

SOBRE AS MÁQUINAS DE TURING ON TURING MACHINES

Rodrigo Canal*

Resumo

O presente trabalho versa sobre o tema da natureza e operação das Máquinas de Turing. O trabalho se insere na literatura secundária sobre filosofia da ciência cognitiva. Procuramos oferecer esclarecimentos a respeito de algumas perguntas básicas, tais como: O que é uma Máquina de Turing? Como funciona? Qual a natureza de, e como é constituída, uma máquina de Turing? O que essa máquina possui? Quem foi seu inventor, quem a projetou? Para que finalidade ela foi projetada? E por que foi construída? Não pretendemos fornecer respostas originais a estas perguntas (citadas acima), nem mesmo oferecer soluções originais aos problemas lógicos e matemáticos que o empreendimento intelectual construído por Alan Turing procurou solucionar com a concepção de Máquinas de Turing. Nem mesmo fornecemos respostas originais e que possam modificar a compreensão que já se tem (profunda e riquíssima) sobre as perguntas anteriormente formuladas a respeito das máquinas de Turing. A abordagem do presente artigo tenta ser, no máximo, informal e intuitiva, fundamentada na pesquisa temático-bibliográfica, que inclui o uso de dicionários e enciclopédias, impressos e virtuais. O trabalho serve como meio, ao mesmo tempo, de introduzir e aproximar o leitor da complexidade dos problemas e temas em que este trabalho está inserido, isto é, sobre a natureza dos processos computacionais e a possibilidade da mente humana em efetuar os processos ditos computacionais.

Palavras-chave: Máquina de Turing. Fisicalismo. Mecanicismo. Processos computacionais. Alan Turing.

Abstract

The present work deals with the theme of nature and operation of Turing machines. The work is part of the secondary literature on the philosophy of cognitive science. We seek to provide clarification on some basic questions such as: What is a Turing machine? How it works? What is the nature of, and how it is formed, a Turing machine? What this machine has? Who was its inventor, who designed it? For what purpose it was designed? And why was it built? We do not intend to provide unique answers to these questions (noted above), even offering original solutions to problems that the logical and mathematical intellectual enterprise built by Alan Turing tried to solve with the concept of Turing machines. Not even original and provide answers that may change the understanding that we already have (deep and rich) on the questions raised earlier about the Turing machines. The approach of this article tries to be, at most, informal and intuitive, based on research-themed literature, which includes the use of dictionaries and encyclopedias, printed and virtual. The work serves as a means at the same time, to enter and bring the reader the complexity of issues and themes in this work is inserted, that is, about the nature of computational processes and the possibility of the human mind to make the so-called computational processes.

* Professor Assistente I da Universidade Federal do Oeste do Pará, email: prof.rodrihocanal@gmail.com.

Keywords: Turing Machine. Physicalism. Mechaniscism computational processes. Alan Turing.

Introdução

Exploraremos neste trabalho o conceito de *Máquinas de Turing*. Um dos objetivos a serem cumpridos por nós será o de fornecer elementos teóricos básicos para a compreensão *do que seja* e de *como funciona* uma Máquina de Turing. Para isso, faremos uma descrição de sua natureza e de seu funcionamento buscando caracterizar quais os aspectos lógico-conceituais dos modelos das Máquinas de Turing. É de se destacar que o próprio modelo de uma *Máquina de Turing*, tal como foi formulado por Alan Mathison Turing¹, é de cunho puramente abstrato.

Alertamos de antemão o leitor que não almejamos nos alargar em outras tantas questões em que está inserida a concepção de Máquina de Turing. A noção de Máquina de Turing foi (e ainda é) parte das pesquisas da Inteligência Artificial (I.A. daqui para frente) sobre a natureza e funcionamento da inteligência humana na atividade de resolver problemas. Com sua concepção de Máquinas de Turing, Turing foi reconhecido como um dos principais contribuidores para o surgimento e desenvolvimento da I.A., na verdade foi um notável precursor desta. Além do mais, a I.A. é apenas um pequeno ramo da gigante *ciência da mente* que é a Ciência Cognitiva² hoje. É de se destacar

¹ Alan Mathison Turing (1912-1954) nasceu em 23 de junho de 1912 em Londres, filho de um oficial britânico, Julius Mathison e Ethel Sara Turing. Seu interesse pela ciência começou cedo, logo que aprendeu a ler e escrever, se distraíndo na faturação de números de hinos religiosos e desenhando bicicletas anfíbias e já em 1928 começou a estudar a Teoria da Relatividade. Com a idade de 24 anos (1936), Alan Turing consagrou-se como um dos maiores matemáticos do seu tempo, quando fez antever aos seus colegas que era possível executar operações computacionais sobre a teoria dos números por meio de uma máquina que tivesse embutidas as regras de um sistema formal. Embora propriamente não existisse ainda materialmente tal máquina, Turing enfatizou desde o início que tais mecanismos poderiam ser construídos pelo homem. Sua descoberta abriu uma nova perspectiva no esforço de formalizar a matemática, e, ao mesmo tempo, marcou fortemente a história da computação. Na II Guerra Mundial, Turing foi trabalhar no Departamento de Comunicações da Gran Bretanha (*Government Code and Cypher School*) em *Buckinghamshire*, com o intuito de quebrar o código das comunicações alemãs. Turing e seus colegas cientistas trabalharam num sistema que foi chamado de Colossus, considerado precursor dos computadores digitais. Durante a guerra, Turing foi enviado aos *EUA* a fim de estabelecer códigos seguros para comunicações transatlânticas entre os aliados. Terminada a guerra, Turing se juntou ao *National Physical Laboratory* para desenvolver um computador totalmente inglês que seria chamado de *ACE (Automatic Computing Engine)*. Decepcionado com a demora da construção desse computador, mudou-se para Manchester e no dia 7 de junho de 1954, suicidou-se durante uma crise de depressão, comendo uma maçã envenenada com cianureto de potássio (POZZA, PENEDO, 2002).

² A expressão *Ciência cognitiva* designa o estudo interdisciplinar da natureza e das operações dos fenômenos mentais: o termo *cognitivo* em *ciência cognitiva* é usado para representar qualquer tipo de operação mental ou estrutura mental (presentes no homem e/ou qualquer outro tipo de animal). A ciência cognitiva investiga os processos de cognição, o que são e como eles operam/funcionam. Isso inclui a pesquisa sobre como a informação é processada (através das capacidades/faculdades mentais tais como a

também que todas estas discussões teóricas fazem parte de um conjunto, complexo e difícil, de problemas que a própria filosofia hoje (a chamada Filosofia da Mente³), aliada a vários tipos de Ciência, procuram dar respostas a respeito da natureza e funcionamento da mente humana. Por conseguinte, apesar de tentar restringir nossos objetivos e pretensões, procuramos, na medida do possível, ter a consciência, bem como dar a impressão neste trabalho, de que todo o aparato conceitual que será exposto aqui não limita-se somente a este campo lógico/formal: questões empírico-conceituais complexas sobre a mente humana o envolvem.

Como falamos, certamente, sem a conceituação e formalização da noção de uma **Máquina de Turing**, não haveria mesmo a possibilidade (futura, pois que a ideia apenas foi abstratamente elaborada por Turing) da construção de máquinas computacionais, tais como as conhecemos (e temos) hoje. Turing previu que seria possível construir máquinas matematicamente mecanicistas para executar determinadas tarefas humanas. Mas sua ideia, digamos metaforicamente, só ficou no papel. Foram seus predecessores que aplicaram sua ideia abstrata de uma máquina computacional, embora tenha sido a mais influente noção de computabilidade de seu tempo.

Para finalizar esta introdução, e antes de adentrarmos às noções importantes (e elementares) para entendermos a concepção de Máquina de Turing, temos que fazer referência (histórica por sinal) a dois artigos publicados pelo lógico-matemático Alan Turing, porque foi em (**On computable numbers, with an application to the**

percepção, linguagem, memória, raciocínio, emoção, etc.), representada e transformada em comportamento. A ciência cognitiva consiste de múltiplas disciplinas de pesquisa, incluindo a *psicologia, inteligência artificial, filosofia, neurociência, lingüística, antropologia, sociologia e educação*. Segundo algumas fontes, o termo foi cunhado por *Hugh Christopher Longuet-Higgins* (1923-2004) ao ter usado a expressão “ciência cognitiva” no relatório *Lighthill* (na verdade a expressão técnica, em inglês, seria *Lighthill report*), publicado em 1973 e compilado pela *Lighthill for the British Science Research Council*: o *Lighthill report* foi resultado da avaliação da pesquisa acadêmica desse conselho no campo da Inteligência Artificial (ver verbete *ciência cognitiva* e *cognitive science*, respectivamente em português e inglês da Wikipedia, disponíveis em <http://pt.wikipedia.org/wiki/Ci%C3%Aancia_cognitiva (português) e <http://en.wikipedia.org/wiki/Cognitive_science> (inglês).

³ Como o próprio nome sugere, o estudo filosófico da mente ou a “disciplina que discute os problemas relacionados com a mente e os fenômenos mentais — a sua existência, a sua natureza, a sua relação com o mundo. Estes problemas possuem estreitas ligações com problemas da *metafísica*, da *filosofia da linguagem*, da *epistemologia* e da *filosofia da ciência*, mas não devem ser confundidos com os problemas empíricos da Psicologia” (**Dicionário Escolar de Filosofia**, acessado 25 de agosto de 2011, encontrado no site: <http://www.defnarede.com/f.html>). A “Filosofia da Mente” estuda questões conceituais metafísicas, epistemológicas e metodológicas sobre a natureza e funcionamento de quaisquer tipos de fenômenos mentais; o aclamado e famigerado problema mente-corpo, por exemplo, é um dos vários problemas investigados neste campo de estudos filosóficos. Em filosofia da mente se explora noções importantes sobre a vida mental em geral, como a noção de consciência, intencionalidade, subjetividade, causalidade mental, identidade pessoal, inconsciente, etc.

Entscheidungsproblem⁴ [1936] e **Computing Machinery and Intelligence**⁵ [1950]) que foram expostos pela primeira vez a ideia de uma Máquina de Turing: uma máquina simples capaz de realizar qualquer cálculo descritível por um procedimento sistemático geral, finito, sem ambiguidades⁶.

1 O problema da decisão (**Entscheidungsproblem**)⁷

Como a própria formulação (do título acima) sugere, problemas de decisões podem ser problemas que são decidíveis ou indecidíveis. Problemas desse tipo (de decisão) preocupam a mente dos matemáticos há muito tempo. Os problemas de decisões podem ser formulados na forma das seguintes questões: “dada uma classe de proposições (em geral infinita) envolvendo objetos matemáticos conhecidos, existirá algum algoritmo que permita saber, para qualquer proposição da classe e no fim de certo número de passos, se a proposição é verdadeira ou falsa?”. [...] “Existirá algum algoritmo que permita saber, para dado número arbitrário, se é primo ou não?” Em termos mais precisos (próprio da lógica-matemática) e genéricos⁸, pode-se dizer que boa parte dos problemas de decisão podem ser reduzidos a questões envolvendo números naturais: “dado um predicado n -ário P nos naturais, existirá um algoritmo que permita

⁴ TURING, Alan. **On computable numbers, with an application to the Entscheidungsproblem**. Proceedings of the London Mathematical Society, 1937. V. 42.

⁵ TURING, Alan. **Computing Machinery and Intelligence**. *Mind* **LIX** (236), p. 433–460, doi:10.1093/mind/LIX.236.433, ISSN 0026-4423, <<http://loebner.net/Prizef/TuringArticle.html>>, retrieved: 2008-08-18.

⁶ Nesses dois artigos (**On computable numbers, with an application to the Entscheidungsproblem** [1936] e **Computing Machinery and Intelligence** [1950]) Turing expôs concepções que hoje são denominadas de “mecanicismo lógico”, em que tentou esclarecer que uma máquina devidamente programada poderia apresentar inteligência o suficiente para confundir um interrogador bem desconfiado quanto às semelhanças entre algumas características humanas e “maquinais”.

⁷ Segundo a **Enciclopédia de termos logico-filosóficos** (BRANQUINHO; MURCHO; GOMES, 2006, p. 515-519): “[...] Existem problemas de decisão largamente conhecidos: 1) O décimo problema de Hilbert (de uma famosa lista de problemas apresentada por Hilbert em 1900): decidir se uma equação polinomial com coeficientes inteiros $P(x_1, \dots, x_n) = 0$ tem soluções inteiras. Após longa luta com esse problema, que envolveu os nomes de M. Davis (1953), H. Putnam (1953), R. M. Robinson (1952) e I. Matijasevic (1970), o problema só foi resolvido em 1970, tendo sido mostrado que é insolúvel. O famoso teorema que afirma isso é por vezes designado por “teorema MRDP” em memória daqueles matemáticos. 2) O problema da palavra para sistemas semi-Thue e Thue. Qualquer desses problemas é insolúvel. 3) O problema de decisão para um dado sistema formal consiste em saber se dada fórmula é ou não um teorema (p. ex., esse problema é solúvel para o cálculo das proposições, mas não para a aritmética de primeira ordem). 4) O PROBLEMA DA PARADA [...]”

⁸ Para que se possa entender adequadamente a formulação, genérica e formal, que dissemos ser comum a boa parte dos problemas da decisão, citamos uma instrução da **Enciclopédia de termos logico-filosóficos** (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519): “[...] Aqui a classe de proposições em jogo é formada pelas proposições da forma $p(n)$, onde $p(x)$ é uma fórmula que exprime que X é primo. Cada vez que se dá um valor a x obtém-se uma proposição concreta, mas o que pretendemos saber é se somos capazes de resolver a questão qualquer que seja x ”.

decidir para cada ênupla de números naturais $\langle X_1 \dots X_n \rangle$ se $P(X_1 \dots X_n)$ é verdadeira ou falsa?” (BRANQUINHO; MURCHO; GOMES, 2006, p. 515-519).

Fornecer uma resposta afirmativa (ou solução positiva) a um problema de decisão, deve consistir em construir e fornecer um algoritmo para resolver o problema – da decisão. Por isso é que ao fornecermos uma resposta afirmativa, ou uma solução positiva, a um problema dessa natureza, podemos dizer que o problema é *solúvel* ou *decidível*. Enquanto que uma resposta negativa, ou uma solução negativa, consiste em mostrar que nenhum algoritmo existe que se resolva esse problema – da decisão: e por isso é que, nesse caso, o problema é *insolúvel* ou *indecidível*.

Turing procurou esclarecer e responder a um problema da decisão dessa natureza, qual seja, o problema proposto por Hilbert⁹, sobre a possibilidade de existir um *algoritmo*¹⁰ - procedimento sistemático finito ou cálculo efetivo - capaz de determinar se uma fórmula do cálculo de predicados clássico seria demonstrável ou não. O problema era conhecido como o “problema da decisão”, porque sua formulação

⁹ A solução de Turing para esse problema já vinha sendo discutido pelos formalistas de sua época, o que Turing fez foi mostrar (e demonstrar), em termos matematicamente precisos, como é que um sistema formal automático, com regras muito simples, pode ser poderoso: segundo Turing, um sistema formal automático é um dispositivo físico que manipula (automaticamente) símbolos de um sistema formal de acordo com as regras. A máquina teórica de Turing foi concebida tanto como um exemplo da sua teoria da computação como uma prova de que certos tipos de máquinas computacionais poderiam, de fato, serem construídas. Turing uniu a matemática e a lógica para construir essa máquina abstrata, tornando possível o que hoje conhecemos como sistemas processadores de símbolos.

¹⁰ A noção de algoritmo é ponto de discussão na filosofia, especificamente em disciplinas como a filosofia da matemática, da ciência cognitiva e filosofia da mente. Etimologicamente, o termo teve origem no sobrenome do matemático persa Al-Khwarizmi (consultar no site <http://pt.wikipedia.org/wiki/Al-Khwarizmi> o termo *algoritmo*), suas contribuições tiveram um grande impacto sobre a linguagem. “Álgebra” é derivado de *al-jabr*, uma das duas operações que Mūsā al-Khwarizm usou para resolver equações quadráticas. O radical de *algarismo* e *algoritmo* vem de **algoritmi**, a forma latina de seu nome. Além do português *algarismo*, seu nome também deu origem ao espanhol *guarismo*. No entanto, Mūsā al-Khwarizm viveu por volta do século IX (por volta de 835) e suas obras somente foram traduzidas no ocidente cristão XII, tendo uma delas recebido o nome *Algorithmi de numero indorum, sobre os algoritmos usando o sistema de numeração decimal* (indiano). Na linguagem comum, um algoritmo é, segundo o **Dicionário Houaiss da língua portuguesa** (ver verbete “algoritmo”), “sequência finita de regras, raciocínios ou operações que, aplicada a um número finito de dados, permite solucionar classes semelhantes de problemas”, ou mesmo um “conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de etapas”. Não há como formalizar lógica e matematicamente, nem mesmo definir em *termos de condições necessárias e suficientes*, de acordo com a **Enciclopédia de termos lógico-filosóficos** (2006, p.26), a noção do que seja um algoritmo. Podemos apenas conceituar intuitivamente: um algoritmo é “uma sequência de instruções ou regras cuja aplicação permite dar uma resposta definitiva a um dado problema”. Segundo ainda a **Enciclopédia de termos lógico-filosóficos** (2006, p.26), um algoritmo é “um processo efetivo que, ao ser aplicado a certo conjunto de símbolos, produz um, e um só, conjunto determinado de símbolos”. Por isso é possível elencar, pelo menos, 5 (cinco) características gerais e cruciais (lógico-conceituais) pertencentes à noção de algoritmo, são elas: “1) Um algoritmo define-se por um conjunto finito de instruções e não pelos poderes causais do agente que segue as instruções; 2) Um agente de computação é capaz de seguir as instruções: não existem instruções ambíguas, mas apenas ordens claras; 3) Para seguir as instruções de um algoritmo é necessário poder computar, armazenar e ler informação; 4) Os algoritmos são discretos: as suas instruções têm de ser apresentadas passo a passo; e 5) A computação que resulta de um algoritmo pode ser levada a cabo de modo determinista” (BRANQUINHO; MURCHO; GOMES, 2006, p.26).

básica foi a seguinte: “dada uma formula qualquer do calculo de predicados clássico, existe um procedimento sistemático finito que permita determinar se essa formula é demonstrável ou não?” (DUPUY, 1995, p. 28). Porém, Turing¹¹ reformulou-o do seguinte modo: “Existe uma Máquina de Turing capaz de computar se determinado procedimento sistemático¹² finito levará a uma demonstração de um teorema ou não?” (BENANTE, 2001, p. 17).

Para alcançar esse objetivo teria (como Turing já era lúcido disso) que ser criado uma *Máquina Universal de Turing* que simularia qualquer outra máquina que desempenhasse um procedimento efetivo¹³. Para os modelos da Máquina de Turing não importa a arquitetura da máquina computacional (que irá computar os símbolos), nem mesmo a constituição “material” dessa máquina que executaria o algoritmo. A ideia de Turing é a de um desprendimento da inteligência (digamos “máquina”) em relação aos elementos de sua composição material: a mente para ele era concebida como um programa de computador cujo suporte físico não seria importante para o entendimento da própria estrutura funcional da mente/programa.

2 Descrição geral das Máquinas de Turing

2.1 Descrição informal

Antes de começarmos, é importante ficar claro qual a natureza de uma descrição do tipo informal: descrevem-se as características de uma dada coisa de forma que não se recorra a conhecimentos técnicos de certos tipos formalizações do campo científico e

¹¹ Provavelmente varias preocupações (problemas) fizeram parte do projeto intelectual que Turing construiu. Não queremos aqui limitar o conjunto de problemas que Turing dedicou sua vida. O que este trabalho faz certamente é dar uma pequena amostra de algumas dessas questões. E certamente uma das preocupações de Turing, uma das coisas que perguntou a si mesmo, foi “O que faz o raciocínio humano quando executa um cálculo?”. E ele respondeu que os cálculos mentais, realizados pelos seres humanos, consistem de operações para transformar números em uma série de estados intermediários que progridem de um para outro de acordo com um conjunto fixo de regras, até que uma resposta seja encontrada. Algumas vezes usamos papel e lápis para não perdermos o estado dos nossos cálculos. As regras da matemática exigem definições mais rígidas que aquelas descritas nas discussões metafísicas sobre os estados da mente humana, e Turing concentrou-se na definição destes estados de tal maneira que fossem claros e sem ambiguidades, para que tais definições pudessem ser usadas para comandar as operações de uma máquina (POZZA, PENEDO, 2002).

¹² Ao definir procedimento sistemático Turing se referia a um conjunto finito de instruções executadas por um agente ou controlador para a resolução de uma dada tarefa. A tarefa seria um insumo ao procedimento sistemático via sua formalização em uma memória de entrada, cuja resolução é explicitada em uma memória de saída.

¹³ Como bem nos mostra Churchland (1995, p.170): “O interessante sobre a Máquina Universal de Turing é que, para todo e qualquer procedimento computacional bem definido, uma máquina universal de Turing pode simular uma máquina que executará esses procedimentos. Ela o faz reproduzindo exatamente o mesmo comportamento de entrada/saída da máquina que esta sendo simulada.”.

filosófico - como cálculos e/ou equações matemáticas ou cálculos e/ou simbolizações lógicas. É o que faremos nesta seção deste trabalho.

Iniciaremos essa descrição com algumas perguntas básicas, servindo como guias para que não percamos o foco: o que é uma Máquina de Turing? Qual a natureza de, e como é constituída, uma máquina de Turing? O que essa máquina possui? Quem foi seu inventor, quem a projetou? Para que finalidade ela foi projetada? Por que foi construída?

A máquina de Turing é um dispositivo teórico conhecido como *máquina abstrata universal*, concebida pelo matemático britânico Alan Turing muitos anos antes de existirem os modernos computadores digitais, tais como os conhecemos hoje. Por isso, é um modelo abstrato (teórico) de um computador, que se restringe apenas aos aspectos lógico-conceituais de seu funcionamento (memória, estados e transições, etc.) e não à sua implementação física. O modelo de máquina de Turing foi primeiramente idealizada por Alan Turing na década de 30, na tentativa de resolver um problema proposto por Hilbert, o bem conhecido *problema da decisão*¹⁴. Turing acreditava que poderia dar uma solução decisiva e definitiva a esse problema. Propôs então que uma definição de “máquina de computação” (do inglês *computing machine*) o faria. Essa concepção de máquina atingiria o âmago da noção de procedimento efetivo – conceito muito importante hoje na I.A. e na ciência computacional. Mas Turing não tinha em mente apenas isso. Como muitos comentadores nos mostram hoje, a ideia central que Turing queria demonstrar e argumentar a favor era a de que uma Máquina de Turing poderia simular o mesmo desempenho da mente humana, em qualquer tipo de sua atividade e não apenas no sentido de calcular. Turing acreditava que a mente funcionava seguindo instruções simples, regida por um conjunto de regras mecânicas, por procedimentos puramente mecânicos.

¹⁴ Numa máquina de Turing pode-se modelar qualquer outro computador digital, pois Turing havia se envolvido na construção de máquinas físicas para quebrar os códigos secretos das comunicações alemãs durante a II Guerra Mundial, tendo utilizado de alguns dos conceitos teóricos desenvolvidos por ele mesmo sobre um modelo de *computador universal*.



Figura 1: Representação artística de uma máquina de Turing, fonte Wikipédia.

Continuando em nossa definição (informal) das máquinas de Turing, ainda temos que explicitar que esta máquina consiste de:

1. Uma fita que é dividida em células, uma adjacente à outra. Cada célula contém um símbolo de algum alfabeto finito. O alfabeto contém um símbolo especial branco (aqui escrito como \rightarrow) e um ou mais símbolos adicionais. Assume-se que a fita é arbitrariamente extensível para a esquerda e para a direita, isto é, a máquina de Turing possui tanta fita quanto é necessário para a computação. Assume-se também que células que ainda não foram escritas estão preenchidas com o símbolo branco¹⁵.

2. Um cabeçote, que pode ler e escrever símbolos na fita e mover-se para a esquerda e para a direita¹⁶.

3. Um registrador de estados, que armazena o estado da máquina de Turing. O número de estados diferentes é sempre finito e há um estado especial denominado estado inicial com o qual o registrador de estado é inicializado¹⁷.

4. Uma tabela de ação (ou função de transição) que diz à máquina que símbolo escrever, como mover o cabeçote (para esquerda e para direita) e qual será seu novo estado, dados o símbolo que ele acabou de ler na fita e o estado em que se encontra. Se não houver entrada alguma na tabela para a combinação atual de símbolo e estado então a máquina pára¹⁸.

¹⁵ Disponível em: <[www.http://pt.wikipedia.org/wiki/Máquina_de_Turing](http://pt.wikipedia.org/wiki/Máquina_de_Turing)>. Acesso em: 21 de Jun. 2011.

¹⁶ Disponível em: <[www.http://pt.wikipedia.org/wiki/Máquina_de_Turing](http://pt.wikipedia.org/wiki/Máquina_de_Turing)>. Acesso em: 21 de Jun. 2011.

¹⁷ Disponível em: <[www.http://pt.wikipedia.org/wiki/Máquina_de_Turing](http://pt.wikipedia.org/wiki/Máquina_de_Turing)>. Acesso em: 21 de Jun. 2011.

¹⁸ Disponível em: <[www.http://pt.wikipedia.org/wiki/Máquina_de_Turing](http://pt.wikipedia.org/wiki/Máquina_de_Turing)>. Acesso em: 21 de Jun. 2011.

2.2 Definição formal

O que faremos aqui é oferecer uma descrição um pouco mais precisa (do ponto de vista lógico-formal) das Máquinas de Turing que possuem fitas (como mostrado mais acima na representação artística de uma máquina de Turing), e *um pouco mais precisa* quer dizer aqui mais técnica do que a descrição informal.

Deste modo, podemos dizer que uma máquina de Turing (com uma fita) é usualmente definida como uma *Tupla* $M = (Q, \Sigma, \Gamma, s, b, F, \delta)$, em que cada um desses símbolos podem ser caracterizados da seguinte forma:

- O símbolo Q representa um conjunto finito de estados;
- O símbolo Σ representa um alfabeto finito de símbolos;
- O símbolo Γ representa o alfabeto da fita (conjunto finito de símbolos);
- Os três seguintes símbolos $s \in Q$ representam uma função que designa o estado inicial da máquina;
- Os três seguintes símbolos $b \in \Gamma$ designam o símbolo branco (o único símbolo que se permite ocorrer na fita infinitamente em qualquer passo durante a computação);
- $F \subseteq Q$ designa o conjunto dos estados finais da máquina;
- $\delta : Q \times \Gamma \Rightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ é uma função parcial chamada

função de transição, onde \leftarrow é o movimento para a esquerda e \rightarrow é o movimento para a direita.

Como podemos ver nessa descrição formal, uma máquina de Turing é um modelo de uma máquina abstrata, quer dizer, é apenas um modelo lógico-conceitual. Tal modelo abstrato foi projetado para realizar computações, efetuar processos computacionais. Os processos computacionais¹⁹ efetuados por uma máquina de Turing consistem em uma manipulação ordenada/sequenciada/programada de símbolos (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519). Por isso, para entendermos

¹⁹ Para entender mais o que são processos computacionais, ver o verbete “linguagem do pensamento” da **Enciclopédia de termos lógico-filosóficos** editado por BRANQUINHO; MURCHO e GOMES, 2006, p. 441.

como opera uma máquina de Turing, é preciso que entendamos os elementos básicos sob os quais opera essa máquina. Vamos a esses elementos, que são os seguintes: um *alfabeto*, um *conjunto de estados* e *conjunto dos movimentos*; reforçando, máquina (abstrata) de Turing é constituída por esses três conjuntos de base (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519).

- O *alfabeto* é um deles, representado pelo símbolo $\mathbf{S} = (\mathbf{S}_0, \mathbf{S}_1, \dots, \mathbf{S}_e)$. No que diz respeito ao alfabeto, podemos entendê-lo enquanto um conjunto finito de símbolos que a máquina é capaz de reconhecer ou que tem a capacidade de reconhecer, ou com o que a máquina de Turing trabalha; \mathbf{S} contém sempre um símbolo, dito o símbolo branco, aqui designado por \mathbf{S}_0 , e os símbolos restantes serão chamados símbolos próprios (há sempre, e no mínimo, um símbolo próprio, de modo que \mathbf{S} tem pelo menos dois elementos) (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519).
- O *conjunto de estados* é outro desses elementos, que podem ser expressos formalmente pelos símbolos $\mathbf{Q} = \{\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_m\}$, são os estados que a máquina de Turing pode assumir, sendo um dos estados \mathbf{q}^{**} (qm se nada for dito em contrário) chamado estado passivo ou terminal ou final, e os restantes, estados ativos. Entre os estados ativos, um deles, denotado pelo símbolo \mathbf{q}^* , diz-se que esse é o estado inicial. Convencionalmente pode-se dizer que o primeiro estado de uma máquina de Turing seja \mathbf{q}_0 (em que \mathbf{q} designa também um conjunto finito com pelo menos dois elementos) (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519).
- Por fim, o último dos conjuntos de bases da máquina de Turing é o *conjunto dos movimentos*, designados com os símbolos $\mathbf{M} = \{\mathbf{e}, \mathbf{d}, \mathbf{p}\}$: o símbolo “e” designa movimento para a esquerda; “d” designa movimento para a direita, e “p” indica ausência de movimento ou permanência na mesma posição (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519)

Continuando, a máquina (abstrata) de Turing, tal como foi concebida (idealmente) por Alan Turing, foi pensada por ele como consistindo fisicamente dos seguintes elementos: 1) uma *fita* (potencialmente), que é *infinita* em ambos os sentidos, isto é, infinita para a esquerda e infinita para a direita e 2) uma *cabeça de leitura/escrita*. Passamos agora a descrever cada um desses dois elementos (BRANQUINHO; MURCHO; GOMES, 2006, p. 515-519).

Designa-se por “fita” de uma máquina de Turing uma fita que pode ser dividida em casas/quadrados/células, em que cada um(a) desses(as) casas/quadrados/células estão escritos um dos símbolos do alfabeto que essa máquina possui e que é capaz de reconhecer (é claro que isso inclui a possibilidade de não poder ter nada escrito nas casas/quadrados/células da fita dessa máquina, possibilidade essa em que a casas/quadrados/células podem estarem em branco, isto é, pode-se dizer que na fita está escrito o símbolo branco). A cabeça de leitura pode estar posicionada, a cada momento do tempo, sobre uma das células da fita (BRANQUINHO; MURCHO; GOMES, 2006, p. 515-519).

A máquina de Turing pode encontrar-se num estado, representado pelos seguintes símbolos $q \in Q$, que é dito o estado da máquina em determinado instante de tempo. Bem, se o estado da máquina é (está) ativo, então a máquina se encontra numa situação que podemos dizer ser uma *situação ativa*. Caso contrário, pode-se dizer que a máquina se encontra em uma situação passiva (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519).

A máquina de Turing sempre poderá ser encontrada numa condição que é comum chamar de *condição finita*: isso quer dizer mais nada que, mesmo que a fita seja infinita (para esquerda e para a direita), a cada instante de tempo somente pode estar inscrito um símbolo próprio num número finito de casas, ou que a cada período de tempo, somente num número finito de casas é que pode estar inscrito um símbolo próprio (BRANQUINHO; MURCHO; GOMES, 2006, p.515-519).

Deste modo, um processo computacional é executado por esse dispositivo (a máquina de Turing) que pode ler/escrever símbolos em uma fita dividida em quadrados. A cabeça de leitura/gravação move-se em qualquer direção ao longo da fita e a um quadrado por vez. O quadrado que é "lido" em cada etapa é conhecido como “quadrado ativo”. A regra que está sendo executada determina o que se convencionou chamar “estado” da máquina (POZZA, PENEDO, 2002).

Numa visão mais técnica, Rubem Benante (2001) defini as características de uma Máquina de Turing da seguinte forma:

- (a) a memória de entrada e de saída que é representada por uma fita, potencialmente infinita para ambos os lados, quadriculada, onde se pode, em cada quadriculado, ler e escrever um símbolo;
- (b) uma cabeça de leitura e gravação sobre essa fita, capaz de alterar e recuperar símbolos, além de se mover para a direita e para a esquerda um quadriculado por vez (ou permanecer em sua posição);

(c) um controlador, que se encontra, a cada passo discreto, em um estado (ou situação) interno distinto, atuando conforme instruções, que são determinadas para todas as combinações de estado e símbolo. Essas instruções indicam: (i) o símbolo que deve ser gravado na fita, (ii) a direção para onde a cabeça de leitura/gravação deve se mover e (iii) o novo estado que o controlador deve assumir. (BENANTE, 2001, p. 61).

Como vimos, este tipo de máquina abstrata possui um *suporte formal* (que deve ser descrito *formalmente*). Tal suporte formal é descrito por Oliveira (2003) da seguinte forma:

- a) um conjunto finito de estados internos ou condições (na terminologia de TURING, 1936) $C=\{c_1, c_2, \dots, c_n\}$
- b) um conjunto finito de símbolos, ou alfabeto $S=\{s_1, s_2, \dots, s_n\}$
- c) um conjunto finito de instruções, regras, ou configurações (terminologia de TURING, 1936) $M=\{m_1, m_2, \dots, m_n\}$ (OLIVEIRA, 2003, p. 21).

Fazendo referencia ainda ao texto de Felipe de Oliveira (2003), podemos notar que o subconjunto $\{ci, cf\} \in C$, onde ci é o estado inicial e cf é o estado final. Cada instrução $mi \in M$ é dada na seguinte forma: $mi=\{cn, sn, sk, de, cj\}$, onde cn é o estado atual da máquina, sn é o símbolo que se encontra no quadrado sob a cabeça de gravação-leitura, sk é o símbolo que a máquina deve gravar no mesmo quadrado, de é a direção que a máquina deve mover a cabeça de gravação-leitura (existem apenas duas opções: direita e esquerda), e cj o novo estado a que a máquina deve passar (OLIVEIRA, 2003, p. 18-21).

Assim sendo, a operação da máquina é determinada pelo conjunto M , que por sua vez pode ser descrito na forma de uma tabela. O que o programador nada mais faz é criar esta tabela determinando o comportamento da máquina em cada situação discreta, estabelecendo anteriormente o conjunto de estados internos C e o alfabeto S . Quando a máquina atingir o estado final cf diz-se que ela chegou à solução do problema proposto, podendo-se ler nos caracteres gravados na fita da máquina a solução. Se o problema não for solucionável em n passos finitos, a máquina não parará. Por exemplo, se programarmos a máquina para calcular o maior número primo possível, ela nunca parará de calcular, visto a memória (fita) ser potencialmente infinita (OLIVEIRA, 2003, p. 18-21).

Neste modelo (de Máquina de Turing), as listas de instruções distintas nunca possuem mais elementos que o produto do numero de símbolos do alfabeto pelo numero de estados internos, e tanto os símbolos como o numero de instruções são finitos. O

procedimento sistemático, efetivo, para a resolução de uma tarefa, nada mais é que uma Máquina de Turing com o devido conjunto de instruções. Um algoritmo, portanto, não é mais que um conjunto finito de instruções correntes com a resolução de determinada tarefa (OLIVEIRA, 2003, p.18-21).

3 Operação das Máquinas de Turing

Mas como funciona, do ponto vista lógico-conceitual, uma Máquina de Turing? Nos utilizaremos novamente da descrição feita por Rubem Benante (2001, p. 60-65).

Nessa descrição de Benante, a Máquina de Turing sempre executa instruções que lhe são impostas e assim opera sobre uma fita, como descrevemos anteriormente. Benante (2001, p. 64-65) explica esse funcionamento, dizendo que para que a Máquina de Turing possa executar as instruções que lhe são inseridas e assim operar a fita, deve seguir os passos de uma *tabela 1* (que é fornecida como um exemplar abaixo):

- 1 - Lembrar seu estado atual Si.
- 2- Ler o símbolo A_j sob a posição atual da cabeça de leitura/gravação.
- 3- Com base em Si e A_j procurar a próxima instrução I_{ij} para executar a instrução I_{ij} na lista, pare.
- 4- Escrever o símbolo Ak na fita, na posição atual da cabeça de leitura/gravação.
- 5- Mover a cabeça de leitura/gravação para a direita ou esquerda, de Mh.
- 6- Fazer do estado da instrução, Sf, o estado atual Si.
- 7- Repetir todos os passos anteriores.

Tabela1: Passos seguidos pelo controlador para executar uma instrução na Máquina de Turing.

Todos esses são os passos seguidos (ou os sistema mecânico que reagem às instruções) para executar uma instrução como uma Máquina de Turing. Benante (2001, p. 60-65) exemplifica isso e nós nos utilizaremos dessa luz novamente para exemplificar essa execução simples de uma Máquina de Turing. Vejamos os procedimentos impostos pela seguinte configuração (abaixo):

Estado Atual	Símbolo Lido	Gravar Símbolo	Mover Cabeça	Mudar estado
Si	Aj	Ak	Mh	Sf

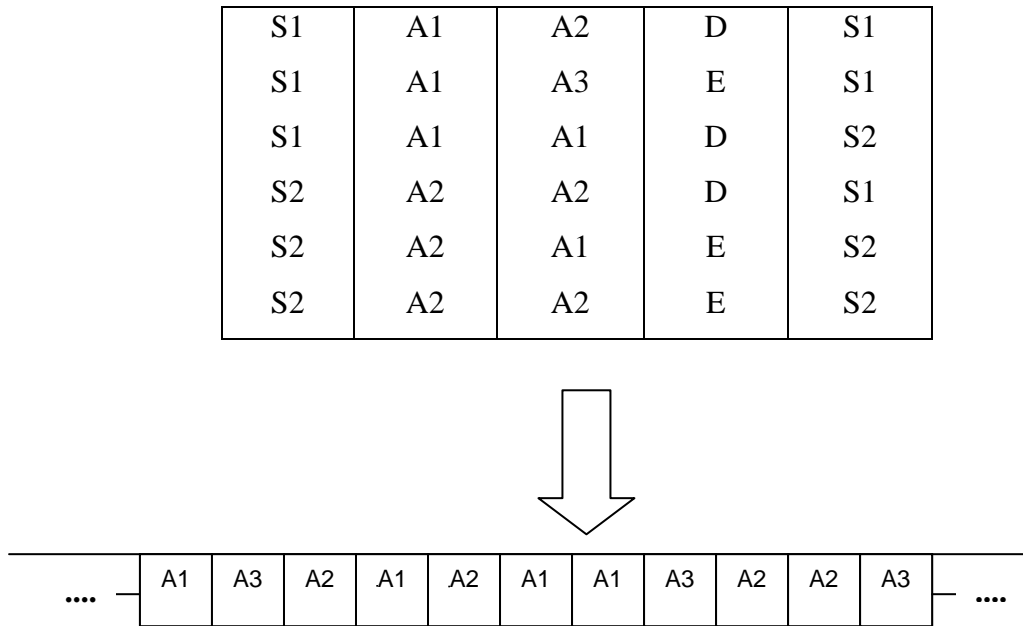


Figura 1: Esquema de uma Máquina de Turing

Este esquema de Benante (2001, p.60-65) (da figura 1) de uma Máquina de Turing exemplifica seu funcionamento: como vemos, temos um alfabeto de três símbolos (**A1 A2 A3**) e dois estados internos (**S1** e **S2**). Existem seis instruções distintas (como falamos acima: o produto de seis por dois) que são as linhas da figura

1. Sendo assim, suponhamos o estado inicial **S1** e iniciemos a execução da instrução (tabela 2, abaixo)

- 1- Estado atual $S_i = S1$
- 2- Ler o símbolo $A_j = A1$ (posição da cabeça de leitura/gravação).
- 3- Próxima instrução $I_{ij} = (S_i, A_j, A_k, M_h, S_f) = (S1, A1, A2, D, S1)$.
- 4- Escrever o símbolo $A_k = A2$ na fita.
- 5- Mover a cabeça de leitura/gravação para a $M_h = D$, direita.
- 6- Fazer do estado da instrução, $S_f = S1$, o estado atual S .
- 7- Repetir todos os passos anteriores.

Tabela 2: execução da primeira instrução da figura 1

Este ciclo (BENANTE, 2001, p. 60-65) novamente representa a execução de uma instrução (realizada pelo controlador), e podemos notar que a Máquina de Turing gravou **A2** na posição inicial e moveu a cabeça para a direita, permanecendo no mesmo estado interno **S1**. Se continuarmos tudo isso de forma análoga, durante 31 ciclos, iremos ver que todos os símbolos do intervalo da fita da figura 1 foram modificados para o símbolo **A3**. Segundo Benante, ficará mais fácil de visualizar e entender seu funcionamento (da máquina) nos orientando pelo diagrama de Benante (2001, p.65) que colocamos a seguir:

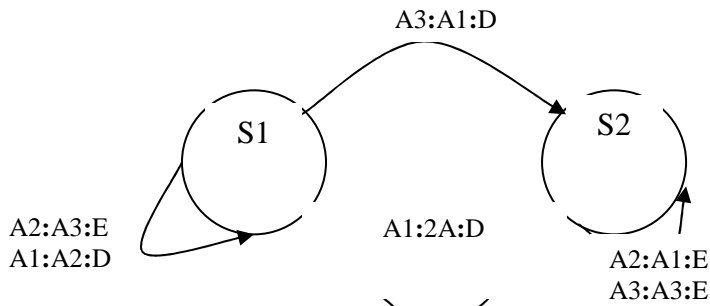


Figura 2: Diagrama da Máquina de Turing representada na figura 1

Como vemos, neste diagrama Benante (2001, p.66) também está representado a forma como uma Máquina de Turing pode funcionar. Os círculos em que estão inseridos os símbolos **S1** e **S2** representam os estados internos da máquina, sendo o número **1** o estado inicial. As setas representam as transições de estados que são realizadas de acordo com as instruções que são inscritas sobre elas. Essas instruções possuem três termos, separados por (:). O primeiro é o símbolo lido na fita da máquina (**Aj**), o segundo é o símbolo a ser gravado na mesma posição (**Ak**), e o terceiro é o movimento da cabeça de leitura e gravação deve fazer (**Mh** **D** para direita e **E** para esquerda). A quintupla que descreve a instrução na Máquina de Turing $I_{ij}=(S_i, A_j, A_k, M_h, S_f)$, fica completa ao se integrar o estado inicial **Si** atual (o número dentro do círculo atual) e a passagem do controle para o novo estado **Sf** (o número dentro do próximo círculo, apontado pela seta). Devemos indicar também (como na figura 1) os dados iniciais na fita, a posição inicial da cabeça de leitura e gravação e o estado inicial, para termos uma descrição precisa.

Nesse exemplo apresentado a máquina nunca irá interromper sua execução por motivo algum, uma vez que ela só pararia se recebesse uma combinação de estado-símbolo que não estivesse pré-programada. No entanto, notemos que na figura 1 (BENANTE, 2001, p. 60-66) todas as seis instruções possíveis foram dadas de antemão. Isso porque, como já falamos, o alfabeto é predeterminado, e assim a máquina nunca poderia encontrar, na sua fita infinita, um símbolo, durante sua permanência num estado qualquer, que não possua instrução prevista para ser executada. Pode até acontecer de um surgimento de um ciclo em que a cabeça de leitura/gravação fique presa em um trecho da fita, andando da esquerda para a direita e vice-versa.

Agora, na figura 3 (BENANTE, 2001, p. 67), podemos exemplificar esse comportamento imaginando uma máquina cuja fita inicial seja composta pelos símbolos

AB, com a cabeça de leitura gravação posicionada inicialmente sobre **A**, no único estado 1, e com as duas instruções, **I1A=(1, A, AD, 1)** E **I1B=(1, B, B, E, 1)**.

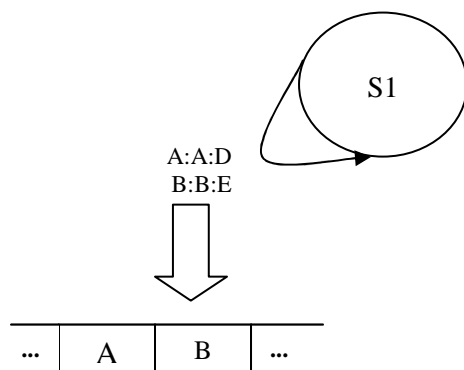


Figura 3: Máquina de Turing em um eterno ciclo

Neste caso a máquina apenas pára se o controlador não conseguir achar a próxima instrução, conforme o passo 3 de seu algoritmo (tabela 1). Benante (2001, p. 67) exemplifica isso da seguinte forma:

[...] suponha que na figura 1, ao invés das reticências que se encontram na fita, se encontrasse o símbolo A4. A máquina pararia assim que lesse o símbolo A4 e tentasse executar qualquer de ambas as instruções iniciadas por (S1, A4, _, _, _) ou (S2, A4, _, _, _). Analogamente, o mesmo aconteceria se alguma instrução válida executada anteriormente remetesse a máquina para um estado inexistente (S3, por exemplo).

Segundo ainda Benante (2001, p.60-68), todos esses desfechos antes citados – falta de instrução para seguir, ou mesmo ciclo repetitivo da cabeça de leitura/gravação – são passíveis de serem previstos e, teoricamente, poderiam ser evitados por via de uma programação mais cuidadosa, uma vez que a fita é dada de antemão.

Passamos agora a apresentar um tipo de explicação mais intuitiva e menos técnica sobre a operação/funcionamento de uma máquina de Turing. Para isso nós temos que pedir ao leitor que imagine agora os símbolos "I" e "-" (**branco**). Suponhamos agora que o dispositivo pode limpar qualquer um deles (o símbolo) quando ele (o dispositivo) os manipula em um quadrado ativo e trocá-lo por outro (isto é, apagar "I" e substituir por "-" e vice-versa). O dispositivo pode mover a cabeça de leitura/gravação para a direita ou esquerda, de acordo com instruções dadas à unidade de controle. As instruções podem limpar um símbolo, escrevê-lo ou deixá-lo como está, de acordo com o símbolo

manipulado. Qualquer tipo de jogo pode ser elaborado usando estas regras, não tendo necessariamente algum significado²⁰.

Dado um quadrado que seja uma posição inicial de uma seção da fita preenchida por quaisquer caracteres (ou brancos), o dispositivo executa ações especificadas por uma lista de regras, seguindo-as, uma a uma por vez, até chegar àquela que force sua parada (se não há uma instrução explícita na tabela para uma determinada configuração da fita, então não há nada que a máquina possa fazer quando alcança aquela configuração, o que encerra, portanto, a execução)²¹. Cada instrução - ou regra - estabelece uma ação a ser executada se houver determinado símbolo no quadrado ativo no tempo em que é lido. Apresentamos agora um exemplo extraído do texto de POZZA e PENEDO²², cujo o procedimento estabelece 4 (quatro) diferentes tipos de regras que uma máquina de Turing pode executar: 1) Substituir - (branco) por símbolo; 2) Substituir símbolo por - (branco); 3) Ir um quadrado para a direita; 4) Ir um quadrado para a esquerda. Eis o exemplo:

"Se houver um I no quadrado ativo, substitua-o por -". Esta instrução faz a máquina executar a segunda ação da lista acima. Para se elaborar um "jogo" nós precisamos fazer uma lista que especifique o número da regra que se deve observar no momento atual, e, de alguma forma, qual será a próxima. Cada regra desta lista será composta pela seguinte sequência: o número da regra - estado da máquina -, um caractere/branco para comparação, próximo estado e ação (novo símbolo que irá para o quadrado ou movimentar para direita [>]/esquerda[<] cabeça de leitura/gravação)²³.

Ainda na clarificação do exemplo extraído do texto de POZZA, PENEDO²⁴, faremos uma explicitação agora de uma lista de regras - código e descrição - que servirá de esclarecimento para entendermos como pode uma máquina de Turing ser programada a desenvolver um determinado "jogo":

1 I 2 -

Estado 1: se há um I no quadrado ativo, substitua-o por - e vá para estado 2;

2 - 3 >

Estado 2: se há um - no quadrado ativo, vá para estado 3 e ande um quadrado a direita;

3 I 3 >

Estado 3: se há um I no quadrado ativo, vá para estado 3 e ande um quadrado a direita;

²⁰ Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de Jun. 2011

²¹ Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de Jun. 2011

²² Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de Jun. 2011.

²³ Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de Jun. 2011.

²⁴ Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de Jun. 2011.

3 - 4 >

Estado 3: se há um - no quadrado ativo, vá para estado 4 e ande um quadrado a direita;

4 I 4 >

Estado 4: se há um I no quadrado ativo, vá para estado 4 e ande um quadrado a direita;

4 - 5 I

Estado 4: se há um - no quadrado ativo, substitua-o por I vá para estado 5;

5 I 5 >

Estado 5: se há um I no quadrado ativo, vá para estado 5 e ande um quadrado a direita;

5 - 6 I

Estado 5: se há um - no quadrado ativo, substitua-o por I vá para estado 6;

6 I 6 <

Estado 6: se há um I no quadrado ativo, vá para estado 6 e ande um quadrado a esquerda;

6 - 7 <

Estado 6: se há um - no quadrado ativo, vá para estado 7 e ande um quadrado a esquerda;

7 I 8 <

Estado 7: se há um I no quadrado ativo, vá para estado 8 e ande um quadrado a esquerda;

8 I 8 <

Estado 8: se há um I no quadrado ativo, vá para estado 8 e ande um quadrado a esquerda;

8 - 1 >

Estado 8: se há um - no quadrado ativo, vá para estado 1 e ande um quadrado a direita;

Note que se houver um branco no quadrado ativo quando os estados forem 1 ou 7, ou se há um I no quadrado ativo quando o estado da máquina é 2, ela pára, pois não saberia o que fazer²⁵.

Podemos notar que neste jogo²⁶ a máquina de Turing é programada a duplicar uma sequência de **I**'s que estejam na fita. Neste caso, se a fita contiver **IIII**, no final conterà **IIIIIIII**. Para que se possa jogar (na verdade executar o programa descrito na lista [citada acima] de regras) é preciso para isso que se especifique uma configuração inicial para a máquina executar em sua fita: por exemplo, qual o quadrado inicial ativo e o estado inicial da máquina. Quando a máquina passa a executar essas regras, ela (a máquina) parte do estado inicial em que se encontra, bem como do respectivo quadrado ativo, seguindo a sequência (lógica) de regras que darão o produto final (POZZA, PENEDO, 2002)²⁷.

Em sua essência, ou seja, em sua natureza lógica-matemática, toda máquina de

²⁵ Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de Jun. 2011.

²⁶ Descrito e disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de Jun. 2011.

²⁷ Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de Jun. 2011.

Turing move-se ou move símbolos, de uma posição para outra em uma fita, da mesma maneira que no exemplo explicitado (acima). Hodiernamente, estes símbolos podem ser impulsos eletrônicos em um microcircuito e a fita uma série de endereços de memória em um chip, mas a ideia básica é a mesma:

Em sua ambição de esclarecer e resolver o problema da decisão de Hilbert, Turing procurou provar que sua hipotética máquina poderia ser vista como uma versão automatizada de um *sistema formal* que é especificado por uma combinação inicial de símbolos (o conjunto de "I"s na fita no início do processo) e as regras (as instruções escritas). Numa máquina de Turing, os movimentos nada mais são do que mudanças de 'estado' dessa máquina e que correspondem a específicos passos de computação. Segundo Pozza e Penedo²⁸, foi Alan Turing que procurou provar pela primeira vez que, para qualquer tipo de sistema formal, existe uma máquina de Turing correspondente que pudesse ser programada para imitar esse mesmo sistema formal; pois que era “este sistema formal genérico”, com a habilidade de imitar qualquer outro sistema formal, o que Turing procurava. Tais sistemas hoje são chamados de “Máquinas de Turing Universais” (POZZA; PENEDO)²⁹.

Considerações Finais

Tentamos, ao longo deste trabalho, mostrar como foi concebida (oque é) e como funciona lógico-conceitualmente a noção de uma Máquina de Turing. Elaboramos descrições do ponto de vista lógico/formal de sua natureza (abstrato/ideal) e funcionamento (mecânico e regido por regras).

Porém, como colocamos na introdução a este trabalho, a noção de Máquina de Turing não se limitaria a apenas este campo descrito por nós. Isso porque a pergunta “é possível construir e configurar um dispositivo puramente físico para que ele possua inteligência?” não foi explorada somente pela IA contemporânea. É de longa data em nossa cultura filosófica, científica.

Por volta da metade do século XVII, o filósofo-matemático alemão Leibniz construiu um dispositivo que podia somar e subtrair por meio de cilindros rotatórios interconectados. Leibniz foi um dos primeiros a crer na possibilidade de uma linguagem perfeitamente lógica em que todo pensamento poderia ser reduzido a simples cálculos.

²⁸ Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>.

²⁹ Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>.

Além de ter sido um empreendimento intelectual produto desse nosso passado intelectual, a noção de Máquina Universal de Turing é responsável por outros desenvolvimentos intelectuais que vieram depois de sua formulação. O que queremos dizer é que a apropriação dessa noção pela I.A. e pelas teorias computacionais ocasionou o surgimento e posterior desenvolvimento do computador moderno, convencional, pelo menos no plano lógico, formal e matemático. Porque um computador moderno nada mais é que uma **Máquina Universal de Turing**. É o que vislumbra Churchland (1995) com a expressão *máquinas de finalidades gerais*, em que aponta que a única diferença entre os computadores reais e as Máquinas Universais de Turing é que aqueles não possuem memória ilimitada, enquanto que estas últimas sim, muito embora a memórias dos computadores convencionais possam ser ampliadas de acordo com as necessidades impostas a eles.

Consequentemente é inegável a influencia da noção de Máquina de Turing sobre certos pontos de vistas acerca da natureza e funcionamento da mente, até mesmo porque Turing teve uma preocupação direta (e não periférica) sobre esse tema. Ele próprio reconhecia que sua concepção de máquinas teria algumas consequências sobre o conhecimento atual do cérebro e da mente humana, assunto tão em voga e polemico hoje: Turing usou o termo *computador (computer)* tendo por referencia seres humanos que eram habilitados e estavam engajados em fazer cálculos. Sendo assim, ele havia pensado mais em computadores humanos (do inglês *human computers*) ao desenvolver a noção de Máquina de Turing que preconizou a própria ideia do computador convencional de hoje. Turing sugeriu que seria possível simplificar e mecanizar processos pelos quais seres humanos (computadores humanos) fazem cálculos, na tentativa de generalizar os mecanismos pelos quais estes mesmos seres humanos fazem cálculos para definir uma classe especial de números.

Muitas das ideias de Turing foram retomadas posteriormente, principalmente pelo funcionalismo conhecido como *funcionalismo da Máquina de Turing*: cujo o pressuposto básico de *função* formulado por Turing serviu de inspiração para os funcionalistas, tais como Hilary Putnam. O pressuposto era que os estados mentais de uma pessoa são vistos como funções que esses mesmos estados desempenham. Os funcionalistas, tal como a ideia de Turing sobre a independência do programa em relação a seu substrato material, desconsideram o cérebro para resolver os problemas da mente e do corpo. Consideram que o teste de Turing seria mais apropriado para resolvê-lo, avaliando a capacidade de uma máquina de responder corretamente a perguntas, pois

que os estados mentais eram entendidos como abstrações de regras e mesmo o raciocínio poderia ser considerado um processo de manipulação de símbolos através de algoritmos.

As consequências dessas teses é que tanto Turing e os funcionalistas aceitariam de bom grado um tipo de dualismo entre máquina (computacional) e programa. Muitos críticos de hoje da IA (e da Ciência Cognitiva em geral) afirmam que tal pressuposto tem raízes, ou pelo menos sérias aproximações, na filosofia cartesiana. Isso porque os adeptos desse pressuposto apostam que apenas as regras abstratas serviriam para estruturar o conhecimento sobre a mente: o comportamento inteligente seria um resultado de uma série de regras a seguir (quando, por exemplo, uma pessoa executa uma ação, ela nada faz mais que seguir um conjunto de regras para executá-la). Daí segue-se que se pudéssemos explicitar tais regras que determinam o, e que são parte do, comportamento inteligente, e conseguíssemos também escrevê-lo em termos de algoritmos, poderíamos implementar tais regras/algoritmos em um computador: é exatamente o algoritmo que contém as regras que possibilita o comportamento inteligente e, logo, é o entendimento do algoritmo que possibilitaria o entendimento desse comportamento – não as máquinas, isto é, onde o próprio algoritmo pode ser realizado.

Ao especificar deste modo a mente em termos puramente formais, isso compromete os funcionalistas com uma forma de dualismo, mas não um dualismo qualquer: o chamado dualismo de propriedades³⁰. Este dualismo de propriedades diria o seguinte: nós somos compostos por duas propriedades distintas, uma lógica (que teria a forma de um algoritmo), e outra física (que seria o “hardware” em que possivelmente se processaria o algoritmo). O algoritmo possuiria uma existência abstrata e desmaterializada, totalmente independente do objeto que o realizaria. Nesse sentido é

³⁰ O dualismo de propriedades é uma posição em filosofia que defende a existência de duas propriedades básicas na realidade: as propriedades físicas e as mentais. O dualismo de propriedades é uma posição filosófica acerca do problema mente-corpo, posição essa que poderíamos situá-la entre o dualismo de substância e o materialismo, já que procura combinar os aspectos plausíveis de ambos o dualismo e o materialismo, tentando evitar e ultrapassar as dificuldades teórico-lógicas dessas duas posições. Tal como os materialistas, defende que há apenas um único tipo de substância: a física. Negam mesmo que existe um tipo de substância imaterial em geral que interage com as substâncias físicas e chegam a negar que a mente seja uma substância imaterial. Mas os dualistas de propriedades, diferentemente dos materialistas, não tentam reduzir a mente à matéria, não afirmam que tudo é unicamente matéria, não reduzem os estados mentais aos estados físicos. De acordo com os dualistas de propriedades, os estados mentais são propriedades irreduzíveis ao físico, à matéria, pois não há como realizar uma análise puramente física dos fenômenos mentais em geral. Deste modo, os dualistas de propriedades defendem que há dois tipos de propriedades básicas resultantes da substância física: as propriedades físicas e as não físicas. Nosso corpo tem propriedades físicas tais como peso e altura, bem como propriedades mentais tais como crenças e desejos.

que se entende que a IA, o Funcionalismo da Máquina de Turing teriam posturas dualistas.

Concluindo, como bem expressou Samuel Guttenplan (1995, p.596), no desenvolvimento de sua máquina, Turing achou que poderia fazer tanto uma contribuição crucial à lógica e à matemática quanto à filosofia da mente.

Referências

- BENANTE, Rubem Carlo. **Alcance e limites das Redes Neurais Artificiais na Ciência Cognitiva**. 2001. Dissertação (Mestrado em Filosofia) – Faculdade de Filosofia e Ciências, Universidade Estadual Paulista, Marília, 2001. *Orientador(a): Prof^a. Dr^a. Maria Eunice Quilici Gonzalez.*
- BRANQUINHO, João; MURCHO, Desidério; GOMES, Nelson Gonçalves. **Enciclopédia de termos lógico-filosóficos**. São Paulo: Martins Fontes, 2006.
- CHURCHLAND, Paul M. **Matéria e Consciência: uma introdução contemporânea à Filosofia da Mente**. Trad. Maria Claro Cescato. São Paulo: Unesp, 2004.
- Dicionário Escolar de Filosofia online**. Disponível em: <<http://www.defnarede.com>>. Acesso em: 22 de set. 2011.
- EPSTEIN, Richard L.; CARNIELI, Walter. **Computability: computable functions, logic, and the foundations of mathematics**. London: An International Thomson Publishing Company London, 2000.
- GOLDSTEIN, Rebecca. **Incompletude: a prova e o paradoxo de Kurt Gödel**. São Paulo: Companhia das Letras, 2011.
- GUTTENPALN, Samuel (Ed). **A companion to the Philosophy of Mind**. Oxford: Blackwell Companions to Philosophy. 1995.
- OLIVEIRA, Luis Felipe de. **As contribuições da ciência cognitiva à composição musical**. 2003. Dissertação (Mestrado em Filosofia) – Faculdade de Filosofia e Ciências, Universidade Estadual Paulista, Marília, 2003. *Orientador(a): Prof. Dr. Edson Sekeff Zampronha.*
- POZZA, Osvaldo Antonio; PENEDO, Sérgio. **A máquina de Turing**. Disponível em: <www.inf.ufsc.br/~barreto/trabaluno/MaqT01.pdf>. Acesso em: 22 de jun.2011.
- WIKIPEDIA: a enciclopédia livre**. Disponível em: <<http://pt.wikipedia.org/wiki/Brasil>>. Acesso em: 22 de set. 2011.
- WIKIPEDIA: the free encyclopedia**. Disponível em: <<http://en.wikipedia.org/wiki>>. Acesso em: 22 de set. 2011.