Definições Sobre Máquina de Turing

Universidade Federal do Cariri

Tony Esaú de Oliveira

Conteúdo

1 Definições Formais sobre Máquina de Turing

1.1 Fita de Turing

Uma *fita* é uma entidade formada por células sequenciais de armazenamento de caracteres.

Definição

Uma *fita* é modelada como uma função $\tau : \mathbb{N} \to \Gamma$, onde $\mathbb{N}$ representa as posições indexadas por naturais e Γ é um alfabeto de caracteres associado a fita, de forma que cada célula $\tau(i)$ para $i \in \mathbb{N}$ armazena um símbolo de Γ. Inicialmente, para $w = w_1 w_2 \cdots w_n \in \Sigma^*$, isto é, uma cadeia de entrada de uma linguagem qualquer no alfabeto Σ, a fita é configurada tal que $\tau(i) = w_i$ para $1 \leq i \leq n$, e $\tau(i) = \sqcup$, para todos os outros $i \in \mathbb{N}$ em que $\sqcup$ representa o caractere vazio, garantindo que apenas um número finito de células contenha símbolos não em branco. Durante a computação, a fita pode ser modificada finitamente em cada passo, mantendo a propriedade de suporte finito, i.e., $|\{i \in \mathbb{N} \mid \tau(i) \neq \sqcup\}| < \infty$.

w_1	w_2	w_3	$\cdots$	w_{n-1}	w_n	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$

$w_i \in \Gamma$,
$\sqcup$ é o símbolo branco, e
$w = w_1 w_2 \cdots w_n \in \Sigma^*$

1.2 Noção Intuitiva da Cabeça de Leitura/Escrita

A *cabeça de leitura* é um apontador de células de uma fita, ou seja, é capaz de guardar a posição, ou endereço, de um caractere do alfabeto Γ armazenado em uma célula de uma fita, permitindo a leitura e modificação desse dado caractere.

				$\downarrow$							
w_1	w_2	$\cdots$	w_{i-1}	w_i	w_{i+1}	$\cdots$	w_n	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$

1.3 Noção de Movimento

Definimos o conjunto de movimentos $\mathcal{M} = \{E, D\}$, em que cada elemento representa uma direção possível para o deslocamento da cabeça de leitura/escrita numa fita f. O símbolo E, ou esquerda, indica que a cabeça deve mover-se uma posição para trás na fita, decrementando sua posição atual i em uma unidade. O símbolo D, ou direita, indica que a cabeça deve mover-se uma posição para frente, incrementando i em uma unidade. Após cada operação de leitura e escrita, uma máquina obrigatoriamente move sua cabeça em uma dessas duas direções, não havendo a opção de permanecer na mesma posição, exceto quando a cabeça estando na posição 1 tenta mover-se à esquerda, ou seja, para uma célula inexistente.

1.4 Definição Formal de Máquina de Turing

Definição

Uma máquina de Turing é uma 7-upla $(Q, \Sigma, \Gamma, \delta, q_0, q_{aceita}, q_{rejeita})$, onde Q, Σ, Γ são todos conjuntos finitos e,

1. Q é um conjunto de estados,

2. Σ é o alfabeto de entrada não contendo o símbolo em branco $\sqcup$,

3. Γ é o alfabeto de fita, onde $\sqcup \in \Gamma$ e $\Sigma \subseteq \Gamma$,
4. $\delta : Q \times \Gamma \to Q \times \Gamma \times \{E, D\}$ é a função de transição,
5. $q_0 \in Q$ é o estado inicial,
6. $q_{aceita} \in Q$ é o estado de aceitação, e
7. $q_{rejeita} \in Q$ é o estado de rejeição, onde $q_{rejeita} \neq q_{aceita}$.

1.5 Configuração de uma Máquina de Turing

Definição

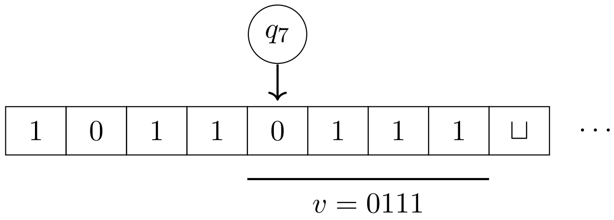
Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{aceita}}, q_{\text{rejeita}})$ uma Máquina de Turing. Uma *configuração* de M é uma tripla $C = (q, \tau, i)$, em que:

1. $q \in Q$ é o estado atual da máquina;
2. $\tau : \mathbb{N} \to \Gamma$ é a função que representa o conteúdo da fita, mapeando cada posição inteira a um símbolo do alfabeto da fita no instante;
3. $i \in \mathbb{N}$ é a posição atual da cabeça de leitura-escrita na fita.

Dessa forma, configuração descreve completamente o estado instantâneo da Máquina de Turing em um determinado momento da computação.

1.5.1 Configuração como uma Cadeia

Configurações são frequentemente representadas de uma maneira especial e compacta. Para um estado $q \in Q$ e duas cadeias $u, v \in \Gamma^*$ sobre o alfabeto de fita Γ, escrevemos uqv para representar a configuração na qual o estado atual é q, o conteúdo atual da fita é uv e a posição atual da cabeça está sobre o primeiro símbolo de v. Por convenção, assumimos que a fita contém apenas brancos ou vazios $\sqcup$ após o último símbolo de v.



Estado: q_7

Configuração: $1011\, q_7\, 0111$

Na figura acima, a cadeia $u = 1011$ representa o conteúdo da fita à esquerda da cabeça de leitura, o estado q_7 indica o estado atual da máquina, e a cadeia $v = 0111$ representa o conteúdo da fita a partir da posição da cabeça inclusivamente. Assim, a notação compacta $1011\, q_7\, 0111$ codifica completamente a configuração $(q_7, \tau, 5)$, onde τ é a função que mapeia as posições da fita aos símbolos $101101111 \sqcup \sqcup \cdots$ e a cabeça está na posição 5.

1.5.2 Configuração Inicial

A configuração inicial consiste na fita com o apontador na posição 1, a entrada escrita nas posições subsequentes, 1 a n, seguida de símbolos em branco infinitamente à direita. A cabeça de leitura/escrita inicia tipicamente na posição 1 , e o estado é q_0, ou seja, $C = q_0 w$, sendo w a cadeia de entrada.

1.5.3 Originação de Configuração

Definição

Sejam $C = (q, \tau, i)$ e $C' = (q', \tau', i')$ duas configurações de uma máquina de Turing M. Dizemos que C *origina* ou *produz* C' em um passo, denotado por $C \vdash_M C'$, se existe uma transição $\delta(q, \tau(i)) = (q', w', d)$ tal que:

1. $\forall k \in \mathbb{N}$, $\tau'(k) = \begin{cases} w', & \text{se } k = i \\ \tau(k), & \text{caso contrário} \end{cases}$;
2. $i' = \begin{cases} i - 1, & \text{se } d = E \\ i + 1, & \text{se } d = D \end{cases}$;
3. O novo estado é q'.

Ou seja, é possível ir de uma dada configuração a outra com exatamente uma transição.

1.5.4 Fecho Originação

Definição

Dizemos que uma configuração C *fecho origina* C', denotado por $C \vdash_M^* C'$, quando existe uma sequência finita, de zero ou mais passos, de configurações $C = C_0, C_1, C_2, \ldots, C_k = C'$ tal que C_i origina C_{i+1} para todo $0 \le i < k$. Quando $k = 0$, temos $C = C'$ é uma originação (fecho) reflexiva.

1.6 Definição Formal da Cabeça de Leitura/Escrita

Dada a definição de Máquina de Turing e Configuração, podemos definir com exatidão qual o comportamento de *cabeça de leitura/escrita* nesse modelo computacional.

Definição

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{aceita}}, q_{\text{rejeita}})$ uma máquina de Turing, com fita $\tau : \mathbb{N} \to \Gamma$. A *cabeça de leitura-escrita* é representada por uma posição $i \in \mathbb{N}$, inicialmente $i = 1$. Em uma configuração $C = (q, \tau, i) \in Q \times (\Gamma^{\mathbb{N}}) \times \mathbb{N}$, a cabeça executa atomicamente:

1. lê $w_i = \tau(i)$;
2. obtém o resultado da transação $\delta(q, w_i) = (q', w_i', E)$ ou $\delta(q, w_i) = (q', w_i', D)$, se houver;
3. atualiza $\tau(i) \leftarrow w_i'$;
4. ajusta $i \leftarrow i - 1$ se δ especifica movimento E ou $i \leftarrow i + 1$ se especifica movimento D;

Em seguida, M transita para o estado q'.

1.7 *Halting*: Situações de Parada de uma Máquina de Turing

Definição

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{aceita}}, q_{\text{rejeita}})$ uma Máquina de Turing com fita infinita à direita. Diz-se que M *para* ou *termina a computação* quando alcança uma configuração (q, τ, i) para a qual não existe transição posterior bem definida, isto é, quando $\delta(q, \tau(i))$ é indefinida. Formalmente, dada uma entrada $w \in \Sigma^*$ e partindo da configuração inicial $C_1 = (q_0, \tau_1, 1)$, a máquina M realiza uma sequência finita de transições

$$C_1 \vdash_M C_2 \vdash_M C_3 \vdash_M \cdots \vdash_M C_k$$

em que $C_k = (q_k, \tau_k, i_k)$ para cada $k \geq 1$, e cada transição $C_k \vdash_M C_{k+1}$ é determinada pela função de transição δ. A computação *para* quando existe um índice $k \geq 1$ tal que $\delta(q_k, \tau_k(i_k))$ não está bem definida.

1.7.1 Estado de Aceitação e Rejeição

Um caso particular importante de parada ocorre quando $q_k \in \{q_{\text{aceita}}, q_{\text{rejeita}}\}$, chamados *estados de parada* ou *estados finais*. Por convenção, não há transições definidas a partir desses estados. Neste caso, dizemos que:

- M *aceita* w, se $q_k = q_{\text{aceita}}$;
- M *rejeita* w, se $q_k = q_{\text{rejeita}}$.

Além disso, se $\delta(q_k, \tau_k(1)) = (q'_k, c, E)$, isto é, se a cabeça está na posição $i_k = 1$ e a transição especifica movimento para a esquerda, então a máquina não pode executar esta transição, pois não existe posição à esquerda de 1 em uma fita infinita apenas à direita, ou seja, a transição está mal definida. Neste caso, a computação, por convenção, *para* imediatamente. Se não existe tal índice k, isto é, se a sequência de configurações é infinita e $\delta(q_i, \tau_i(i_i))$ está sempre definida para todo $i \geq 1$, dizemos que M *não para* ou que a computação *entra em laço* sobre a entrada w.

1.7.2 Configuração de Parada

Uma configuração $C = (q, \tau, i)$ é chamada de *configuração de parada* ou *configuração final* se $\delta(q, \tau(i))$ não está definida. Em particular, toda configuração (q, τ, i) com $q \in \{q_{\text{aceita}}, q_{\text{rejeita}}\}$ é uma configuração de parada.

Dessa forma, uma máquina de Turing M aceita a entrada w se uma sequência de configurações $C_1, C_2, \ldots, C_k$ existe, em que:

1. C_1 é a configuração inicial de M sobre a entrada w;
2. cada C_i origina C_{i+1}, isto é, $C_i \vdash_M C_{i+1}$ para $1 \leq i < k$;
3. C_k é uma configuração de aceitação, ou seja, $C_k = (q_{\text{aceita}}, \tau_k, i_k)$ para alguma fita τ_k e posição i_k.

1.8 Reconhecibilidade e Decidibilidade

1.8.1 Linguagem Turing-Reconhecível

A coleção de cadeias que M aceita é a *linguagem de* M, ou a *linguagem reconhecida por* M, denotada por $L(M)$:

$$L(M) = \{w \in \Sigma^* \mid M \text{ aceita } w\}.$$

Definição

Uma linguagem é dita *Turing-reconhecível*, ou *recursivamente enumerável*, se alguma Máquina de Turing a reconhece.

Uma máquina de Turing pode falhar ao processar uma cadeia de entrada w de duas maneiras distintas. A primeira forma de falha ocorre por *rejeição explícita*, quando a máquina alcança o estado q_{rejeita} e para imediatamente, indicando que a entrada não pertence à linguagem reconhecida. A segunda forma de falha ocorre quando a máquina entra em um *laço infinito*, continuando sua computação indefinidamente sem nunca alcançar um estado de parada. Neste caso, a máquina não aceita nem rejeita explicitamente a entrada, ela simplesmente nunca termina sua execução. Portanto, para toda entrada $w \in \Sigma^*$, exatamente uma das três situações deve ocorrer: M aceita w, M rejeita w, ou M entra em loop sobre w.

1.8.2 Linguagem Turing-Decidível

Às vezes, distinguir uma máquina que está em laço de uma que está meramente levando um tempo longo é difícil. Por essa razão, preferimos máquinas de Turing que param sobre todas as entradas; tais máquinas nunca entram em laço. Essas máquinas são chamadas *decisores*, porque elas sempre tomam uma decisão de aceitar ou rejeitar. Um decisor que reconhece alguma linguagem também é dito *decidir* essa linguagem.

Definição

Uma linguagem é dita *Turing-Decidível*, *recursiva* ou simplesmente decidível se alguma Máquina de Turing a decide.

1.9 Níveis de Descrição de uma Máquina de Turing

Existem três níveis principais de descrição para especificar uma máquina de Turing, cada um com seu propósito e grau de detalhe. A escolha do nível de descrição depende do contexto e do objetivo da apresentação: provas teóricas geralmente exigem descrições formais completas, enquanto discussões conceituais podem se beneficiar de descrições em linguagem natural.

1.9.1 Descrição em Linguagem Natural

A descrição em linguagem natural apresenta a operação da máquina de Turing de forma intuitiva e acessível, sem entrar em detalhes técnicos minuciosos. Neste nível, explicamos o comportamento geral da máquina e sua estratégia de processamento usando português ou outra língua natural, focando na ideia principal do algoritmo. Por exemplo, podemos descrever uma máquina que reconhece a linguagem $\{0^n 1^n \mid n \geq 0\}$ dizendo: "a máquina varre a fita da esquerda para a direita, marcando um 0 e depois um 1 correspondente, repetindo esse processo até que todos os símbolos sejam marcados ou até detectar um desequilíbrio". Este tipo de descrição é útil para comunicar a intuição por trás do funcionamento da máquina, mas não fornece detalhes suficientes para uma implementação completa.

1.9.2 Descrição de Implementação

A descrição de implementação detalha o funcionamento da máquina de forma mais precisa, especificando claramente como a fita é manipulada, quais símbolos são escritos e lidos, e como os estados são utilizados, mas ainda sem enumerar explicitamente toda a função de transição. Neste nível intermediário, descrevemos os passos principais do algoritmo de forma estruturada, frequentemente usando pseudocódigo ou uma sequência de instruções bem definidas. Por exemplo:

- 1. Varra a fita da esquerda para a direita procurando o primeiro 0 não marcado. Se não houver, vá para o passo 4;

– 2. Marque esse 0 com o símbolo $\times$;

– 3. Continue varrendo para a direita até encontrar o primeiro 1 não marcado. Se não houver, rejeite. Marque esse 1 com o símbolo $\circ$ e retorne ao início da fita. Volte ao passo 1;

– 4. Varra a fita para verificar se restam símbolos 1 não marcados. Se sim, rejeite; caso contrário, aceite.

Este nível é suficiente para que alguém com conhecimento de máquinas de Turing possa implementar a máquina, mas ainda evita a especificação completa de todos os estados e transições.

1.9.3 Descrição Formal

A descrição formal é o nível mais detalhado e rigoroso, especificando completamente a 7-upla $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{aceita}}, q_{\text{rejeita}})$ que define a máquina de Turing. Neste nível, enumeramos explicitamente todos os elementos: o conjunto finito de estados Q, o alfabeto de entrada Σ, o alfabeto de fita Γ, a função de transição $\delta : Q\times\Gamma \to Q\times\Gamma\times\{E, D\}$, frequentemente representada como uma tabela ou conjunto de regras, o estado inicial q_0, o estado de aceitação q_{aceita} e o estado de rejeição q_{rejeita}. Por exemplo, poderíamos especificar $Q = \{q_0, q_1, q_2, q_3, q_{\text{aceita}}, q_{\text{rejeita}}\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \times, \circ, \sqcup\}$, e listar cada transição como $\delta(q_0, 0) = (q_1, \times, D)$, $\delta(q_1, 0) = (q_1, 0, D)$, e assim por diante. Este nível de descrição é essencial para provas matemáticas rigorosas e para demonstrar formalmente que uma linguagem específica é Turing-reconhecível ou Turing-decidível, pois elimina qualquer ambiguidade sobre o comportamento da máquina.

Referências

SIPSER, Michael. Introdução à Teoria da Computação. 2. ed. São Paulo: Cengage Learning, 2005. ISBN 978-85-221-0499-4.

LEWIS, Harry R; PAPADIMITRIOU, Christos H. Elementos de Teoria da Computação. 2 ed. Porto Alegre: Bookman, 2004. ISBN 978-85-7307-534-2.