INTRODUCAO 'AS MAQUINAS DE TURING

Comparacao com computadores:

==> um modelo matematico simples de um computador
Semelhancas: le e escreve em posicoes arbitrarias de memoria
Diferencas: sem limite no tamanho de memoria

==> define uma classe de linguagens (conjuntos recursivamente enumeraveis - a maquina para se a palavra pertence a linguagem, porem pode entrar em loop caso contrario)

==> define uma classe de funcoes que ele computa (funcoes parcialmente recursivas - dada a entrada para a funcao, a maquina pode ou nao parar)

==> e' uma formalizacao aceita para um "procedimento efetivo"
(obs: "procedimento efetivo" = que pode ser resolvido algoritmicamente)

Exemplos de problemas para os quais nao existem algoritmos:

- Ideia: existem mais problemas que algoritmos
- determinar se o complemento de uma CFL e' vazia

Um modelo formal de um procedimento efetivo de ter:
1) descricao finita
2) o procedimento deve ser composto de passos discretos que podem ser executados mecanicamente

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

Ideia:

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

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

Cada posicao da fita contem um simbolo. Existe uma posicao mais a esquerda, mas a fita e' infinita a direita.

Note que em um passo a maquina, dependendo do simbolo de entrada e o estado atual pode:

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

Tabela de transicoes para a TM acima:

estado	0	1	X	Y	B
	Simbolos na fita				
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	-	-	-	-	-

Formalmente:

Def: uma maquina de Turing e′ uma tupla de 7 elementos
M=(Q, Sigma, Gama, delta, q0, B, F), onde
Q e′ um conjunto finito de estados
Gama e′ um conjunto finito de simbolos da fita
B e′ um simbolo em Gama, e′ o branco
(indica posicoes indica vazias na fita,
que sao as que estao preenchidas com esse simbolo)
Sigma subconjunto de Gama e′ o conjunto de simbolos de entrada
delta e′ um funcao de Q x Gama --> Q x Gama x {L,R}
q0 em Q e′ o estado inicial
F subconjunto de Q e′ o conjunto de estados finais.

Def: a DESCRICAO INSTANTANEA de uma maquina M e′ descrita por A1 q A2, onde q e′ o estado atual e A1 e A2 sao strings em Gama*, que e′ o conteudo da fita do inicio da fita ate′ o simbolo nao Branco mais a direita. O simbolo sendo lido e′ o simbolo mais a esquerda de A2.

Def: a relacao |- (resulta) para uma maquina de Turing M e′ definida por:

1) X1 X2 ... Xi-1 q Xi ... Xn |- X1 X2 .. Xi-2 p Xi-1 Y Xi+1 ... Xn se
   delta(q,Xi) = (p, Y, L).
   . se Xi-1 = n (ou seja, Xi...Xn e′ uma sequencia de B) entao Xi = B
   . se i = 1 entao a computacao aborta, pois a cabeca da fita nao pode
   se mover mais para a esquerda

2) X1 X2 ... Xi-1 q Xi ... Xn |- X1 X2 .. Xi-1 Y p Xi+1 ... Xn se
   delta(q, Xi) = (p, Y, R)

|-* e′ o fecho transitivo e reflexivo de |-

Def: A linguagem ACEITA por uma maquina de Turing
M=(Q, Sigma, Gama, delta,q0, B, F) e′ definido por:

{w em Sigma* | q0 w |-* A1 p bA2 para algum p em F, A1 e A2 em Gama* e
b em Gama tal que delta(p,b) e′ indefinido}

assumindo que se w pertence a linguagem a maquina para;
porem, se w nao pertence a linguagem a maquina pode entrar em loop.

```
Exemplo da computacao de 0011 na maquina acima:
  q0 0011 |- X q1 011 |- X0 q1 11 |- X q2 0Y1 |- q2 X0Y1 |- X q0 0Y1 |-
  XX q1 Y1 |- XXY q1 1 |- XX q2 YY |- X q2 XYY |- XX q0 YY |- XXY q3 Y |-
  XXYY q3 |- XXYYBq4

Exemplo de diferentes tipos de nao aceitacao:
  L = {a,b,c}* - {ab}{a,b,c}*  (todas as palavras que nao comecam com ab):

1) para sempre se w em L ou nao.  Se w comeca com ab, nem e' preciso
   processar a palavra inteira.

           a/a, R          b/b, L
    ((q0)) --------> ((q1)) --------> (q2)

2) so' para se w em L

           a/a, R
    ((q0)) --------> ((q1))
          <-------
           b/b, L
```

```
Exemplo de utilizacao da M.T para computar funcoes:
---------------------------------------------------
Subtracao: m - n igual a:  m - n se m >= n
                           0 se m < n


representacao 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 numeros
resultado:
   se m>=n no final da computacao a fita deve conter m-n 0's
   se m< n no final da computacao a fita deve conter apenas B's


Note que neste exemplo nao existe estado final -- a maquina sempre para
quando o resultado esta' na fita.


Ideia:
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 ate' encontrar um B.  Repete.
2) a maquina para se:
  . procurando o primeiro 0 depois do 1, encontra um B
    -> quer dizer que os n 0's ja' foram transformados em 1's.
    como o primeiro 0 ja' foi transformado em B, voltar, transformando os
    n 1's em brancos e quando encontrar um B (no inicio da fita),
    transforma-lo de volta em 0.
  . no inicio do ciclo, ao inves de encontrar um 0, encontra um 1
    -> quer dizer que m <= n.  Entao transformar todos os 0's e 1's em B's.
```

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

Exemplo da computacao 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