Máquinas Universais

Andrey Souto Maior
Arthur Manuel Bandeira
Gabriel Vinícius Papa Belini

Programa

- Introdução
  - Máquinas Universais
- Máquina de Turing
  - Computação de Funções
  - Enumeração de Conjuntos
  - Decidibilidade de Conjuntos
- Máquina de Post
- Máquina com Pilhas
- Máquina Norma
- Exercício

Introdução

- Máquinas Universais

"Se for possível representar qualquer algorítmo como um programa em tal máquina, então esta é denominada **Máquina Universal**"

Introdução

- **Evidência Interna**

  “Demonstração de que qualquer extensão das capacidades da máquina proposta não aumenta o seu poder computacional”

- **Evidência Externa**

  “É o exame de outros modelos que definem a noção de algoritmo, juntamente com a prova de que são, no máximo, computacionalmente equivalentes”

Introdução

- ➢ Auto-referência

“É um fenômeno em língua natural ou linguagem formal que consiste de uma oração ou fórmula que refere-se a si mesma diretamente ou através de alguma oração ou fórmula intermediária, ou por meio de alguma codificação.”

Máquina de Turing

Uma máquina de Turing, **M**, é uma óctupla,

$$\mathbf{M} = (E, \Sigma, \Gamma, \#, \beta, \delta, i, F),$$

em que:

- E : é o conjunto de estados;
- $\Sigma \subseteq \Gamma$: é o alfabeto de entrada;
- Γ : é o alfabeto da fita ou alfabeto auxiliar;
- # : é o símbolo marcador de ínicio da fita;
- β : é o símbolo branco;
- $\delta : E \times \Gamma \rightarrow E \times \Gamma \times \{E, D\}$ é a função de transição;
- i : é o estado inicial;
- $F \subseteq E$ é o conjunto dos estados finais.

Máquina de Turing

- Computação de Funções

  Uma função parcial:

$$\mathbf{f}: (\Sigma^*)^n \rightarrow \Sigma^*$$

é dita *Turing-Computável* ou simplesmente *Computável* se existe uma Máquina de Turing **M** que computa **f**.

Máquina de Turing

- Computação de Funções
  - Exemplo

    Considere a função:

$$quadrado: \{1\}^* \rightarrow \{1\}^*$$

tal que associa o valor natural n, representado em unário, ao valor n^2.

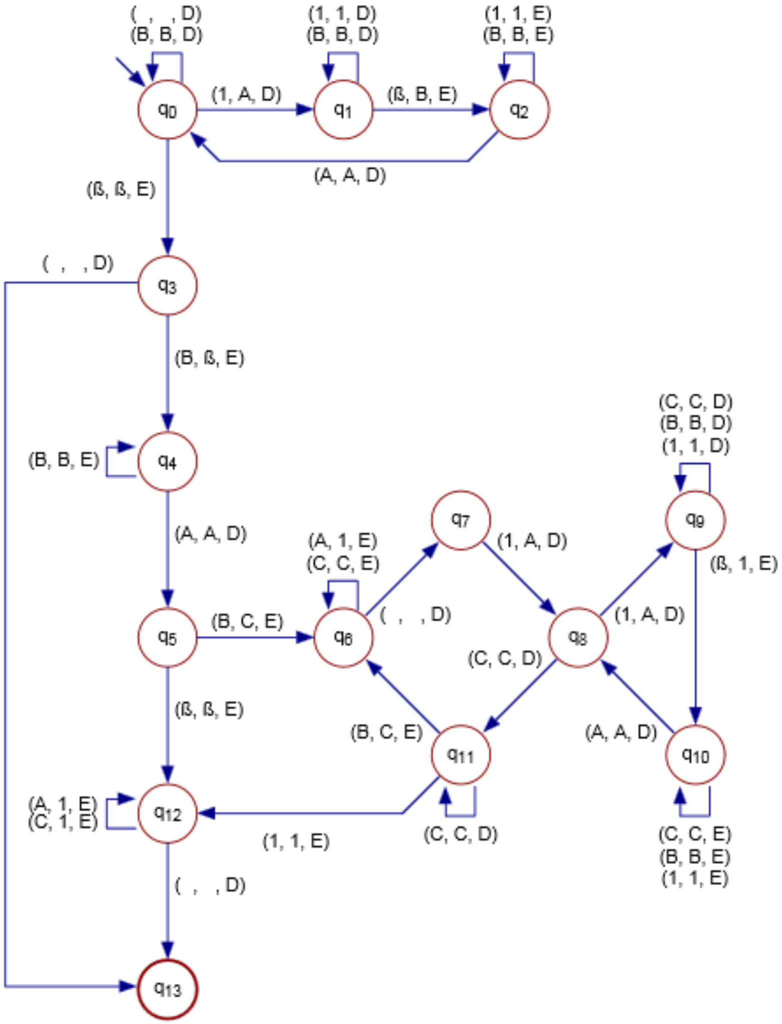
$$Quadrado = (\{q_0, q_1, q_2, \ldots , q_{13}\}, \{1\}, \{1, A, B, C, \#, \beta\}, \#, \beta, \delta, q_0, \{q_{13}\})$$

onde δ é representada pela tabela:

Máquina de Turing

δ	#	1	A	B	C	β
q_0	$(q_0, \#, D)$	(q_1, A, D)		(q_0, B, D)		(q_3, β, E)
q_1		$(q_1, 1, D)$		(q_1, B, D)		(q_2, B, E)
q_2		$(q_2, 1, E)$	(q_0, A, D)	(q_2, B, E)		
q_3	$(q_{13}, \#, D)$			(q_4, β, E)		
q_4			(q_5, A, D)	(q_4, B, E)		
q_5				(q_6, C, E)		(q_{12}, β, E)
q_6	$(q_7, \#, D)$		$(q_6, 1, E)$		(q_6, C, E)	
q_7		(q_8, A, D)				
q_8		(q_9, A, D)			(q_{11}, C, D)	
q_9		$(q_9, 1, D)$		(q_9, B, D)	(q_9, C, D)	$(q_{10}, 1, E)$
q_{10}		$(q_{10}, 1, E)$	(q_8, A, D)	(q_{10}, B, E)	(q_{10}, C, E)	
q_{11}		$(q_{12}, 1, E)$		(q_6, C, E)	(q_{11}, C, D)	
q_{12}			$(q_{12}, 1, E)$		$(q_{12}, 1, E)$	
q_{13}	$(q_{13}, \#, D)$					

Máquina de Turing



Máquina de Turing

- Enumeração de Conjuntos
  - **Linguagem Recursivamente Enumerável (LRE)**
    - Uma linguagem aceita por uma Máquina de Turing é dita Recursivamente Enumerável
    - Palavras de qualquer LRE podem ser enumeradas por uma Máquina de Turing

Máquina de Turing

- Enumeração de Conjuntos
  - Exemplo

Considere a linguagem:

$$\text{Duplo\_Bal} = \{a^n b^n \mid n \geq 0\}$$

A máquina de Turing:

$MT\_Duplo\_Bal = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \{a, b, A, B, \#, \beta\}, \#, \beta, \delta, q0, \{q4\})$

onde δ é representada pela tabela:

Máquina de Turing

δ	#	a	b	A	B	β
q_0	$(q_0, \#, D)$	(q_1, A, D)			(q_3, B, D)	$(q_4, β, D)$
q_1		(q_1, a, D)	(q_2, B, E)		(q_1, B, D)	
q_2		(q_2, a, E)		(q_0, A, D)	(q_2, B, E)	
q_3					(q_3, B, D)	$(q_4, β, E)$
q_4						

Máquina de Turing

- ➢ Decidibilidade de Conjuntos

  - **Tese de Church-Turing**

    “Se uma função é efetivamente computável, então ela é computável por uma Máquina de Turing”

Máquina de Turing

1. Para cada instância de *P* deve existir pelo menos uma palavra de Σ^* que a represente.

2. Cada palavra de Σ^* deve representar no máximo uma instância de *P.*

3. Para cada palavra $w \in \Sigma^*$, deve ser possível determinar se ela representa ou não alguma instância de *P*.

Máquina de Turing

- Máquina de Turing Universal
    - O primeiro passo para se construir uma Máquina de Turing que simule qualquer Máquina de Turing é conceber uma representação para Máquinas de Turing

Máquina de Turing

➢ Exemplo

O alfabeto usado na representação será {0,1}. Seja uma MT qualquer $M = (E, \Sigma, \Gamma, \#, \beta, \delta, i, F)$, onde $E = \{e_1, e_2, \dots, e_n\}$ e $\Gamma = \{a_1, a_2, \dots, a_k\}$. Suponha $e_1 = i$, $a_1 = \#$, $a_2 = \beta$ e lembre-se que $\Sigma \subset \Gamma$.

A representação dos estados e símbolos do alfabeto é dada por: $E = \{e_1 = 1, e_2 = 11, \dots, e_n = 1^n\}$, $\Gamma = \{a_1 = 1, a_2 = 11, \dots, a_k = 1^k\}$. A direção da movimentação do cabeçote é representada por D = 1, E = 11.

Máquina de Turing

Supondo que $F = \{f_1, f_2, \ldots , f_p\}$ e designando por $R<x>$ a representação de x, seja x um estado, um símbolo de Γ ou direção de movimentação do cabeçote, a representação de M tem os seguintes componentes:

- F é representado por uma lista das representações dos estados finais separados por 0, ou seja,

$$R<F> = R<f_1>0R<f_2>0...R<f_p>;$$

- Cada transição t da forma $\delta(ei,aj) = [e'_i, a'_j, d]$ é representada pela palavra

$$R<t> = R<e_i>0R<a_j>0R<e'_1>0R<a'_j>0R<d>.$$

Máquina de Turing

Finalmente, sendo t_1, t_2, …, t_s as transições de M, uma representação de M é:

$$R\langle M\rangle = R\langle F\rangle 00R\langle t_1\rangle 00R\langle t_2\rangle 00...R\langle t_s\rangle.$$

Máquina de Turing

- Exemplo

$$M = (\{0, 1\}, \{a, b\}, \{a, b, \#, \beta\}, \#, \beta, \delta, 0, \{0, 1\})$$

com δ contendo somente as duas transições:

t_1: $\delta(0, a) = [1, a, D]$
t_2: $\delta(1, b) = [0, b, E]$

Máquina de Post

Uma Máquina de Post, **M**, é uma máquina determinística representada pela tripla:

$$\mathbf{M} = (\Sigma, D, \#),$$

onde:

- Σ : alfabeto de entrada;
- D : programa ou diagrama de fluxos;
- # : símbolo auxiliar

Máquina de Post

O diagrama de fluxos pode possuir diferentes estados:

- INÍCIO: Representa o estado inicial da Máquina, é onde devemos começar a processar a string. Será representado por uma elipse;
- ACEITAÇÃO: Estado o qual se atingido indica que a *string* processada foi aceita. Será representado por uma elipse;
- REJEIÇÃO: Estado o qual se atingido indica que determinada *string* não é aceita pela máquina. Será também representado por uma elipse.

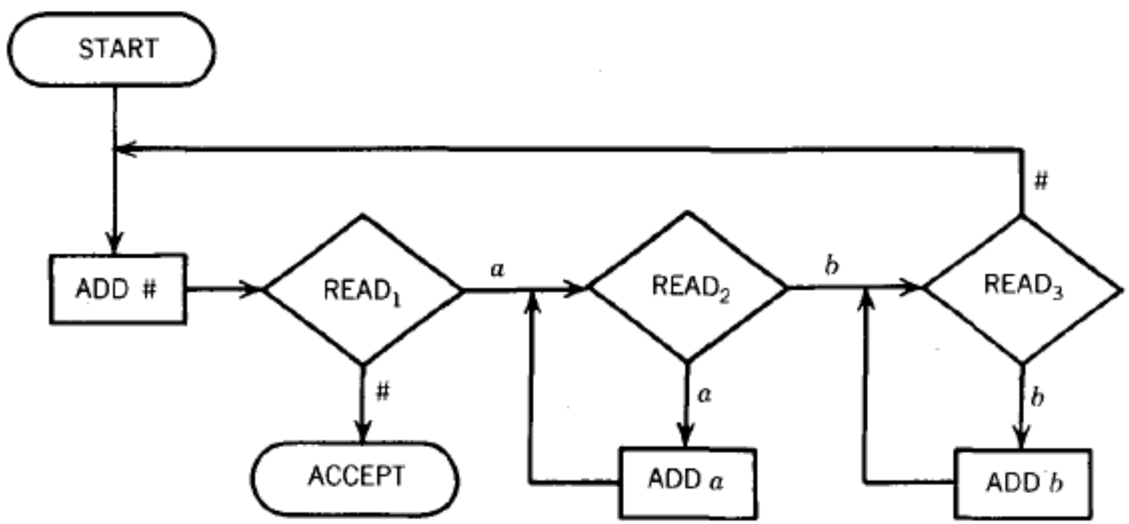
Máquina de Post

- LEITURA: Estado o qual consome um símbolo da string, e toma caminhos diferentes dependendo do símbolo consumido.
- ADIÇÃO: Estado que adiciona determinado símbolo ao final da *string* restante, será representado por um losango.

Ao processar uma *string* utilizando a Máquina de Post sabemos se ela foi aceita se ao processarmos seus símbolos por meio do diagrama de fluxo conseguirmos atingir um estado de ACEITAÇÃO, não é necessário consumir todos os símbolos da palavra para que ela seja aceita. Caso a *string* fique “presa” em algum estado ou atinja um estado de REJEIÇÃO a palavra é rejeitada.

Máquina de Post

Vamos exemplificar o processamento de algumas *strings* utilizando a Máquina Post a seguir: (Λ, aabb, ab, aba)



Máquina de Post

- Equivalência com a Máquina de Turing
  - Máquina de Turing ≤ Máquina de Post

    A estrutura de fita da Máquina de Turing é simulada em Post, usando a variável X, onde a posição corrente da cabeça corresponde à primeira posição da fila

  - Máquina de Post ≤ Máquina de Turing

    A variável X da Máquina de Post é simulada em Turing, usando a estrutura de fita, onde a primeira posição da fila corresponde à posição corrente da cabeça da fita

Máquinas com Pilhas

Uma Máquina com Pilhas, **M**, é uma dupla:

$$\mathbf{M} = (\Sigma, D),$$

onde:

- Σ : alfabeto de entrada;
- D : programa ou diagrama de fluxos;

Máquinas com Pilhas

➢ Autômatos com Duas Pilhas

Um Autômato com Duas Pilhas, **APD**, é uma sêxtupla:

APD = $(E, \Sigma, \Gamma, \delta, i, F)$

- E : é o conjunto de estados;
- $\Sigma \subseteq \Gamma$: é o alfabeto de entrada;
- Γ : é o alfabeto auxiliar;
- $\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \rightarrow E \times (\Gamma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\})$;
- i : é o estado inicial;
- $F \subseteq E$ é o conjunto dos estados finais.

Máquinas com Pilhas

- Equivalência com a Máquina de Turing
  - Máquina de Turing ≤ Autômato com Duas Pilhas

    A estrutura de fita da Máquina de Turing é feita usando a pilha 1 para simular o conteúdo da fita à esquerda da cebeça da fita e a pilha 2 o conteúdo à direita.

  - Autômato com Duas Pilhas ≤ Máquina de Turing
    - A palavra de entrada corresponde às primeiras posições da fita da Máquina de Turing;
    - A pilha 1 corresponde às células ímpares da fita, após a palavra de entrada;
    - A pilha 2 corresponde às células pares da fita, após a palavra de entrada;

Máquina Norma

Uma Máquina Norma (Numer Theoretic Register Machine), **M**, é uma sétupla:

$$\mathbf{M} = (N\infty, N, N, ent, sai, \{ad_k, sub_k\}, \{zero_k\}),$$

onde:

- $N\infty$: cada elemento do conjunto de valores de memória denota uma configuração de seus infinitos *registradores*, os quais são denotados por: A, B, X, Y, …
- ent : função de entrada $N \rightarrow N\infty$ que carrega no registrador denotado o valor de entrada, inicializando todos os demais registradores com zero;
- sai : função de saída $N\infty \rightarrow N$ que retorna o valor corrente do registrador denotado;

Máquina Norma

- ad_k : é uma família de operações indexada pelos registradores onde, para cada registrador K, tem-se que: $ad_k : N^\infty \rightarrow N^\infty$ adiciona um à componente correspondente ao registrador K;
- sub_k : é uma família de operações indexada pelos registradores onde, para cada registrador K, tem-se que: $sub_k : N^\infty \rightarrow N^\infty$ subtrai um à componente correspondente ao registrador K, se o seu valor for maior que zero;
- $zero_k$: é uma família de operações indexada pelos registradores onde, para cada registrador K, tem-se que: $zero_k : N^\infty \rightarrow$ {verdadeiro, falso} resulta em verdadeiro, se a componente correspondente ao registrador K for zero e resulta em falso, caso contrário

Máquina Norma

A Máquina Norma utiliza apenas três operações, sendo elas:

- Adição do valor 1 à algum registrador;
- Subtração do valor 1 à algum registrador;
- Atribuição do valor 0 à algum registrador.

A Máquina Norma, mesmo sendo relativamente simples, possui poder computacional que é no mínimo o de um computador atual.

Máquina Norma

Representação de números racionais utilizando máquina norma:

- Se $r = a/b$, podemos denotar o par ordenado (a,b) como sendo sua representação, exemplo: $r = 0,5 = \frac{1}{2} = (1,2)$

Partindo dessa representação podemos definir operações matemáticas entre números racionais das seguintes formas:

- Adição: $(a,b) + (c,d) = (a*d + b*c, b*d)$
- Subtração: $(a,b) - (c,d) = (a*d - b*c, b*d)$
- Multiplicação: $(a, b) \times (c, d) = (a*c, b*d)$
- Divisão: $(a,b) / (c, d) = (a*d, b*c)$

Máquina Norma

Faremos alguns exemplos de operações com números racionais utilizando as definições mostradas no slide anterior:

1. Adição: 0,5 + 0,25 = ?
2. Subtração: 0,75 - 0,5 = ?
3. Multiplicação: 4,5*2,5 = ?
4. Divisão: (1,4) / (1,2) = ?

Máquina Norma (Exemplos)

Alguns exemplos de operações utilizando registradores:

1)Atribuição do valor 0 em A:

A := 0 (Código em Pascal)

2) Soma de duas variáveis:

A:= A + B (Código em Pascal)

3) Soma de dois valores preservando B:

C := A (Código em Pascal)

A := A + B (Código em Pascal)

Máquina Norma

- Equivalência com a Máquina de Turing
  - Máquina de Turing ≤ Máquina Norma

    A estrutura de fita da Máquina de Turing é simulada em Norma usando uma estrutura de arranjo unidimensional

  - Máquina Norma ≤ Máquina de Turing
    - O conteúdo de cada registrador (valor natural) é implementado de forma unária em Turing
    - O registrador X ocupa as células ímpares da fita, e Y, as pares

Referências Bibliográficas

➢ COHEN, Daniel I. A.. **Introduction to Computer Theory.** 2. ed. New York: John Wiley & Sons, 1997. 634 p.

➢ DIVÉRIO, Tiajarú Asmuz; MENEZES, Paulo Blauth. **Teoria da Computação:** Máquinas Universais e Computabilidade. 2. ed. Porto Alegre: Sagra Luzzatto, 2004. 205 p.

➢ VIEIRA, Newton José. **Introdução aos Fundamentos de Computação:** Linguagens e Máquinas. São Paulo: Pioneira Thomson Learning, 2006. 319 p.

➢ AUTORREFERÊNCIA. 2013. Disponível em: <http://pt.wikipedia.org/wiki/Autorreferência>. Acesso em: 30 abr. 2014.

➢ CASILLO, Danielle. **Teoria da Computação:** Máquinas Universais. Disponível em: <https://www2.ufersa.edu.br/portal/view/uploads/setores/166/arquivos/CienciaComputacao/MáquinasUniversais.pdf>. Acesso em: 30 abr. 2014.

➢ **TESE de Church-Turing**. 2013. Disponível em: <http://pt.wikipedia.org/wiki/Tese_de_Church-Turing>. Acesso em: 30 abr. 2014.

Referências Bibliográficas

- **NORMA register machine**. Disponível em: <http://everything2.com/title/NORMA+register+machine> Acesso em: 2 maio 2014.
- CASILLO, Danielle. **Teoria da Computação:** Máquina de Registradores Norma. Disponível em: <http://www2.ufersa.edu.br/portal/view/uploads/setores/166/arquivos/CienciaComputacao/M%C3%A1quina%20de%20Registradores%20-%20Norma.pdf>. Acesso em: 2 maio 2014.
- CASILLO, Danielle. **Teoria da Computação:** Máquina de Post. Disponível em: <http://www2.ufersa.edu.br/portal/view/uploads/setores/166/arquivos/CienciaComputacao/M%C3%A1quina%20de%20Post.pdf>. Acesso em: 2 maio 2014.