Máquinas de Turing

Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática – CEEI

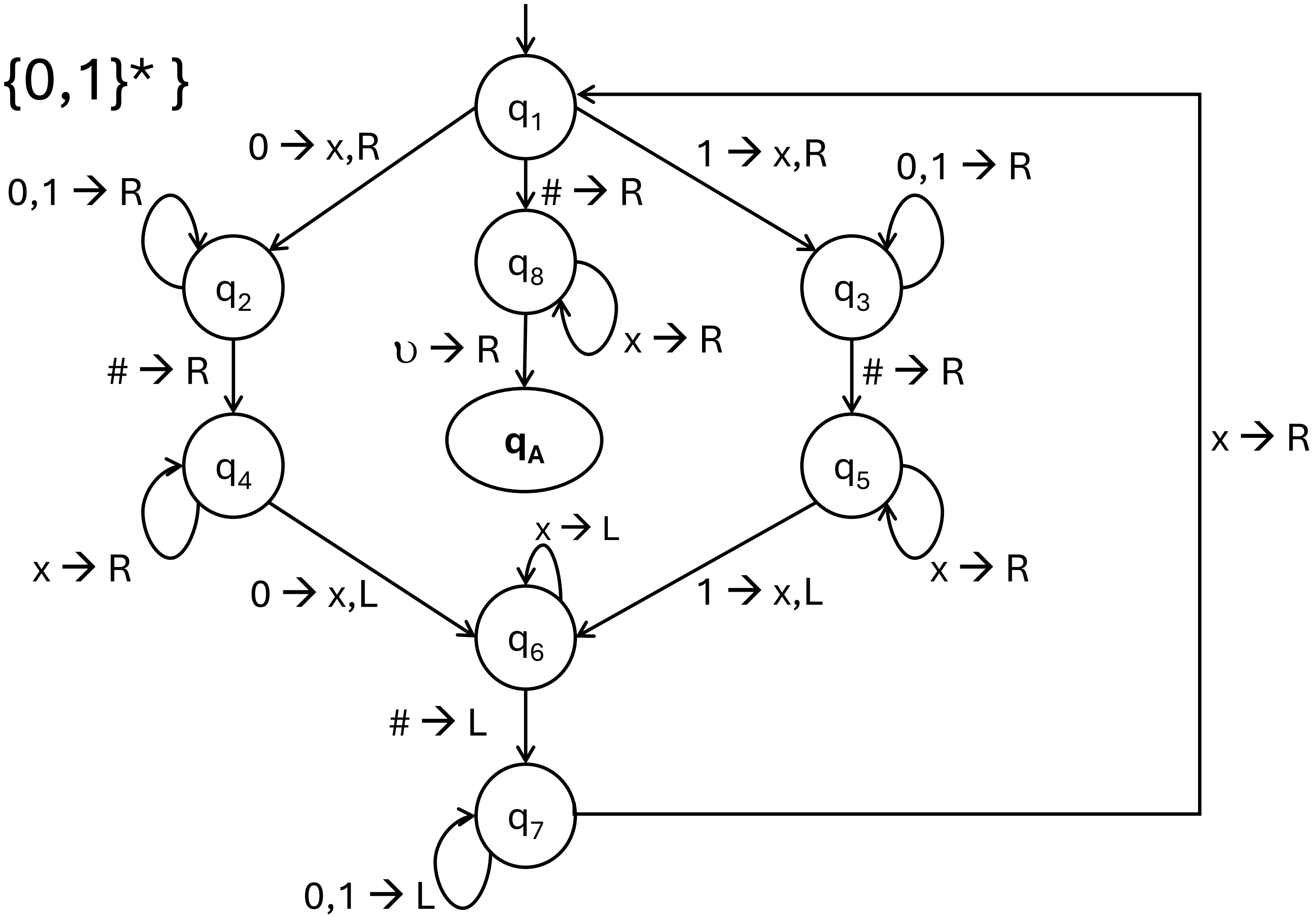
Departamento de Sistemas e Computação – DSC

Professor: Andrey Brito

Período: 2023.2

Exemplo de MT

$L = \{ w\#w \mid w \in \{0,1\}^* \}$



Transições omitidas levam para o estado de rejeição.

Outro exemplo

- E como seria uma MT para uma linguagem que tem as palavras no formato $0^n1^n2^n$, com $n > 0$?

Definição formal

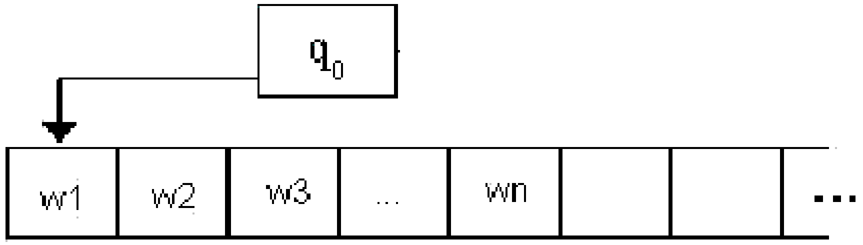
- Uma MT é uma 7-tupla <Q, Σ, Γ, δ, q_0,q_A,q_R>
  - Q é o conjunto de estados
  - Σ é o alfabeto de entrada (não contendo um símbolo especial B – "branco")
  - Γ é o alfabeto da fita onde B $\in \Gamma$ e $\Sigma \subset \Gamma$
  - δ: Q x $\Gamma \rightarrow$ Q x Γ x {E,D} é a função de transição
  - $q_0 \in$ Q é o estado inicial
  - $q_A \in$ Q é o estado de aceitação
  - $q_R \in$ Q é o estado de rejeição

Configurações

- Uma representação do "status" atual da máquina (estado + memória)
  - Se a MT fosse um programa, isso seria onde o programa está na execução dele e qual o valor de todos os objetos (ou seja, o que você vê em um depurador)

- Inicial: q_0w
  - Controle (um autômato) no estado q_0
  - O cabeçote no começo

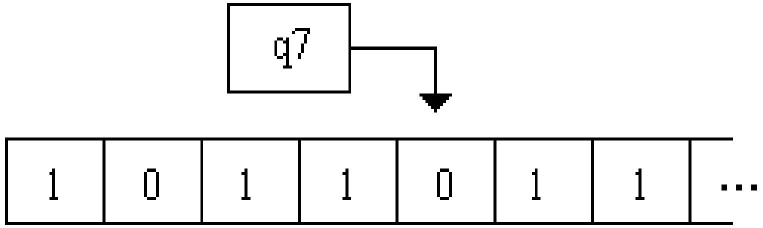
$q_0\ w \quad \cong$



Configurações

- Inicial: q_0w
- Intermediária: uq_iv
  - $q_i \in Q$
  - $u,v \in \Gamma^*$

$1011q_7011 \quad \cong$



Outro exemplo de MT

$L = \{ 0^{2^n} : n \geq 0 \}$

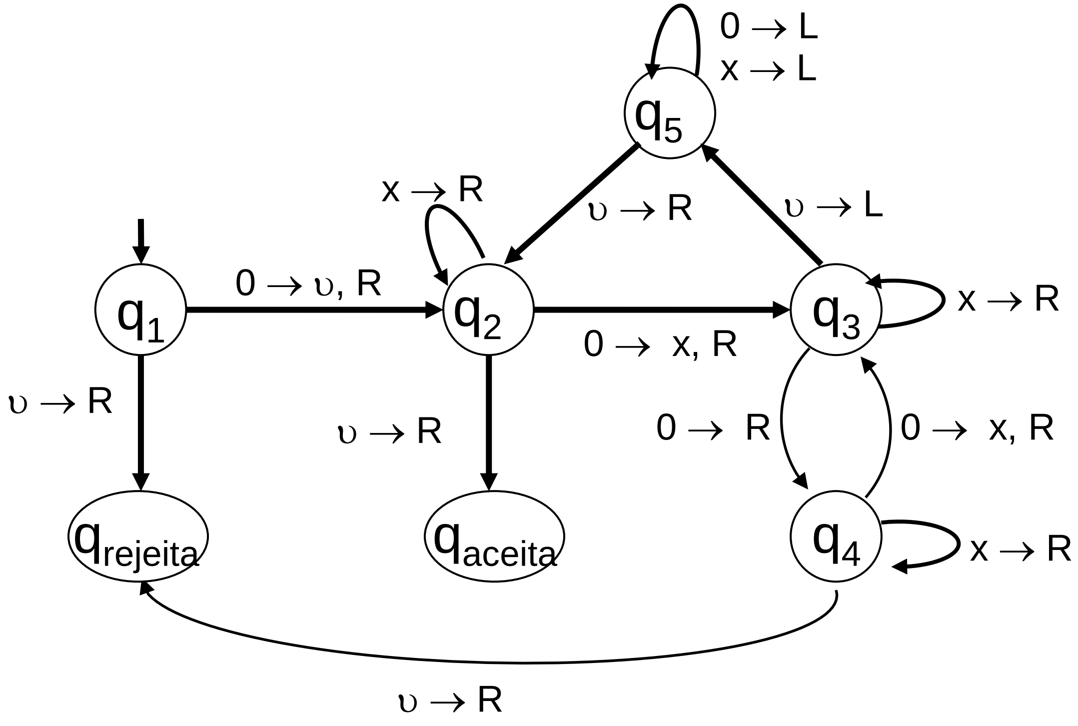
M:

$Q = (q_1, q_2, q_3, q_4, q_5, q_{aceita}, q_{rejeita})$

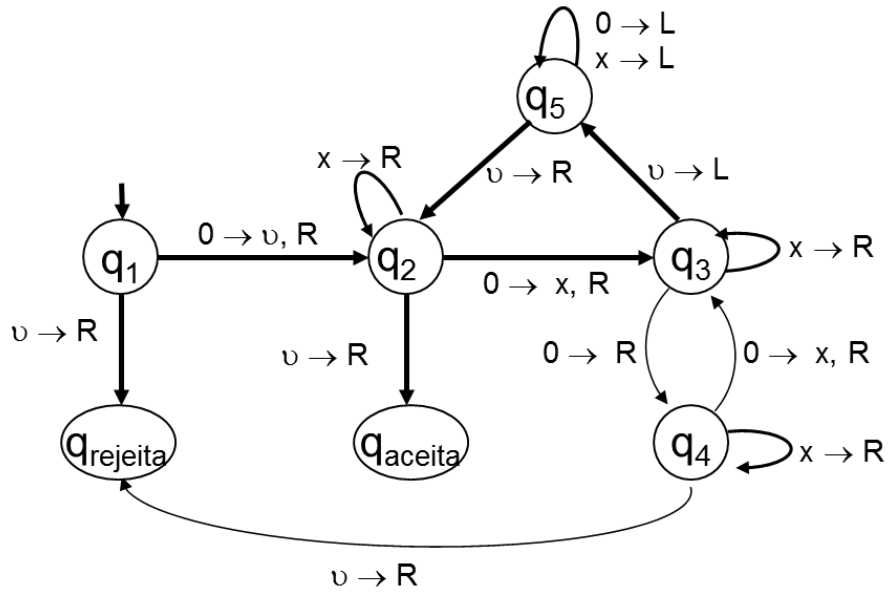
$\Sigma = \{0\}$

$\Gamma = \{0, x, \upsilon\}$

$q_0 = q_1$



Execução com entrada 0000: sequência de configurações



$q_1 0000$
$\upsilon q_2 000$
$\upsilon x q_3 00$
$\upsilon x 0 q_4 0$
$\upsilon x 0 x q_3 \upsilon$
$\upsilon x 0 q_5 x \upsilon$
$\upsilon x q_5 0 x \upsilon$

$\upsilon q_5 x 0 x \upsilon$
$q_5 \upsilon x 0 x \upsilon$
$\upsilon q_2 x 0 x \upsilon$
$\upsilon x q_2 0 x \upsilon$
$\upsilon x x q_3 x \upsilon$
$\upsilon x x x q_3 \upsilon$
$\upsilon x x q_5 x \upsilon$

$\upsilon x q_5 x x \upsilon$
$\upsilon q_5 x x x \upsilon$
$q_5 \upsilon x x x \upsilon$
$\upsilon q_2 x x x \upsilon$
$\upsilon x q_2 x x \upsilon$
$\upsilon x x q_2 x \upsilon$
$\upsilon x x x q_2 \upsilon$

$\upsilon x x x \upsilon q_{aceita}$

Sequência de configurações

- A configuração C_1 **produz** a configuração C_2 se a máquina pode ir de C_1 a C_2 em um único passo

- Por exemplo, seja $C_1 = ua\, \mathbf{q_i}\, bv$ e $C_2 = u\, \mathbf{q_j}\, acv$
  - u e v $\in \Gamma^*$; a, b e c $\in \Gamma$
  - C_1 produz C_2 se $\delta(q_i,b) = (q_j,c,E)$

- Note que se a máquina está na extremidade esquerda da fita $\mathbf{q_i}\, bv$ produziria $\mathbf{q_j}\, cv$

Aceitação

- Uma máquina de Turing M aceita a entrada w se uma sequência de configurações $C_1,C_2,\ldots,C_k$ existe onde
  - C_1 é a configuração inicial de M na entrada w,
  - Cada C_i produz C_{i+1}, e
  - C_k é uma configuração de aceitação

- O conjunto de entradas aceitas por M é L(M), a linguagem de M

Variações da MT

Calculadora de função

- Ao invés de SIM ou NÃO, usar a MT para fazer algo mais
- Já fizemos alguns cálculos, por exemplo f(x) = Verdadeiro, se $x = 2^n$ com $n >= 0$, e f(x)=Falso, caso contrário
    - Mas agora queremos que a MT resolva uma função, deixando o resultado na fita
    - Computar uma função f com uma MT M: "para todo $w \in \Sigma^*$, $M(w) = f(w)$"
    - Função recursiva: existe uma MT que computa a função
- Variante de MT, termina deixando o resultado da computação na fita e entrando no estado de aceitação
- (Aqui) Especialmente útil como parte de uma MT maior

Exemplos – Calculadora de função

- Funções de strings:
  - Mover um pedaço da palavra (ex., 0101# → #0101)
  - Ou achar o ponto central de uma palavra (ex., 010010 → 010#010)

Exemplos – Calculadora de função

- Fazer um cálculo?
  - $f(x) = x + 1$
  - $f(x,y) = x/2+y$

Brainstorming – Calculadora de função

- Fazer um cálculo
  - f(x) = x + 1
    - A entrada, um inteiro x, vem escrita na fita na forma de 0s
    - Codificação: $|w| = x$ - Solução: acrescente um 0 a mais na entrada
  - f(x,y) = x/2+y
    - Codificação: entrada é $w_x 1 w_y$, onde $w_x, w_z \in \{0\}^*$ e $|w_x| = x$ e $|w_y| = y$
    - O "1" é o separador - Solução: cancele metade dos 0s de w_x e desloque os 0s de w_y

Brainstorming – Calculadora de função

- Como seria a ideia de uma máquina que calcula $f(x) = 2.x+1$?
- E para $f(x) = x^2+1$?

Que outras variações poderíamos ter?

- E se a fita fosse mais fácil de usar?
- E se tivéssemos múltiplas fitas?
- E se tivéssemos não determinismo?

Adicionando capacidade a uma MT: Cabeçote

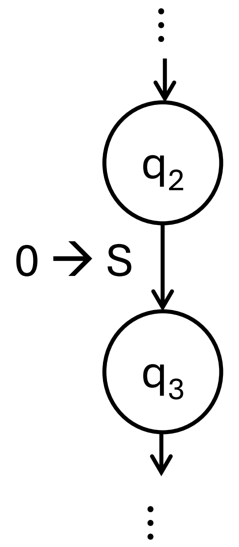
- Poderíamos permitir que o cabeçote ficasse parado, ou seja: δ: Q x $\Gamma \rightarrow$ Q x Γ x {E,D,S}

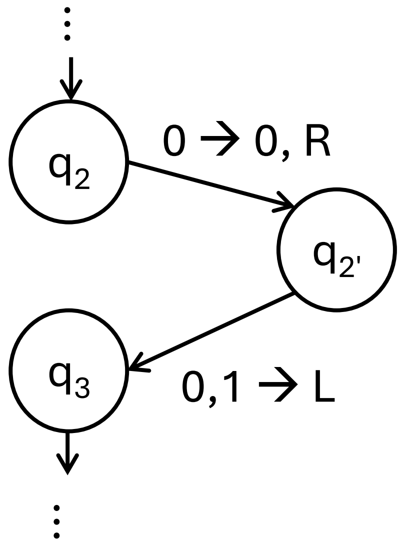
- Isso muda o poder da máquina?

Adicionando capacidade a uma MT: Cabeçote

- Poderíamos permitir que o cabeçote ficasse parado, ou seja: δ: Q x $\Gamma \rightarrow$ Q x Γ x {E,D,S}

- Isso muda o poder da máquina? Não.





A principal ideia aqui é: uma transição em um tipo de máquina sofisticada pode ser equivalente a um conjunto de transições na máquina mais simples.

Adicionando capacidade a uma MT: Cabeçote

- E se fosse possível rebobinar a fita?
- E se a fita fosse infinita para os dois lados?

Adicionando capacidade a uma MT: Cabeçote

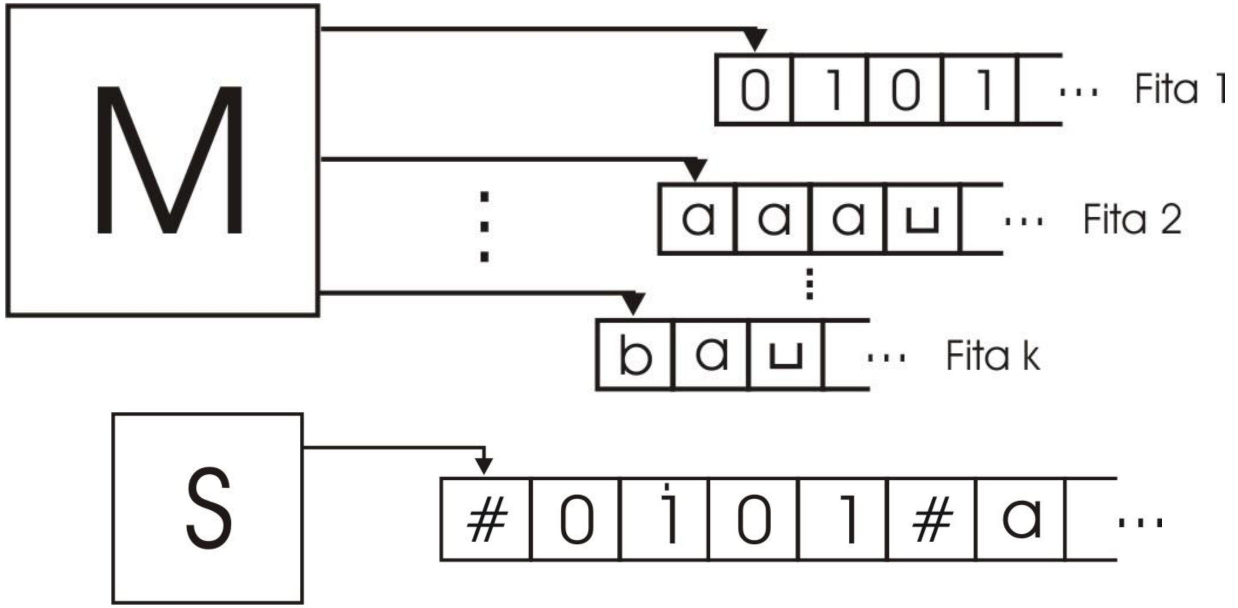
- E se fosse possível rebobinar a fita?
- E se a fita fosse infinita para os dois lados?

Pense em um caderno que tem um número infinito de páginas e você consegue mexer na espiral.

Uma variante mais complexa

- Teorema: Toda MT multifita M tem uma MT S de uma só fita equivalente

Para que ajudaria uma MT multifita?



Pense em um caderno que tem um número infinito de páginas e vários marcadores de página.

Intuição da prova

- Se M tem 3 fitas, nós queremos armazenar tudo em uma só fita

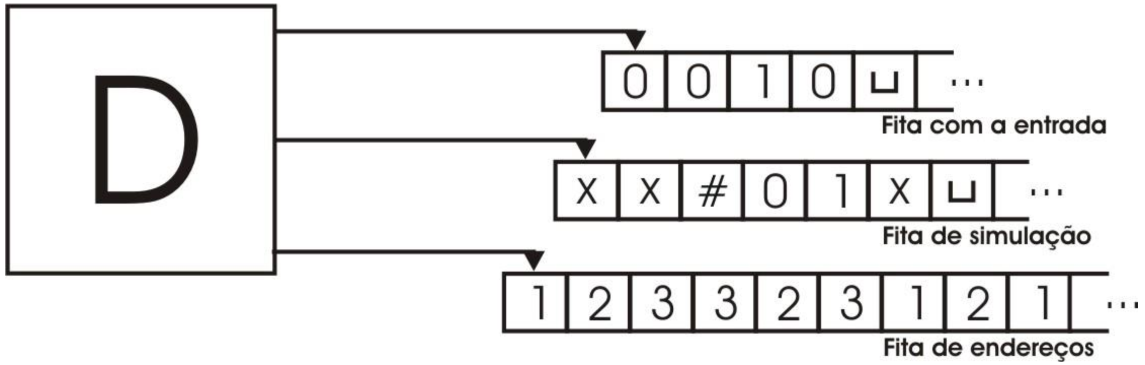
a	**b**	**#**	υ	**#**	υ	**#**	υ	υ	…

- Do que precisamos?
  - Quando o autômato estiver no último símbolo de uma fita virtual e mover para a direita, precisamos abrir espaço (basta mover todos os caracteres para esquerda)
  - Precisamos manter a posição de todas as cabeças (marcando o símbolo em cada "fita virtual" onde a cabeça estaria)

Máquinas não-determinísticas e determinísticas também são equivalentes

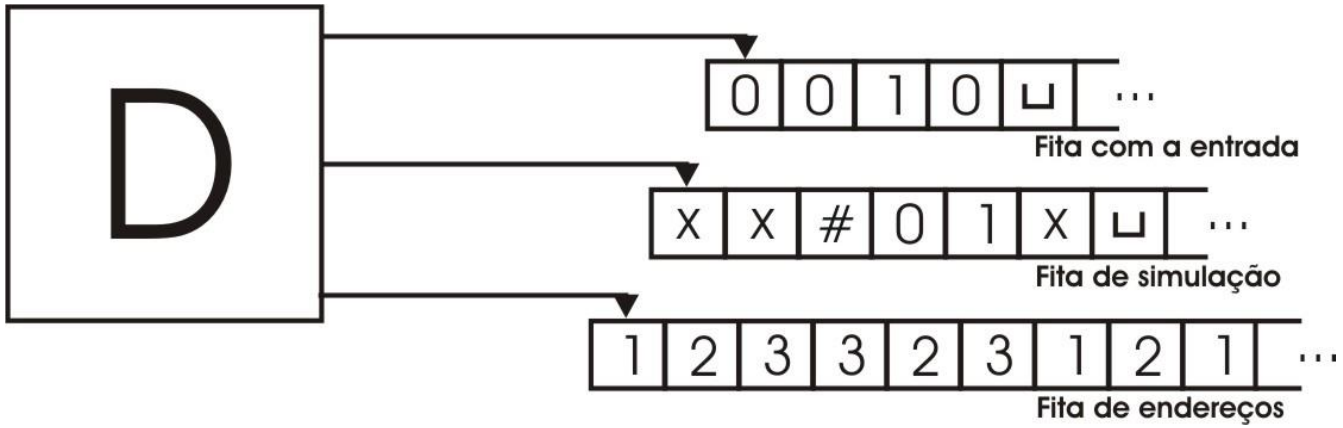
Máquinas não-determinísticas e determinísticas também são equivalentes

- Para uma máquina não determinística N, sua versão determinística D tem 3 fitas e é composta por várias pequenas máquinas
- D_1 copia entrada para fita 2 e gera uma lista com as possíveis tentativas na fita 3
- D_2 é uma versão determinística de N, com uma fita a mais
  - Usa a fita 2 para executar N, mas como só pode seguir uma transição, olha na posição atual da fita 3 para saber qual escolha ela toma (e avança o cabeçote 3 para a próxima decisão)
  - Se D_2 chega no estado de aceitação, aceita, se D_2 chega no estado de rejeição ou nenhuma transição é possível, D vai para o próximo estágio...



Máquinas não-determinísticas e determinísticas também são equivalentes

- Resultado:
  - D testa todos os caminhos não determinísticos, mas somente um de cada vez
  - A fita 1 guarda uma cópia da entrada
  - A fita 2 é a memória de trabalho
  - A fita 3 é o que registra que caminhos já foram tomados e que caminhos a execução atual vai seguir
  - É importante notar também que ela não pode tentar uma execução por tempo indeterminado, pois corre o risco de nunca parar



Estudo dirigido

Estudo dirigido – Compensação de aulas

- Faça uma máquina que decide a linguagem www com w $\in$ {0,1}*, em passos intermediários
  - Máquina 1: recebe www e transforma em w'w'w' onde w' é a palavra w com o seu último símbolo marcado com um '
  - Máquina 2: recebe w'w'w' e transforma em w#w#w
  - Máquina 3: recebe w#w#w e aceita se as 3 partes são iguais
- Faça uma máquina que compute f(x,y) = x/2+y, deixando o resultado no início da fita e a fita toda limpa
- Cada uma das duas máquinas contará como 3 horas-aula e precisa ser entregue em papel A4 durante a chamada no dia 30/4
  - Ou enviado digitalizado por e-mail antes e entregue em papel na aula seguinte