Máquina de Turing

Seguindo nos mecanismos reconhecedores das linguagens formais, o mecanismo reconhecedor das linguagens enumerável recursivamente temos a Máquina de Turing.

Uma Máquina de Turing tem a representação similar aos mecanismos vistos anteriormente, com um diferencial de que a fita que antes era utilizada somente para leitura, agora pode ser utilizada para escrita também, e também podemos movimentar o cabeçote de leitura para direita e para esquerda também.

Assim, uma Máquina de Turing é formada então pela fita, que é utilizada para entrada da palavra a ser testada, e pode também ser utilizada para saída. Temos o cabeçote de leitura e escrita, que se movimenta de acordo com a função de transição, sendo direcionado para direita ou esquerda. E ainda a MT é formada pela função de transição, que faz com que o cabeçote se movimente para direita ou esquerda, faz com que ocorra a mudança de estado e faz a leitura do símbolo ao qual o cabeçote está indicando na fita.

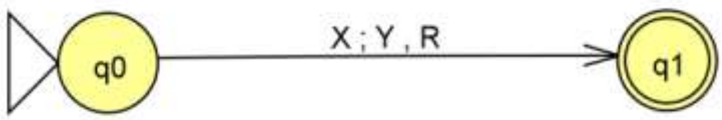
As Máquinas de Turing podem ser divididas em dois tipos: reconhecedoras e transdutoras. As MT reconhecedoras tem como entrada uma palavra, e a resposta da máquina é SIM/NÃO ou ACEITA/REJEITA a entrada para determinada máquina. Por exemplo, se a palavra “aaabbb” for dada como entrada para uma máquina que aceita a linguagem $L = \{a^nb^n / n \geq 0\}$ a resposta deve ser ACEITA, pois a palavra faz parte da linguagem. Caso a palavra de entrada seja “bababa” para a mesma máquina a resposta deve ser NÃO, pois a palavra “bababa” não faz parte da linguagem $L = \{a^nb^n / n \geq 0\}$. Já por sua vez, as MT transdutoras tem como entrada uma palavra, e a sua saída é escrita na própria fita de entrada. Por exemplo, se tivermos uma Máquina de Turing que leia um texto qualquer e deixe-o em CAIXA ALTA, e com a entrada “turing” a resposta da máquina deve ser “TURING” e deverá estar escrita na fita utilizada para leitura/escrita.

A fita de uma MT é infinita para os dois lados, sendo assim uma MT tem memória infinita, diferente de qualquer computador atual existente, isto possibilita realizar com uma máquina de Turing qualquer problema que possa ser resolvido com um computador, sendo então a MT o maior poder computacional existente.

Um modo de simplificar o funcionamento da MT é utilizar a fita infinita para a direita e finita para a esquerda, iniciando com um símbolo que delimita o marcador de início da fita (<), assim ao testar uma palavra na fita, a palavra começará a partir da segunda posição e se estenderá até o final da palavra, completando a fita com um símbolo especial, o símbolo branco (β), por exemplo, ao inserir a palavra “aaabbb” na fita da MT, a palavra resultante será “<aaabbbββββ...”

Estes dois símbolos especiais, símbolo marcador de início de fita e símbolo branco fazem parte da descrição formal de uma Máquina de Turing, que é formada por {E, Σ, δ, i, F, B, Γ,<}, na qual:

- E é o conjunto finito de estados
- Σ é o conjunto finito de símbolos do alfabeto
- δ é a função de transição
  - δ: E x (Σ U Γ) → E x (Σ U Γ) x (direita/esquerda)
- i é o estado inicial (i ϵ E)
- F é o conjunto de estados finais (F C E)
- B é o símbolo de branco
- < é o marcador de início da fita
- Γ são os símbolos auxiliares da fita (B ϵ Γ, < ϵ Γ)



A representação gráfica de uma MT pode ser vista na Figura acima, na qual X representa o símbolo lido da fita, Y representa o símbolo escrito na fita e R representa o sentido do movimento do cabeçote da fita (direita ou esquerda).

Se pensarmos em uma Máquina de Turing transdutora que recebe uma palavra formada por {a, b} e transforma a palavra em uma palavra somente com letras maiúsculas, por exemplo, para a entrada "<abbabaβββ...>" deve ser gerado como saída a seguinte palavra "<ABBABAβββ...>". Para montarmos uma MT que altera as letras "a" e "b" para maiúsculas devemos ler a letra minúscula e alterar para maiúscula até o final da palavra. Isto pode ser feito na MT representado na Figura abaixo, nas duas transições em loop de s0

Para a MT acima, que transforma uma palavra de {a, b} em maiúscula, a descrição formal é formada por:

- E = {s0, s1}
- Σ = {a, b}
- δ é a função de transição

	a	b	A	B	<	□
s0	s0, A, r	s0, B, r	x	x	x	s1, □, l
s1	x	x	x	x	x	x

- i = s0
- F = {s1}
- □ é o símbolo de branco
- < é o marcador de início da fita
- Γ = {□, <, A, B}, são os símbolos auxiliares da fita

Para a MT acima, o alfabeto da fita é formado por {a, b}, o conjunto de estados E = {s0, s1} são os estados existentes na MT, o estado inicial é o s0, e o conjunto de estados finais é formado por {s1}, o símbolo que delimita o marcador de início de fita é o <, e o símbolo que indica fita em branco é o □. Os símbolos auxiliares da fita são símbolos que podem ser escritos na fita para auxiliar a função da MT, Γ = {□, <, A, B} e por fim a função de transição, representada pela tabela acima na qual nas linhas temos todos os estados existentes, e nas colunas os símbolos do alfabeto da fita e os símbolos auxiliares da fita. A célula marcada como "s0, A, R" está na linha do estado "s0" e coluna do símbolo "a", isto indica que temos uma transição a partir de "s0" com o símbolo "a", esta transição tem como destino "s0", sobre-escreve o símbolo "a" com "A" e move o cabeçote para direita (right). Células com X indica que não tem transição com o símbolo da coluna da tabela, por exemplo, a partir do estado "s1" não temos nenhuma transição, assim a linha "s1" da tabela é preenchida com X em todas as colunas.

Para o processamento de uma palavra, “<abbbaaββββ...” por exemplo, o cabeçote da fita começa na primeira posição à direita do marcador de início da fita, tal como é mostrado abaixo

<	a	b	b	b	a	a	□	□	□	...
	↑									

Processando o primeiro “a”, alteramos para “A”, e movemos o cabeçote para direita

<	A	b	b	b	a	a	□	□	□	...
		↑								

Para os símbolos seguintes, cada símbolo (a ou b) é alterado para (A ou B) e o cabeçote movimenta-se para a direita até chegar no símbolo branco (□)

<	A	B	b	b	a	a	□	□	□	...
			↑							

<	A	B	B	b	a	a	□	□	□	...
				↑						

<	A	B	B	B	a	a	□	□	□	...
					↑					

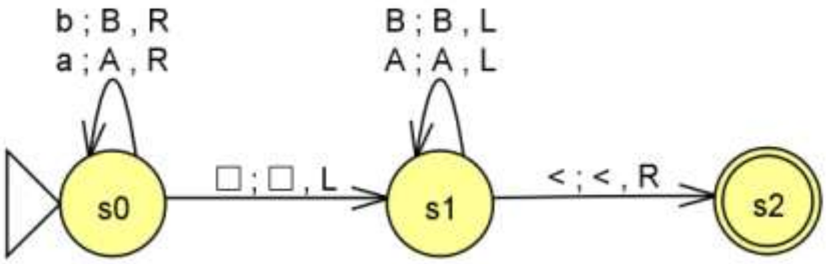
<	A	B	B	B	A	a	□	□	□	...
						↑				

<	A	B	B	B	A	A	□	□	□	...
							↑			

Quando o cabeçote chega no primeiro símbolo em branco (□), a transição de “s0” para “s1” é realizada, lendo o símbolo em branco, escrevendo ele novamente e movendo o cabeçote para a esquerda (left), e alterando o estado atual da MT para o estado “s1” que é o estado final, tendo como resultado na fita a seguinte palavra:

<	A	B	B	B	A	A	□	□	□	...
						↑				

Nas Máquinas de Turing é comum finalizar o processamento de uma função ou reconhecimento de uma palavra e deixarmos o cabeçote da MT na mesma posição na qual ele começa, ou seja, a direita do símbolo marcador de início de fita. Para uma MT que volte o cabeçote para a posição inicial, teríamos como resultado a seguinte MT



Na MT acima, o processamento da palavra “<abbbaaββββ...” se dá do modo que é representado abaixo

<	a	b	b	b	a	a	□	□	□	...
	↑									

<	A	b	b	b	a	a	□	□	□	...
		↑								

<	A	B	b	b	a	a	□	□	□	...
			↑							

<	A	B	B	b	a	a	□	□	□	...
				↑						

<	A	B	B	B	a	a	□	□	□	...
					↑					

<	A	B	B	B	A	a	□	□	□	...
						↑				

<	A	B	B	B	A	A	□	□	□	...
							↑			

<	A	B	B	B	A	A	□	□	□	...
						↑				

<	A	B	B	B	A	A	□	□	□	...
					↑					

<	A	B	B	B	A	A	□	□	□	...
				↑						

<	A	B	B	B	A	A	□	□	□	...
			↑							

<	A	B	B	B	A	A	□	□	□	...
		↑								

<	A	B	B	B	A	A	□	□	□	...
	↑									

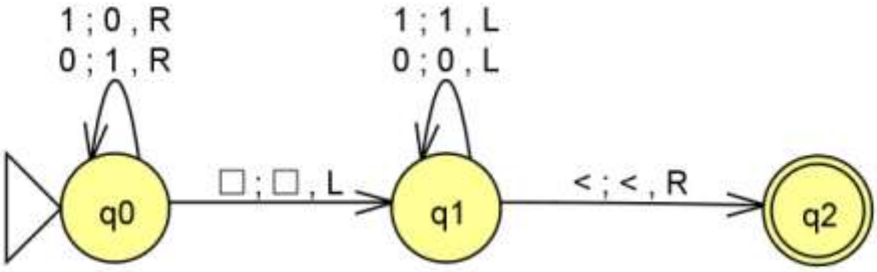
<	A	B	B	B	A	A	□	□	□	...
↑										

<	A	B	B	B	A	A	□	□	□	...
	↑									

Outro exemplo de Máquina de Turing Transdutora pode ser uma MT que receba como entrada um número binário, e devolva como saída o complemento binário da entrada, ou seja, a MT troca 0 por 1 e 1 por 0.

O funcionamento da MT acima trabalha fazendo as trocas de 0 por 1 e de 1 por 0 e, avançando para a direita com o cabeçote até encontrar o primeiro símbolo de branco.

Após isso a MT somente retorna o cabeçote para a posição inicial da fita (à direita do símbolo marcador de início). A MT que realiza o complemento binário pode ser visto na Figura abaixo, e o processamento de uma palavra “10100βββ...”



<	1	0	1	0	0	β	β	...
	↑							

<	0	0	1	0	0	β	β	...
		↑						

<	0	1	1	0	0	β	β	...
			↑					

<	0	1	0	0	0	β	β	...
				↑				

<	0	1	0	1	0	β	β	...
					↑			

<	0	1	0	1	1	β	β	...
						↑		

<	0	1	0	1	1	β	β	...
					↑			

<	0	1	0	1	1	β	β	...
				↑				

<	0	1	0	1	1	β	β	...
			↑					

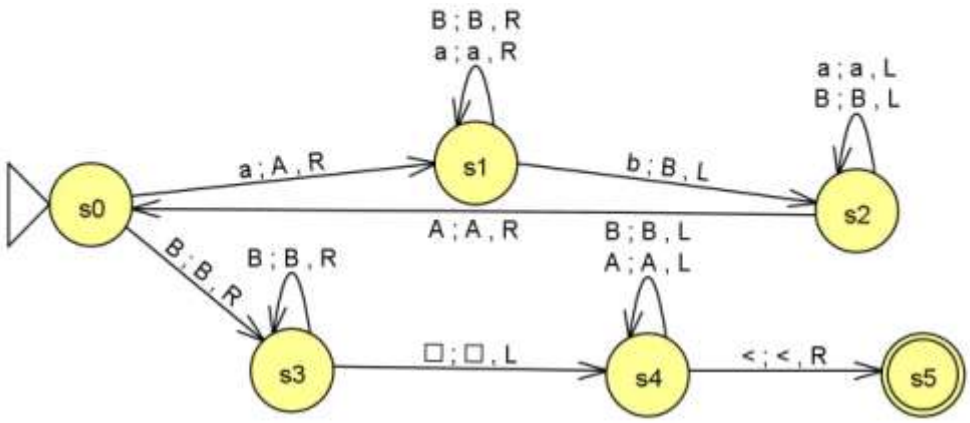
<	0	1	0	1	1	β	β	...
		↑						

<	0	1	0	1	1	β	β	...
	↑							

<	0	1	0	1	1	β	β	...
↑								

<	0	1	0	1	1	β	β	...
	↑							

Para MT reconhecedoras, por exemplo, a linguagem $L = \{a^n b^n \ / \ n > 0\}$ é formada por palavras nas quais a primeira metade da palavra é formada somente por “a” em n quantidades e a segunda metade é formada somente por “b” também em n quantidades, ou seja, a quantidade de “a” inicial deve ser a mesma de “b”. Para garantir a mesma quantidade de “a” e “b” em uma MT podemos utilizar a própria fita de entrada da palavra como memória auxiliar para ir “contando” a quantidade de símbolos. Esta contagem de símbolos é feita de tal forma que, ao encontrar um “a” alteramos ele para “A”, que indica que este já foi contado, e então buscamos encontrar um “b” na fita, e quando este “b” é encontrado alteramos para “B” para indicar que também foi contado. Ao marcar um “a” e um “b” sabemos que a quantidade de símbolos é igual e podemos iniciar o processo novamente, buscando um a, marcando-o e depois buscando um b, e marcando-o também. Esta MT pode ser vista na Figura abaixo



Na MT, ao encontrar um “a” e alterarmos para “A”, passamos de s0 para s1, e estando no estado s1, avançamos o cabeçote para direita quando encontramos “a” ou “B” na fita sem alterá-la, e ao encontrar um “b”, alteramos para “B”, mudamos de estado para s2 e a partir deste ponto voltamos o cabeçote para começar uma nova busca por “a” e “b”. Este retorno do cabeçote é feito até encontrar um primeiro “A”, uma vez que ao encontrar este “A” indica que a esquerda já marcamos todos os “a”s e não precisamos voltar até o marcador de início da fita.

Assim, em s2, quando encontramos um “a” ou um “B” andamos o cabeçote para a esquerda sem alterar o símbolo e ao encontrar o primeiro “A”, movemos o cabeçote para direita, voltando para s0. Caso a fita contenha mais símbolos “a”, será passado pelos estados s1 e s2, voltando para s0 ao final do processamento. Quando todos os símbolos foram contabilizados (alterados para “A” e “B”), voltamos para o estado s0 e não encontramos mais “a” para ir ao estado s1, assim encontramos um “B”, avançamos com o cabeçote para a direita para garantir que não tem mais nenhum “b” a mais na fita, e ao encontrar o símbolo em branco (□), retornamos o cabeçote para a posição inicial (à direita do marcador de início de fita).

Para o processamento da palavra “aaabbbββ” é realizado da seguinte forma (a posição em negrito indica a posição do cabeçote, e cada linha da tabela indica o processamento da função de transição)

<	**a**	a	a	b	b	b	β	...
<	A	**a**	a	b	b	b	β	...
<	A	a	**a**	b	b	b	β	...
<	A	a	a	**b**	b	b	β	...
<	A	a	**a**	B	b	b	β	...
<	A	**a**	a	B	b	b	β	...
<	**A**	a	a	B	b	b	β	...
<	A	**a**	a	B	b	b	β	...

<	A	A	**a**	B	b	b	β	...
<	A	A	a	**B**	b	b	β	...
<	A	A	a	B	**b**	b	β	...
<	A	A	a	**B**	B	b	β	...
<	A	A	**a**	B	B	b	β	...
<	A	**A**	a	B	B	b	β	...
<	A	A	**a**	B	B	b	β	...
<	A	A	A	**B**	B	b	β	...
<	A	A	A	B	**B**	b	β	...
<	A	A	A	B	B	**b**	β	...
<	A	A	A	B	**B**	B	β	...
<	A	A	A	**B**	B	B	β	...
<	A	A	**A**	B	B	B	β	...
<	A	A	A	**B**	B	B	β	...
<	A	A	A	B	**B**	B	β	...
<	A	A	A	B	B	**B**	β	...
<	A	A	A	B	B	B	**β**	...
<	A	A	A	B	B	**B**	β	...
<	A	A	A	B	**B**	B	β	...
<	A	A	A	**B**	B	B	β	...
<	A	A	**A**	B	B	B	β	...
<	A	**A**	A	B	B	B	β	...
<	**A**	A	A	B	B	B	β	...
<	A	A	A	B	B	B	β	...
<	**A**	A	A	B	B	B	β	...

Quando processamos uma palavra pertencente à linguagem, tal como a “aaabbb” vista acima, paramos no estado final s5 na MT indicando que a palavra foi aceita para a linguagem da MT. Quando testamos uma palavra que não pertence à linguagem, “aab” por exemplo, encontramos o segundo “a” da palavra porém não encontramos o segundo “b” a partir do estado s1, assim não atingimos o estado final e não reconhecemos a palavra “aab” como pertencente à linguagem da MT. Do mesmo modo, para a palavra “abb", encontramos o primeiro “a” e primeiro “b”, ao voltar para s0 para procurar novos “a” não encontramos, passamos para o estado s3 pois temos “B” na fita, porém encontramos um “b”, que indica que não foi alterado pois não temos a mesma quantidade de “a” e de “b”, assim travamos o processamento no estado s3 e não aceitamos a palavra para a linguagem da MT, pois não temos a mesma quantidade de símbolos.

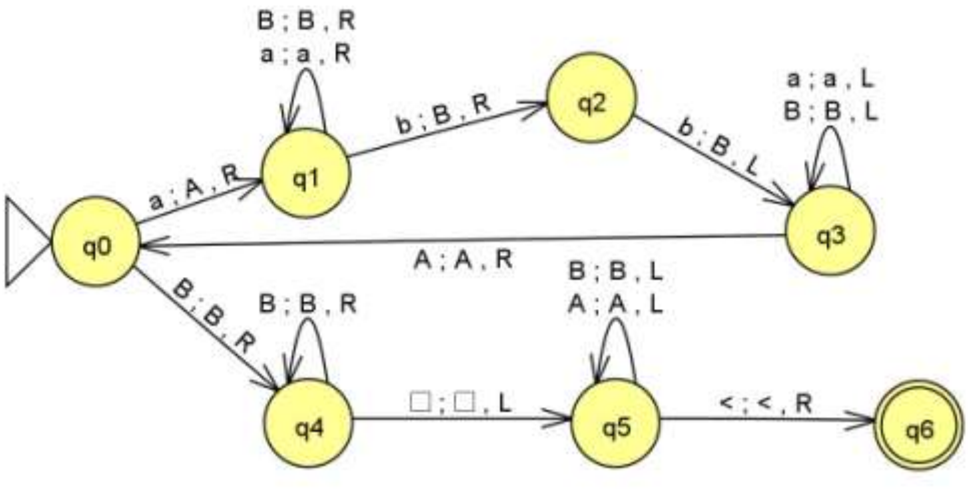
Ainda para a MT da Figura acima, a sua descrição formal pode ser representada por:

- E = {s0, s1, s2, s3, s4, s5}
- Σ = {a, b}
- δ é a função de transição

	a	b	A	B	<	□
s0	s1,A,D	X	X	s3,B,D	X	X
s1	s1,a,D	s2,B,E	X	s1,B,D	X	X
s2	s2,a,E	X	s0,A,D	s2,B,E	X	X
s3	X	X	X	s3,B,D	X	s4,□,E
s4	X	X	s4,A,E	s4,B,E	s5,<,D	X
s5	X	X	X	X	X	X

- i = s0
- F = {s5}
- □ é o símbolo de branco
- < é o marcador de início da fita
- Γ = {□, <, A, B}, são os símbolos auxiliares da fita

Para a linguagem L = $\{a^n b^{2n} / n > 0\}$ a MT precisa achar um “a”, e depois procurar dois “b”s pois na linguagem temos “b” elevado à “dois vezes n”. Assim, ao encontrarmos um “a” e dois “b”s, voltamos para o estado inicial e começamos tudo de novo até não encontrar mais o símbolo “a”, e também garantindo que não temos mais nenhum símbolo “b” na fita. A MT que reconhece essa linguagem pode ser vista na Figura abaixo



Para palavras que pertencem à linguagem, {abb, aabbbb, aaabbbbbb, aaaabbbbbbbb, ...} a MT para no estado final q6. Quando não encontramos dois “b”s para cada “a”, paramos o processamento no estado q2, e quando temos um “b” a mais do que a quantidade que devemos ter para reconhecer uma palavra, a MT para no estado q4 e não avança para o estado final q6, não aceitando corretamente a palavra que não pertence à linguagem.

Bibliografia

VIEIRA, NEWTON JOSÉ. **Introdução aos Fundamentos da Computação**. São Paulo. Pioneira Thomson Learning. 2006;

MENEZES, P. B.; DIVERIO, T. A.; **Teoria da Computação: Máquinas Universais e Computabilidade**; 3ª edição Bookman 2011

MENEZES, P. B. ; **Linguagens Formais e Autômatos**. 6ª edição. Ed. Artmed. 2011

SIPSER M. **Introdução à Teoria da Computação**. 2 ed. Cengage Learning.2007

MORET, B. M. **"Theory of Computation".** Addison-Wesley, 1998.

HOPCROFT, JOHN E.; ULLMAN, JEFFREY D.; MOTWANI, RAJEEV **Introdução à Teoria de Autômatos, Linguagens e Computação** Ed.Campus 2002

SILVA, FLAVIO SOARES CORRÊA; MELO, ANA CRISTINA VIEIRA; **Modelos Clássicos de Computação** Ed. Thomson 1ª Edição 2006