitem que se apenas uma parte do padrão para esse item for apresentada, o resto do padrão será completado automaticamente, lembrando o funcionamento da mente humana na atividade de reconhecimento de padrões que envolvem ruído.

O objetivo básico de uma RNA baseada em memória associativa é, portanto, aprender e armazenar um conjunto de padrões e poder recuperá-los posteriormente. A função primária desse tipo de rede é recuperar um padrão armazenado na memória, em resposta à apresentação de uma entrada cuja versão esteja incompleta ou com ruído, já que a memória tem a capacidade de correção de erros.

Uma outra característica importante na memória associativa é que esse tipo de memória *não* é armazenado em um *neurônio individual*, mas é uma propriedade de *toda a rede*. Esta é outra importante característica dos modelos do tipo memória associativa, e também parece assemelhar-se à forma de armazenamento da memória humana.

4.3.3 Memória: computador versus associativa versus humana

Uma das funções mais importantes do cérebro humano é o estabelecimento e a recuperação da memória. Tendo sido apresentadas a memória do computador digital e a memória associativa de uma Rede Neural Artificial, tentar-se-á fazer uma simples comparação entre três tipos de memória: a memória humana, a memória do computador, e a memória associativa de uma RNA. Essa comparação foi incluída para que se pudesse analisar se, por meio de algum tipo particular de memória, há possibilidade de se dotar a máquina de comportamento inteligente.

No caso humano, sabe-se que a memória acessa informações por meio de seu conteúdo e não através do local em que ela está armazenada no cérebro. Tais características podem ser notadas facilmente através de simples exemplos:

- é possível recuperar todos os aspectos da descrição de uma pessoa, lembrando-se apenas de um ou dois de seus aspectos característicos;
- é possível lembrar-se de uma única coisa através de uma variedade de formas: cheiro, cor, formato, entre outros;

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

- é possível completar uma palavra a partir de fragmentos dela; ou seja, existe a capacidade de se lidar com ruídos ou falhas.

As características acima, no entanto, são muito diferentes em um computador convencional, em que dados específicos são localizados em áreas específicas de sua memória. Em computadores convencionais, os dados (ou parte deles) são designados para uma posição de memória. A RAM usa a posição de memória para a recuperação dos dados armazenados, de modo que o conteúdo real daquilo que se encontra armazenado seja irrelevante para a busca. Como foi bem exposto por Gardner (1995, p.143), " (...) os computadores são deliberadamente construídos para serem cegos para conteúdo". Assim, se for dada apenas uma informação parcial, ou contendo algum erro sobre a localização de onde os dados estão disponíveis na memória, esses dados não poderão ser recuperados por completo. Enquanto isso, a memória humana não precisa de tabelas ou etiquetas de endereços pré definidas para recuperar itens armazenados, e, na maioria dos casos, qualquer característica de um item pode trazê-lo por inteiro à mente humana. Portanto, nos computadores, a função de recuperação dos dados é feita de forma muito limitada.

Tais divergências entre as memórias do computador e a do homem são compartilhadas com Kovács (1997, p.133), que afirma

> Nesta nossa era digital, tendemos a associar memória ao RAM (um registrador de acesso aleatório) ou ao disco rígido dos nossos computadores pessoais, sendo ambos basicamente imensas tabelas com os dados primários, os bytes, diretamente registrados em seus elementos. Memória no sentido neurológico pouco tem a ver com uma tabela endereçável. Aprender um comportamento, como dirigir um automóvel, por exemplo, decorar uma tabela de dados como a tabuada, ou registrar uma seqüência de eventos vividos, é de alguma forma registrado na própria conectividade dos circuitos cerebrais e passam, em um certo sentido, a fazer parte da sua própria estrutura.

unesp

Se por um lado, em um sistema de computador convencional a informação é mantida em um ou vários endereços e sua recuperação depende de se saber o endereço, a rede de Hopfield, sendo uma rede com memória associativa, é mais flexível e, de certa forma, capaz de armazenar certas 'recordações' ou padrões de maneira similar à do cérebro, já que os dados são acessados com base na análise de seu conteúdo. Assim sendo, um padrão completo pode ser obtido mesmo estando a rede apenas com sua informação parcial. Na memória associativa, a especificação de um item elimina a presença de cópias desse item, o que ocorre quando se trata de 'posições de memória'. Além disso, existe um grau de estabilidade no sistema e, mesmo que algumas conexões entre seus neurodos sejam cortadas, a memória recordada não será corrompida – a rede irá buscar pela 'melhor suposição', dada a sua propriedade de generalizar.

Nesse contexto, Costa (1999, p.31) afirma que fenômeno semelhante é observado no cérebro, quando diz:

> Durante uma vida comum, muitos neurônios morrem, mas o cérebro não sofre uma perda catastrófica de recordações individuais. O cérebro é bastante robusto neste aspecto (até a morte, um cérebro comum pode perder 20 por cento de seus neurônios originais).

De forma resumida, o que se pretendeu esclarecer é que a *memória associativa*, tal como vista na rede desenvolvida por Hopfield, difere das memórias de um computador convencional. Essencialmente, enquanto no computador a informação é recuperada a partir da especificação da posição da memória onde os dados estão armazenados, em uma memória associativa a informação é recuperada a partir da análise de seu conteúdo, ao se especificar

unesp

uma parte ou algumas propriedades da informação desejada; e é dessa maneira que a memória humana trabalha. Apesar da semelhança, a memória associativa parece constituir apenas um dos elementos da riquíssima memória humana. Esta envolve muitas outras habilidades, como é o caso do processo de esquecimento.

Tendo sido desenvolvido o estudo proposto nesta dissertação, serão apresentados, a seguir, os resultados obtidos sobre a relevância da memória para a simulação do comportamento inteligente nos modelos mecânicos.

unesp

NOTAS FINAIS

Tomando como ponto de partida a considerável importância que os modelos mecânicos têm como instrumento explicativo para a Ciência Cognitiva, uma vez que esta procura explicar a atividade cognitiva através da criação de modelos, neste trabalho levantamos a dúvida quanto à possibilidade de as máquinas serem instrumentos apropriados para esse tipo de modelagem.

Na busca de esclarecimento para essa dúvida, primeiramente destacamos a afirmação de Turing de que, dotando-se as máquinas de uma grande capacidade de memória, além de métodos de alocação e economia de memória, elas seriam bons modelos do comportamento inteligente.

A seguir, as idéias de Searle e Dreyfus foram introduzidas para que opiniões contrárias às teorias de Turing a respeito das limitações das máquinas, fossem conhecidas. Diante dessa discordância, passamos a levantar argumentos a fim de constatar se a hipótese de Turing é realmente procedente e, assim, verificar se, e até que ponto, as possibilidades dos modelos mecânicos inteligentes dependem da memória da máquina.

Para essa verificação, foi feito o estudo de máquinas utilizadas nas duas linhas de pesquisa representacionistas da Ciência Cognitiva, e também o da noção de memória nessas máquinas, a fim de se obter argumentos convincentes

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

sobre a decisiva relevância da memória na eficiência dos modelos.

Abordando a primeira linha de pesquisa da Ciência Cognitiva, a IA, as máquinas analisadas foram a MEF, a MT e o computador digital.

A inclusão da MEF e da MT foi feita com o intuito de investigar até que ponto a capacidade de computar das máquinas da IA é afetada pela capacidade de memória disponível. Conforme visto, essas duas máquinas abstratas diferem-se, basicamente, pela capacidade de memória, pois a MT é considerada uma MEF sem limite em razão do acréscimo de uma memória ilimitada proporcionado por sua fita. Através de exemplos em que a memória é fator crítico para a computação, foi verificado que tudo o que é feito pela MEF pode ser feito pela MT, mas o contrário não é possível. Assim, pudemos constatar que a memória constitui um elemento decisivo na capacidade de computação de uma máquina.

Uma das novidades introduzidas através desta dissertação encontra-se na seção 2.3. Trata-se do argumento usado para justificar o estudo das máquinas com capacidade de memória infinita, as quais nem poderiam ser construídas pelo simples fato de o nosso universo temporal ser finito. Na justificativa apresentada, mostramos que a *obrigação lógica* é o que faz o infinito ser incluído na teoria dos autômatos.

Também foi possível constatar que a caracterização da memória nos computadores e nas RNAs ilustra bem a diferença existente entre as duas linhas de pesquisa representacionistas da Ciência Cognitiva, referenciadas neste trabalho.

Na primeira linha de pesquisa, conforme indicamos no Cap. 3, a máquina utilizada para implementar os modelos é o computador digital, e nele, a memória apresenta um *caráter estático*. Resumidamente, argumentamos que os dados, nessa máquina, são armazenados em locais específicos na memória, chamados endereços de memória, e são recuperados através do número do endereço, sem que o conteúdo real desses dados sejam utilizados no processo. A menos que ocorra alguma danificação na máquina, cada detalhe é reproduzido exatamente como foi gravado. Não há programa que a capacite a abstrair ou correlacionar idéias. Simplificando, podemos dizer que, por causa de seu caráter estático de armazenamento, a memória do computador funciona somente como um *depósito de dados*.

No exame da segunda linha de pesquisa da Ciência Cognitiva, vimos que a memória da RNA não possui um lugar específico. Ela é distribuída, é associada às alterações nos pesos das conexões, e emerge continuamente a cada momento em que o sistema ocorre, o que permite que ela seja reconhecida como possuidora de um *caráter dinâmico*.

Explorada a noção de memória nas máquinas utilizadas em cada linha de pesquisa representacionista da Ciência Cognitiva, o desafio passou a ser o de verificar se a memória – estática ou dinâmica - dotaria a máquina da capacidade de pensar e, assim, procuramos examinar até que ponto Turing está correto em acreditar que a capacidade suficiente de memória resolve a polêmica da inteligência nas máquinas.

unesp

Com fundamento nos estudos acima resumidos, completamos o exame da memória nos modelos mecânicos, entendendo que, enquanto limitada a *funções computáveis* (todas as funções para as quais existe um algoritmo que as compute) a memória realmente constitui um elemento decisivo nos modelos mecânicos da inteligência. Sendo assim, a *relevância da memória nas máquinas* está em permitir a obtenção de um domínio razoável dentro do universo computável. Esse entendimento considera que a memória resolve o problema do *computável*, pois ela garante que toda função computável pode ser simulada mecanicamente. Não resolve, porém, o problema do *não computável*, ou seja, a memória não garante que as funções para as quais não se tem algoritmo possam ser simuladas em um máquina.

Para chegar a essa conclusão, analisamos algumas funções – a de adição, a de multiplicação, a de reconhecimento de linguagem, a de decidir se uma computação é finita, entre outras – a fim de verificar a execução das mesmas em uma MT. Nessa análise, utilizando o *problema da multiplicação*, verificamos que a aquisição de memória implica na possibilidade de execução de uma função computável. Se *apenas* o elemento memória constituísse obstáculo para a máquina desenvolver as atividades cognitivas humanas, certamente não haveria dificuldades intransponíveis para a MT. No entanto, não existe solução para o *problema da parada* em uma MT infinita. Analogamente, podemos tomar qualquer exemplo de função não computável, tal como saber se uma sentença da aritmética é verdadeira ou falsa, que a MT será incapaz de solucioná-la.

Diante disso, foi possível constatarmos que, desde que todos os modelos baseados em regras são, por definição, computáveis e, portanto, desqualificados para as funções não computáveis, haverá funções que, independente da capacidade de memória, não poderão ser realizadas pelas máquinas programadas com regras explícitas. Conseqüentemente, modelos implementados nessas máquinas envolverão, mesmo com memória ilimitada à disposição, somente funções computáveis.

Quanto às RNA não possuidoras de regras explícitas, acreditamos que, dentro do limite das funções computáveis, a conclusão a respeito das possibilidades *oferecidas* pela memória seja equivalente à dos modelos baseados em regras. Isso por considerarmos que, enquanto redutíveis a funções computáveis e implementáveis em máquinas do tipo MT, os modelos mecânicos da IA e do Conexionismo são equivalentes.

Nos últimos anos, os pesquisadores das RNA têm procurado avançar na possibilidade de implementar funções no domínio do não computável, contudo as evidências experimentais constatadas até o momento não são suficientes para decidir definitivamente essa questão. Como exemplo, temos experiências feitas com o objetivo de estudar manifestações mentais, tais como a consciência e o livre arbítrio, utilizando-se da teoria quântica, o que constitui uma das possibilidades pensadas por pesquisadores para a implementação de funções não computáveis em RNA.

Se o aspecto tecnológico envolvendo a capacidade de memória da máquina é aqui entendido como um fator fundamental, mas não excludente

para que se obtenha progressos nos modelos mecânicos, é porque, sob nosso ponto de vista, ainda há outros aspectos a serem considerados: filosóficos, sociais e até psicológicos. Estes seriam de grande valia para que o risco de uma simplificação grosseira no tratamento dos processos mentais não ocorra.

Quanto ao *aspecto social*, por exemplo, acreditamos que a dificuldade da máquina em lidar com esse aspecto seja um incômodo para a modelagem das atividades cognitivas. Concordamos com a argumentação central de Vygotsky (1994) na qual ele afirma que as funções mentais superiores estão ligadas à interação social. Seguindo as idéias desse autor, entendemos que é essencial, para uma boa aprendizagem, que o contexto social em que ocorre a cognição seja incorporado ao modelo, pois a falta de interação com o ambiente não acontece sem perdas. Enquanto o aspecto social não for considerado nos modelos cognitivos, estará faltando neles um componente essencial do comportamento humano inteligente.

Um outro fator que o referencial do modelo humano possui e que não sabemos ao certo como funciona, mas que parece oferecer vantagem em relação à memória das máquinas, refere-se ao processo de *esquecimento*. Enquanto os neurônios biológicos estão propensos a falhas e têm a capacidade de esquecer como uma característica positiva, foi verificado que é importante que a memória de um computador opere perfeitamente a fim de que a máquina execute de forma correta suas atividades. No computador, idealmente é desejável que os *bits* de informações armazenados não sejam perdidos. Supomos, no entanto, que a criação de programas não sensíveis à falha de algumas informações aumentaria a

unesp

capacidade de realização de tarefas dos computadores. Esta idéia é baseada em uma estatística feita por Rich & Knight (1993), que demonstram ser muito mais fácil construir um circuito integrado com *bilhões* de componentes funcionando *corretamente* , ou seja, dentro do aceitável, mas com possibilidade de erro, do que com *milhões* de componentes funcionando *perfeitamente*, ou seja, sem nenhuma possibilidade de erro.

O esquecimento também é algo desejado em uma RNA, pois sua capacidade para lidar com universais, para generalizar em grande forma, é afetada pelo fato de ela não possuir o esquecimento seletivo. Além disso, se as RNA pretendem ser uma boa simulação do modelo humano, é necessário modelar os aspectos fundamentais do comportamento biológico, os quais envolvem, inclusive, o esquecimento. À medida que o ser biológico reage às mudanças que ocorrem no ambiente, se o ambiente sofre alterações, esse ser também sofrerá. Dessa forma, as RNA, na tentativa de imitar o modelo natural, biológico, devem, igualmente, ser capazes de esquecer seletivamente. Se isso não acontecer, será difícil a adaptação das redes às mudanças circunstanciais do ambiente.

Sugerimos, diante de nosso posicionamento em relação a sistemas artificiais que envolvem modelos mecânicos, especificamente ao que se refere à memória, que a hipótese provável utilizada por Turing para ligar memória à inteligência seja revisada.

Talvez, por não ser um termo totalmente esclarecido, o conceito de inteligência utilizado por Turing esteja equacionado com o acúmulo de

informação e, dessa forma, quanto maior o conhecimento, maior a inteligência de um indivíduo.

As pesquisas de Vygotsky (1994) relacionadas com memória sugerem que, somente em crianças muito pequenas o ato de pensar é, antes de mais nada, determinado por sua memória acumulativa. Ele diz que, na criança, o conteúdo do ato de pensar não é determinado tanto pela estrutura lógica do conceito em si quanto pelas suas lembranças concretas. As representações do mundo, nessa fase, ainda não possuem o caráter de abstração, mas baseiam-se na lembrança de exemplos concretos. Ele declara que essa dependência da memória é característica definitiva apenas dos primeiros estágios do desenvolvimento cognitivo. Suas pesquisas mostram que, já no final da infância, as relações envolvendo a memória invertem a direção, e assim, nos adolescentes, o processo de lembrança consiste em estabelecer e encontrar relações lógicas entre elementos.

Acreditamos que considerar somente a memória como fator para desenvolver o aprendizado, a inteligência e o conhecimento, seria uma grande simplificação da atividade cognitiva, pois consideramos que eles envolvem uma série de fatores interligados. É difícil 'limpar' o processo cognitivo e considerar apenas um dos fatores relevantes, no caso específico, a memória, sem atentar para os demais.

Sendo assim, é possível que a versão de Turing represente um conceito fragmentado de inteligência. Mais uma vez declaramos nosso entendimento de que é óbvio que a memória é um fator necessário para uma boa

unesp

definição de inteligência, contudo não suficiente, ou excludente. Por exemplo, não seria possível decidir corretamente o movimento de uma peça de xadrez se não houvesse o armazenamento das regras do jogo. No entanto, apenas o acúmulo de conhecimento – sem o envolvimento de elementos sociais, das sutilezas do universo emocional, das intuições, da criatividade, da vontade, entre outros – não garante a manifestação da inteligência. Ficam faltando fatores que envolvam a complexidade do ser humano. Podemos exemplificar essa idéia citando os computadores, que são excelentes máquinas de lembrar. Na verdade, cremos que esta é uma das coisas que eles melhor fazem; ainda não demonstram, porém, comportamento social inteligente.

Em resumo, no presente trabalho, procuramos investigar a relevância da memória nos modelos mecânicos da inteligência. Nossa pretensão é não deixar dúvida de que a quantidade limitada de memória impossibilita algumas máquinas de realizarem possíveis computações, certamente prejudicando seu desempenho e os modelos nelas implementados. Também, não nos parece ser possível afirmar que, a partir da memória, seja concebida e construída uma máquina genuinamente capaz de aprender, e que possa tornar legítimos os modelos do comportamento humano inteligente a ponto de aceitá-los como sendo a realidade das atividades cognitivas. Apesar de nossa descrença no sucesso dos modelos mecânicos quanto ao que depende simplesmente da memória, temos por certo que, da procura do aperfeiçoamento desses modelos, ninguém esteja pensando em desistir!

unesp

BIBLIOGRAFIA

ALCALDE, E., GARCIA, M., PEÑUELAS, S. *Informática básica*. São Paulo: Makron Books, 1991.

ALEKSANDER, I. *Neural computing architetures*: the design of brain like machines. s.l.: North Oxford Academic Publisher, 1989.

ANDLER, D. (*Org.*) *Introdução às ciências cognitivas*. São Leopoldo: Ed. Unisinos, 1988.

BITTENCOURT, G. *Inteligência artificial*: ferramentas e teorias. Campinas: Instituto de Computação, UNICAMP, 1996.

BORKO, H. *Computer applications in the behavioral sciences*. London: Prentice Hall Internacional, 1962.

BURSEN, H. A. *Dismantling the memory machine*: a philosophical investigation of machine theories of memory. Dordrecht: D.Reidel Publishing Company, 1978.

BUTTON, G. *et al.*. *Computadores, mentes e conduta*. São Paulo: Ed. Unesp, 1998.

CAUDILL, M., BUTLER, C. *Naturally intelligent systems*. Massachussets: MIT Press, 1990.

_____ *Understanding neural networks*: computer explorations. Massachussets: MIT Press, 1992. Volume 1: basic networks.

COPELAND, B. J. *Artificial intelligence*: a philosophical introduction. s.l.: Blackwell Cambridge, 1995.

CORREA, C. P. Q. A nova ciência da mente revisitada. *Cadernos de trabalho do Centro de Ciências do Homem*. UENF, Fundação Estadual Norte Fluminense, n. 4, p.49-68, 1999.

COSTA, E. J. X. A rede de Hopfield. In: CABRAL JÚNIOR, E. F. *Redes neurais artificiais*: um curso teórico e prático para engenheiros e cientistas. São Paulo: EDUSP, 1999. cap. 2, p.31-42.

CUTLAND, N. *Computability*: na introduction to recursive functions. Cambridge: University Press, 1992.

DEL MASSO, M.C.S. *Orientação para o trabalho:* proposta curricular para deficientes mentais. São Paulo, 2000. 190p. Tese (Doutorado em Educação) – Faculdade de Educação – USP.

DICIONÁRIO de informática Microsoft Press. Trad. da 3.ed. americana. Rio de Janeiro: Campus, 1998.

DREYFUS, H. *What computers can't do*: a critique of artificial reason. London: The MIT Press, 1972.

_____ *What computers still can't do:* a critique of artificial reason. London: The MIT Press, 1992.

DROBOT, V. *Formal languages and automata theory*. Computer Science Press, 1989.

DUPUY, J.-P. *Nas origens das ciências cognitivas*. São Paulo: Ed. Unesp, 1996.

FERREIRA, A. B. H. *Dicionário Aurélio básico da língua portuguesa.* Rio de Janeiro: Nova Fronteira, 1975. p.902.

GARDNER, H. *A nova ciência da mente*. São Paulo: Editora da Universidade de São Paulo, 1995.

GENNARI, M. C. *Minidicionário de informática*. São Paulo: Saraiva, 1999.

GONZALES, M. E .Q. Auto-organização e perspectivismo: algum acréscimo à Ciência Cognitiva. In: *Encontro com as ciências cognitivas.* Marília: Marília publicações: Faculdade de Filosofia e Ciências, 1998. v.2, 3-13p.

GONZALES, M. E .Q. et al. *Encontro com as ciências cognitivas.* 2.ed. Marília publicações: Faculdade de Filosofia e Ciências, 1997.

HEBB, D. O. *Organization of Behavior*: a neuropsychological theory. NY: s.n., 1949.

HOPFIELD, J. *Neural networks and physical systems with emergent colective computational abilities*. USA, v.79, p.2554-2558, 1982.

JOHNSON-LAIRD, P.N. *The computer and the mind*: an introduction to cognitive science. Great Britain: Fontana Press, 1988.

KHALFA, J. (*Org.*). *A natureza da inteligência:* uma visão interdisciplinar. São Paulo: Ed. Unesp, 1996.

KOHONEN, T. *Self-organization and associative memory*. 2.ed. Berlin: Heidelberg, 1987. 284p.

KOVÁCS, Z. L. *Redes neurais artificiais*: fundamentos e aplicações. 2.ed. São Paulo: Ed. Acadêmica, 1996.

_____ *O cérebro e sua mente*: uma introdução à neurociência computacional. São Paulo: Ed. Acadêmica, 1997.

KURZWEIL, R. *The age of spiritual machines*: when computers exceed human intelligence. New York: Viking, 1999.

LEVINE, D., LEVEN, S. *Motivation, emotion and goal direction in neural networks*. Lawrence Erlbaum Associates, 1992.

LÉVY, P. *A máquina universo*: criação, cognição e cultura informática. Porto Alegre: ArtMed, 1998.

LURIA, A.R. *The Mind of a Mnemonist*: a little book about a vast memory. New York: Harvard Univ Pr., 1988.

MICROSOFT PRESS. *Dicionário de Informática*. Rio de Janeiro: Campus, 1998.

MINSKY, M. L. *Computation*: finite and infinite machines. Prentice-Hall, 1967.

NORTON, P. *Introdução à informática*. São Paulo: Makron Books, 1996.

PINKER, S. *Como a mente funciona*. São Paulo: Companhia das Letras, 1998.

PROUDFOOT, D., COPELAND, B. J. Alan Turing's forgotten ideas in computer science. *Scientific American*. p.77-81, Apr. 1999.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

RICH, E., KNIGHT, K. *Inteligência artificial*. 2.ed. São Paulo: Makron Books, 1994.

ROGERS, H. *Theory of recursive functions and effective computability*. s.l.: McGraw-Hill, 1947.

RUMELHART, D. E., McCLELLAND, J. L. *Parallel distributed processing*: explorations in the microstructure of cognition. Volume 1: foundations. The MIT Press, 1986.

SCHANK, R., BIRNBAUM, L. Aumentando a inteligência. In: KHALFA (*Org.*) *A natureza da inteligência:* uma visão interdisciplinar. São Paulo: Ed. Unesp, 1996.

SEARLE, J. R. *A redescoberta da mente*. São Paulo: Martins Fontes, 1997.

_____ *O mistério da consciência*. São Paulo: Paz e Terra, 1998.

SOARES, A. B. *et al.*. Sistemas de memória. *Cadernos de trabalho do Centro de Ciências do Homem*. UENF, n.1, p.16-27, 1997.

SOARES, A. *O que são ciências cognitivas*. São Paulo: Brasiliense, 1993.

TANENBAUM, A. S. *Sistemas operacionais modernos*. Rio de Janeiro: Prentice Hall do Brasil, 1995.

TEIXEIRA, J.F. *Mentes e máquinas:* uma introdução à ciência cognitiva. Porto Alegre: Artes Médicas, 1998.

THAGARD, P. *Mente*: introdução à ciência cognitiva. Porto Alegre: ArtMed, 1998.

TORRES, G. *Hardware*. 2.ed. Rio de Janeiro: Axcel Books, 1998.

TREFIL, J. *Somos diferentes?*: um cientista explora a inteligência ímpar da mente humana. Rio de Janeiro: Rocco, 1999.

TURING, A. On computable numbers, with an application to the Entscheidungsproblem. 1936. In: DAVIS, M. *The undecidable*: basic papers on undecidable propositions, unsolvable problems and computable functions. New York: Raven Press Books, 1965.

_____ Computadores e Inteligência (art.). s.l.:s.n., 1950.

VON ECKARDT, B. *What is cognitive science?*. 5.ed. Londres: A Bradford Book The MIT Press, 1996.

VYGOTSKY, L.S. *A formação social da mente*: o desenvolvimento dos processos psicológicos superiores. São Paulo: Martins Fontes, 1994.

WASSERMAN, P. D. *Neural computing*: theory and practice. New York: Van Nostrand Reinhold, 1989.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

BUENO, L. P. P. *The study of memory in mechanical models of intelligence*. Marília, 2000. 187p. Dissertação (Mestrado em Filosofia – Área de Concentração: Ciência Cognitiva e Filosofia da Mente) – Faculdade de Filosofia e Ciências, Campus de Marília, Universidade Estadual Paulista "Júlio de Mesquita Filho".

ABSTRACT

The creation of mechanical models is the methodological strategy used by Cognitive Science to simulate (and so explain) intelligent behavior. However, it doesn't constitute unanimity amongst cognitive scientists, because there are divergent opinions about the possibility of machines being appropriated to this simulation. In this context, our purpose is to verify the importance of the memory of machines so that some mental processes can be adequately simulated in mechanical models. In order to get subsidies to this elucidation, it is made a study about the machines envolved in the Cognitive Science representacionalist lines of research, and more specifically, the notion of memory is analised in these machines. Finally, we expose our point of view about the possible amplification of the eficiency of intelligent models in the presence of appropriated memory.

Keywords: Memory; mechanical models; Cognitive Science.