Generalizações de MTs e Fechamento

Luís Felipe

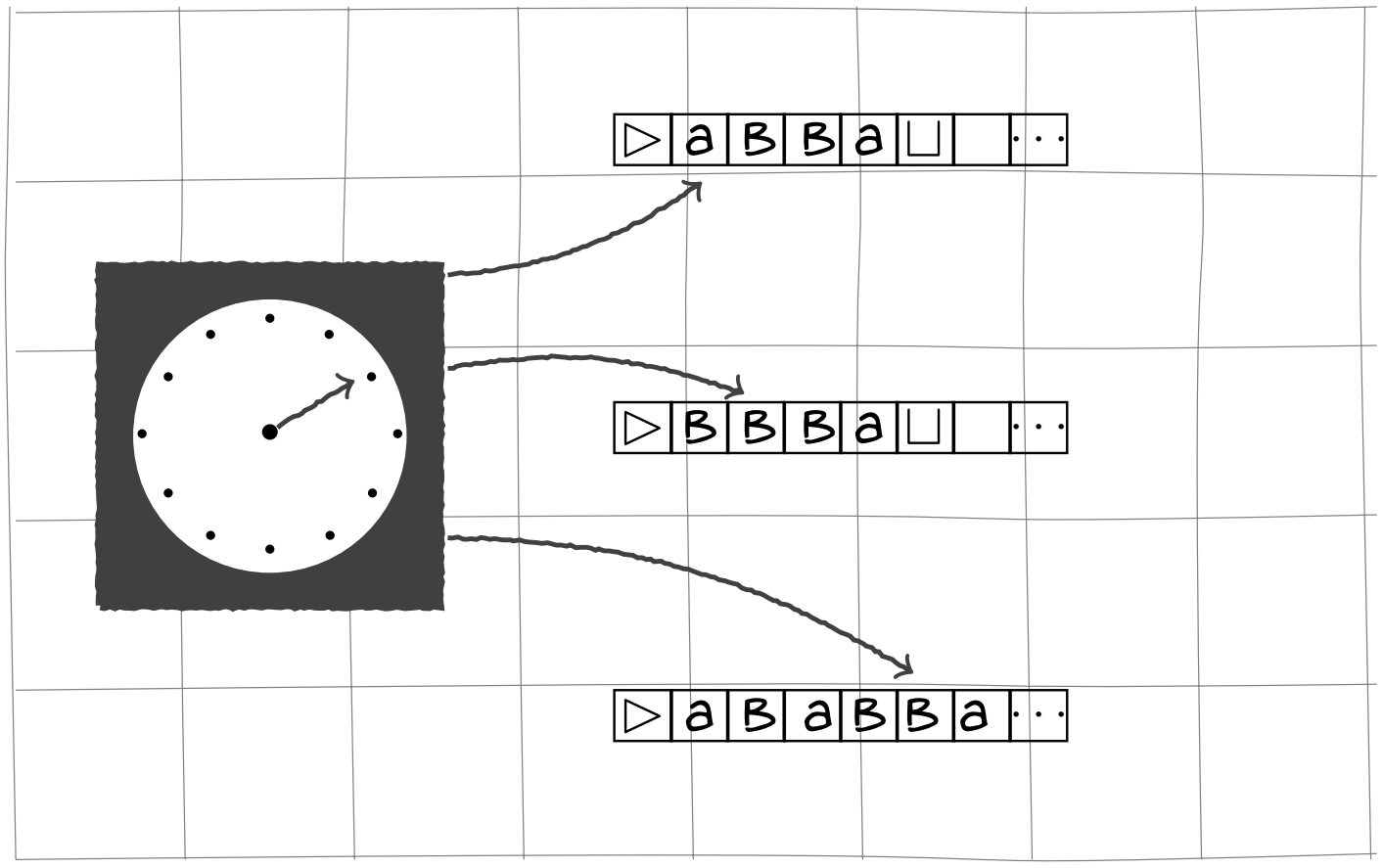
UFF

14 de Junho de 2023

Luís Felipe

14/06/23

Máquina de Turing multifitas



Luís Felipe

14/06/23

Composição

- MT comum com várias fitas
- Cada fita tem seu próprio cabeçote para leitura e escrita
- A função de transição permite ler, escrever e mover cabeçotes de fitas distintas simultaneamente
  - $\delta(q_1, a_1, a_2, \ldots, a_k) = (q', b_1, b_2, \ldots, b_k), b_i \in (\Sigma \setminus \{\triangleright\}) \cup \{\leftarrow, \rightarrow\}$
- Seriam as MT multifitas mais poderosas do que as MTs comuns?
  - Não! Como podemos mostrar esta equivalência?

Luís Felipe

14/06/23

Equivalência

Teorema: Toda MT multifita tem uma MT de fita única equivalente.

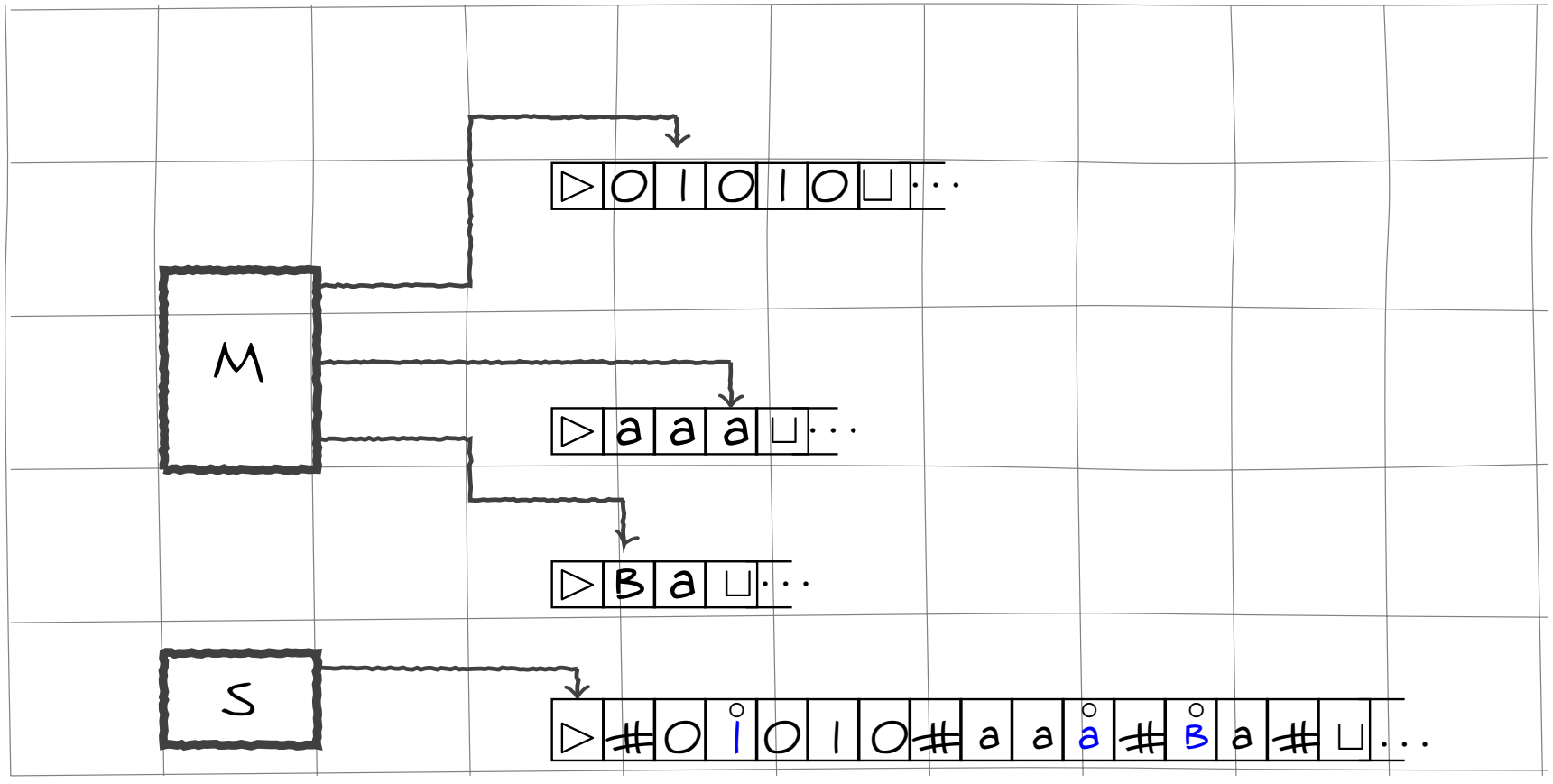
Ideia: Construir uma MT que simule a MT multifita. Esta MT:

- Utiliza o símbolo # como delimitador para separar o conteúdo de diferentes fitas.
- Para marcar onde está o cabeçote em cada fita, usamos uma marca (bolinha) acima do símbolo no qual está posicionado o cabeçote, em cada fita.
- Estes símbolos marcados são novos símbolos adicionados ao alfabeto da fita.

Luís Felipe

14/06/23

Esquema equivalência



Luís Felipe

14/06/23

Como simular a MT original?

Na nova MT:

- 1ª etapa: O cabeçote deve varrer toda a fita desde $\triangleright$ até encontrar o $(k+1)$-ésimo símbolo $\#$
  - Durante esta varredura, a máquina deve armazenar em seu estado os símbolos com ponto que ela encontrar, pois estes são os símbolos sobre os quais os cabeçotes da máquina original estariam.
- 2ª etapa: O cabeçote retorna até o $\triangleright$ e, novamente, varre toda a fita até encontrar o $(k+1)$-ésimo símbolo $\#$.
  - Durante esta varredura, a máquina deve executar, em cada parte da fita, a ação definida pela função de transição da máquina original para cada uma das suas fitas.

Luís Felipe

14/06/23

Concluindo a simulação

O que falta analisar?

- Se a ação for mover o cabeçote para a casa mais a esquerda da fita associada, então marque o símbolo #.
- Se a ação for mover o cabeçote para ␣ tal que não haja símbolo do alfabeto da entrada a direita, então estaríamos sobre #. Dessa forma, faça um shift-right na fita em # e ponha neste lugar ␣̊ (␣ com bolinha)

Luís Felipe

14/06/23

Máquina de Turing não-determinística

E se tivermos mais de uma possibilidade de transição numa MT em detrimento de um par (q, σ), para $q \in Q$ e $\sigma \in \Sigma$?!

Formalmente, uma máquina de Turing não-determinística (MTND) é um 6-upla $M = (\Sigma_0, \Sigma, Q, q_0, F, \delta)$, onde:

1. Σ_0 é o alfabeto da entrada, satisfazendo $\Sigma_0 \cap \{\triangleright, \sqcup, \leftarrow, \rightarrow\} = \emptyset$
2. Σ é o alfabeto da fita, satisfazendo $\Sigma_0 \cup \{\triangleright, \sqcup\} \subseteq \Sigma$ e $\Sigma \cap \{\leftarrow, \rightarrow\} = \emptyset$
3. Q é o conjunto finito de estados
4. $q_0 \in Q$ é o estado inicial
5. $F \subseteq Q$ é o conjunto de estados finais
6. δ é a função de transição não-determinística $\delta : (Q \setminus F) \times \Sigma \to \mathcal{P}(Q \times ((\Sigma \setminus \{\triangleright\}) \cup \{\leftarrow, \rightarrow\}))$

Obs.: Só mudamos a função de transição em relação a MT determinística

Luís Felipe

14/06/23

Aceitação e decisão

- Uma MTND aceita (ou semi-decide) uma linguagem $L \subseteq \Sigma^*$ se para qualquer palavra $w \in L$:
  - $w \in L \leftrightarrow$ existe pelo menos uma computação tal que: $(q_0, \triangleright w) \vdash \cdots \vdash (h, v)$, onde $h \in F$.
- Uma MTND com estados finais S e N decide uma linguagem $L \subseteq \Sigma^*$ se para qualquer palavra $w \in \Sigma^*$, a máquina para com entrada w em S ou N, e:
  - $w \in L \leftrightarrow$ existe pelo menos uma computação tal que: $(q_0, \triangleright w) \vdash \cdots \vdash (S, v)$. E
  - $w \notin L \leftrightarrow$ toda computação é da forma $(q_0, \triangleright w) \vdash \cdots \vdash (N, v)$.

Luís Felipe

14/06/23

Equivalência entre MTND e MT

- Apesar do não-determinismo ser um recurso poderoso, esse novo recurso não representa um ganho de poder computacional.

- Qualquer linguagem semi-decidida ou decidida e qualquer função computada por uma MTND pode ser semi-decidida, decidida ou computada, respectivamente, por uma MT.

- Faremos uma transformação de uma MTND para uma MT com 3 fitas. Como vimos que toda MT com k fitas pode ser convertido numa MT de fita única, então estamos feitos.

Luís Felipe

14/06/23

Árvores de computações da MTND

- Vamos garantir que, para cada entrada, existe uma ordenação (e enumeração) para todas possíveis transições que a função de transição permite para aquela entrada.

- Regras da árvore de computações:

  1. Cada nó: Uma configuração
  2. Filho de um nó: Uma configuração seguinte à do nó
  3. Raiz: Configuração inicial

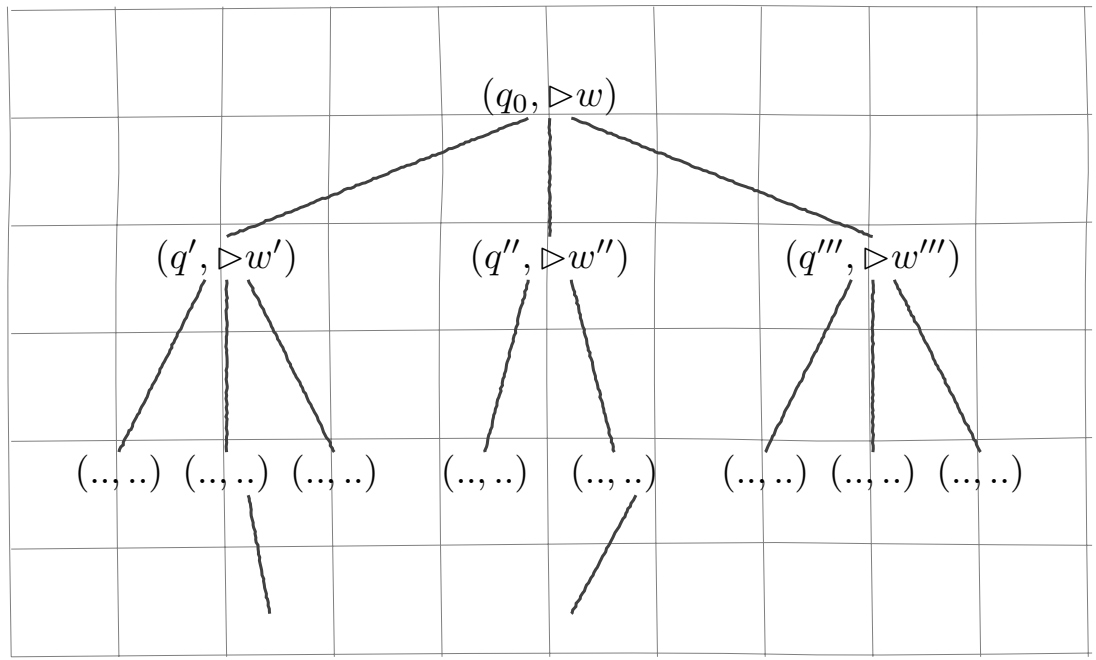
- Vamos percorrer a árvore em nível. Busca em largura na árvore.

Luís Felipe

14/06/23

Computação

Seja $(q_0, \triangleright w)$, temos assim o seguinte esquema:



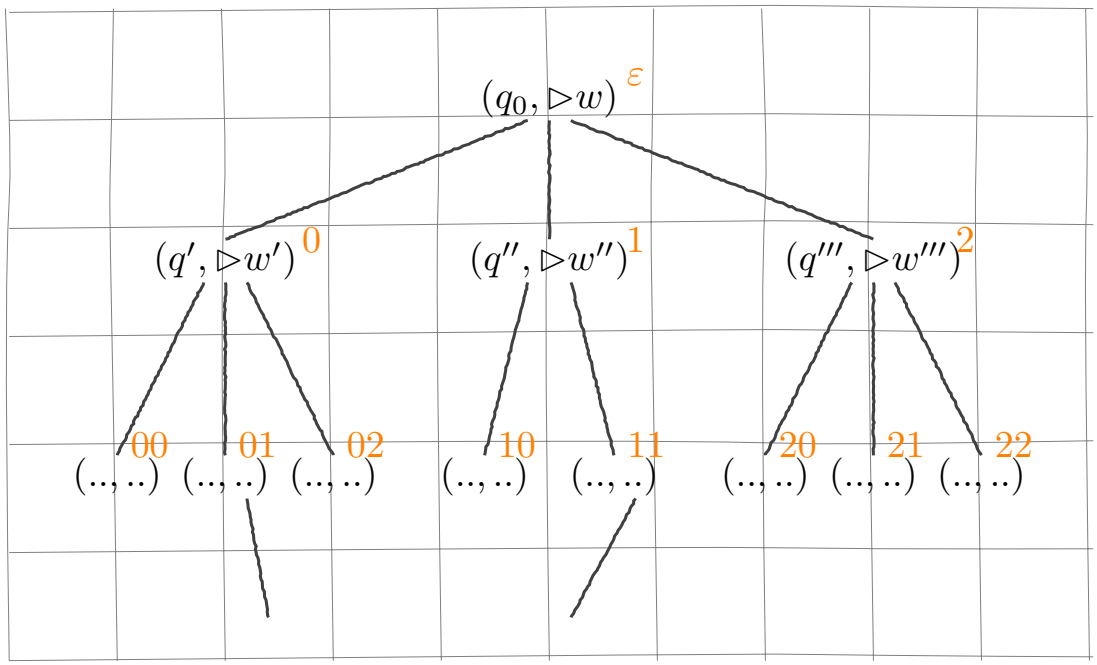
Luís Felipe

14/06/23

Endereçamentos

Forma de endereçar nós na árvore:

- Seja b o número máximo de transições que a função de transição oferece para entrada w.
- Todo nó pode ser endereçado por um número em função b
- Cada nó em um nível n é endereçado por um número de n algarismos.



Luís Felipe

14/06/23

MT com 3 fitas

Simulação da MTND pela MT satisfazendo:

- 1ª fita: Cópia permanente da entrada w
- 2ª fita: Fita onde são similadas as computações da MTND.
- 3ª fita: Endereçamento do ramo da computação a ser simulado.

Após cada simulação, incrementa o endereçamento (em largura) na 3ª fita e copia novamente o conteúdo da 1ª fita para a 2ª.

- Dessa forma, a sequência de endereçamentos na 3ª fita é: $0, 1, 2, \cdots, b, 00, 01, 02, \cdots, bb, \cdots$

Luís Felipe

14/06/23

Equivalente, mas lento

- Apesar de toda MTND ter uma MT associada, esta MT pode executar um número de passos exponencialmente maior do que a MTND. Analisemos o pior caso:

  Teremos (no pior caso): b^0 nós no nível 0
  b^1 nós no nível 1
  b^2 nós no nível 2
  $\vdots$
  b^n nós no nível n

- Portanto, no pior caso, a MT vai percorrer até o último nó mais a direita da árvore, ou seja, a MT vai percorrer o seguinte número de nós:
  - $b^0 + b^1 + b^2 + \cdots + b^n = \frac{b^{n+1}-1}{b-1} = O(b^n)$.
- A MTND alcança esse nó com n transições. Isso implica uma perda de desempenho exponencial.

Luís Felipe

14/06/23

Fechamento por operações

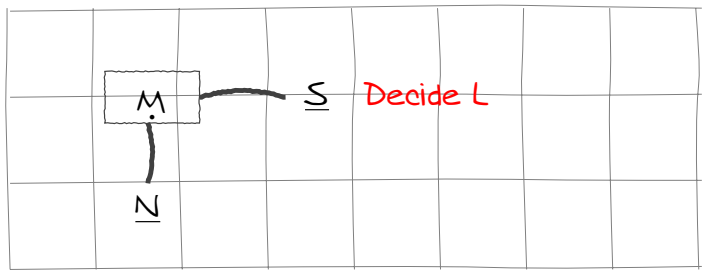
1. Recursivas e Recursivamente Enumeráveis por união
   - M_1 aceita (ou decide) L_1
   - M_2 aceita (ou decide) L_2
   - Queremos uma MT que aceite (ou decida) $L_1 \cup L_2$
   - Ideia: MT com 2 fitas
     - $M = (\Sigma_0, \Sigma, Q, q_0, F, \delta)$
     - $M_1 = (\Sigma_0, \Sigma, Q_1, q_0^1, F_1, \delta_1)$
     - $M_2 = (\Sigma_0, \Sigma, Q_2, q_0^2, F_2, \delta_2)$
     - $Q = Q_1 \times Q_2, \ \ q_0 = (q_0^1, q_0^2)$
     - $F = F_1 \times Q_2 \cup Q_1 \times F_2$ (Para aceitação)
     - Para decisão, devemos transformar os estados finais em S e N
     - $\delta((q_1, q_2), a_1, a_2) = ((q_1', q_2'), b_1, b_2)$ tal que $\delta_1(q_1, a_1) = (q_1', b_1)$ e $\delta_2(q_2, a_2) = (q_2', b_2)$

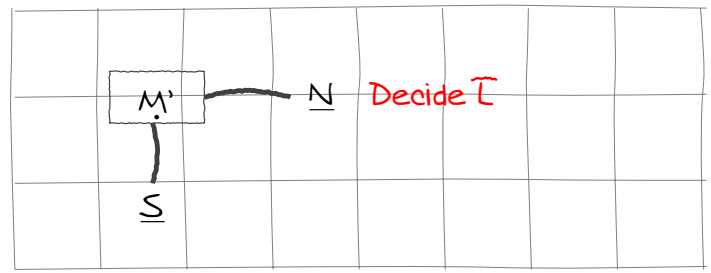
Luís Felipe

14/06/23

Fechamento por operações

3. Recursivas e Recursivamente Enumeráveis por interseção
   - $F = F_1 \times F_2$ (Para Aceitação)
   - Para decisão, (S_1, S_2) é o novo S e $(N_1, q'), q' \in Q_2$ ou $(q', N_2), q' \in Q_1$ conduzem a N.
4. Recursivas por complemento





5. Recursivas por diferença
   - $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$