GENERALIZAÇÕES DE M.T.

- VEREMOS QUE EM VÁRIAS DIREÇÕES, NÃO GANHAMOS PODER COMPUTACIONAL AO CONSIDERARMOS M.T. MAIS POTENTES

EX:
- FITA INFINITA DOS DOIS LADOS
- MULTIPLAS FITAS
- MULTIPLOS CABEÇOTES DE LEITURA OU ESCRITA
- MOVIMENTAÇÃO MENOS RESTRITA DO CABEÇOTE

⇝ QUALQUER LINGUAGEM DECIDÍVEL, SEMI-DECIDÍVEL, OU FUNÇÃO COMPUTÁVEL POR UM MODELO DE M.T. TAMBÉM O É EM OUTRO MODELO.

Aula de Hoje

- Multiplas Fitas
- M.T. Não-Determinística
- Fechamento
- Tese de Church-Turing

M.T. COM MÚLTIPLAS FITAS

- A FUNÇÃO DE TRANSIÇÃO $\delta : (Q - F) \times \Sigma^{K} \longrightarrow Q \times \left((\Sigma - \{\triangleright\}) \cup \{\leftarrow, \rightarrow\} \right)^{K}$

  NÚMERO DE FITAS

- VAMOS REPRESENTAR O CONTEÚDO DAS K FITAS EM ÚNICA FITA
- USAMOS O SÍMBOLO # PARA SEPARAR O CONTEÚDO DAS FITAS
- PARA INDICAR EM QUE POSIÇÃO ESTÁ CADA CABEÇOTE DOBRAMOS OS SÍMBOLOS DO ALFABETO, CRIANDO UMA CÓPIA DE CADA SÍMBOLO COM UM • ACIMA

$\triangleright$	#	$\dot{a}$	b	c	#	a	$\dot{b}$	a	a	#	c	c	$\dot{c}$	#	$\sqcup$	$\sqcup$	...		

$$\Sigma = \{a, b, c\} \rightsquigarrow \dot{\Sigma} = \{\dot{a}, \dot{b}, \dot{c}\}$$

TRANSIÇÕES:

- SÃO FEITAS DUAS VARREDURAS

→ A MÁQUINA PERCORRE A FITA ATÉ $(k+1)$-ÉSIMO #

E GUARDAR OS SÍMBOLOS COM • EM SEUS ESTADOS

→ PARA CADA ESTADO $r \in Q_0$ E K-TUPLA $(a_1, \dots, a_k) \in \Sigma$

CRIAMOS OS ESTADOS $q_r, q_{ra_1}, q_{ra_1a_2}, \dots, q_{ra_1 \dots a_k}$

→ NA SEGUNDA VARREDURA EXECUTAMOS AS OPERAÇÕES EM CADA PARTE

• MOVER O CABEÇOTE SIGNIFICA MUDAR A POSIÇÃO DE •

→ ENTENDEMOS O # COMO $\triangleright$, QUE LIMITA O CABEÇOTE DE ANDAR P/ ESQ.

→ SE ACABAMOS DE MOVER P/ ESQ. E ENCONTRAMOS #, O PRÓXIMO MOVIMENTO DEVE SER DE MOVER P/ DIR

→ SE ACABAMOS DE MOVER P/ DIR. E ENCONTRAMOS #, DEVEMOS FAZER UM SHIFT DE TODA A PARTE DA FITA SEGUINTE.

M.T. NÃO-DETERMINÍSTICA

- A FUNÇÃO DE TRANSIÇÃO: $\Delta : (Q - F) \times \Sigma \longrightarrow \mathcal{P}(Q \times ((\Sigma \setminus \{\triangleright\}) \cup \{\leftarrow, \rightarrow\}))$

  ⇝ SE A MÁQUINA ESTÁ NO ESTADO q E LÊ σ, ELA ESCOLHE DE FORMA NÃO-DETERMINÍSTICA $(q', \sigma') \in \Delta(q, \sigma)$

  ⇝ CONJUNTO DE POSSÍVEIS TRANSIÇÕES

DEF: UMA M.T. NÃO-DETERMINÍSTICA M ACEITA (OU SEMI-DECIDE) $L \subseteq \Sigma_0^*$ SE, PARA CADA $w \in \Sigma_0^*$, TEMOS QUE $w \in L$ SE E SOMENTE SE AO MENOS UMA COMPUTAÇÃO EM M TERMINA (PARA) EM UM ESTADO FINAL.

DEF: UMA M.T. NÃO-DETERMINÍSTICA M DECIDE $L \subseteq \Sigma_0^*$ SE, PARA CADA $w \in \Sigma_0^*$, TEMOS QUE M SEMPRE PARA, E SE $w \in L$, ENTÃO EXISTE UMA COMPUTAÇÃO EM M QUE PARA NO ESTADO DE ACEITAÇÃO (S).

ISSO IMPLICA QUE, SE $w \notin L$, ENTÃO TODAS AS COMPUTAÇÕES EM M PARAM NO ESTADO DE REJEIÇÃO.

DEF: UMA M.T. NÃO-DETERMINÍSTICA M COMPUTA UMA FUNÇÃO ϕ SE PARA TODO $w \in \Sigma_0^*$ M SEMPRE PARA E, PARA TODAS COMPUTAÇÕES, TERMINA COM $\phi(w)$ NA FITA.

ISTO NÃO TRAZ GANHO COMPUTACIONAL

→ PODEMOS SIMULAR UMA MTND COM UMA MTD

→ TODA LINGUAGEM ACEITA/DECIDIDA/COMPUTÁVEL POR UMA MTND TAMBÉM É POR MTD

SIMULANDO MTND POR MTD

- 3 FITAS

⇝ VAMOS CONSIDERAR QUE AS TRANSIÇÕES SÃO ORDENADAS

↳ É POSSÍVEL POIS O CONJ. DE TRASIÇÕES É FINITO

↳ ORDENAMOS POR SÍMBOLOS $C_b\{1, \dots, b\}$ → O TAMANHO DO MAIOR CONJ. DE TRASIÇÕES. A MTND DADA

⇝ VAMOS SIMULAR DE FORMA ORDENADA TODAS AS POSSÍVEIS TRANSIÇÕES DE M

- A PRIMEIRA FITA GUARDA UM BACKUP DA ENTRADA
- A SEGUNDA FITA RECEBE, PARA CADA SIMULAÇÃO, UMA CÓPIA DA ENTRADA
- A TERCEIRA FITA GUARDA A SEQUÊNCIA DE ESCOLHAS FEITAS

EX: 2374 INDICA UMA SEQ. DE PROFUNDIDADE 4 QUE ESCOLHEU

$2^{\underline{a}}$, $3^{\underline{a}}$, $7^{\underline{a}}$, $4^{\underline{a}}$ ...

- NÃO PODEMOS FAZER UMA "BUSCA EM PROFUNDIDADE" POIS PODE HAVER COMPUTAÇÕES INFINITAS.

- ENTÃO EXECUTAREMOS TODAS AS "BUSCAS" DE PROF. 1, DEPOIS DE PROF. 2, ...

- A ORDEM DAS PALAVRAS NA TERCEIRA FITA É

$$
\begin{array}{ccccccc}
1 & 11 & 21 & & \beta 1 & 111 & 1111 \\
2 & 12 & 22 & & \beta 2 & \vdots & \vdots \\
3 & 13 & \vdots & \cdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & & & & & \\
\beta & 1\beta & 2\beta & & \beta\beta & \beta\beta\beta & \beta\beta\beta\beta
\end{array}
$$

- SE HÁ UMA COMPUTAÇÃO NÃO-DETERMINÍSTICA COM n PASSOS, TEREMOS UMA COMPUTAÇÃO DETERMINÍSTICA β^n PASSOS

FECHAMENTO

- COMPLEMENTO, UNIÃO, INTERSEÇÃO, DIFERENÇA

DECIDÍVEIS (REC)

1) COMPLEMENTO: BASTA TROCAR S POR N $\quad L \quad \overline{L}$

2) UNIÃO: RODAR EM PARALELO M_1 E M_2 $\quad Q_U = Q_1 \times Q_2$

   $\rightsquigarrow$ CRIAR UMA MÁQUINA COM DUAS FITAS E

   ACEITAR UMA PALAVRA SE ATINGE S EM PELO MENOS UMA

   MÁQUINA ((S,S), (S,N), (N,S))

3) INTERSEÇÃO: RODAR EM PARALELO

   $\rightsquigarrow$ CRIAR UMA MÁQUINA COM DUAS FITAS E

   ACEITAR UMA PALAVRA SE ATINGE S NAS DUAS MÁQUINAS (S,S)

4) $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$

SEMI-DECIDÍVEIS

UNIÃO E INTERSEÇÃO $\rightsquigarrow$ (s,s)

(s,q) $\rightsquigarrow$

(q,s)

(s,q) OU (q,s)

(s,s)

COMPLEMENTO E DIFERENÇA) RE NÃO É FECHADA POR COMPLEMENTO OU DIFERENÇA

TEOREMA: SE L E $\bar{L}$ SÃO RE, ENTÃO L E $\bar{L}$ REC

PROVA: PODEMOS DECIDIR L COM UMA MÁQUINA QUE RODA M_L E $M_{\bar{L}}$ EM PARALELO E ACEITA w SE PARA EM S DE M_L, E REJEITA w SE PARA EM S DE $M_{\bar{L}}$.

Tese de Church-Turing

- M.T. é um modelo simples
  - Pode estudar o que pode
  - e o que não pode ser computado
- Múltiplas fitas, cabeçotes, não-determinismo
  - não aumenta seu poder computacional
- Há outros modelos equivalentes
  - λ-cálculo
  - funções μ-computáveis
- Adota M.T. como modelo teórico de algoritmo

- Não pode ser provada
- Mas pode ser desprovada!
  Basta apresentar modelo mais poderoso

Tese de Church-Turing: um problema de decisão é algoritmicamente computável se e somente se existe uma M.T. que o decide ou computa