Máquina de Turing

Teoria da Computação

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

21 de março de 2023

[0] Slides baseados no livro LEWIS, Harry R.; PAPADIMITRIOU, Christos H. Elements of the Theory of Computation. ACM SIGACT News, v. 29, n. 3, p. 62-78, 1998.

Definição

- Vimos em LFA que nem os autômatos finitos e nem os autômatos de pilha podem ser considerados modelos gerais de computação, uma vez que não são capazes de reconhecer linguagens simples tais como $\{a^n b^n c^n : n \geq 0\}$
- Conheceremos dispositivos capazes de reconhecer essa e outra linguagens mais complexas: as Máquinas de Turing
- Apesar de serem mais genéricas que os autômatos previamente estudados, seu aspecto básico é similar
- Uma máquina de Turing consiste em um controle finito, uma fita e um cabeçote que pode ser utilizado para efetuar leituras ou gravações na fita

O que veremos?

- Veremos que, por mais elementares que as máquinas de Turing pareçam ser, nenhuma tentativa de fortalecê-la se mostra eficaz
- Estudaremos variantes da Máquina de Turing que utilizam de muitas fitas, máquinas com dispositivo de memória mais sofisticados que podem ser lidos ou gravados em regime de acesso aleatório semelhante aos computadores reais
- Podemos converter em máquina de Turing básica equivalente qualquer qualquer máquina estendida
- Portanto, qualquer computação realizada por uma máquina de Turing mais sofisticada podem também ser realizada com máquinas de Turing básicas

O que veremos?

- Definiremos um dispositivo gerador que é uma generalização da gramática livre de contexto que é equivalente à Máquina de Turing
- De uma outra perspectiva, discutiremos também a questão da computabilidade de uma função numérica ($2^x + x^2$) e concluiremos que tal notação se mostra equivalente às máquinas de Turing
- Visão amplamente aceita de que qualquer forma aceitável de expressão das ideias contidas em um algoritmo é, em última instância, equivalente à que se pode obter empregando-se uma máquina de Turing

Definição

Uma máquina de Turing consiste de um controle de estados finitos associado a uma unidade de fita

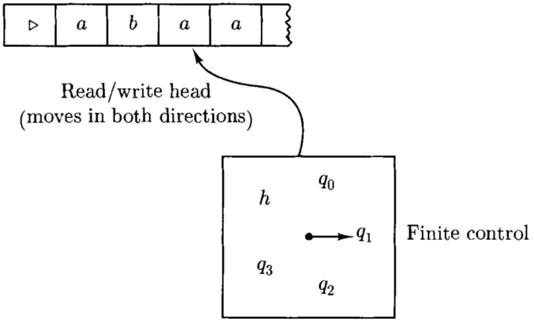

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Definição

- A comunicação entre as duas é proporcionada por um simples cabeçote responsável pela leitura dos símbolos da fita e também utilizado para modificar os símbolos encontrados na fita
- A unidade de controle opera em passos discretos; a cada passo, ela realiza duas operações, que dependem do seu estado atual e do símbolo da fita corretamente lido pelo cabeçote de leitura/gravação
  - (1) Levar a unidade de controle para um novo estado
  - (2a) Gravar um símbolo na célula corretamente apontada pelo cabeçote substituindo algum símbolo lá encontrado, ou então
  - (2b) Mover cabeçote de leitura/gravação para apontar uma célula à esquerda ou à direita na fita em relação à posição corrente

Definição

- A fita apresenta-se fisicamente delimitada em sua extremidade esquerda mas estende-se infinitamente para a direita
- Vamos admitir que a extremidade esquerda da fita esteja sempre marcada por um símbolo especial, denotado por $\triangleright$
- Convencionamos que sempre que o cabeçote encontrar um símbolo $\triangleright$, imediatamente a sua posição seja movida para a direita
- Iremos utilizar os símbolos especiais $\leftarrow$ e $\rightarrow$ para denotar o movimento do cabeçote para a esquerda e para a direita respectivamente

Definição

- Uma máquina de Turing é alimentada gravando-se previamente a cadeia de entrada nas células mais a esquerda da fita imediatamente à direita do símbolo $\triangleright$
- O restante da fita fica preenchido, nesse momento, com símbolos que representam espaços em branco aqui denotados por $\sqcup$
- A máquina é livre para modificar o conteúdo da sua fita de entrada de qualquer maneira que se considere apropriada
- Dado que a máquina pode mover seu cabeçote somente uma célula de cada vez, conclui-se que, após qualquer computação finita, apenas um número finito de células da fita terá sido visitado

Definição formal

Uma máquina de Turing é uma quíntupla $(K, \Sigma, \delta, s, H)$ onde

- K é um conjunto finito de estados
- Σ é o alfabeto de entrada, que contém os símbolos $\sqcup$e $\triangleright$ mas que não contém os símbolos $\leftarrow$ e $\rightarrow$
- $s \in K$ é o estado inicial
- $H \subseteq K$ é o conjunto de estados de parada
- δ, a função de transição, é uma função de $(K - H) \times \Sigma$ para $K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$, tal que
  - (a) para todos os $q \in K - H$, se $\delta(q, \triangleright) = (p, b)$, então $b = \rightarrow$
  - (b) para todos os $q \in K - H$ e $a \in \Sigma$, se $\delta(q, a) = (p, b)$, então $b \neq \triangleright$

Definição formal

- Se $q \in K - H$, $a \in \Sigma$ e $\delta(q, a) = (p, b)$, então M, quando no estado q e tendo lido o símbolo corrente a, transitará para o estado p e
  - (1) se b for um símbolo contido em Σ, M irá substituir na fita o símbolo corrente a pelo símbolo b ou,
  - (2) se b for um dos símbolos $\leftarrow$ o $\rightarrow$, M moverá sua cabeça na direção convencionada para o símbolo b

Observações

- Uma vez que δ é uma função, a operação de M é determinística e irá parar somente quando M transitar para algum dos estados de parada
- O marcador da extremidade esquerda da fita ($\triangleright$) nunca será apagado
- M nunca ultrapassará a extremidade esquerda de sua fita
- M nunca gravará um símbolo ($\triangleright$)
- δ não é definida nos casos em que o estado corrente pertence ao conjunto H: sempre que M alcança um estado de parada, sua operação para imediatamente

Exemplo

$M = (K, \Sigma, \delta, s, \{h\})$, onde:
$K = \{q_0, q_1, h\}$, $\Sigma = \{a, \sqcup, \triangleright\}$ e $s = q_0$
sendo δ dado pela seguinte tabela:

q, σ		$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$
q_1	$\triangleright$	$(q_1, \rightarrow)$

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Máquinas de Turing

Exemplo

$M = (K, \Sigma, \delta, s, \{h\})$, onde:
$K = \{q_0, h\}$, $\Sigma = \{a, \sqcup, \triangleright\}$, $s = q_0$, $H = \{h\}$, e δ uma função de transição especificada na seguinte tabela:

q,	σ	$\delta(q, \sigma)$
q_0	a	$(q_0, \leftarrow)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Máquinas de Turing

Configuração de uma máquina de Turing

- Para caracterizar o andamento da computação de uma máquina de Turing, é necessário especificar o seu estado, o conteúdo da sua fita de entrada e a posição do cabeçote sobre a fita
- Particionaremos a fita em duas partes: a parte à esquerda da célula que estiver sendo lida e a parte à direita dessa célula
- Convenção: A segunda parte da cadeia nunca termina com um espaço em branco

Definição - Configuração

Uma configuração da máquina de Turing $M = (K, \Sigma, \delta, s, H)$ é algum membro de $K \times \triangleright\Sigma^* \times (\Sigma^*(\Sigma - \{\sqcup\}) \cup \{\varepsilon\})$

Configurações válidas

$(q, \triangleright a, aba)$, $(h, \triangleright \sqcup\sqcup\sqcup, \sqcup a)$ e $(q, \triangleright \sqcup a \sqcup\sqcup, \varepsilon)$

Configurações não válidas

$(q, \triangleright baa, abc\sqcup)$ e $(q, \sqcup aa, ba)$

Configuração de parada

Uma configuração cujo estado componente seja o estado H de parada será chamada configuração de parada

Configurações válidas

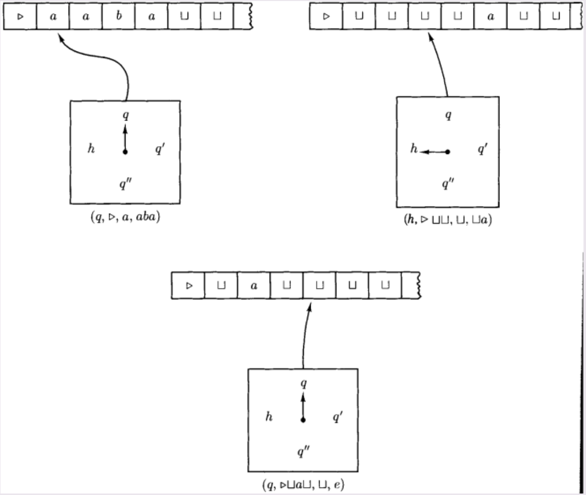

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Configuração - Notação simplificada

- Denotamos por $w\underline{a}u$ o conteúdo da fita na configuração (q, wa, u); o símbolo sublinhado indica a posição do cabeçote
- Dessa forma, as três configurações mostradas na figura anterior ficariam assim:
  - $(q, \rhd\underline{a}aba)$
  - $(h, \rhd\sqcup\sqcup\underline{\sqcup}\sqcup a)$
  - $(q, \rhd\sqcup a\sqcup\underline{\sqcup})$

Definição

Seja $M = (K, \Sigma, \delta, s, H)$ uma máquina de Turing. Sejam duas configurações de M, $(q_1, w_1\underline{a_1}u_1)$ e $(q_2, w_2\underline{a_2}u_2)$, onde $a_1, a_2 \in \Sigma$. Então

$$(q_1, w_1\underline{a_1}u_1) \vdash_M (q_2, w_2\underline{a_2}u_2)$$

se e somente se, para algum $b \in \Sigma \cup \{\leftarrow, \rightarrow\}$, $\delta(q_1, a_1) = (q_2, b)$ e também

1. $b \in \Sigma$, $w_1 = w_2$, $u_1 = u_2$ e $a_2 = b$, ou então
2. $b = \leftarrow$, $w_1 = w_2a_2$ e
   - (a) $u_2 = a_1u_1$, se $a_1 \neq \sqcup$ ou $u_1 \neq \varepsilon$, ou
   - (b) $u_2 = \varepsilon$ se $a_1 = \sqcup$ e $u_1 = \varepsilon$, ou ainda
3. $b = \rightarrow$, $w_2 = w_1a_1$ e
   1. (a) $u_1 = a_2u_2$, ou
   2. (b) $u_1 = u_2 = \varepsilon$, e $a_2 = \sqcup$

Para ilustrar esses casos, sejam $w, u \in \Sigma^*$, onde u não termina em um espaço $\sqcup$, e sejam $a, b \in \Sigma$.

Exemplos - Ilustração dos casos

1. $\delta(q_1, a) = (q_2, b)$
   Exemplo: $(q_1, w\underline{a}u) \vdash_M (q_2, w\underline{b}u)$
2. $\delta(q_1, a) = (q_2, \leftarrow)$
   Exemplo para (a): $(q_1, wb\underline{a}u) \vdash_M (q_2, w\underline{b}au)$
   Exemplo para (b): $(q_1, wb\underline{\sqcup}) \vdash_M (q_2, w\underline{b})$
3. $\delta(q_1, a) = (q_2, \rightarrow)$
   Exemplo para (a): $(q_1, w\underline{a}bu) \vdash_M (q_2, wa\underline{b}u)$
   Exemplo para (b): $(q_1, w\underline{a}) \vdash_M (q_2, wa\underline{\sqcup})$

Definição

Para uma Máquina de Turing M arbitrária, seja $\vdash_M^*$ o fechamento transitivo reflexivo de $\vdash_M$; diz-se que a configuração C_1 resulta na configuração C_2 se $C_1 \vdash_M^* C_2$. Uma computação em M é uma sequência de configurações $C_0, C_1, ..., C_n$, para algum $n \geq 0$, tal questão

$$C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M ... \vdash_M C_n$$

Diz-se, neste caso, que a computação é de comprimento n, ou que possui n passos, e denota-se $C_0 \vdash_M^n C_n$

Exemplo

$M = (K, \Sigma, \delta, s, \{h\})$, onde:
$K = \{q_0, q_1, h\}$, $\Sigma = \{a, \sqcup, \triangleright\}$ e $s = q_0$
sendo δ dado pela seguinte tabela:

$q,$	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$
q_1	$\triangleright$	$(q_1, \rightarrow)$

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Considere-se a máquina de Turing M descrita no exemplo anterior. Se M for iniciada na configuração $(q_1, \rhd\underline{\sqcup}aaaa)$, sua computação será

Exemplo

$$
\begin{aligned}
(q_1, \rhd\underline{\sqcup}aaaa) &\vdash_M (q_0, \rhd\sqcup\underline{a}aaa)\\
&\vdash_M (q_1, \rhd\sqcup\underline{\sqcup}aaa)\\
&\vdash_M (q_0, \rhd\sqcup\sqcup\underline{a}aa)\\
&\vdash_M (q_1, \rhd\sqcup\sqcup\underline{\sqcup}aa)\\
&\vdash_M (q_0, \rhd\sqcup\sqcup\sqcup\underline{a}a)\\
&\vdash_M (q_1, \rhd\sqcup\sqcup\sqcup\underline{\sqcup}a)\\
&\vdash_M (q_0, \rhd\sqcup\sqcup\sqcup\sqcup\underline{a})\\
&\vdash_M (q_1, \rhd\sqcup\sqcup\sqcup\sqcup\underline{\sqcup})\\
&\vdash_M (q_0, \rhd\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup})\\
&\vdash_M (h, \rhd\sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup})
\end{aligned}
$$

Notação para as Máquinas de Turing

- Máquinas básicas
- Regras para combinar máquinas

Máquinas Básicas

- Definido um alfabeto fixo Σ para nossas máquinas, para cada $a \in \Sigma \cup \{\rightarrow, \leftarrow\} - \{\triangleright\}$, definimos uma máquina de Turing $M_a = (\{s, h\}, \Sigma, \delta, s, \{h\})$ onde, para cada $b \in \Sigma - \{\triangleright\}$, $\delta(s, b) = (h, a)$
- Naturalmente, $\delta(s, \triangleright)$ deve ser sempre $(s, \rightarrow)$
- A única ação que essa máquina executa é a de gravar o símbolo a, se $a \in \Sigma$ parando imediatamente

Regras para combinar máquinas

- As conexões entre uma máquina e outra somente são percorridas no momento em que a primeira máquina termina a sua execução;
- A outra máquina é então iniciada a partir do seu estado inicial com a fita e o cabeçote exatamente na mesma situação deixada pela primeira máquina

$$\begin{tikzcd} M_1 \arrow[r, "a"] \arrow[d, "b"] & M_2 \\ M_3 & \end{tikzcd}$$

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Regras para combinar máquinas

- Sejam $M_1 = (K_1, \Sigma, \delta_1, s_1, H_1)$, $M_2 = (K_2, \Sigma, \delta_2, s_2, H_2)$ e $M_3 = (K_3, \Sigma, \delta_3, s_3, H_3)$ três máquinas de Turing
- A máquina de Turing combinada $M = (K, \Sigma, \delta, s, H)$, onde:
  - $K = K_1 \cup K_2 \cup K_3$
  - $s = s_1$
  - $H = H_2 \cup H_3$
  - Para cada $\sigma \in \Sigma$ e $q \in K - H, \delta(q, \sigma)$ é definida como segue:
    (a) Se $q \in K_1 - H_1$, então $\delta(q, \sigma) = \delta_1(q, \sigma)$
    (b) Se $q \in K_2 - H_2$, então $\delta(q, \sigma) = \delta_2(q, \sigma)$
    (c) Se $q \in K_3 - H_3$, então $\delta(q, \sigma) = \delta_3(q, \sigma)$
    (d) Se $q \in H_1$, então $\delta(q, \sigma) = s_2$, se $\sigma = a$ e $\delta(q, \sigma) = s_3$, se $\sigma = b$, caso contrário $\delta(q, \sigma) \in H$

Máquinas básicas

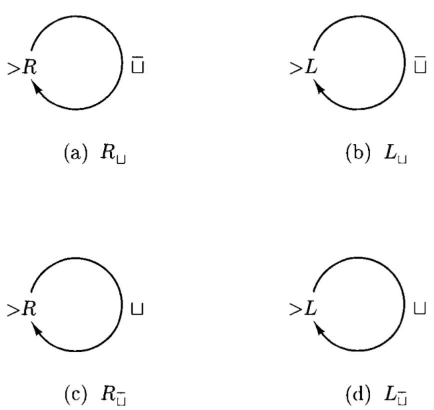

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Máquina de cópia - Transforma $\sqcup w \underline{\sqcup}$ em $\sqcup w \sqcup w \underline{\sqcup}$

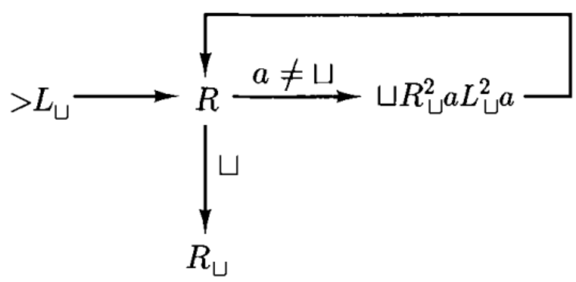

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Máquina de shift - Transforma $\sqcup w \underline{\sqcup}$ em $\sqcup\sqcup w \underline{\sqcup}$

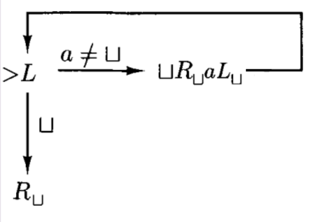

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Computações usando Máquinas de Turing

- Máquinas de Turing foram propostas com a promessa de que elas poderiam substituir, como reconhecedores de linguagem, todos os tipos de autômatos vistos em LFA
- Veremos como elas devem ser utilizadas para realizar tarefas computacionais

Convenção Importantíssima!!

A cadeia de entrada, isenta de espaços em branco, é gravada à direita do símbolo $\triangleright$, com um espaço em branco a sua esquerda e com espaços em branco a sua direita.

Destaque

Ou seja, depois de $\triangleright$, SEMPRE teremos um $\sqcup$ antes da palavra w ser escrita.
Qualquer palavra w sempre será escrita da seguinte maneira: $\triangleright\sqcup w$

Definição

- Seja $M = (K, \Sigma, \delta, s, H)$ uma Máquina de Turing, tal que $H = \{y, n\}$ consiste em dois estados distintos de parada (y e n simbolizam "sim" e "não" respectivamente)
- Qualquer configuração de parada cuja componente de estado é y é chamada configuração de aceitação
- Componente de estado é n é chamada configuração de rejeição
- Dizemos que M aceita uma entrada $w \in (\Sigma - \{\sqcup, \triangleright\})^*$, se $(s, \triangleright \underline{\sqcup} w)$ levar a uma configuração de aceitação
- Dizemos que M rejeita w se $(s, \triangleright \underline{\sqcup} w)$ levar a uma configuração de rejeição

Definição

- Seja $\Sigma_0 \subseteq \Sigma - \{\sqcup, \triangleright\}$ um alfabeto, chamado alfabeto de entrada de M
- Fixando Σ_0 como um subconjunto de $\Sigma - \{\sqcup, \triangleright\}$, permitimos que nossas máquinas de Turing utilizem símbolos adicionais durante a sua computação
- Dizemos que M decide a linguagem $L \subseteq \Sigma_0^*$, se, para qualquer cadeia $w \in \Sigma_0^*$, se $w \in L$, então M aceita w; e se $w \notin L$, então M rejeita w
- Por fim, dizemos que a linguagem L é recursiva se houver uma máquina de Turing que a decide
- Uma máquina de Turing decide uma linguagem L se, quando iniciada com a entrada w, ela sempre para em um estado de parada que corresponde à resposta correta à entrada w

Exemplo - Sequências de a's, b's e c's (nessa ordem) onde a quantidade de a's é igual a quantidade de b's e igual a quantidade de c's em cada sequência

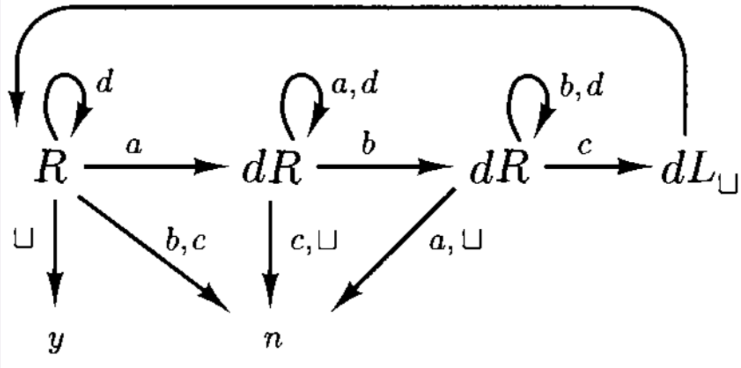

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Observações

- Com outros reconhecedores, duas opções podem ocorrer: ou a máquina aceita ou então a máquina rejeita a entrada
- Uma máquina de Turing, apesar de ter apenas dois estados de parada (y e n), pode simplesmente deixar de parar
- Dado uma máquina de Turing, ela pode ou não decidir uma linguagem e não há um modo óbvio de dizer se ela faz isso

Máquinas de Turing podem produzir saídas mais elaboradas do que um simples "sim" ou "não"

Funções Recursivas - Definição

- Seja $M = (K, \Sigma, \delta, s, \{h\})$ uma máquina de Turing,
- $\Sigma_0 \subseteq \Sigma - \{\sqcup, \triangleright\}$ um alfabeto;
- e $w \in \Sigma_0^*$
- Suponha que M para ao operar sobre a entrada w e que $(s, \triangleright \underline{\sqcup}\, w) \vdash_M^* (h, \triangleright \underline{\sqcup}\, y)$ para algum $y \in \Sigma_0^*$
- Então y é dito saída de M para a entrada w e é denotado $M(w)$
- Note que $M(w)$ é definida somente se M para ao operar sobre a entrada w e isso de fato, acontece para configurações da forma $(h, \triangleright \underline{\sqcup}\, y)$, com $y \in \Sigma_0^*$

Funções Recursivas

- f função qualquer de Σ_0^* para Σ_0^*
- Dizemos que M computa a função f se, para todos os $w \in \Sigma_0^*$, $M(w) = f(w)$
- Assim, para todo $w \in \Sigma_0^*$, M acaba por parar na entrada w e, nessa ocasião, sua fita contém a cadeia $\triangleright \sqcup f(w)$
- Uma função f é dita recursiva, se houver uma máquina de Turing M que computa f

Exemplo - Função Sucessora - $succ(n) = n + 1$

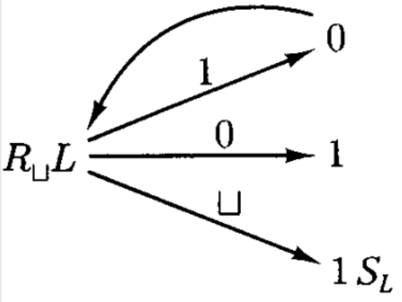

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Funções recursivamente enumeráveis

- Pelo fato de uma máquina de Turing decidir uma linguagem ou computar uma função, pode-se considerar razoável interpretá-la como um algoritmo que executa corretamente alguma tarefa computacional
- Seja $M = (K, \Sigma, \delta, s, H)$ uma máquina de Turing
- Seja $\Sigma_0 \subseteq \Sigma - \{\sqcup, \triangleright\}$ um alfabeto
- E seja $L \subseteq \Sigma_0^*$ uma linguagem
- Dizemos que M semidecide L se para qualquer cadeia $w \in \Sigma_0^*$, $w \in L$ se e somente se M para em resposta à entrada w
- Uma linguagem L é recursivamente enumerável se e somente se existir uma máquina de Turing M que semidecide L

Linguagens recursivamente enumeráveis

- Quando M é acionada com a entrada $w \in L$, exige-se que pare ao final
- Entretanto, quando $w \in \Sigma_0^* - L$, então M nunca deve atingir o estado de parada
- Escreve-se $M(w) = \nearrow$, se M falhar em parar em resposta à entrada w

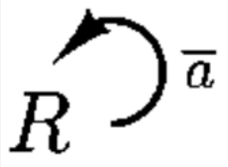

Figura: Fonte: Livro Elementos de Teoria da Computação - Papadimitriou

Linguagens recursivamente enumeráveis

- Existe uma diferença importante entre os autômatos visto em LFA e máquinas de Turing que semidecidem
- Os autômatos finitos sempre param após ler toda a entrada
- Assim é um dispositivo computacionalmente útil, um algoritmo
- Com relação às maquinas de Turing que semidecidem, se $w \notin L$, então nunca saberemos se esperamos o suficiente para obter uma resposta
- Máquinas de Turing que semidecidem não são algoritmos

Teorema

Se uma linguagem é recursiva, então ela é recursivamente enumerável.

- Será possível sempre transformar cada máquina de Turing que semidecide uma linguagem em um algoritmo real para decidir a linguagem?
- Veremos que há linguagens recursivamente enumeráveis que não são recursivas

Máquinas de Turing

Teorema

Se L é uma linguagem recursiva, então o seu complemento $\overline{L}$ também é recursivo

Prova

Se L é decidida pela máquina de Turing $M = (K, \Sigma, \delta, s, \{y, n\})$, então $\overline{L}$ é decidida pela máquina $M' = (K, \Sigma, \delta', s, \{y, n\})$, idêntica à M exceto que inverte os papéis de y e n, ou seja, δ' é definida como se segue:

$$\delta'(q, a) = \begin{cases} n & \text{se } \delta(q, a) = y \\ y & \text{se } \delta(q, a) = n \\ \delta(q, a) & \text{caso contrário} \end{cases}$$

Considerações finais

- É claro que $M'(w) = y$ se e somente se $M(w) = n$ e, portanto, M' decide $\overline{L}$
- Seria a classe das linguagens recursivamente enumeráveis também fechada sob o complemento?
- Veremos que não

Próxima Aula

O que vem por aí?

- Extensões das máquinas de Turing
- Máquinas de Turing de acesso aleatório
- Máquinas de Turing não-determinísticas
- Gramáticas
- Funções numéricas

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

Máquina de Turing

Teoria da Computação

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

21 de março de 2023