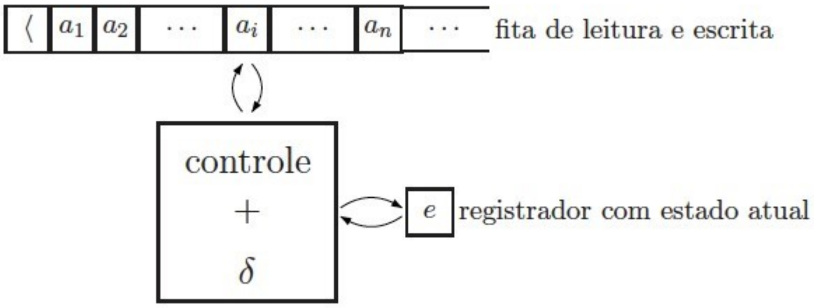
Máquinas de Turing

Na aula anterior, vimos novamente as limitações dos formalismos para processamento de linguagens. Embora a adição da pilha tenha aumentado o poder computacional dos APs com respeito aos AFs, as limitações de poder acessar somente o elemento do topo da pilha e a leitura unidirecional da fita de entrada restringem aquilo que podemos computar nessas máquinas.

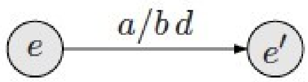
Na década de 1930, Alan Turing propôs um novo formalismo com capacidade superior ao das máquinas que vimos anteriormente. A proposta de Turing foi de que a fita de entrada funcionasse também como uma memória ilimitada. Inicialmente, essa memória continha a palavra de entrada a ser reconhecida, porém o acesso à fita poderia ser feito sequencialmente em ambas as direções, sendo permitida a leitura e escrita de símbolos. Esse é o modelo teórico seguido pelas máquinas atuais. A figura abaixo mostra essa arquitetura de forma esquemática.



Note que a arquitetura é muito próxima de autômatos finitos. A grande diferença nesse caso é que a fita de entrada é usada tanto para leitura quanto escrita, servindo como memória adicional, e o cabeçote de leitura/escrita pode ser movimentado tanto para direita quanto para a esquerda (embora, consideramos nesse modelo que a fita é limitada à esquerda; i.e. ela possui um marcador início).

Dessa forma, a função de transição deve levar em consideração o símbolo sob o cabeçote e o estado armazenado no registrador, para, então, transitar para um novo estado, escrevendo um novo símbolo na fita de entrada e movimentando o cabeçote em alguma direção (esquerda ou direita).

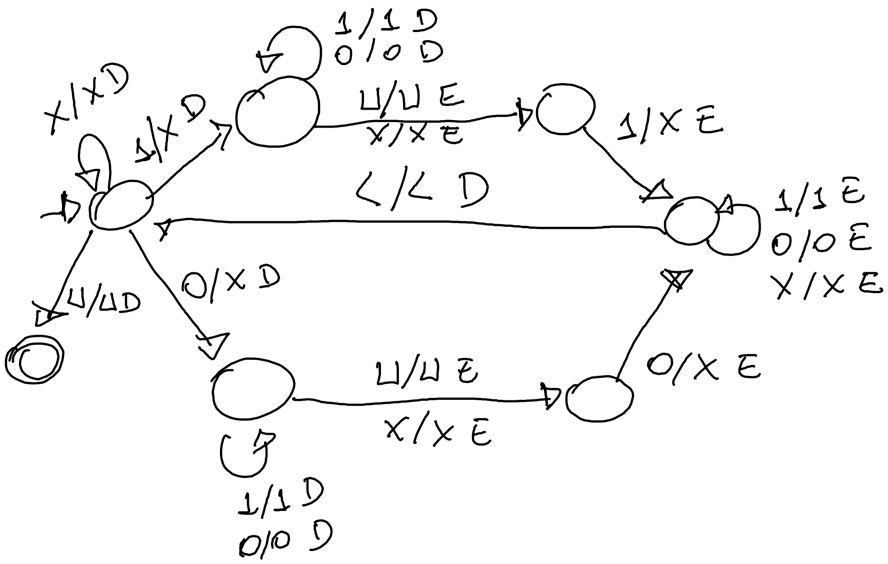
As transições da máquina de Turing são da seguinte forma:



Em que:

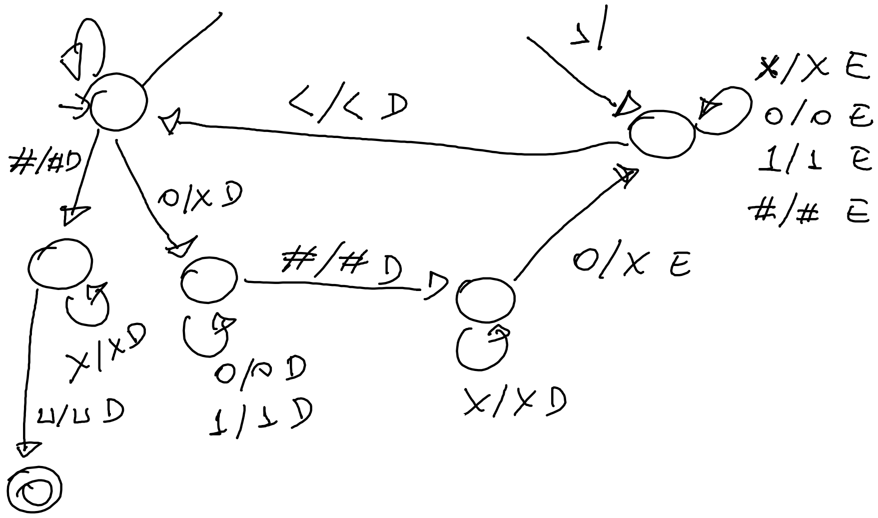
- $e, \quad e'$ são estados da máquina;
- a é o símbolo atualmente sob o cabeçote de leitura/escrita
- b é o símbolo a ser escrito na fita
- d é a direção em que o cabeçote deve ser movido.

☆ Formalmente, uma máquina de Turing determinística é uma sêxtupla $MT = (Q, \Sigma, \quad \Gamma, \quad \delta, \quad i, \quad F)$ em que:



1/1 D
0/0 D
X/X D
1/X D
⊔/⊔ E
X/X E
1/X E
</< D
1/1 E
0/0 E
X/X E
⊔/⊔D
0/X D
⊔/⊔ E
X/X E
0/X E
1/1 D
0/0 D


0/0 D
X/X D
#/#D



- Q, i, F e Σ são como nos AFs;
- Γ é o alfabeto da fita (conjunto de símbolos que podem ser escritos na fita $\Sigma \subseteq \Gamma$; além disso inclui dois símbolos especiais que não fazem parte do alfabeto da linguagem $\langle$ e $\sqcup$ que denotam respectivamente o início da fita e célula em branco)
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D\}$ é a função de transição, uma função parcial, que, para um dado estado e símbolo da fita, indica o próximo estado, símbolo a ser escrito e direção de movimentação do cabeçote.

Inicialmente, a palavra de entrada está escrita a partir da primeira posição após o início da fita e todos os símbolos após a palavra são espaço em branco. O cabeçote se encontra posicionado sob o primeiro símbolo da palavra.

Outro ponto é $\delta(e, \langle\ ) = [e', \langle, D]$. Em outras palavras, chegando-se no início da fita, o marcador deve ser reescrito e o cabeçote movido obrigatoriamente para direita.

Segue um exemplo de uma MT.

Exemplo: Construa uma MT para reconhecer a linguagem $\{xx^R \mid x \in \{0,1\}^*\}$.

Uma configuração instantânea de uma MT é um par $[e, x\underline{a}y]$ em que:

- e é o estado atual da máquina
- $x \in \Gamma^*$ é o conteúdo da fita à esquerda do cabeçote
- $a \in \Gamma$ é o símbolo sob o cabeçote
- $y \in \Gamma^*$ é o conteúdo da fita à direita do cabeçote, excluindo-se os espaços em branco.

A configuração inicial de uma máquina seria $[i, \langle \underline{a_1} a_2 \ldots a_n]$, considerando que a palavra de entrada fosse $a_1 a_2 \ldots a_n$. Se a palavra de entrada for λ, então representamos a configuração inicial por $[i, \langle \underline{\sqcup}]$.

A relação *resulta em*, para designar que uma configuração instantânea resulta em outra, é definida por:

- $\delta(e,a) = [e', b, D] \Rightarrow [e, \quad x\underline{a}cy] \ \vdash \ [e', \quad xb\underline{c}y] \vee [e, x\underline{a}] \ \vdash \ [e', xb\underline{\sqcup}]$
- $\delta(e,a) = [e', b, E] \Rightarrow [e, \quad xc\underline{a}y] \ \vdash \ [e', \quad x\underline{c}by]$
- Se $\delta(e,a)$ for indefinido, então a máquina para a computação na configuração $[e, x\underline{a}y]$.

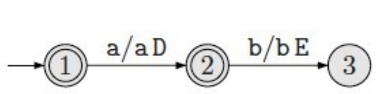
Com base nessa definição de configuração instantânea e a relação entre elas, podemos definir a linguagem aceita por uma máquina de Turing.

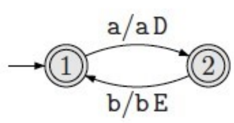
Seja a MT $M = (Q, \Sigma, \Gamma, \delta, i, F)$. A linguagem reconhecida pela máquina é:

- $L(M) = \quad \{w \in \Sigma^* | \quad [i, \langle \underline{w}] \ \vdash^* \ [e, x\underline{a}y] \wedge e \in F \quad \wedge \quad \delta(e,a) \quad é \quad indefinido\}$

☆ Repare que a máquina de Turing pode entrar em loop. Portanto, para que ela reconheça uma palavra, ela deve terminar sua computação, isto é parar, em um estado final. Caso ela entre em loop ou pare em estado não final, ela não reconhece a palavra.

Por exemplo, as máquinas abaixo reconhecem a mesma linguagem (a linguagem de a's e b's que não possui o prefixo ab). A da esquerda para com qualquer palavra, enquanto a da direita entra em loop quando não reconhece uma palavra.





Isso implica que, ao contrário das máquinas anteriores, a MT não precisa consumir todos os símbolos de uma palavra para decidir se a reconhece ou não. Tão logo ela detecte um padrão que leve à aceitação ou não de uma palavra, ela pode interromper a computação.

Além disso, é perceptível que a MT é capaz de reconhecer tanto LRs quanto LLCs. Ela pode simular um AF simplesmente lendo a fita da esquerda para direita e proibindo movimentações à esquerda. Da mesma forma, um AP também pode ser simulado por ela. Nesse caso, podemos usar a continuação da fita após o término da palavra de entrada como pilha. Embora seja trabalhoso construir esse simulador, percebe-se que é perfeitamente possível.

As máquinas de Turing dão origem a duas classes de linguagens:

- **Linguagens recursivamente enumeráveis** (LRE) para as quais existe uma MT que a reconhece. Também são chamadas de **Turing-reconhecíveis**.
- **Linguagens recursivas** (LRec) para as quais existe uma MT que a reconheça e

sempre pare com qualquer palavra do alfabeto. Essas linguagens são também chamadas de **Turing-decidíveis**.

Como as MTs reconhecem LRs e LLCs, tem-se que $LR \subset LLC \subset LRec \subset LRE$. Essa hierarquia das linguagens é conhecida como **Hierarquia de Chomsky**. Há evidências que as máquinas de Turing possuem o maior poder computacional e que, portanto, o universo computável se resume às LREs. Retomaremos essa discussão mais à frente no curso.

Segue outro exemplo de MT.

Exemplo: Construa uma máquina de Turing que reconheça $\{w \# w \mid w \in \{0,1\}^*\}$.