CI059: Introdução à Teoria da Computação

Profa. Carmem Hara

Departamento de Informática/UFPR

1 de agosto de 2023

Roteiro

- Informações da disciplina
- Áreas da Teoria da Computação
- Linguagens formais

Cap. 1 do livro Introdução à Teoria de Autômatos, Linguagens e Computação

Informações Gerais

- Página Web: www.inf.ufpr.br/carmem/ci059
- Email: carmemhara@ufpr.br
- Horário de atendimento: terça-feira, 15:30-16:30

Bibliografia

- **Livro Texto**:
  Introdução à Teoria de Autômatos, Linguagens e Computação
  John E. Hopcroft, Jeffrey D. Ullman, Rajeev Motwani
  Editora Campus, 2003
- Notas de aula do Prof. Murilo:
  https://www.inf.ufpr.br/murilo/teoria/notes/itc-22-2.pdf

Bibliografia Adicional

- *Languages and Machines: An Introduction to the Theory of Computer Science*
  Thomas Sudkamp
  Addison-Wesley, Second Edition, 1998
- Elementos de Teoria da Computação
  Harry F. Lewis, C. H. Papadimitriou
  2a Edicao, Editora Bookman
- Theory of Computation
  Wood, D.
  Ed. John Wiley & Sons, 1987
- Introdução aos Fundamentos da Computação
  Newton José Vieira
  Ed. Cengage Learning; 1ª edição, 2006

Avaliação

- 2 provas:
  - Primeira (P1): 03/outubro/2023, terça-feira
  - Segunda (P2): 30/novembro/2023, quinta-feira
  - Exame Final: 07/dezembro/2023, quinta-feira
- Média = P1*0.5 + P2*0.5

Áreas da Teoria da Computação

O estudo da teoria da computação começou com as perguntas:

- COMO? as linguagens são definidas
  $\longrightarrow$ Linguagens formais
- O QUE? conseguimos computar
  $\longrightarrow$ Computabilidade
- QUANTO CUSTA? computar
  $\longrightarrow$ Complexidade

Linguagens Formais

- Estudo de gramáticas de Noam Chomsky
- Classes de linguagens (Hierarquia de Chomsky)
  - Regulares
  - Livres de Contexto
  - Sensíveis a contexto
  - Recursivamente enumeráveis

- Características de um algoritmo
  - completo: sempre produz um resultado
  - finito: tem uma quantidade finita de instruções
  - determinístico: sempre produz o mesmo resultado para a mesma entrada
- necessidade de um modelo formal e abstrato de computação
  Exemplos: funções recursivas, cálculo lambda, maquinas RAM, máquinas de Turing
- provado que são todos equivalentes

Tese de Church-Turing

Um problema tem solução algoritmica se e somente se pode ser resolvido por uma máquina de Turing que pára para todas as entradas possíveis.

- existem problemas com solução computacional e sem solução computacional
- Exemplos de problemas indecidíveis:
  - determinar se um programa termina para todas as entradas possíveis
  - determinar se dois programas computam a mesma função

- *problemas tratáveis:* com solução em tempo polinomial
- *problemas intratáveis:* com solução em tempo exponencial tem solução com recursos ilimitados

Classes de problemas:

- *Classe P:* que podem ser resolvidos em tempo polinomial por uma máquina de Turing determinística
- *Class NP:* que podem ser resolvidos em tempo polinomial por uma máquina de Turing NÃO determinística
  Ex: Problema do caixeiro viajante: procura de um circuito no grafo que possua a menor distância, começando numa cidade qualquer, entre várias, visitando cada cidade precisamente uma vez e regressando à cidade inicial

Uma linguagem e' um conjunto de palavras sobre um alfabeto.

Linguagens formais X Linguagens naturais

- sintaxe bem definida
- semântica não ambígua
  Ex: "Jose cumprimentou Joao. Ele não tinha pressa."

Alfabeto

Alfabeto (Σ): é um conjunto finito de símbolos indivisíveis.

Geralmente usamos letras do início do alfabeto (a,b,c) e dígitos numéricos para representar símbolos do alfabeto.

- $\Sigma_1 = \{$a,b,c,s$\}$
- $\Sigma_2 = \{0, 1\}$ (alfabeto binário)

Palavra

Palavra: uma sequência finita de elementos de um alfabeto.

Geralmente usamos letras no final do alfabeto (u, v, w, x, y, z) para representar palavras.

- Ex. em Σ_1: casa, aba, aaa, b
- Ex. em Σ_2: 0011, 0^3

- tamanho da palavra ($|w|$): quantidade de símbolos do alfabeto em w
  Ex: $|\text{casa}| = 4$
- palavra vazia (ϵ): $|\epsilon| = 0$

Operações sobre Palavras, Subpalavras

- Concatenação ($x.y$): todos os símbolos da palavra x seguidos por todos os símbolos da palavra y
  Ex: casa.ca = casaca
  Ex: casa.ϵ = casa
- Reverso (w^R): Ex: $(\text{casa})^R$ = asac
  Definição recursiva:
  1. se $w = \epsilon$ então $w^R = \epsilon$
  2. se $w = x.a$ então $w^R = a.x^R$
- Subpalavras (Ex: w=casa)
  - se $w = w_1.w_2.w_3$ então w_2 é uma **subpalavra** de w
    {ϵ, c, a, s, ca, as, sa, cas, asa, casa}
  - se $w = w_1.w_2$ então w_2 é um **sufixo** de w
    {ϵ, a, sa, asa, casa}
  - se $w = w_1.w_2$ então w_1 é um **prefixo** de w
    {ϵ, c, ca, cas, casa}

Σ^*: Fecho Kleene do Alfabeto

$\boxed{\Sigma^*\text{: conjunto de todas as palavras sobre o alfabeto } \Sigma}$

Pode ser definido recursivamente da seguinte forma:

1. $\epsilon \in \Sigma^*$
2. se $w \in Sigma^*$ e $a \in \Sigma$ então $w.a \in Sigma^*$

Linguagem

$$\boxed{\text{Linguagem sobre um alfabeto } \Sigma \text{ é um subconjunto de } \Sigma^*}$$

Exemplos considerando o alfabeto binário $\Sigma = \{0, 1\}$

- $L_1 = \{0, 1, 01\}$
- $L_2 = \{0^n\ 1^n \mid n \geq 0\}$
- $L_3 = \emptyset$
- $L_4 = \{\epsilon\}$
- $L_5 =$ palavras que começam com 0 e tem comprimento par
  Pode ser definida recursivamente da seguinte forma:
  1. $00 \in L_5$ e $01 \in L_5$
  2. se $w \in L_5$ então $w00$, $w11$, $w01$, e $w10$ também pertencem a L_5

Operações sobre Linguagens: Operações de Conjuntos

- União: $L_1 \bigcup L_2$
  $\{w \mid w \in L_1 \text{ ou } w \in L_2\}$
- Diferença: $L_1 - L_2$
  $\{w \mid w \in L_1 \text{ mas } w \notin L_2\}$
- Intersecao: $L1 \bigcap L_2$
  $\{w \mid w \in L_1 \text{ e } w \in L_2\}$

Exemplo:

- $L_1 = \{\epsilon, 0, 1, 01, 10\}$
- $L_2 = \{0^n\ 1^n \mid n \geq 0\}$

$L_1 \bigcup L_2 =?$
$L_1 - L_2 =?$
$L_1 \bigcap L_2 =?$

Operações sobre Linguagens: Concatenação $L_1.L_2$

$L_1.L_2 = \{u.v \mid u \in L_1 \text{ e } v \in L_2\}$

Exemplo:

$L_1 = \{0, 1\}$
$L_2 = \{00, 01, 10, 11\}$
$L_3 = \{w \in \{0, 1\}^* \mid |w| = 5\}$
$L_4 = \{0y \mid y \in 0, 1^*\}$

- $L_1.L_2 = \{000, 001, 010, 011, 100, 101, 110, 111\}$
  $= \{w \in \{0, 1\}^* \mid |w| = 3\}$
- $L_3.L_3 = \{w \in \{0, 1\}^* \mid |w| = 10\}$
- $L_3.L_4 = \{w \in \{0, 1\}^* \mid \text{ o sexto símbolo de } w \text{ é } 0\}$
- $L_4.L_3 = \{em\{0, 1\}^* \mid w \text{ começa com } 0 \text{ e } |w| \geq 6\}$

Entenda por que:

- $\{0, 1\}.\{\} = \{\}$
- $\{0, 1\}.\{\epsilon\} = \{0, 1\}$

Operações sobre Linguagens: Concatenação - Notação L^n

L^n representa a operação de concatenar a linguagem L com ela mesma n vezes.

L^n pode ser definida recursivamente da seguinte forma:

1. $L^0 = \{\epsilon\}$
2. $L^n = L^{n-1}.L$

Exemplos:
$L_2 = \{00, 01, 10, 11\}$
$L_3 = \{w \in \{0,1\}^* \mid |w| = 5\}$

- $L_2^0 = L_3^0 = \{\epsilon\}$
- $L_2^3 = L_2.L_2.L_2 = \{w \in \{0,1\}^* \mid |w| = 6\}$
- $L_3^2 = L_3.L_3 = \{w \in \{0,1\}^* \mid |w| = 10\}$

Operações sobre Linguagens: Fecho Kleene L^*

O Fecho Kleene (L^*) é a operação de concatenar a linguagem L com ela mesma zero ou mais vezes.

L^* pode ser definido recursivamente da seguinte forma:

1. ϵ pertence a L^*
2. se $xinL^*$ e $y \in L$ então $x.y \in L^*$

Exemplos:

$L_2 = \{00, 01, 10, 11\}$
$L_3 = \{w \in \{0,1\}^* \mid |w| = 5\}$
$L_5 = \{0\}$

- $L_2^* = \{w \in \{0,1\}^* \mid |w|$ é zero ou é par$\}$
- $L_3^* = \{w \in \{0,1\}^* \mid |w|$ é zero ou é um múltiplo de 5$\}$
- $L_5^* = \{O^n \mid n \geq 0\}$

Mais Exemplos com o alfabeto binário: $\Sigma = \{0, 1\}$

- $L_6 = \{0, 1\}^*$: linguagem com todas as palavras formadas por 0s e 1s
- $L_7 = \{00\}$: linguagem com uma única palavra 00

1. Linguagem das palavras que contém o substring 00
   $\{0, 1\}^*.\{00\}.\{0, 1\}^*$
2. Linguagem das palavras que começam com 00
   $\{00\}.\{0, 1\}^*$
3. Linguagem das palavras que terminam com 00
   $\{0, 1\}^*.\{00\}$

Operações sobre Linguagens: L^+

L^+ é a operação de concatenar a linguagem L com ela mesma **uma** ou mais vezes. Ou seja, $L^+ = L^*.L$

Perguntas:

- $L^* - \{\epsilon\} = L^+$?
- $L^+ \bigcup \{\epsilon\} = L^*$?