

Máquina de Post.

Criada em 1936 por Emil Leon Post.



Em 1936, antes do aparecimento dos computadores, os eminentes matemáticos Alan Turing e Emil Post publicaram, respectivamente, os artigos

“Sobre os números computáveis com a aplicação ao problema da **solucionabilidade**”.

Estes dois trabalhos propõem um conceito rigoroso (matemático) de *algoritmo*.

O computador abstrato (ou matemático) proposto por Post, é mais simples, que o de Turing no que respeita às instruções elementares. Esta simplicidade tem um custo: os algoritmos do computador matemático de Post exigem, em geral, mais memória e maior quantidade de passos do que os correspondentes algoritmos de Turing.

Definição da Máquina de Post

Uma *Máquina de Post* é uma tripla: $M = (\Sigma, D, \#)$

onde:

Σ *alfabeto de símbolos de entrada;*

D *programa ou diagrama de fluxos* construído a partir de componentes elementares denominados partida, parada, desvio e atribuição;

$\#$ *símbolo auxiliar.*

Constituição de uma máquina de Post.

Uma Máquina de Post consiste de duas partes:

Variável X.

Trata-se de uma variável do tipo fila e é utilizada como entrada, saída e memória de trabalho.

A variável X não possui tamanho nem limite fixos. Seu comprimento é igual ao comprimento da palavra corrente armazenada.

Os símbolos podem pertencer ao alfabeto de entrada ou a $\{ \# \}$, único símbolo auxiliar.

Inicialmente, o valor de X é a palavra de entrada. Caso X não contenha símbolos, a entrada é vazia, representada por ϵ .

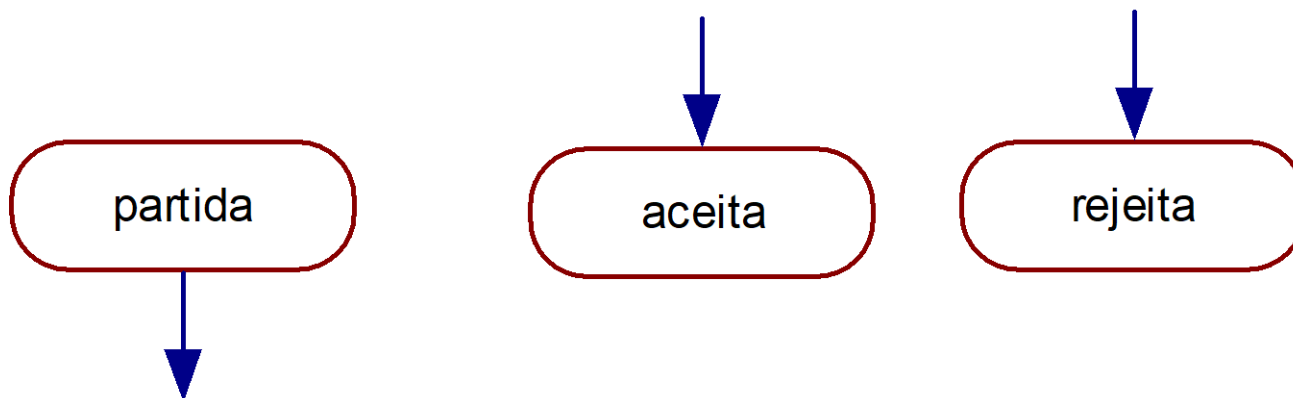
Programa.

É uma seqüência finita de instruções, representado como um diagrama de fluxos (espécie de fluxograma), no qual cada vértice é uma instrução.

As instruções podem ser de quatro tipos: **partida, parada, desvio (leitura com teste) e atribuição.**

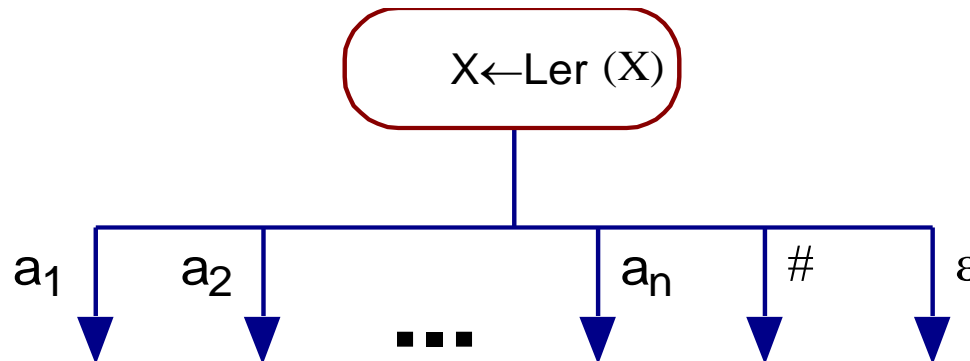
Componentes elementares de um diagrama de fluxos.

- **Partida.** Existe somente uma instrução de início em um programa.
- **Parada.** Existem duas alternativas de instruções de parada em um programa, uma de aceitação e outra de rejeição.



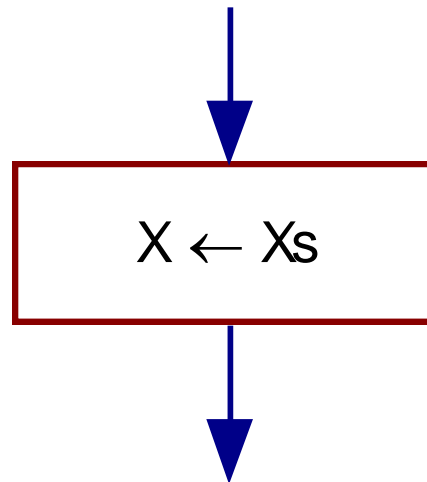
Desvio (ou leitura com teste). $X \leftarrow \text{ler}(X)$


- denota o comando que lê o símbolo mais à esquerda da palavra armazenada em X , retirando o primeiro símbolo.
- É uma instrução composta de uma leitura do símbolo à esquerda (início da fila), excluindo-o da fila e desviando o fluxo do programa de acordo com o símbolo lido;
- fluxo do programa é determinado de acordo com o símbolo mais à esquerda da palavra.
- deve ser prevista a possibilidade de X conter a palavra vazia.
- Portanto, é um desvio condicional, e trata-se de uma função total, estando definida para todos os valores do domínio.
- Se o cardinal de Σ é n , então existem $n+2$ arestas de desvios condicionais, pois se deve incluir as possibilidades $\#$ e ϵ .



Atribuição. $X \leftarrow Xs$

- É uma instrução de concatenação, gravando o símbolo indicado (pertencente a $\Sigma \cup \{ \# \}$) à direita da palavra armazenada na variável X (fim da fila).
- A operação de atribuição é representada a seguir, supondo que $s \in \Sigma \cup \{ \# \}$.

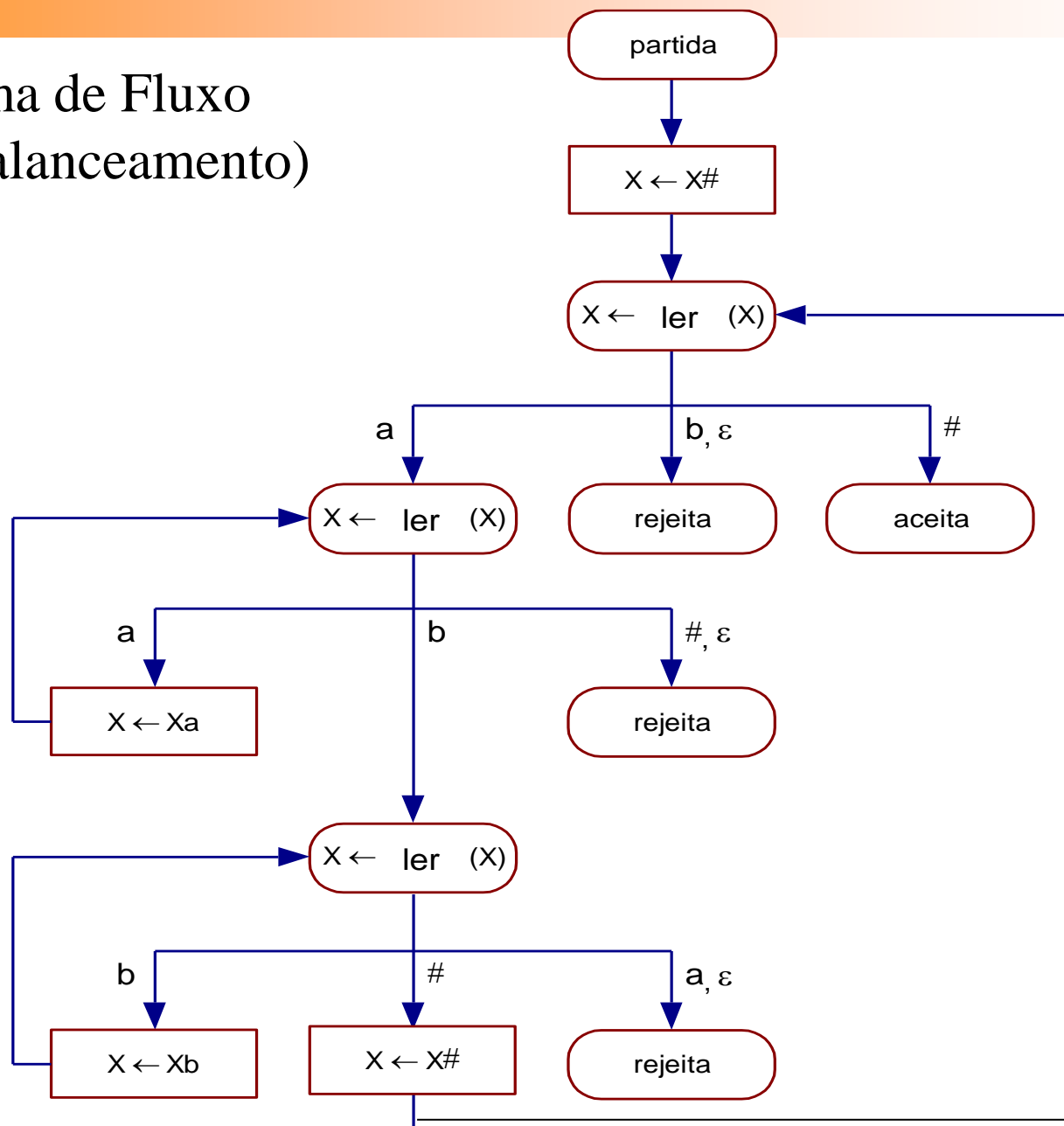




Exemplo de uma máquina de Post. (Duplo balanceamento).

- Considere a seguinte linguagem $\text{Duplo_Bal} = \{ a^n b^n \mid n \geq 0 \}$.
- A Máquina de Post: $\text{Post_Duplo_Bal} = (\{ a, b \}, D, \#)$.
- $\text{ACEITA}(\text{Post_Duplo_Bal}) = \text{Duplo_Bal}$
- $\text{REJEITA}(\text{Post_Duplo_Bal}) = \Sigma^* - \text{Duplo_Bal}$
- $\text{LOOP}(\text{Post_Duplo_Bal}) = \emptyset$.

Diagrama de Fluxo (Duplo Balanceamento)



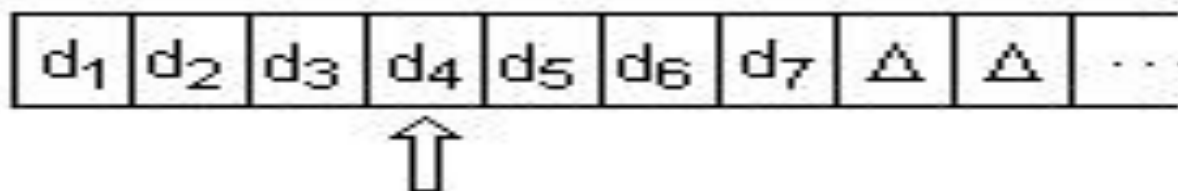
Explicação do Diagrama de Fluxo (duplo balanceamento)

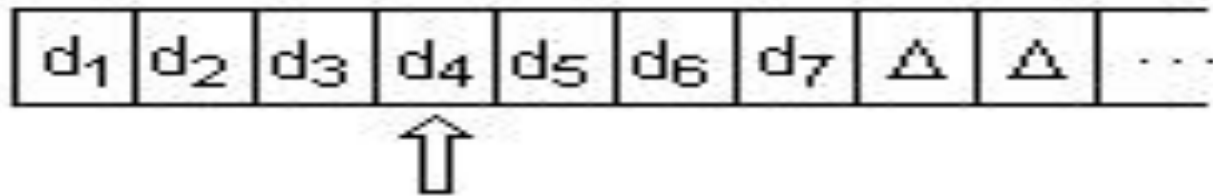
- algoritmo lê e remove o primeiro símbolo **a**;
- realiza uma varredura circular em busca do correspondente **b**.
- Essa varredura é realizada através de sucessivas leituras (e remoções), armazenando o símbolo lido à direita de **X**.
- Ao encontrar o **b**, este é removido, e uma nova varredura circular é realizada até o fim da palavra de entrada (identificado pelo símbolo auxiliar **#**, atribuído a **X** no início do processamento).
- Este ciclo é repetido até restar a palavra vazia ou ocorrer alguma condição de rejeição.

Equivalência: Post x Turing

- **Teorema** : A classe das Máquinas de Post sobre Σ têm a mesma potência computacional que a classe das Máquinas de Turing sobre Σ isto é, para toda Máquina de Post sobre Σ existe uma equivalente Máquina de Turing sobre Σ e vice-versa.

- Prova : 1ª toda Máquina de Turing pode ser simulada por uma Máquina de Post sobre $\Sigma = \{a,b\}$
- O conteúdo da fita e posição da cabeça da fita em qualquer estágio de computação da Máquina de Turing são expressas como os valores de x na Máquina de Post.
- Exemplo: Se em algum estágio de computação da Máquina de Turing a fita é da forma:



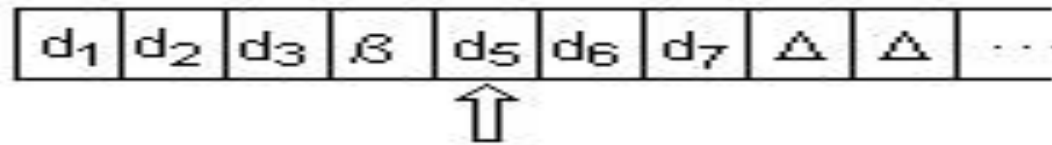


onde $d_i \in \Sigma \cup \{ \Delta \}$ e a cabeça de leitura lê o símbolo d_4 , então esta situação é expressa na Máquina de Post por:

$$x = d_4 d_5 d_6 d_7 \# d_1 d_2 d_3$$

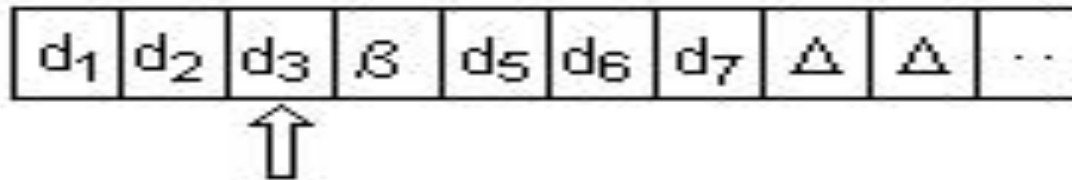
ou seja, a cadeia infinita de Δ 's é ignorada de tal forma que o símbolo mais a esquerda de x é aquele lido pela cabeça de leitura da Máquina de Turing. O símbolo especial $\#$ é usado para indicar o ponto de quebra da string.

- Se $x = d_4 d_5 d_6 d_7 \# d_1 d_2 d_3$ e a próxima instrução da Máquina de Turing é (d_4, β, R) ; então o conteúdo de x é trocado na Máquina de Post para



$x = d_5 d_6 d_7 \# d_1 d_2 d_3 \beta$

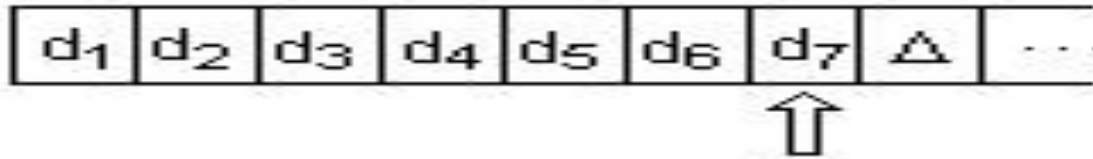
- Se $x = d_4 d_5 d_6 d_7 \# d_1 d_2 d_3$ e a próxima instrução for (d_4, β, L) então



$x = d_3 \beta d_5 d_6 d_7 \# d_1 d_2$

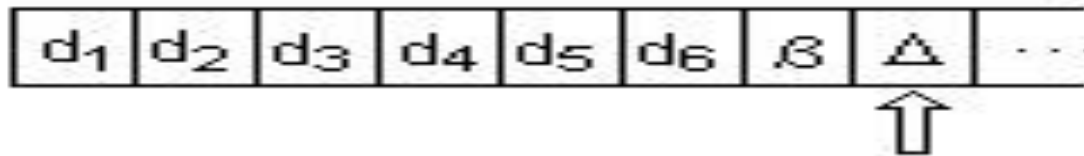
Entretanto existem 2 casos especiais :

1º caso :



Quando $x = d_7 \#d_1 d_2\dots d_6$

e a próxima instrução é $(d_7 \beta, R)$

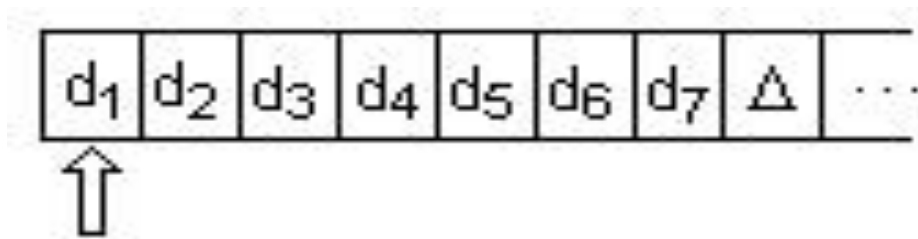


$\#d_1 d_2\dots d_6 \beta$ ($\#$ é o símbolo mais a esquerda de x)

Isto significa que o próximo símbolo a ser lido pela cabeça da Máquina de Turing é o 1º Δ (branco) à direita de d_7 . Portanto neste caso, troca-se x por $\Delta \#d_1\dots d_6 \beta$.

2º caso :

Quando se atinge uma situação onde $x = d_1...d_7\#$ ($\#$ é o símbolo mais a direita de x)



e a próxima instrução da Máquina de Turing é (d_1, β, L)

Este caso acontece quando a cabeça de leitura da fita da Máquina de Turing lê o símbolo mais a esquerda da fita e é pedido para se fazer movimento para a esquerda. Portanto, neste caso deve-se ir para uma parada Reject na Máquina de Post.

A Máquina de Post obtida é sobre $\Sigma \cup \{ \Delta \}$ e não sobre Σ .

(\leq volta) Para toda Máquina de Post, existe uma Máquina de Turing equivalente. (é bem mais simples).

O valor corrente de x durante a computação da Máquina de Post; por exemplo $x = d_1 d_2 d_3 d_4 \# d_5 d_6$ é expressa na fita da Máquina de Turing como:

$\Delta \Delta...d_1 d_2 d_3 d_4 \# d_5 d_6 \Delta \Delta...$

isto é necessário para que a Máquina possa atingir a 1ª letra da string de entrada através de movimentos à esquerda sem encontrar um halt reject.



ALUNOS:

Douglas.

Fernando (Kill).

Larissa.

Nelson.

Wellington.

CIÊNCIA DA COMPUTAÇÃO 3^a ANO
(NOTURNO)