

Introdução às Máquinas de Turing (TM)

Comparação com computadores:

- um modelo matemático simples de um computador
 - Semelhanças: lê e escreve em posições arbitrárias de memória
 - Diferenças: sem limite no tamanho de memória

Uma TM define uma classe de linguagens denominada linguagens **Rekursivamente Enumeráveis (RE)**

(ou conjuntos recursivamente enumeráveis)

- a máquina para se a palavra pertence a linguagem, porém pode processar indefinidamente (i.e. não parar) caso contrário

TMs definem a classe de funções que elas computam (**funções parcialmente recursivas**)

- dada a entrada para a função, a máquina pode ou não parar)

TM é uma formalização aceita para um "procedimento efetivo"

(obs: "procedimento efetivo" = que pode ser resolvido algoritmicamente)

Ideia:

- existem mais problemas que algoritmos
- TM como modelo matemático simples usado para provar que certos problemas não são computáveis
- Exemplos de problemas para os quais não existem algoritmos (i.e. problemas indecidíveis):
 - determinar se o complemento de uma CFL é vazia
 - outros: ver item 7.4.5 do capítulo sobre propriedades de linguagens livres de contexto

Um modelo formal de um procedimento efetivo deve ter:

1. descrição finita
2. o procedimento deve ser composto de passos discretos que podem ser executados mecanicamente

Exemplo: $L = \{0^n 1^n \mid n > 0\}$

Ideia:

- 1) troca o primeiro 0 por X
- 2) move a direita (pulando 0's e Y's) até achar o primeiro 1
- 3) Se encontrar, troca 1 por Y
se não encontrar (achar B ao invés de 1) rejeita a palavra
- 4) volta a esquerda (pulando Y's e 0's) até achar o X
- 5) se a direita existe um 0, voltar a 1
se a direita existe um Y (acabaram os 0's), ir até o final verificar se ainda existem 1's. Se existir rejeita, senão aceita.

(FIG 8.10 do livro, no arquivo jflap MT1.jff)

Cada posição da fita contém um símbolo. Existe uma posição mais à esquerda, mas a fita é infinita à direita.

Note que em um passo a máquina, dependendo do símbolo de entrada e o estado

atual pode:

- 1) mudar de estado
- 2) escrever um símbolo na fita (por cima do símbolo lido)
- 3) mover a cabeça de leitura para a direita ou esquerda

Tabela de transições para a TM acima:

estado	Símbolos na fita				
	0	1	X	Y	B
q0	(q1,X,R)	-	-	(q3,Y,R)	-
q1	(q1,0,R)	(q2,X,R)	-	(q1,Y,R)	-
q2	(q2,0,L)	-	(q0,X,R)	(q2,Y,L)	-
q3	-	-	-	(q3,Y,R)	(q4,B,R)
q4	-	-	-	-	-

OBS: - significando HALT

Formalmente:

Def: uma máquina de Turing é uma tupla de 7 elementos

$M=(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, onde

Q é um conjunto finito de estados

Γ é um conjunto finito de símbolos da fita

B é um símbolo em Γ , e' o branco

(indica posições vazias na fita,

que são as que estão preenchidas com esse símbolo)

Σ subconjunto de Γ e' o conjunto de símbolos de entrada

δ é um função de $Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L,R\}$

q_0 em Q e' o estado inicial

F subconjunto de Q e' o conjunto de estados finais.

Def: a Configuração INSTANTÂNEA (ID) de uma máquina M é descrita por

$$\alpha_1 q \alpha_2,$$

onde q é o estado atual e α_1 e α_2 são strings em Gama^* , que é o conteúdo da fita do início da fita até o símbolo não Branco mais a direita.

O símbolo sendo lido é o símbolo mais a esquerda de α_2 .

Def: a relação $\vdash$ (resulta) para uma máquina de Turing M é definida por:

para $\text{delta}(q, X_i) = (p, Y, L)$ temos:

$$X_1 X_2 \dots X_{i-1} q X_i \dots X_n \quad \vdash \quad X_1 X_2 \dots X_{i-2} p X_{i-1} Y X_{i+1} \dots X_n$$

$$\text{se } i=1 \text{ fica:} \quad q X_1 X_2 \dots X_n \quad \vdash \quad p B Y X_2 \dots X_n$$

$$\text{se } i=n \text{ e } Y=B \text{ fica:} \quad X_1 X_2 \dots X_{n-1} q X_n \quad \vdash \quad X_1 X_2 \dots X_{n-2} p X_{n-1}$$

para $\text{delta}(q, X_i) = (p, Y, R)$ temos:

$$X_1 X_2 \dots X_{i-1} q X_i \dots X_n \quad \vdash \quad X_1 X_2 \dots X_{i-1} Y p X_{i+1} \dots X_n \text{ se}$$

$$\text{se } i=n \text{ fica:} \quad X_1 X_2 \dots X_{n-1} q X_n \quad \vdash \quad X_1 X_2 \dots X_{n-1} Y p B$$

$$\text{se } i=1 \text{ e } Y=B \text{ fica:} \quad q X_1 X_2 \dots X_n \quad \vdash \quad p X_2 \dots X_n$$

Def: $\vdash^*$ é o fecho transitivo e reflexivo de $\vdash$

Def: A linguagem ACEITA por uma máquina de Turing $M=(Q, \text{Sigma}, \text{Gama}, \text{delta}, q_0, B, F)$ é definida por:

$$L(M) = \{ w \text{ em } \text{Sigma}^* \mid \begin{array}{l} q_0 w \vdash^* \alpha_1 p b \alpha_2 \text{ para algum } p \text{ em } F, \alpha_1 \text{ e } \alpha_2 \text{ em } \text{Gama}^* \text{ e} \\ b \text{ em } \text{Gama} \text{ tal que } \text{delta}(p, b) \text{ é indefinido} \end{array} \}$$

assumindo que se w pertence a linguagem a máquina para;
porem, se w não pertence a linguagem a máquina pode processar indefinidamente.

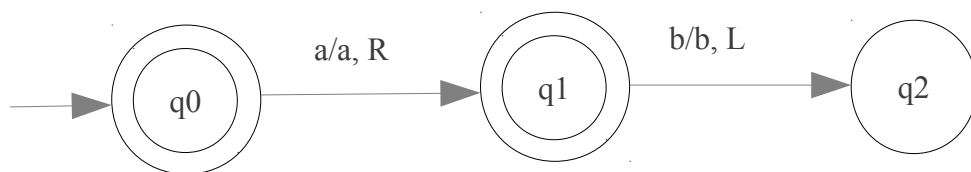
Exemplo da computação de 0011 na máquina acima:

q_0 0011 |— X q_1 011 |— X0 q_1 11 |— X q_2 0Y1 |— q_2 X0Y1 |— X q_0 0Y1 |—
 XX q_1 Y1 |— XXY q_1 1 |— XX q_2 YY |— X q_2 XYY |— XX q_0 YY |—
 XXY q_3 Y |— XXY q_3 |— XXYB q_4

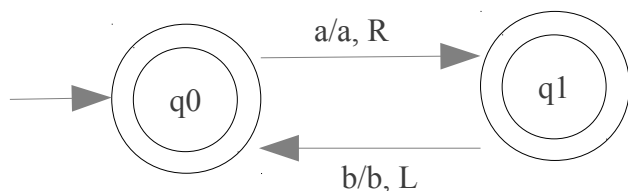
Exemplo de diferentes tipos de não aceitação:

$L = \{a,b,c\}^* - \{ab\}\{a,b,c\}^*$ (todas as palavras que não começam com ab):

1) a máquina para, se w em L ou não. Se w começa com ab, nem é preciso processar a palavra inteira.



2) só para se w em L



Exemplo de utilização da M.T para computar funções:

Subtração própria (chamada monus), $m - n$ igual a:

- $m - n$ se $m \geq n$
- 0 se $m < n$
-

representação de m e n :

qualquer numero i pode ser representado por 0^i , portanto podemos representar m e n por: $0^m 1 0^n$, usando 1 como o separador de números resultado:

- se $m \geq n$ no final da computação a fita deve conter $m-n$ 0's
- se $m < n$ no final da computação a fita deve conter apenas B's

Note que neste exemplo não existe estado final — a máquina sempre para quando o resultado está na fita.

Ideia (pseudo-codigo) do algoritmo:

1) troca o primeiro 0 por B e procura a direita pelo primeiro 0 depois do 1. Troca este 0 por 1 e volta a esquerda até encontrar um B. Repete.

2) a máquina para se:

- procurando o primeiro 0 depois do 1, encontra um B
 - → quer dizer que os n 0's já foram transformados em 1's. como o primeiro 0 já foi transformado em B, voltar, transformando os n 1's em brancos e quando encontrar um B (no início da fita), transforma-lo de volta em 0.
- no início do ciclo, ao invés de encontrar um 0, encontra um 1
 - → quer dizer que $m \leq n$.
Então transformar todos os 0's e 1's em B's.

(FIG 8.12 do livro, no arquivo jflap MT2.jff)

Exemplo da computação de 0010:

q0 0010 |− B q1 010 |− B0 q1 10 |− B01 q2 0 |−
B0 q3 11 |− B q3 011 |− q3 B011 |− B q0 011 |−
BB q1 11 |− BB1 q2 1 |− BB11 q2 |− BB1 q4 1 |−
BBq4 1 |− Bq4 |− B0 q6
