

FUNDAMENTOS TEÓRICOS DA COMPUTAÇÃO

--- MÁQUINAS DE TURING ---

Variantes



Multifitas

MT com multifitas

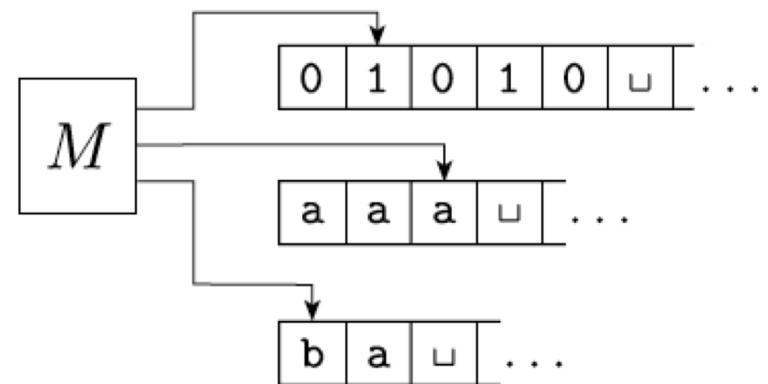
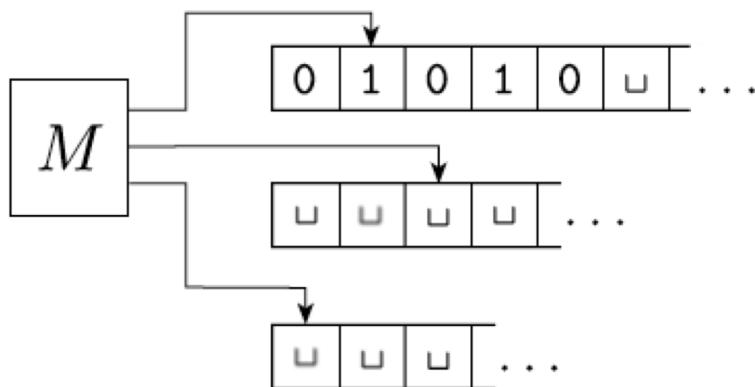
Uma MT multifitas é uma MT padrão com k fitas

Cada fita tem seu próprio cursor de leitura/escrita independente

Inicialmente, $\omega = \omega_1\omega_2 \dots \omega_n$ aparece na fita 1 e as outras fitas estão em branco

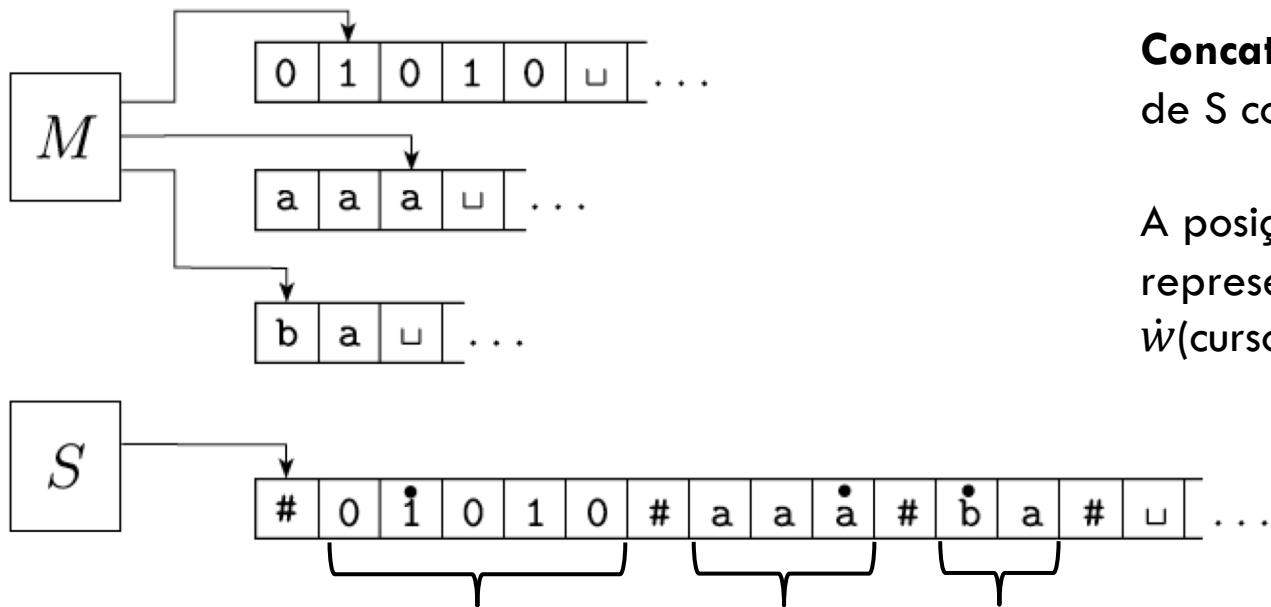
A função de transição é redefinida: $\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{E, D, P\}^k$

Com isso: $\delta(q, a_1a_2 \dots a_k) = \left(p, \underbrace{b_1b_2 \dots b_k}_{k}, E, D, \dots P \right)$



MT com multifita

- As MT multifita aparentam ser mais poderosas que as MT padrão, mas na verdade **não são**; elas são equivalentes
- É possível **converter** uma MT multifita **em uma** MT de fita única



Concatenar as k fitas na fita de S com o delimitador $\# \notin \Gamma$

A posição de **cada cursor** é representada por um símbolo \dot{w} (cursores virtuais)

MT com multifita

Primeiro colocamos ω na fita de S na codificação proposta, isto é:

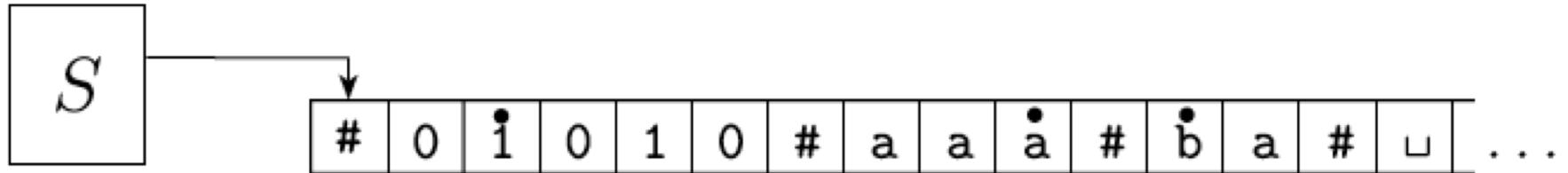
$$\# \dot{\omega}_1 \omega_2 \cdots \omega_n \# \sqcup \# \sqcup \cdots \#$$

Para simular

$$\delta(q, a_1 a_2 \cdots a_k) = \left(p, b_1 b_2 \cdots b_k, \underbrace{E, D, \cdots P}_{\text{}} \right)$$

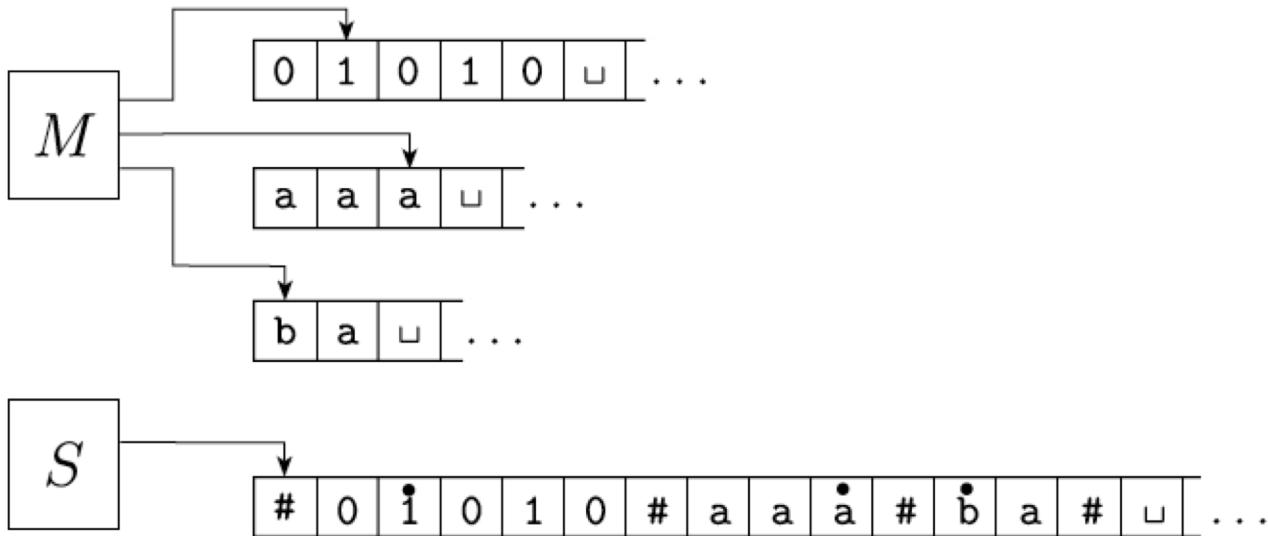
A máquina S verifica cada cursor virtual para definir qual transição deve ser aplicada

MT com multifita



Suponha que exista a transição $\delta(q, 1, a, b) = (p, 0, b, a, D, E, P)$ e, quando executada, o resultado é que a fita transforma-se em:
#00̄10#aàb#àa#

MT com multifita



Uma exceção ocorre quando algum cursor virtual para sobre o i -ésimo $\#$. A fita precisa deslocar todas as células depois dessa posição um para a direita, e inserir um \sqcup
Assumimos que $i = 3$, a fita transforma-se em:

#0i010#aaà \sqcup #ba#

MT Não-Determinística

MT não determinísticas

Em uma MT não-determinística, a computação pode proceder de diversas maneiras

Dado um estado q e símbolo a lido podemos ir para vários outros estados, escrever diferentes símbolos e mover para diferentes direções

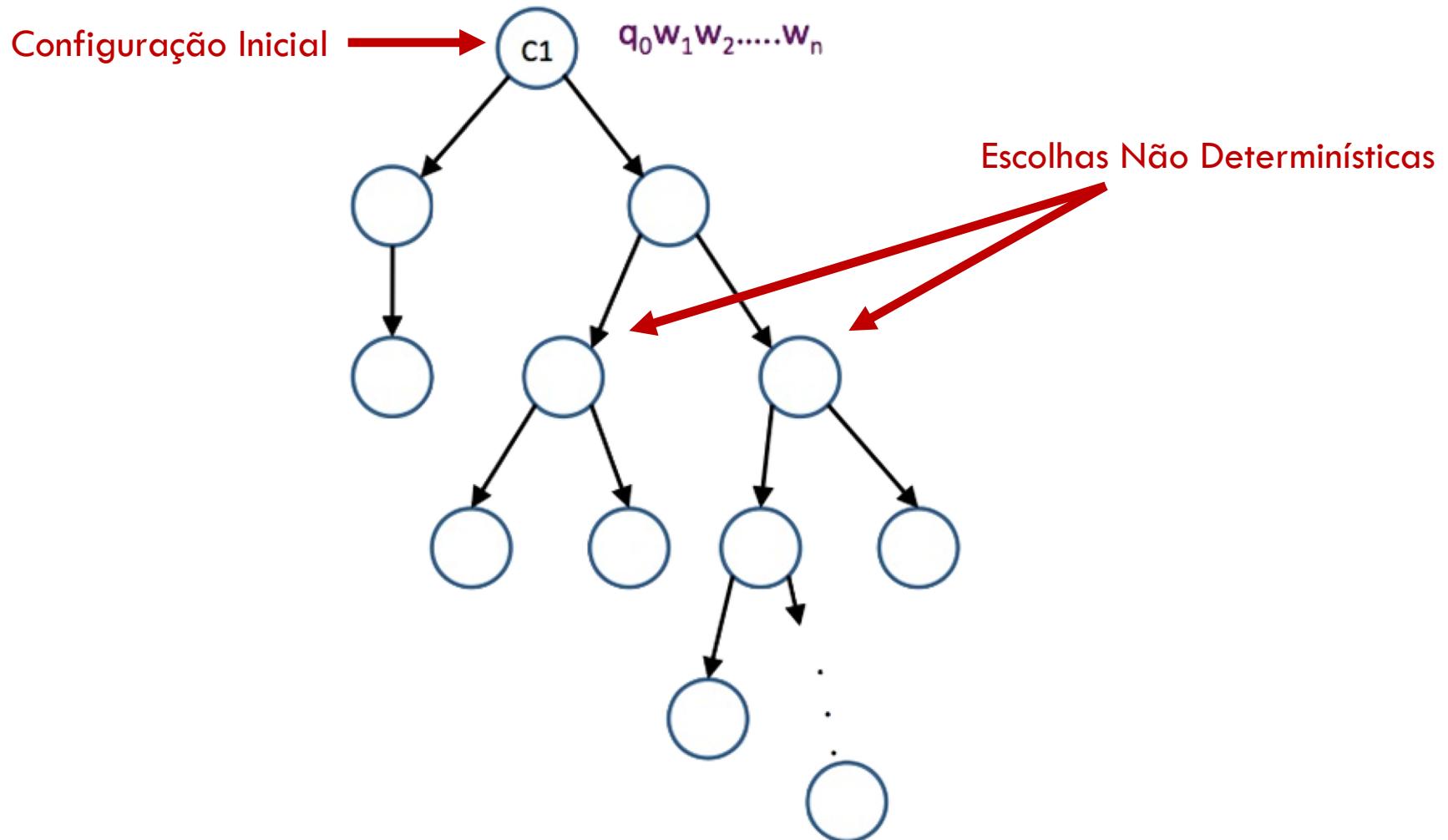
A função de transição tem a forma:

$$\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{E, D, P\}}$$

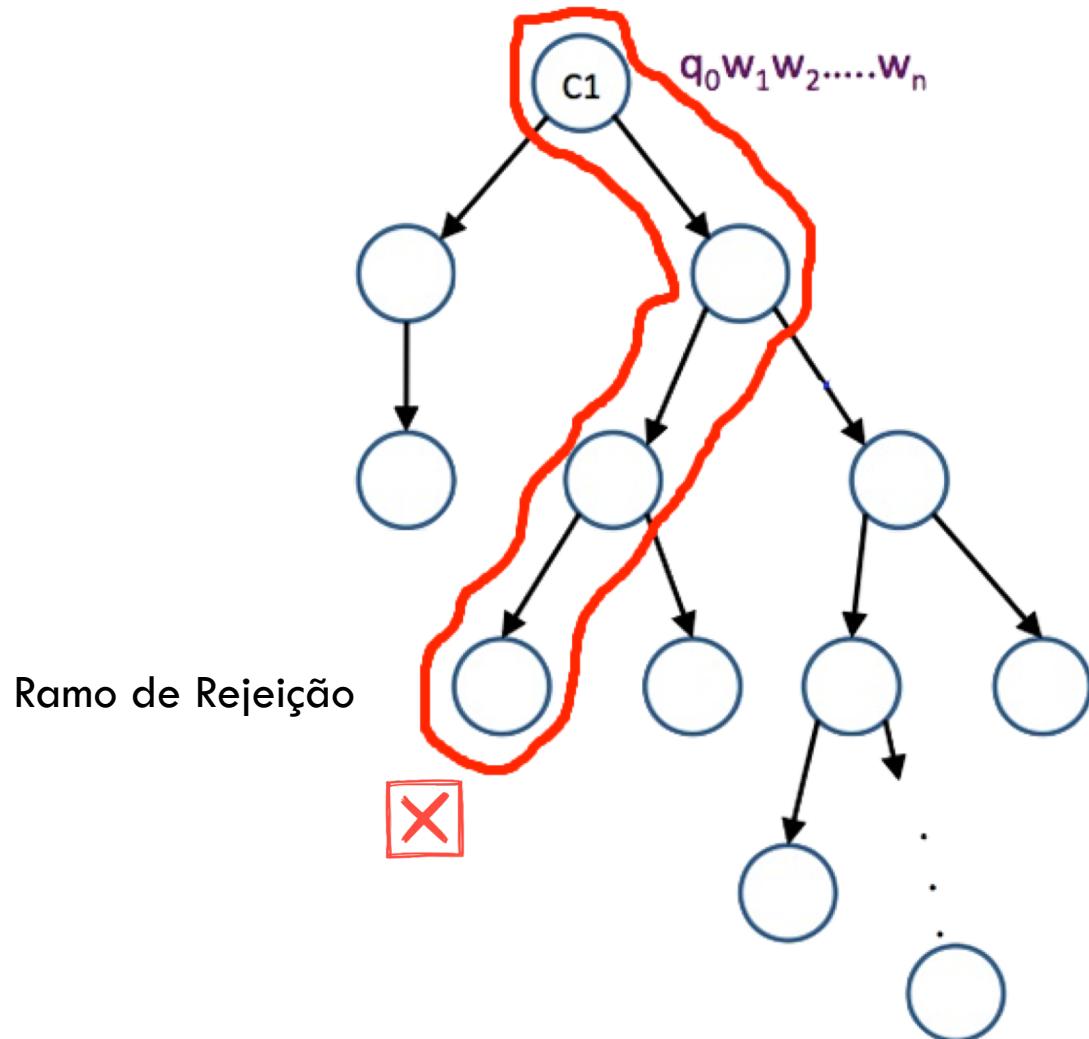
Ou seja, $\delta(q, a) = \{(p_i, b_i, D), (p_j, b_j, E), \dots\}$

As MTs não-determinísticas **aparentam** ser mais poderosas que as MT padrão, mas na verdade **não são**; elas são equivalentes. É possível **converter** uma MT não-determinística em **uma** MT padrão

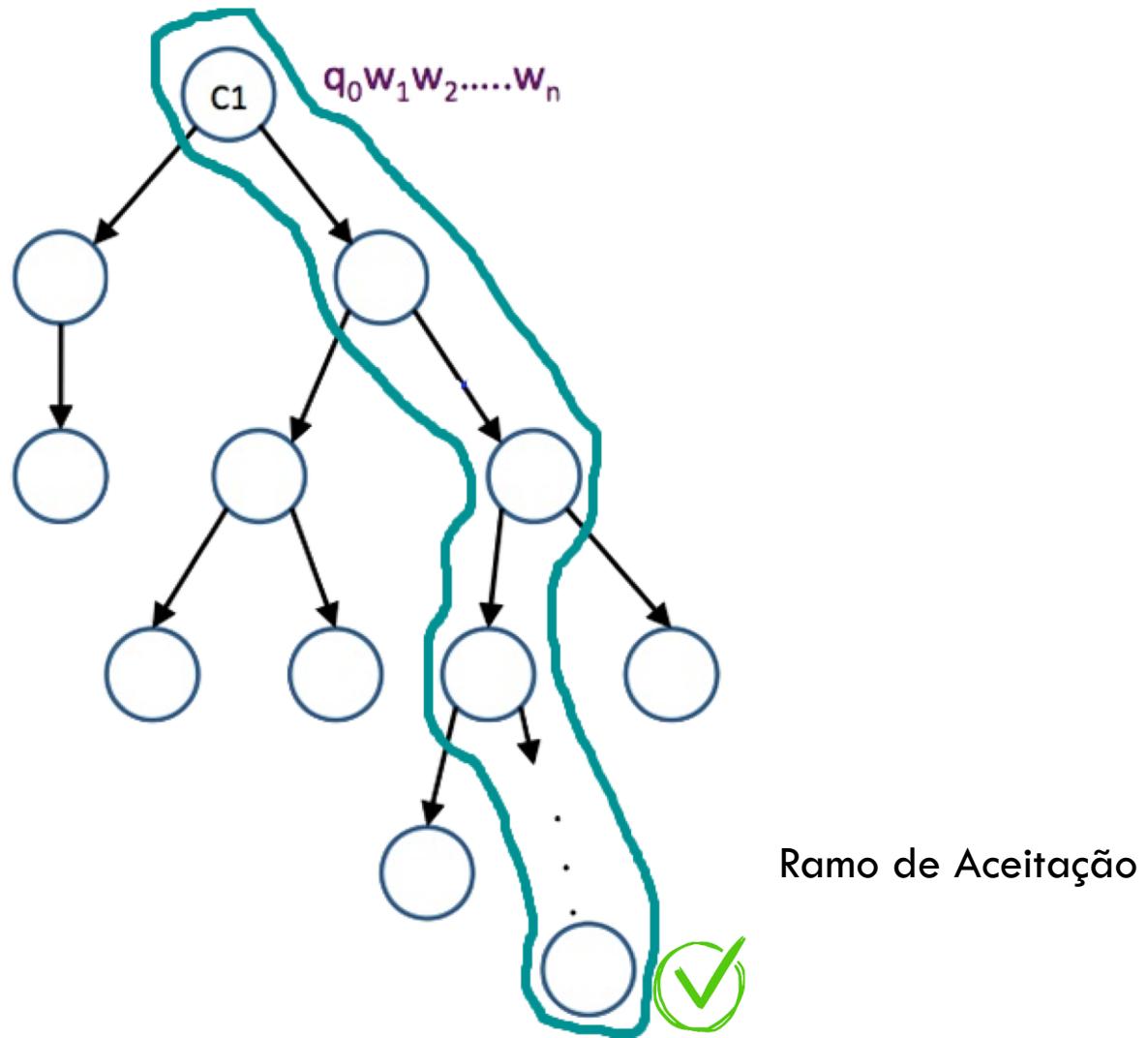
MT não determinísticas



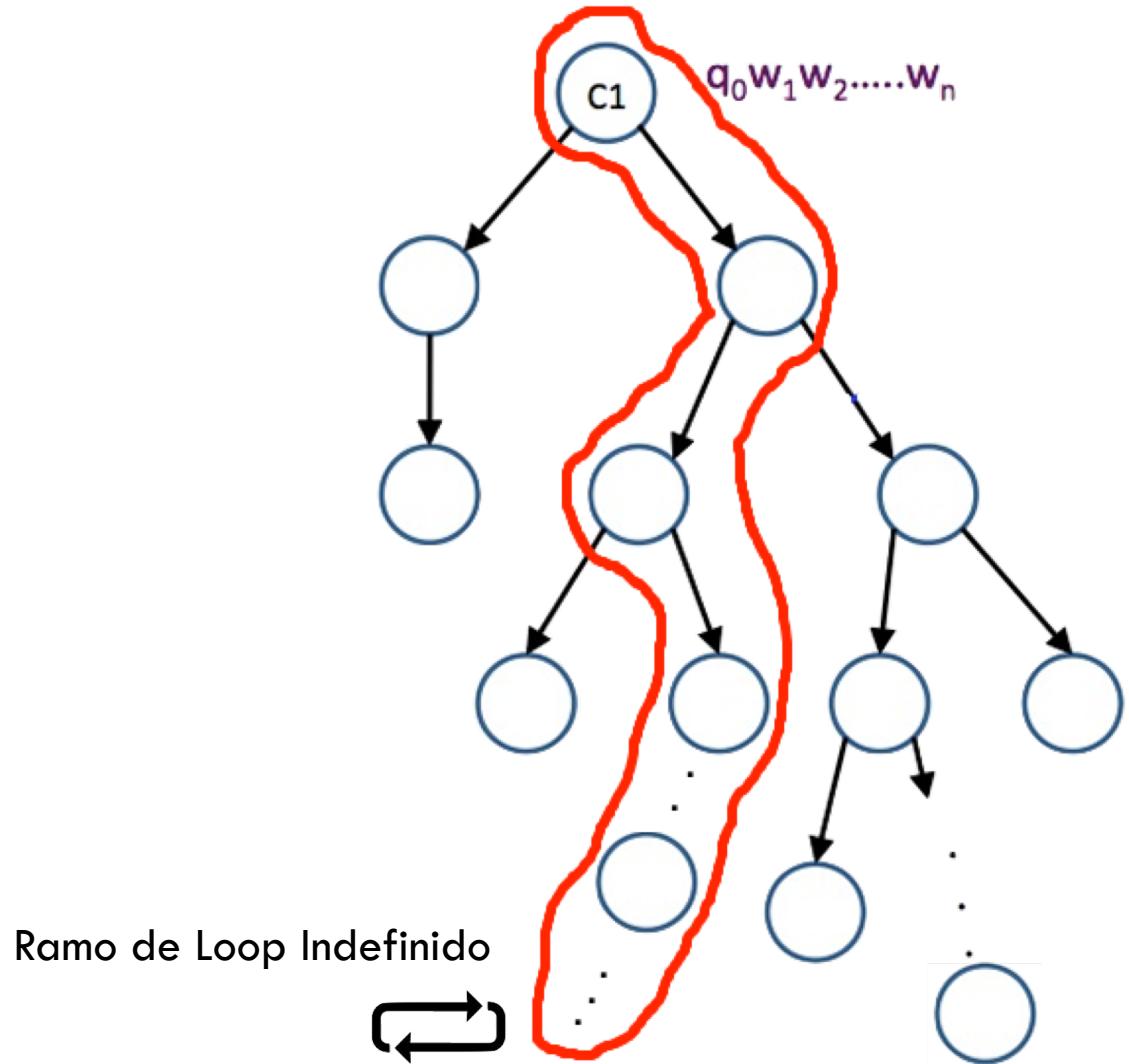
MT não determinísticas



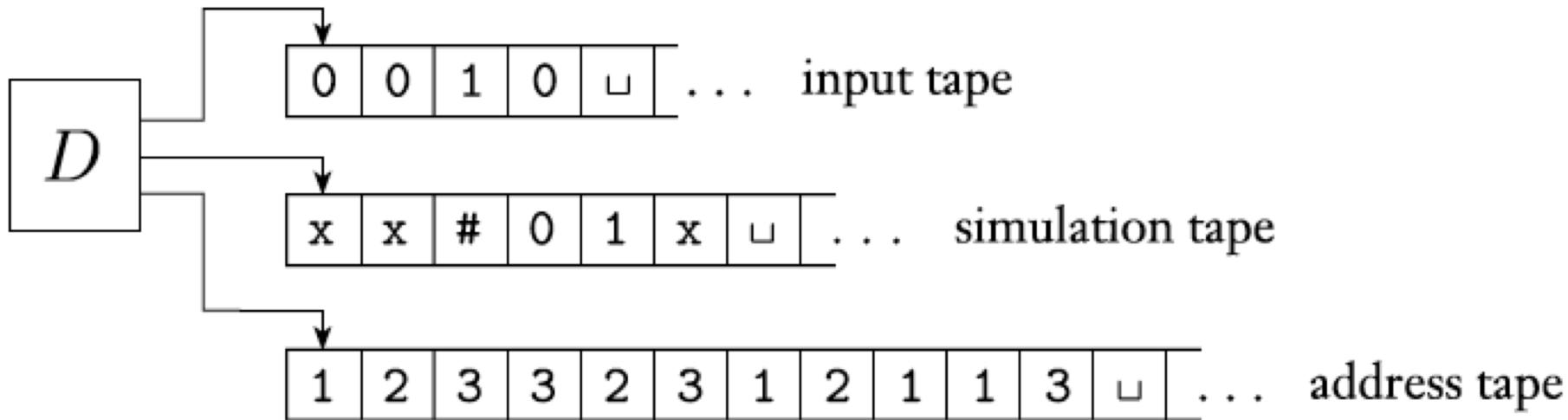
MT não determinísticas



MT não determinísticas



Conversão

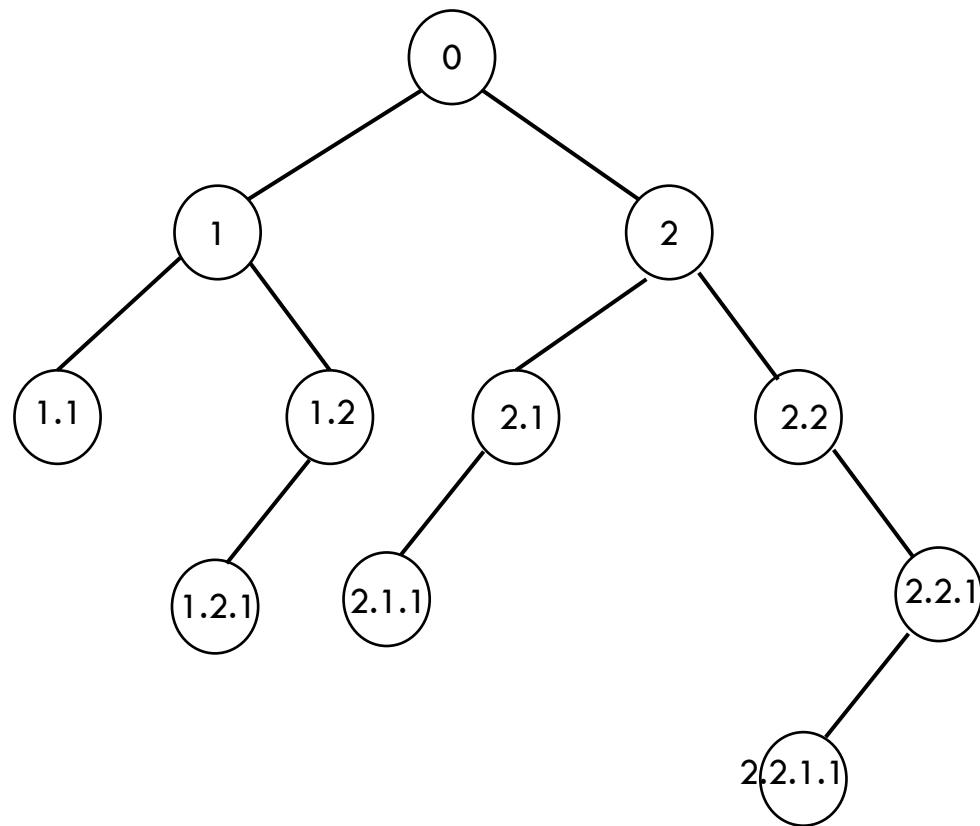


Seja N uma MT não determinística e D uma MT determinística com três fitas. A MT D vai simular todas as computações não determinísticas. A fita 1 conterá a cadeia de entrada (só leitura). A fita 2 é a fita de trabalho que também guarda uma cópia da fita de N em uma computação. A fita 3 guarda a posição de D na árvore de computação.

Conversão

Como percorrer as árvores da computação?

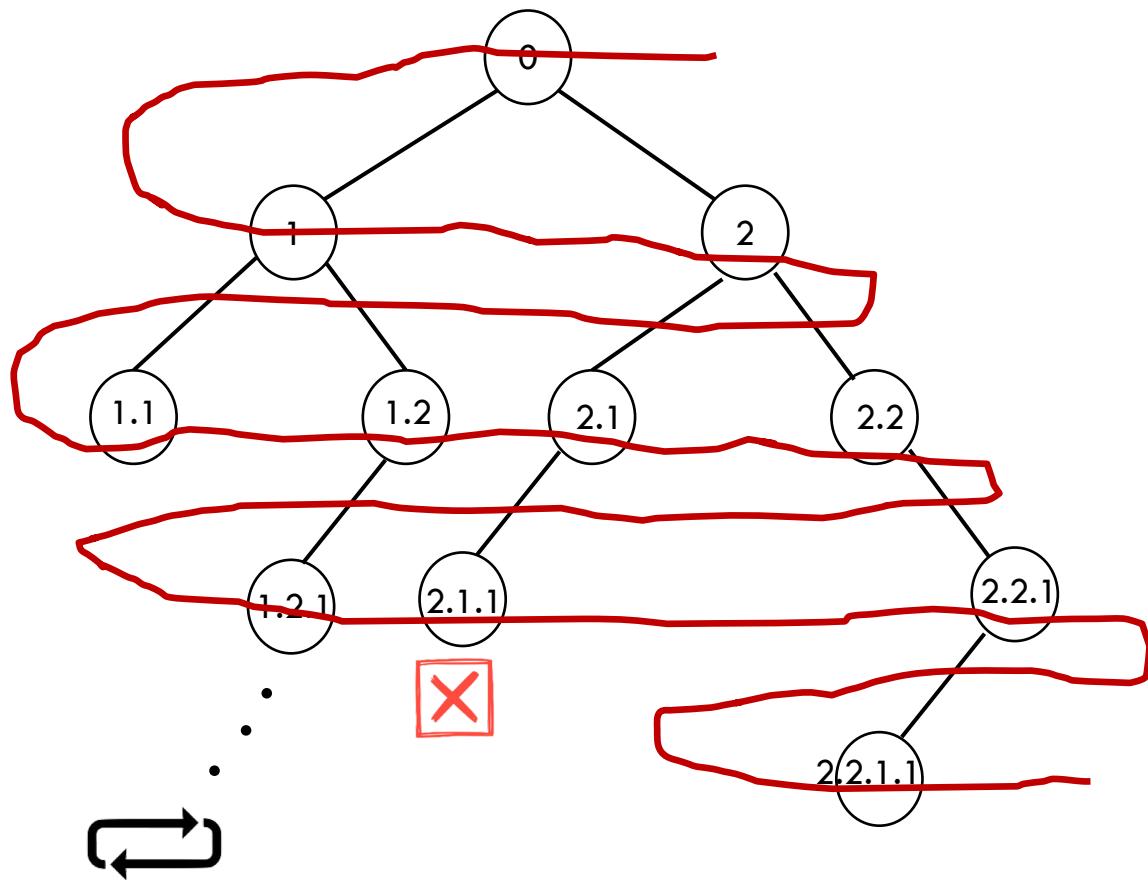
Enumera
todos os nós
da árvore



Conversão

Como percorrer as árvores da computação?

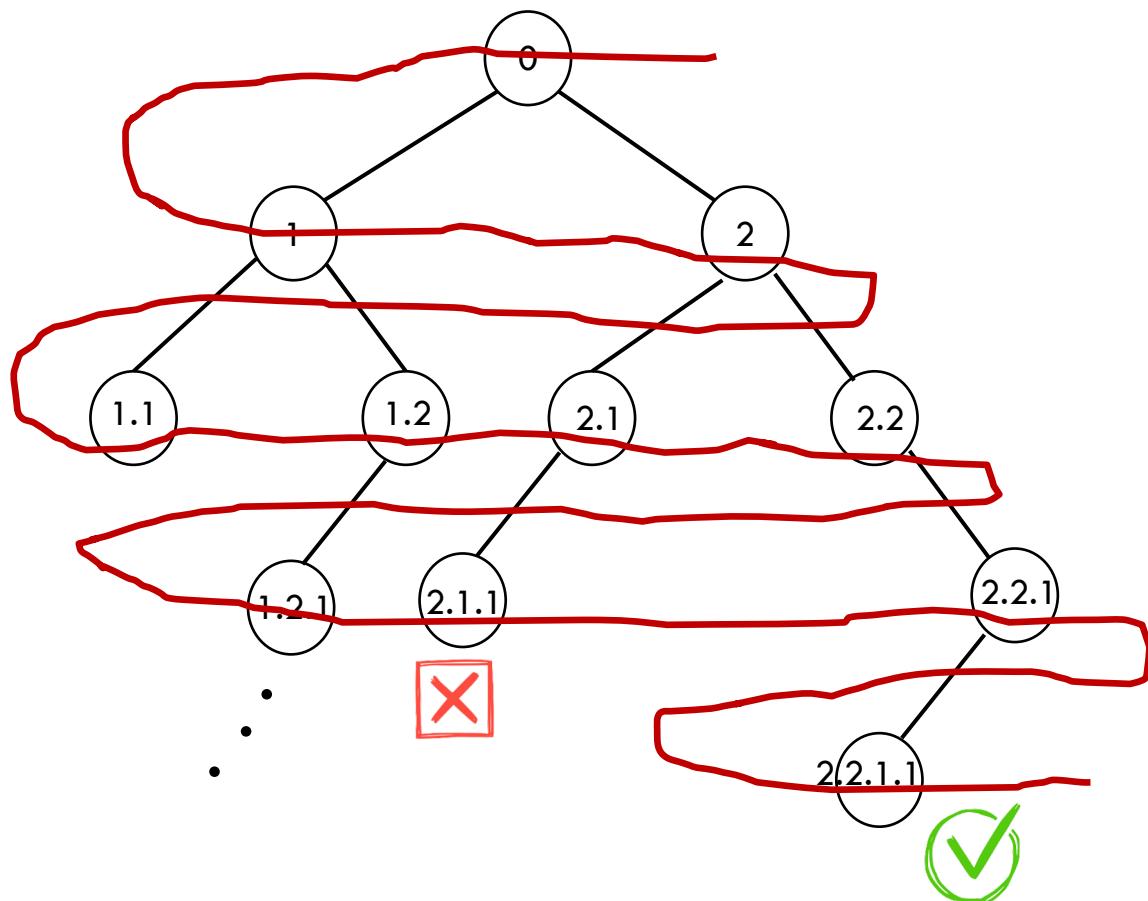
Faz-se uma
busca em
largura (e
não em
profundidade)



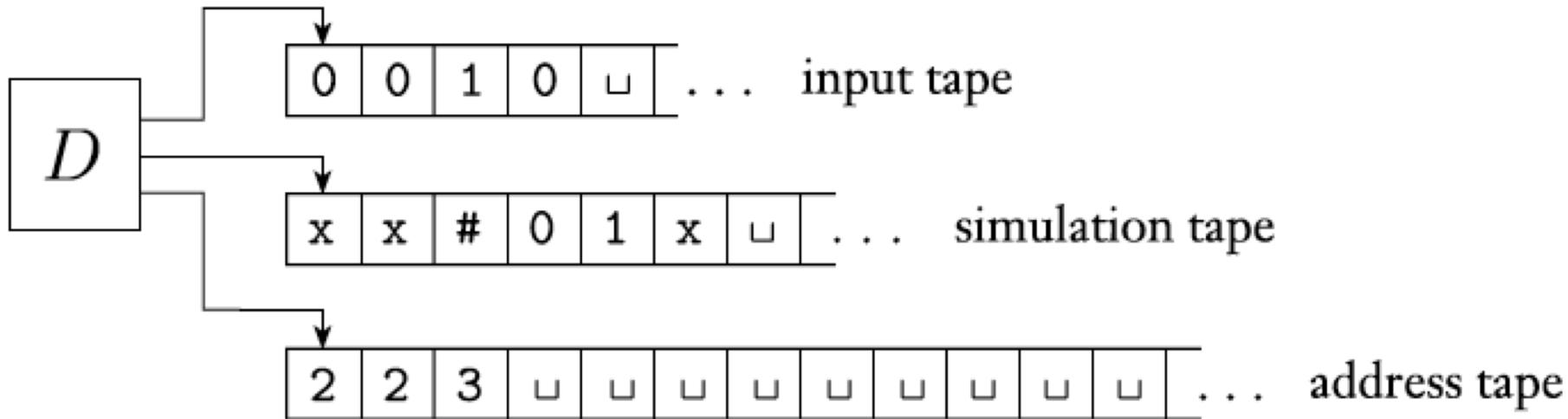
Conversão

Como percorrer as árvores da computação?

Para na
primeira
aceitação



Conversão



A fita 3 indica qual o ramo da computação de N estamos simulando atualmente e utiliza a fita 2 de trabalho

A partir da fita 3 e da cadeia de entrada (fita 1) é possível escrever na fita 2

No exemplo, coloca na fita 2 as decisões 2, 2 e 3 (nesta sequência) da função de transição

Conversão

Sobre a cadeia de entrada w

1. Inicialmente, a fita 1 tem a entrada w e as fitas 2 e 3 estão vazias
2. Copie a fita 1 para a fita 2
3. Use a **fita 2** para simular N com a entrada w sobre um ramo de sua computação não determinística. Antes de cada passo de N , consulte o próximo símbolo na fita 3 para determinar qual escolha fazer entre aquelas permitidas pela função de transição de N . Se **não restam** mais símbolos na fita 3, ou se essa escolha não determinística for inválida, **aborte** esse ramo indo para o passo 4. Também vá para o passo 4 se uma configuração de **rejeição** for encontrada. Se uma configuração de aceitação for encontrada, **aceite a entrada**
4. Substitua a cadeia na fita 3 pela próxima cadeia na ordem lexicográfica. Simule o próximo ramo da computação de N indo para o passo 2



MT e Computadores

Outros modelos de computação

Muitos outros modelos para computação têm sido propostos: cálculo- λ , sistemas de Post, funções μ -recursivas, e outros, são todos Turing-equivalentes

Todos compartilham: acesso irrestrito a memória ilimitada. Todos os modelos com essas características são equivalentes em poder

Embora diferentes, a classe de algoritmos que eles descrevem permanece a mesma. É **similar** ao que vemos em **linguagens de programação**, ou seja, todo programa que pode ser escrito em **Python** pode também ser escrito em **Java**, **C**, **C++**, ...

Um computador pode **simular** uma máquina de Turing (exceto que a fita é infinita) e uma máquina de Turing consegue **fazer tudo o que um computador faz**

Modelo de computador

Parece com uma arquitetura básica de um computador moderno

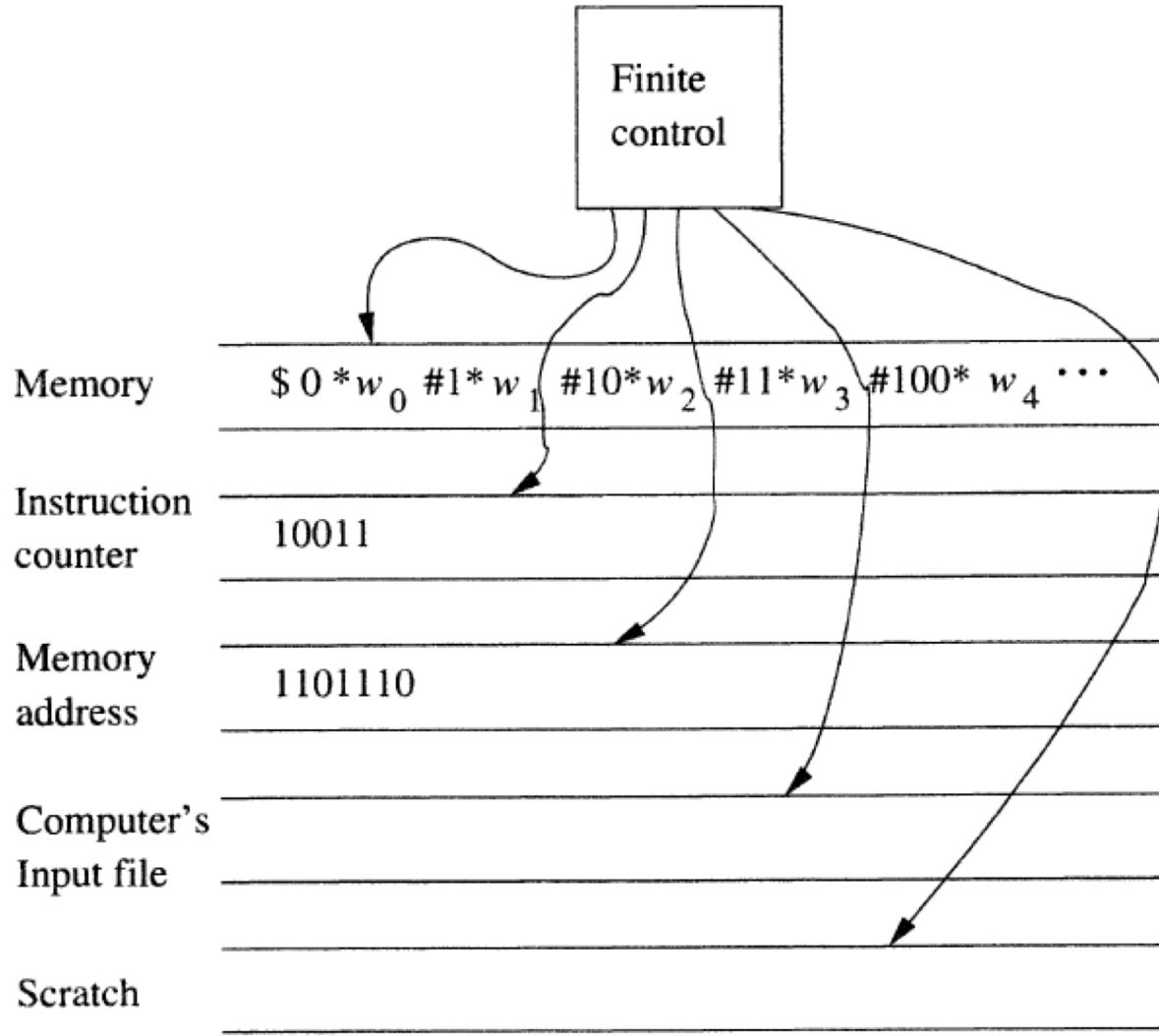
Memória: sequência longa de palavras, cada uma com um endereço

Programa:

- Sequência de palavras, cada uma com um endereço;
- Cada palavra representa uma instrução simples (read, write, load, store, ...)
- Cada instrução envolve um número limitado de palavras
- Cada instrução altera o valor de no máximo uma palavra

Contador de instruções: indica a próxima instrução a ser executada

Modelo de computador



Modelo de computador

Ciclo de instruções do computador:

- A máquina de Turing procura na fita 1 o local apontado pelo conteúdo da fita 2;
- Interpreta o conteúdo da instrução;
- Se a instrução usa o valor de um endereço, copiamos este endereço para a fita 3;
- A MT então executa a instrução, podendo ou não usar a fita rascunho;
- A MT incrementa o conteúdo da fita 2 e recomeça o ciclo

Não é uma prova completa e formal, mas passa a ideia da possibilidade de simular um computador em uma máquina de Turing

