

Equivalência Norma-Turing (1/3)

- Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, \beta, F, \phi)$ simulado por Norma
 - Fita: codificada como arranjo unidimensional em X ;
 - Símbolo de uma célula: codificado como natural. $\Sigma = \{a_1, a_2, \dots, a_n\}$, o símbolo a_i é codificado como o natural i , e os símbolos especiais β e ϕ como zero e $n+1$, respectivamente.
 - Estados: para cada estado em Turing existe uma correspondente instrução rotulada em Norma. O rótulo inicial é 0 e para cada $q_i \in F$, f é rótulo final;
 - Estado Corrente: o estado corrente de M é simulado em Norma usando o registrador Q , o qual assume valores em $\{0, 1, \dots, n\}$ (correspondendo aos estados q_0, q_1, \dots, q_n);

Norma simula Turing (2/3)

Cabeça da Fita: é simulada usando o registrador C de Norma, o qual contém a posição corrente do arranjo em X e é inicializado com o valor 1.

Função Programa: simulada por um programa Norma

$\Pi(q_u, ar) = (q_v, as, m)$ onde m assume valores em $\{E, D\}$

u: **faça** $A := 2^Q \times 3^{X(C)}$ **vá para** End_A
...
a: **faça** $X(C) := s$ **vá para** a_1 grava na fita
 a_1 : **faça** ad_C **vá para** a_2 move a cabeça
 a_2 : **faça** $Q := v$ **vá para** End_Q novo estado

Norma simula Turing (3/3)

Rótulo final:

f: **faça** $Y := X$ **vá para** fim

...

fim:

Codificação: o conteúdo da fita é codificado em X como arranjo unidimensional.

Decodificação:

É o inverso da codificação acima.

Turing simula Norma (1/3)

• Norma simulada por Turing

- Registrador X: codificado em unário na células pares da fita de M .
- Registrador Y: o registrador Y é armazenado na fita em unário, mas nas células ímpares (excetuando-se a primeira, que contém o marcador de início de fita ϕ).
- Rótulos: cada rótulo r corresponde um estado q_r de Turing. Aos rótulos inicial e finais correspondem os estados inicial e finais, respectivamente.

Turing simula Norma (2/3)

Programa:

ADIÇÃO r: faça adK vá_para s

SUBTRAÇÃO r: faça subK vá_para s

TESTE r: se zeroK vá_para s senão vá_para t

Turing simula Norma (3/3)

Codificação: conteúdo inicial do registrador X é codificado em unário nas células pares da fita de M;

Decodificação: é o inverso da codificação acima.

Máquina de Post

Variável X: variável do tipo fila (uma leitura exclui o dado lido), sem limite fixo, cujo valor inicial é a palavra de entrada.

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

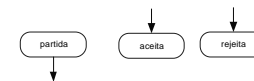
Σ alfabeto de símbolos de entrada;

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

símbolo auxiliar.

Componentes elementares de um diagrama de fluxos

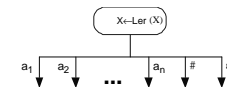
- **Partida e Parada (aceitação ou rejeição)**



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

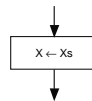
- É uma instrução composta de uma leitura do símbolo à esquerda (início da fila), excluindo-o da fila e desviando o fluxo do programa de acordo com o símbolo lido;

- Se o cardinal de Σ é n, então existem n+2 arestas de desvios condicionais, pois se deve incluir as possibilidades # e ϵ .



Componentes Elementares Post

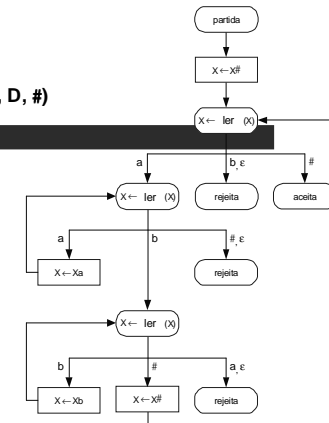
- **Atribuição.** $X \leftarrow Xs$



Em um diagrama de fluxos existe 1 partida, mas podem existir diversas instruções de parada (aceitação ou rejeição)

X pode conter a palavra vazia ϵ

Exemplo Post_Duplo_Bal = ({ a, b }, D, #)



Post simula Turing (1/3)

- **Fita:** $\epsilon \ a_1 \ a_2 \ a_3 \ a_4 \ \dots \ a_n \ \beta \ \dots$

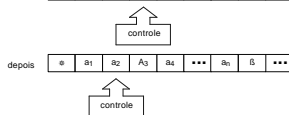


$X = a_3 a_4 \dots a_n \# a_1 a_2$

- **Movimento para a Esquerda:**

antes $\epsilon \ a_1 \ a_2 \ a_3 \ a_4 \ \dots \ a_n \ \beta \ \dots$

$X = a_3 a_4 \dots a_n \# a_1 a_2$



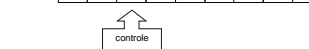
$X = a_2 A_3 a_4 \dots a_n \# a_1$

Post simula Turing (2/3)

- **Movimento para a Direita:**

antes $\epsilon \ a_1 \ a_2 \ a_3 \ a_4 \ \dots \ a_n \ \beta \ \dots$

$X = a_2 a_3 a_4 \dots a_n \# a_1$



depois $\epsilon \ a_1 \ A_2 \ a_3 \ a_4 \ \dots \ a_n \ \beta \ \dots$

$X = a_3 a_4 \dots a_n \# a_1 A_2$

Post simula Turing (3/3)

- **Estados**
 - *Estado Inicial.* Simulado pela instrução partida;
 - *Estados Finais.* Simulados pela instrução aceita;
 - *Demais Estados.* Cada estado corresponde a uma instrução desvio (leitura com teste);
- **Condições de Rejeição:** simuladas em Post por rejeita.

Turing simula Post (1/2)

- **Variável X**

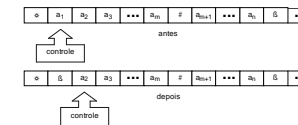
$X = a_1 a_2 a_3 \dots a_m \# a_{m+1} \dots a_n$



- **Desvio:** $X \leftarrow \text{ler}(X)$

conteúdo da variável X:

$X = a_1 a_2 a_3 \dots a_m \# a_{m+1} \dots a_n$



leitura e remoção resulta em:
 $X = a_2 a_3 \dots a_m \# a_{m+1} \dots a_n$

Turing simula Post (2/2)

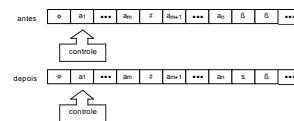
- **Atribuição:** $X \leftarrow Xs$

Para o conteúdo de X

$X = a_1 \dots a_m \# a_{m+1} \dots a_n$

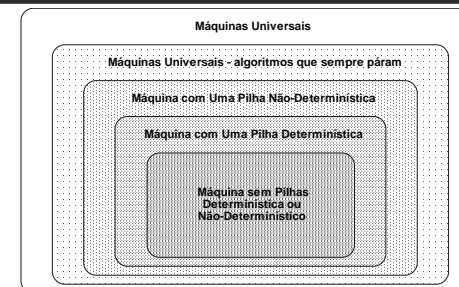
resulta em

$X = a_1 \dots a_m \# a_{m+1} \dots a_n s$



- **Partida:** estado inicial
- **Aceita:** estado final
- **Rejeita:** uma condição excepcional de parada (como um movimento inválido).

Classes de Máquinas



Classes de Linguagens

Linguagens Regulares: Máquinas sem Pilha.

- editores de texto, protocolos de comunicação.
- não é possível fazer qualquer tipo de balanceamento de tamanho não-predefinido
- tempo de reconhecimento de uma palavra é diretamente proporcional ao comprimento da entrada (eficiente)

Linguagens Livres do Contexto Determinísticas: Autômatos Determinísticos com Uma Pilha.

- não é possível reconhecer a linguagem $\{ ww^r \mid w \text{ pertence a } \{ a, b \}^* \}$
- tempo de reconhecimento de uma entrada é diretamente proporcional ao dobro do tamanho da entrada;

Classes de Linguagens

Linguagens Livres do Contexto: Autômatos Não-Determinísticos com Uma Pilha

- Algol e Pascal.
- linguagens muito simples não pertencem a essa classe $\{ a^n b^m c^n \mid n \geq 0 \}$
 $\{ ww \mid w \text{ é palavra sobre os símbolos } a \text{ e } b \}$
- tempo de processamento proporcional ao tamanho da entrada elevado ao cubo

Classes de Linguagens

Linguagens Recursivas:

- todas as linguagens que podem ser reconhecidas mecanicamente e existe um algoritmo de reconhecimento que sempre pára para qualquer entrada.
- inclui a grande maioria das linguagens aplicadas.
- ineficientes, tanto em termos de tempo de processamento como de recursos de memória

Linguagens Enumeráveis Recursivamente: Máquinas Universais

- todas as linguagens que podem ser reconhecidas mecanicamente.