b) $R_2 = \{ \langle a, a \rangle, \langle a, b \rangle, \langle b, a \rangle, \langle b, b \rangle, \langle b, c \rangle \}$
c) $R_3 = \{ \langle a, a \rangle, \langle b, b \rangle, \langle b, c \rangle, \langle c, b \rangle, \langle a, c \rangle, \langle c, a \rangle \}$
d) $R_4 = A \times A$
e) $R_5 = \varnothing$

Exercício 6.20 Em que condições a relação $\varnothing: A \to A$ é uma relação de equivalência?

Exercício 6.21 Seja R uma endorrelação em $\mathbf{N}^2$ definida por:

$$\langle a, b \rangle R \langle c, d \rangle \Leftrightarrow a + b = c + d$$

Demonstre que R é uma relação de equivalência.

Exercício 6.22 Seja R uma endorrelação em $\mathbf{N}^2$ definida por:

$$\langle a, b \rangle R \langle c, d \rangle \Leftrightarrow a * d = b * c$$

Demonstre que R não é uma relação de equivalência.

Exercício 6.23 As *relações de equivalência* são fechadas para as seguintes operações sobre conjuntos (ou seja, a operação de duas relações de equivalência resulta em uma relação de equivalência)? Justifique a sua resposta:

a) União;
b) Intersecção;
c) Complemento;
d) Diferença;
e) Produto Cartesiano.

Exercício 6.24 Seja $A = \{ 1, 2, 3 \}$. Para cada uma das seguintes relações de equivalência, apresente a partição induzida (ou seja, o correspondente conjunto quociente):

a) $\langle A, = \rangle$
b) $\langle \mathbf{P}(A), = \rangle$
c) $\varnothing: \varnothing \to \varnothing$
d) $A^2: A \to A$

Exercício 6.25 Sejam $A = \{ x \mid 0 \leq x \leq 10 \}$ e R uma endorrelação em A definida por:

$$x R y \Leftrightarrow (\exists k \in \mathbf{Z})\ (x - y = 4k)$$

Qual o conjunto quociente A/R?

Exercício 6.26 Complemente a implementação do sistema proposto no Capítulo 4 - Relações (o qual permite definir relações e tratar construções correlatas) de tal forma que também seja capaz de:

- verificar as propriedades de uma endorrelação;
- para um conjunto de propriedades selecionadas, calcular o fecho de uma endorrelação.

7 Cardinalidade de Conjuntos

Entende-se por *cardinalidade* de um conjunto uma medida de seu tamanho. Até o momento, a cardinalidade de conjuntos vem sendo tratada de maneira informal ou semi-formal. Por exemplo, a seguinte expressão foi usada com alguma freqüência:

número de elementos de um conjunto

Também já foi afirmado que:

um conjunto pode possuir um número finito ou infinito de elementos

Nesse contexto, foi afirmado que um conjunto é:

a) *Conjunto finito* se pode ser denotado por extensão, ou seja, listando exaustivamente todos os seus elementos;
b) *Conjunto infinito*, caso contrário.

Assim, em um estudo mais formal sobre a cardinalidade de conjuntos, as seguintes perguntas surgem naturalmente:

- como definir formalmente a cardinalidade de um conjunto?
- quando dois conjuntos possuem o mesmo cardinal?
- o que é um cardinal infinito?
- existe mais de um, ou seja, existem diferentes cardinais infinitos?
- nesse caso, existe uma ordem de cardinais infinitos?

Essas e outras perguntas, além de relevantes, são muito importantes no estudo da Computação e Informática. Entretanto, um estudo mais completo ou aprofundado do assunto foge um pouco do escopo deste livro. Assim, este capítulo aborda a questão com ênfase nos conceitos, resultados e interpretações mais usados em Computação e Informática, sem deixar de lado a precisão formal.

Uma conseqüência importante do estudo da cardinalidade na Computação e Informática é que, computacionalmente falando, existem mais problemas não-solucionáveis do que problemas solucionáveis. Tal resultado é baseado no cardinal de todos os *problemas solucionáveis*, ou seja, problemas para os quais existe pelo menos um algoritmo (uma *Máquina de Turing*) capaz de solucioná-lo. De fato, o conjunto de todos os problemas solucionáveis está diretamente relacionado com o conceito de *discreto* (em oposição ao termo *contínuo*), justificando a denominação *Matemática Discreta*. Embora a noção intuitiva e o modelo formal da *Máquina de Turing* tenham sido apresentados em uma seção de leitura complementar, sua leitura é fortemente recomendada.

7.1 Cardinalidade Finita e Infinita

Cardinalidade é definida usando funções bijetoras. É interessante observar que o uso da bijeção é intuitivo e comum em diferentes civilizações e em todas as épocas. Por exemplo:

- para definir a cardinalidade de conjuntos, era usual fazer uma bijeção entre o conjunto de objetos em questão (por exemplo, um pequeno rebanho de ovelhas) com um subconjunto de dedos das mãos;
- quando os dedos da mão não eram suficientes para expressar uma cardinalidade maior, era usado, por exemplo, um conjunto de pedras ou de nós em cordas para estabelecer a bijeção;
- observe que se trata de uma bijeção qualquer, pois não existe uma definição de qual dedo (pedra ou nó) corresponde a qual objeto. Ou seja, a preocupação era saber se *existe* uma bijeção;
- claramente, se sobra (respectivamente, falta) um dedo (ou pedra ou nó), então falta (respectivamente, sobra) um objeto.

Definição 7.1 - Cardinalidade Finita, Cardinalidade Infinita
A *Cardinalidade* de um conjunto A, representada por:

$$\#A$$

é como segue:

a) *Finita* se existe uma bijeção entre A e o conjunto $\{1, 2, 3, \ldots, n\}$, para algum $n \in \mathbf{N}$. Nesse caso, afirma-se que (como fica o caso em que $n = 0$?):

$$\#A = n$$

b) *Infinita* se existe uma bijeção entre A e um subconjunto próprio de A. ❑

Portanto, um conjunto A é um:

- *conjunto finito* (ou seja, possui uma cardinalidade finita) se for possível representá-lo por extensão, como já introduzido;
- *conjunto infinito* se for possível retirar alguns elementos de A e, mesmo assim, estabelecer uma bijeção com A.

EXEMPLO 7.1 - Cardinalidade Infinita do Conjunto dos Números Inteiros
A função $f: \mathbf{Z} \to \mathbf{N}$ tal que, para qualquer $a \in \mathbf{Z}$:

se $a \geq 0$, então $f(a) = 2a$ — não-negativos são associados aos pares
se $a < 0$, então $f(a) = |2a| - 1$ — negativos são associados aos ímpares

onde $|2a|$ é o módulo (valor absoluto) de $2a$, é bijetora (por quê?). Claramente, $\mathbf{N}$ é subconjunto próprio de $\mathbf{Z}$. Logo, $\mathbf{Z}$ é infinito. ❑

7.2 Conjunto Contável e Não-Contável

É importante destacar que nem todos os conjuntos infinitos possuem a mesma cardinalidade, o que contradiz a noção intuitiva da maioria das pessoas.

Definição 7.2 - Conjunto Contável, Conjunto Não-Contável, Conjunto Enumerável
Um conjunto A é dito:

a) *Finitamente Contável* se for finito;
b) *Infinitamente Contável* ou *Enumerável* se existe uma bijeção entre o conjunto A e um subconjunto infinito de $\mathbf{N}$. Neste caso, a bijeção é denominada *enumeração* de A;
c) *Não-Contável*, caso contrário. ❑

O termo *conjunto contável* usualmente significa finitamente ou infinitamente contável. Observe que um conjunto é infinitamente contável se for possível enumerar seus elementos como uma seqüência infinita:

$$\langle a_1, a_2, a_3, \ldots \rangle$$

EXEMPLO 7.2 - Conjunto Contável
Os seguintes conjuntos são (infinitamente) contáveis ou enumeráveis:

Z (ver a função bijetora do EXEMPLO 7.1 - Cardinalidade Infinita do Conjunto dos Números Inteiros);
Q (prova sugerida como exercício). ❑

A prova de que um conjunto é não-contável pode ser realizada usando o método da *Diagonalização de Cantor* proposta por Georg Cantor (1845-1918). Esse método é freqüentemente aplicado em provas no contexto da Computação e Informática.

Teorema 7.3 - Conjunto Não-Contável
O seguinte conjunto é não-contável:

$$S = \{x \in \mathbf{R} \mid 0 < x < 1\}$$

Prova: (*por absurdo - Diagonalização de Cantor*)

Suponha que S é contável. Então, existe uma enumeração de S e, portanto, seus elementos podem ser enumerados em uma seqüência infinita como segue:

$$\langle s_1, s_2, s_3, \ldots \rangle$$

Claramente, qualquer número $s \in S$ pode ser representado como uma seqüência infinita de decimais, ou seja, dada por uma seqüência infinitamente contável de dígitos como segue:

$$s = 0{,}d_1\, d_2\, d_2 \ldots d_n \ldots$$

Por exemplo, $\pi / 10 = 0{,}31415\ldots$ Portanto, os componentes de $\langle s_1, s_2, s_3, \ldots \rangle$ podem ser representados da seguinte forma (observe a diagonal destacada):

$$s_1 = 0{,}\mathbf{d_{1\,1}}\, d_{1\,2}\, d_{1\,3} \ldots d_{1\,n} \ldots$$
$$s_2 = 0{,}d_{2\,1}\, \mathbf{d_{2\,2}}\, d_{2\,3} \ldots d_{2\,n} \ldots$$
$$s_3 = 0{,}d_{3\,1}\, d_{3\,2}\, \mathbf{d_{3\,3}} \ldots d_{3\,n} \ldots$$
$$\ldots$$
$$s_n = 0{,}d_{n\,1}\, d_{n\,2}\, d_{n\,3} \ldots \mathbf{d_{n\,n}} \ldots$$
$$\ldots$$

Seja r um número real construído como segue:

$$r = 0{,}e_1\, e_2\, e_3 \ldots e_n \ldots$$

sendo que, para $i \in \{1, 2, 3, \ldots\}$, a componente e_i é construída a partir da diagonal como segue:

$e_i = 1$, caso $d_{ii} \neq 1$
$e_i = 2$, caso $d_{ii} = 1$

Claramente, $r \in S$. Entretanto, r é diferente de qualquer número de $\langle s_1, s_2, s_3, \ldots \rangle$ pois difere pelo menos no dígito destacado na diagonal. Portanto, $r \notin S$, o que é uma contradição!

Logo, é um absurdo supor que S é contável. Portanto, S é não-contável. ❑

Neste momento, a prova de que o conjunto dos números reais é não-contável é simples. Antes considere a seguinte definição.

Definição 7.4 - Conjuntos Equipotentes
Dois conjuntos A e B são ditos *Equipotentes* quando existe uma função bijetora entre A e B. ❑

Logo, conjuntos equipotentes são aqueles que têm a mesma cardinalidade. Portanto:

- todos conjuntos enumeráveis são equipotentes;
- todo conjunto indexado é equipotente ao seu conjunto de índices.

Teorema 7.5 - R é um Conjunto Não-Contável
O conjunto dos números reais **R** é não-contável.

Prova: (*direta*)
Basta provar que **R** é equipotente a $S = \{ x \in \mathbf{R} \mid 0 < x < 1 \}$. De fato, seja $f: S \to \mathbf{R}$ tal que:

se $0 < s \leq 1/2$, então $f(s) = (1/2s) - 1$
se $1/2 \leq s < 1$, então $f(s) = (1/(2s-2)) + 1$

a qual é uma bijeção (tal verificação é sugerida como exercício). ❑

EXEMPLO 7.3 - Conjunto Não-Contável
Os seguintes conjuntos são não-contáveis:

I (conjuntos dos números irracionais);
C (conjunto dos números complexos);
$\mathbf{R}^2$ ❑

7.3 Cardinalidade dos Conjuntos Não-Contáveis

O que dizer sobre a cardinalidade dos conjuntos não-contáveis? Terão todos a mesma "quantidade" de elementos? A resposta é *não*, ou seja:

nem todos os conjuntos não-contáveis têm a mesma cardinalidade

Estendendo o conceito de contagem ao infinito, pode-se dizer que um conjunto A tem, pelo menos, tantos elementos quanto um conjunto B, ou seja, que:

$$\#A \leq \#B$$

quando existe uma função injetora $f: A \rightarrowtail B$. Para que a relação $\leq$ entre as cardinalidades seja uma relação de ordem parcial, ela deve ser:

- *reflexiva*, ou seja, $\#A \leq \#A$. De fato, basta considerar a função identidade $id_A: A \to A$, a qual é injetora;
- *transitiva*, ou seja, se $\#A \leq \#B$ e $\#B \leq \#C$, então $\#A \leq \#C$. De fato, já foi visto que a composição de funções injetoras é uma função injetora;
- *anti-simétrica*, ou seja, se $\#A \leq \#B$ e $\#B \leq \#A$, então $\#A = \#B$. Isso é garantido pelo *Teorema de Schröder-Bernstein*, enunciado a seguir, o qual não será demonstrado.

Teorema 7.6 - Schröder-Bernstein
Sejam A e B dois conjuntos tais que existem duas funções injetoras:

$$f_1: A \rightarrowtail B \quad \text{e} \quad f_2: B \rightarrowtail A$$

Então existe uma função bijetora:

$$g: A \leftrightarrow B$$ ❑

Um resultado que garante que existem cardinais não-contáveis em quantidade infinita é o fato de que o conjunto das partes de um conjunto tem sempre cardinalidade maior que este. Esse fato é conhecido como *Teorema de Cantor*, apresentado a seguir. Na demonstração, lembre que:

$$n < m \Leftrightarrow n \leq m \wedge n \neq m$$

Teorema 7.7 - Cantor
Seja C um conjunto e 2^C o conjunto das partes de C. Então:

$$\#C < \#2^C$$

Prova:
A prova é dividida em duas partes:

- para mostrar que $\#C \leq \#2^C$, basta apresentar uma função injetora $f: C \rightarrowtail 2^C$
- para mostra que $\#C \neq \#2^C$, basta mostrar que *não* existe uma função bijetora $g: C \leftrightarrow 2^C$

Parte 1: (*direta*)
Seja $f: C \to 2^C$ uma função tal que, para todo $s \in C$, vale:

$$f(s) = \{ s \}$$

Claramente f é injetora (por quê?). Portanto, $\#C \leq \#2^C$.

Parte 2: (*por absurdo*)
Suponha que existe uma função bijetora $g: C \leftrightarrow 2^C$. Seja o seguinte subconjunto A de C:

$$A = \{ a \in C \mid a \notin g(a) \}$$

Como g é uma função bijetora, em particular, é uma função sobrejetora. Portanto existe $c \in C$ tal que $g(c) = A$. Assim, existem dois casos:

Caso 1: $c \in A$

$c \in A \Rightarrow$	$g(c) = A$
$c \in g(c) \Rightarrow$	pela definição de A
$c \notin A$	

Caso 2: $c \notin A$

$c \notin A \Rightarrow$	$g(c) = A$
$c \notin g(c) \Rightarrow$	pela definição de A
$c \in A$	

o que é uma *contradição* (um elemento não pode pertencer e não pertencer a um conjunto ao mesmo tempo)! Ou seja, é absurdo supor que existe tal função bijetora. Logo, *não* existe uma função bijetora entre C e 2^C. Portanto, $\#C \neq \#2^C$. ❑

A partir desse resultado, pode-se pensar em uma classe infinita de números cardinais.

Definição 7.8 - Cardinal
Um *Cardinal* é uma classe de equivalência de conjuntos equipotentes. ❑

Dessa forma, a classe dos cardinais é a classe de todas as classes de equivalência dos conjuntos equipotentes. Observe a semelhança com o Paradoxo de Russell se a classe de todos os cardinais for tomada como um cardinal. Portanto, tal classe *não* é um conjunto.

Em geral utiliza-se a primeira letra do alfabeto hebraico $\aleph$ (lê-se "alef") com índices para indicar cardinais infinitos conhecidos. De especial interesse para Ciência da Computação é o cardinal do conjunto dos números naturais **N**, denotado por $\aleph_0$. Assim, o cardinal de qualquer conjunto contável (infinito) é $\aleph_0$. De fato:

$\aleph_0$ é o menor cardinal dos conjuntos infinitos

Define-se $\aleph_{i+1}$ como sendo o menor cardinal maior que $\aleph_i$. Prova-se que o conjunto das partes de **N** é equipotente ao conjunto **R**. Considerando que 2^k denota o cardinal do conjunto das partes de conjuntos com cardinalidade k, pode-se afirmar que $2^{\aleph_0}$ é a cardinalidade do conjunto dos números reais, ou seja, é a cardinalidade do *continuum*.

Entretanto, resta a questão se $\aleph_{i+1} = 2^{\aleph_i}$. Essa asserção é conhecida como *Hipótese do Continuum* e foi formulada como hipótese por Cantor uma vez que ele não conseguiu obter sua prova a partir dos axiomas que estabeleceu para a sua *Teoria dos Conjuntos*. Atualmente, primeiro devido a Gödel (década de 1930) e depois a Cohen (década de 1950), sabe-se que esse fato é independente da Teoria dos Conjuntos, ou seja, tanto a Hipótese do *Continuum*, quanto sua negação são consistentes com o restante da Teoria dos Conjuntos, não gerando nenhuma contradição à adição de cada uma em separado à mesma.

7.4 Cardinal do Conjunto de Todos os Problemas Solucionáveis

O estudo dos cardinais é de fundamental importância em Computação e Informática, com especial interesse no estudo dos problemas que podem ser resolvidos usando um sistema computador. De fato, prova-se que existem $\aleph_0$ programas que podem ser definidos em qualquer linguagem de programação de propósitos gerais (como a linguagem Pascal ou a linguagem C). Como, por exemplo, existem $2^{\aleph_0}$ funções de **N** para **N**, conclui-se que existem infinitas funções que não podem ser representadas algoritmicamente, ou seja, que não são computáveis (em um sistema computador). Portanto, pode-se afirmar que:

- existem infinitos, mas contáveis, *problemas solucionáveis*, ou seja, problemas para os quais existe pelo menos um algoritmo (uma *Máquina de Turing*) capaz de solucioná-lo;
- existem infinitos, e não-contáveis, *problemas não-solucionáveis*, ou seja, para os quais não existe qualquer algoritmo (programa) capaz de solucioná-los.

Adicionalmente, como existe uma função (injetora) de inclusão de **N** para **R**, e não existe função injetora de **R** para **N**, pode-se afirmar que $\#\mathbf{N} < \#\mathbf{R}$. Portanto, o cardinal dos problemas não-solucionáveis é maior que o cardinal dos problemas solucionáveis.

7.5 Leitura Complementar: Máquina de Turing

Quando da introdução do Autômato Finito no Capítulo 6 - Funções Parciais e Totais, foi destacada a limitação da capacidade de solucionar problemas deste modelo. Como ilustração, prova-se que não existe autômato finito capaz de reconhecer construções tão simples como palíndromos de qualquer tamanho, qualquer número de parênteses aninhados ou balanceados (como em uma expressão aritmética), etc.

No Capítulo 2 - Lógica e Técnicas de Demonstração, foi dito que a Máquina de Turing é aceita como uma formalização do conceito de algoritmo computável, e conseqüentemente, do que é possível solucionar em um sistema computador. No Capítulo 3 - Álgebra de Conjuntos, algumas questões relacionadas com a capacidade de solucionar problemas foram discutidas.

Entretanto, o conceito de Máquina de Turing não foi introduzido. Adicionalmente, a seguinte pergunta destaca-se: qual a diferença fundamental entre os dois modelos, de tal forma que o poder computacional da Máquina de Turing seja tão maior que o do Autômato Finito?

A resposta a essa questão ilustra o porquê de a *Matemática Discreta* possuir como ênfase os estudos matemáticos baseados em conjuntos contáveis, *finitos* ou *infinitos* (conceito apresentado no Capítulo 1 - Introdução e Conceitos Básicos). De fato:

a) A definição de *Autômato Finito*, como o próprio nome indica, é baseada na noção de estados *finitos* e *predefinidos* que o sistema pode assumir. Portanto, é baseada na noção de conjunto *finitamente contável*;
b) Em contrapartida, a definição de *Máquina de Turing* é baseada na noção de estados possíveis finitos, mas *não-predefinidos*, o que implica uma noção de "tão grande quanto necessário". Portanto, é baseada na noção de conjunto *infinitamente contável*.

7.5.1 Noção Intuitiva da Máquina de Turing

A *Máquina de Turing*, proposta por Alan Turing em 1936, é um mecanismo simples que formaliza a idéia de uma pessoa que realiza cálculos. Lembra, em muito, os computadores atuais, embora tenha sido proposta anos antes do primeiro computador digital. Apesar de sua simplicidade, o modelo Máquina de Turing possui, no mínimo, o mesmo poder computacional de qualquer computador de propósito geral.

O ponto de partida de Turing foi analisar a situação na qual uma pessoa, equipada com um instrumento de escrita e um apagador, realiza cálculos em uma folha de papel, organizada em quadrados.

Inicialmente, suponha que a folha de papel contém somente os dados iniciais do problema. O trabalho da pessoa pode ser resumido em seqüências de operações simples como segue:

- ler um símbolo de um quadrado;
- alterar um símbolo em um quadrado;
- mover os olhos para outro quadrado.

Quando é encontrada alguma representação satisfatória para a resposta desejada, a pessoa termina seus cálculos. Para viabilizar esse procedimento, as seguintes hipóteses são aceitáveis:

- a natureza bidimensional do papel não é um requerimento essencial para os cálculos. Pode ser assumido que o papel consiste de uma fita infinita organizada em quadrados;
- o conjunto de símbolos pode ser finito, pois se pode utilizar seqüências de símbolos;
- o conjunto de estados da mente da pessoa durante o processo de cálculo é finito. Mais ainda, entre esses estados, existem dois em particular: "estado inicial" e "estado final", correspondendo ao início e ao fim dos cálculos, respectivamente;
- o comportamento da pessoa, a cada momento, é determinado somente pelo seu estado presente e pelo símbolo para o qual sua atenção está voltada;
- a pessoa é capaz de observar e alterar o símbolo de apenas um quadrado de cada vez, bem como de transferir sua atenção somente para um dos quadrados adjacentes.

7.5.2 Modelo e Exemplo

Essa noção de uma pessoa calculando pode ser vista como uma máquina, constituída de três partes, como segue:

a) *Fita*. Usada simultaneamente como dispositivo de entrada, de saída e de memória de trabalho;
b) *Unidade de Controle*. Reflete o estado corrente da máquina. Possui uma unidade de leitura e gravação (cabeça da fita), a qual acessa uma célula da fita de cada vez e movimenta-se para a esquerda ou para a direita;
c) *Programa* ou *Função de Transição*. Função que define o estado da máquina e comanda as leituras, as gravações e o sentido de movimento da cabeça.

A fita é finita à esquerda e *infinita* (tão grande quanto necessário) à direita, sendo dividida em células, cada uma armazenando um símbolo. Os símbolos podem pertencer ao alfabeto de entrada, ao alfabeto auxiliar ou ainda ser "branco" ou "marcador de início de fita". Inicialmente, a palavra a ser processada (ou seja, a informação de entrada para a máquina) ocupa as células mais à esquerda após o marcador de início de fita, ficando as demais com "branco", como ilustrado na Figura 7.1, onde β e ◎ representam "branco" e "marcador de início de fita", respectivamente.

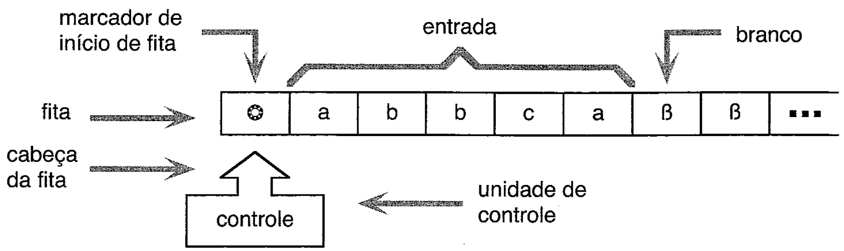

Figura 7.1 Fita e unidade de controle de uma Máquina de Turing

A unidade de controle possui um número *finito* e *predefinido* de estados. A *cabeça da fita* lê o símbolo de uma célula de cada vez e grava um novo símbolo. Após a leitura/gravação (a gravação é realizada na mesma célula de leitura), a cabeça move uma célula para a direita ou para a esquerda. O símbolo gravado e o sentido do movimento são definidos pelo programa.

O programa é uma função que, dependendo do estado corrente da máquina e do símbolo lido, determina o símbolo a ser gravado, o sentido do movimento da cabeça e o novo estado.

EXEMPLO 7.4 - Máquina de Turing - Duplo Balanceamento

Considere a Máquina de Turing ilustrada na Figura 7.2, definida sobre o alfabeto $\Sigma = \{a, b\}$ e tendo como alfabeto auxiliar (de trabalho) $\{A, B\}$, na qual:

- nodos representam *estados* da máquina, os quais são em número *finito*;
- arcos representam *transições* ou *computações atômicas*, as quais são em número *finito*. O conjunto de todas as transições constitui o programa da Máquina de Turing;
- o estado q_0 é dito *estado inicial* e é representado de forma diferenciada (no caso, o nodo destino de uma seta sem origem);
- o estado q_4 é dito *estado final* e é representado de forma diferenciada (no caso, o traço da circunferência do nodo é mais forte). ❑

A função programa considera o estado corrente e o símbolo lido da fita para determinar o novo estado, o símbolo a ser gravado e o sentido de movimento da cabeça, onde esquerda e direita são representados por E e D, respectivamente. A interpretação como um grafo é ilustrada na Figura 7.3.

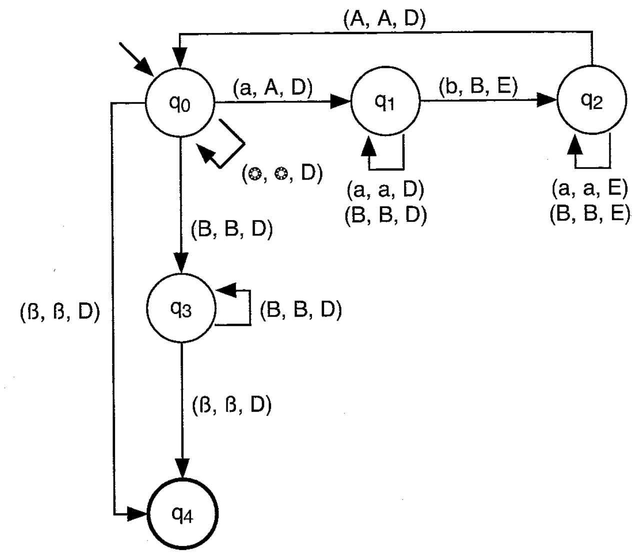

Figura 7.2 Máquina de Turing

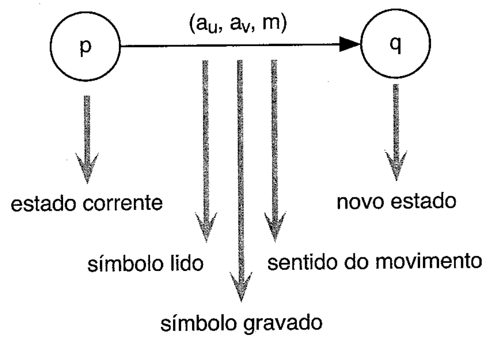

Figura 7.3 Interpretação da função programa como um grafo

O processamento de uma Máquina de Turing, para uma palavra de entrada w armazenada na fita, consiste na sucessiva aplicação da função programa a partir do estado inicial q_0 e da cabeça posicionada na célula mais à esquerda da fita até ocorrer uma condição de parada. O processamento de M para a entrada w pode parar ou ficar em *loop* infinito. A parada pode ser de duas maneiras: aceitando ou rejeitando a entrada w. As condições de parada são as seguintes:

a) A máquina assume um estado final: a máquina pára e a palavra de entrada é aceita;
b) A função programa é indefinida para o argumento (símbolo lido e estado corrente): a máquina pára e a palavra de entrada é rejeitada;
c) O argumento corrente da função programa define um movimento à esquerda e a cabeça da fita já se encontra na célula mais à esquerda: a máquina pára e a palavra de entrada é rejeitada.

A Máquina de Turing ilustrada na Figura 7.2 aceita a seguinte linguagem definida sobre o alfabeto $\Sigma = \{ a, b \}$, conhecida como "duplo balanceamento":

$$L_1 = \{ \varepsilon, ab, aabb, aaabbb, \dots \}$$

Essa linguagem é um exemplo clássico e de fundamental importância no estudo das linguagens, pois permite estabelecer uma analogia com linguagens que possuem duplo balanceamento em sua estrutura como, por exemplo:

a) *Linguagens Bloco-Estruturadas*, como a linguagem de programação Pascal, na qual cada bloco é determinado pelo trecho de programa contido entre as palavras `begin` e `end`;

b) Linguagens com parênteses balanceados, ou seja, um número qualquer de parênteses em um texto, eventualmente encadeados (aninhados), de tal forma a garantir que, para cada parêntese aberto (respectivamente, fechado), existe um correspondente parêntese fechado (respectivamente, aberto), como as expressões aritméticas, presentes na maioria das linguagens de programação.

Observe que, analogamente ao autômato finito, o programa de uma Máquina de Turing pode ser definido como uma função parcial. De fato, supondo um conjunto finito de estados Q e um conjunto de símbolos S (incluindo o alfabeto Σ, os símbolos especiais β e ⊙ e o alfabeto auxiliar), a *função programa* pode ser definida como segue:

$$\delta: Q \times S \to Q \times S \times \{ E, D \}$$

Para a Máquina de Turing da Figura 7.2, a correspondente função programa é representada na forma de matriz na Figura 7.4, na qual cada estado (linha) e símbolo lido (coluna) determina um novo estado, o símbolo gravado e movimento da cabeça (célula da matriz).

δ	⊙	a	b	A	B	β
q_0	(q_0, ⊙, D)	(q_1, A, D)			(q_3, B, D)	(q_4, β, D)
q_1		(q_1, a, D)	(q_2, B, E)		(q_1, B, D)	
q_2		(q_2, a, E)		(q_0, A, D)	(q_2, B, E)	
q_3					(q_3, B, D)	(q_4, β, D)
q_4						

Figura 7.4 Máquina de Turing - função programa na forma de matriz

O algoritmo apresentado reconhece o primeiro símbolo a, o qual é marcado como A, e movimenta a cabeça da fita à direita, procurando o b correspondente, o qual é marcado como B. Esse ciclo é repetido sucessivamente até identificar, para cada a, o seu correspondente b. Adicionalmente, o algoritmo garante que qualquer outra palavra que não esteja na forma de um duplo balanceamento é rejeitada. A Figura 7.5 ilustra a seqüência do processamento da Máquina de Turing em questão para a entrada $w = aabb$.

Observação 7.9 - Máquina de Turing × Algoritmo

Ao longo de todo o livro, tem sido afirmado que a *Máquina de Turing* é aceita como uma formalização do conceito de *algoritmo*. Entretanto, também é usual considerar que o conceito de algoritmo corresponde a uma Máquina de Turing que sempre pára para qualquer entrada. Nesse caso, uma máquina que eventualmente fica processando indefinidamente (em *loop* infinito) não seria considerada um algoritmo. ❑

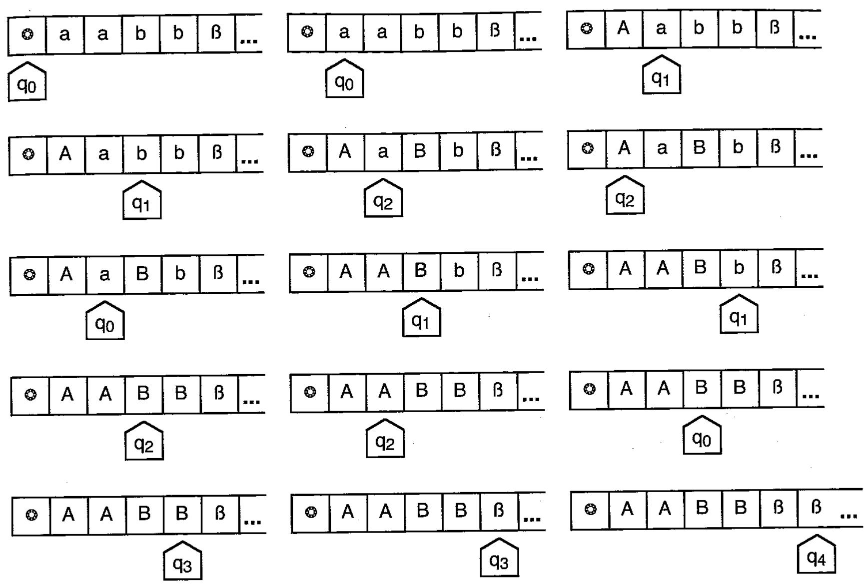

Figura 7.5 Seqüência de processamento de uma Máquina de Turing

7.5.3 Cardinal do Conjunto de Todas as Máquinas de Turing

Já foi afirmado que existem infinitos, mas contáveis, problemas solucionáveis, ou seja, problemas para os quais existe pelo menos uma Máquina de Turing capaz de solucioná-lo. De fato, o conjunto de todas as Máquinas de Turing Δ é equipotente ao conjunto dos números naturais. A prova pode ser facilmente entendida como segue (o detalhamento é sugerido como exercício e pode ser encontrado na maioria dos livros de Teoria da Computação):

- dado que uma função programa de uma máquina de Turing é tal que
$$\delta \subseteq (Q \times S) \times (Q \times S \times \{ E, D \})$$
- então, cada par de δ possui como primeira componente um par (estado e símbolo) e como segunda componente uma terna (estado, símbolo e sentido do movimento);
- portanto, cada par de δ contém um total de cinco informações, todas com possibilidades finitas de valores;
- logo, supondo que o cardinal de δ é n (uma função programa sempre é finita), 5n componentes definem toda a função programa;
- pelo *Teorema Fundamental da Aritmética*, sabe-se que cada número natural é univocamente decomposto em seus fatores primos;
- assim, as 5n componentes de δ podem ser codificadas univocamente como um número natural. Como, para diferentes funções programas, tem-se diferentes números naturais, essa construção caracteriza uma função injetora de Δ para **N**.

Portanto, $\#\Delta \leq \#\mathbf{N}$. Como $\#\Delta$ é infinito, e sabendo-se que $\#\mathbf{N} = \aleph_0$ é o menor cardinal dos conjuntos infinitos, então Δ e **N** são conjuntos equipotentes.

7.6 Exercícios

Exercício 7.1 Afirma-se que a cardinalidade de um conjunto é finita, se existe uma bijeção entre A e o conjunto $N = \{1, 2, 3, \ldots, n\}$, para algum $n \in \mathbf{N}$. Supondo $n = 0$:

a) Qual é o conjunto A?

b) Qual é a correspondente função bijetora?

Exercício 7.2 Prove que, de fato, a seguinte função é bijetora:

$f: \mathbf{Z} \to \mathbf{N}$ tal que:

se $a \geq 0$, então $f(a) = 2a$ — positivos são associados aos pares

se $a < 0$, então $f(a) = |2a| - 1$ — negativos são associados aos ímpares

Exercício 7.3 Prove que **R** é um conjunto infinito, apresentando uma bijeção com um subconjunto próprio de **R**.

Exercício 7.4 Prove que **Q** é contável.

Sugestão: considere **Q** como sendo um conjunto de pares e, para construir uma função injetora de **Q** para **N**, lembre-se de que a decomposição de um número em seus fatores primos é única.

Exercício 7.5 Prove que a função $f: S \to \mathbf{R}$ apresentada na prova do Teorema 7.5 - **R** é um Conjunto Não-Contável é de fato uma bijeção.

Dica: na investigação se f possui inversa, considere a seguinte função $g: \mathbf{R} \to S$ tal que:

se $x \geq 0$, então $g(x) = 1 / (2x + 2)$

se $x < 0$, então $g(x) = (1 / (2x - 2)) + 1$

Exercício 7.6 Prove que:

a) Nem sempre a intersecção de conjuntos não-contáveis é não-contável;

b) Nem sempre a diferença de conjuntos não-contáveis é não-contável.

Exercício 7.7 Prove que:

a) A união de conjuntos contáveis é contável;

b) O produto cartesiano de conjuntos contáveis é contável.

Exercício 7.8 Prove que, se $A \subseteq B$ e A é infinito, então B é infinito.

Exercício 7.9 Lembre-se de que, se existe uma função injetora $f: A \rightarrowtail B$, então $\#A \leq \#B$. Sabendo que **R** é não-contável, prove que são conjuntos não-contáveis:

a) $\mathbf{R}^2$

b) $\mathbf{N} \times \mathbf{R}$

c) **C** (conjunto dos números complexos)

Exercício 7.10 Suponha o alfabeto $\Sigma = \{a, b\}$. Desenvolva Máquinas de Turing, determinísticas ou não, que aceitem as seguintes linguagens:

a) $\{\varepsilon, a, b\}$

b) $\{w \in \Sigma^* \mid$ w tem o mesmo número de símbolos a e b$\}$

c) $\{w \in \Sigma^* \mid$ o décimo símbolo da direita para a esquerda de w é a$\}$

d) $\{waw \mid w \in \Sigma^*\}$

Exercício 7.11 Detalhe a prova de que o conjunto de todas as Máquinas de Turing é equipotente ao conjunto dos números naturais.

8 Indução e Recursão

8.1 Princípio da Indução Matemática

O *Princípio da Indução Matemática* é uma técnica para lidar com tipos de dados que têm uma *relação de boa-ordem*, isto é, uma relação onde todo subconjunto não-vazio do tipo de dado tem um elemento mínimo segundo essa relação de ordem. Um exemplo típico é o conjunto dos números naturais (na realidade um tipo de dado). Dada uma boa ordem, pode-se aplicar indução para provar propriedades que valem para todo elemento do tipo de dado.

Por simplicidade, o tipo de dados que tem uma relação de boa-ordem considerado é o conjunto dos números naturais **N** (ou qualquer outro conjunto isomorfo a **N**).

Um exemplo simples que ilustra o Princípio da Indução Matemática é o *efeito dominó* (veja a Figura 8.1): uma fila sem fim de peças do jogo dominó para a qual, ao derrubar a primeira peça, todas as demais peças são derrubadas em cadeia. Para estar certo de que tal fato ocorre, suponha que são verdadeiras as seguintes proposições:

a) a primeira peça é derrubada na direção das demais;

b) se qualquer peça está suficientemente próxima da seguinte na fila, então, ao ser derrubada, fará com que a sua vizinha seguinte também seja derrubada.

Então:

- pelo item a), a primeira peça é derrubada;
- pelo item b), a segunda peça é derrubada;
- pelo item b), a terceira peça é derrubada;
- pelo item b), a quarta peça é derrubada;
- e assim sucessivamente.

Portanto, para i tão grande quanto se queira, pode-se afirmar que a i-ésima peça é derrubada. Logo, para qualquer n, pode-se afirmar que a n-ésima peça é derrubada.

O Princípio da Indução Matemática pode ser resumido como segue:

se o início é correto e se coisa alguma pode dar errada,
então sempre será correto