mac.1 Introdução

tur:mac:int:sec

O que significa dizer que uma função, digamos, de $\mathbb{N}$ para $\mathbb{N}$ é *computável*? Entre as primeiras respostas, e a mais conhecida, está a de que uma função é computável se puder ser computada por uma máquina de Turing. Essa noção foi apresentada por Alan Turing em 1936. As máquinas de Turing são um exemplo de *modelo de computação*—uma maneira matematicamente precisa de definir a ideia de um "procedimento computacional." O que exatamente isso significa é debatido, mas há amplo consenso de que as máquinas de Turing são uma forma de especificar procedimentos computacionais. Embora o termo "máquina de Turing" evoque a imagem de uma máquina física com partes móveis, estritamente falando uma máquina de Turing é um construto puramente matemático e, como tal, idealiza a ideia de um procedimento computacional. Por exemplo, não impomos restrição alguma aos requisitos de tempo ou memória de uma máquina de Turing: máquinas de Turing podem computar algo mesmo que a computação exija mais espaço de armazenamento ou mais passos do que há átomos no universo.

explanation

Talvez seja melhor pensar em uma máquina de Turing como um programa para um tipo especial de mecanismo imaginário. Esse mecanismo consiste em uma *fita* e uma *cabeça de leitura-escrita*. Na nossa versão das máquinas de Turing, a fita é infinita em uma direção (à direita) e é dividida em *quadrados*, cada um dos quais pode conter um símbolo de um *alfabeto* finito. Esses alfabetos podem conter qualquer número de símbolos diferentes, mas nos contentaremos em geral com três: $\triangleright$, 0 e 1. Quando o mecanismo é iniciado, a fita está vazia (ou seja, cada quadrado contém o símbolo 0), exceto o quadrado mais à esquerda, que contém $\triangleright$, e um número finito de quadrados que contêm a *entrada*. A qualquer momento, o mecanismo está em um de um número finito de *estados*. No início, a cabeça examina o quadrado mais à esquerda e em um *estado inicial* especificado. A cada passo da execução do mecanismo, o conteúdo do quadrado atualmente examinado, junto com o estado em que o mecanismo se encontra e o programa da máquina de Turing, determinam o que acontece em seguida. O programa da máquina de Turing é dado por uma função parcial que recebe como entrada um estado q e um símbolo σ e produz uma tripla $\langle q', \sigma', D\rangle$. Sempre que o mecanismo está no estado q e lê o símbolo σ, substitui o símbolo no quadrado atual por σ', a cabeça move-se à esquerda, à direita ou permanece no lugar conforme D seja L, R ou N, e o mecanismo passa ao estado q'.

Por exemplo, considere a situação em Figura 1. A parte visível da fita da máquina de Turing contém o símbolo de fim de fita $\triangleright$ no quadrado mais à esquerda, seguido de três 1's, um 0 e mais quatro 1's. A cabeça está lendo o terceiro quadrado da esquerda, que contém um 1, e está no estado q_1—dizemos que "a máquina está lendo um 1 no estado q_1." Se o programa da máquina de Turing retornar, para a entrada $\langle q_1, 1\rangle$, a tripla $\langle q_2, 0, N\rangle$, a máquina substituiria o 1 no terceiro quadrado por um 0, deixaria a cabeça de leitura/escrita onde está e passaria ao estado q_2. Se então o programa retornar $\langle q_3, 0, R\rangle$ para a entrada $\langle q_2, 0\rangle$, a máquina sobrescreveria o 0 com outro 0 (efetivamente, dei-

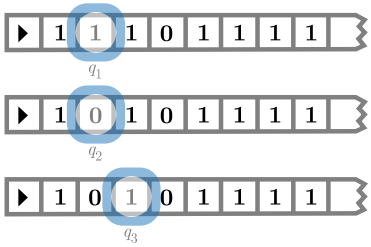

Figura 1: Uma máquina de Turing executando seu programa.

tur:mac:int:fig:tm

xando inalterado o conteúdo da fita sob a cabeça de leitura/escrita), moveria um quadrado à direita e entraria no estado q_3. E assim por diante.

Dizemos que a máquina *para* quando encontra algum estado, q_n, e símbolo, σ, tais que não há instrução para $\langle q_n, \sigma \rangle$, ou seja, a função de transição para a entrada $\langle q_n, \sigma \rangle$ é indefinida. Em outras palavras, a máquina não tem instrução a executar e, nesse ponto, cessa a operação. A parada às vezes é representada por um estado de parada específico h. Isso será demonstrado com mais detalhes adiante.

digression

A beleza do artigo de Turing, "On computable numbers," está em que ele apresenta não só uma definição formal, mas também um argumento de que a definição captura a noção intuitiva de computabilidade. Pela definição, deve ficar claro que toda função computável por uma máquina de Turing é computável no sentido intuitivo. Turing oferece três tipos de argumento de que o inverso é verdadeiro, ou seja, que toda função que naturalmente consideraríamos computável é computável por tal máquina. São eles (nas palavras de Turing):

1. Um apelo direto à intuição.

2. Uma prova da equivalência de duas definições (caso a nova definição tenha maior apelo intuitivo).

3. Dar exemplos de grandes classes de números que são computáveis.

Nosso objetivo é tentar definir a noção de computabilidade "em princípio," ou seja, sem levar em conta limitações práticas de tempo e espaço. Claro, com a definição mais ampla de computabilidade em vigor, pode-se então considerar computação com recursos limitados; isso forma o cerne do assunto conhecido como "complexidade computacional."

Observações históricas Alan Turing inventou as máquinas de Turing em 1936. Embora seu interesse na época fosse a decidibilidade da lógica de primeira ordem, o artigo tem sido descrito como um trabalho definitivo sobre os fundamentos do projeto de computadores. No artigo, Turing concentra-se em números reais computáveis, ou seja, números reais cujas expansões decimais são

computáveis; mas observa que não é difícil adaptar suas noções a funções computáveis sobre os números naturais, e assim por diante. Note que isso foi cinco anos inteiros antes de o primeiro computador de propósito geral em funcionamento ser construído em 1941 (pelo alemão Konrad Zuse na sala de estar dos pais), sete anos antes de Turing e seus colegas em Bletchley Park construírem o Colossus que quebrava códigos (1943), nove anos antes do ENIAC americano (1945), doze anos antes do primeiro computador britânico de propósito geral—a Manchester Small-Scale Experimental Machine—ser construída em Manchester (1948), e treze anos antes dos americanos testarem o BINAC (1949). A Manchester SSEM tem a distinção de ser o primeiro computador com programa armazenado—as máquinas anteriores tinham de ser recabeada manualmente para cada nova tarefa.

Créditos das Imagens

Referências Bibliográficas