Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologias
Engenharia da Computação

Bruno Feres de Souza

Disciplina: Linguagens Formais e Autômatos
Código: EECP0020

22 de fevereiro de 2021

Conteúdo programático

- Elementos de matemática discreta
- Conceitos básicos de linguagens
- Linguagens regulares e autômatos finitos
- Linguagens livres de contexto e autômatos de pilha
- Linguagens sensíveis ao contexto e Máquinas de Turing com fita limitada
- Linguagens recursivas e Máquinas de Turing com finta infinita
- Linguagens recursivamente enumeráveis

Sumário

- Contextualização
- Símbolos, cadeias e alfabetos
- Linguagens
- Hierarquia de Chomsky

Conceitos didáticos

- Segundo o dicinário **Michaelis**, linguagem pode ser definida como:
  - Faculdade que tem todo homem de comunicar seus pensamentos e sentimentos.
  - Conjunto de sinais falados, escritos ou gesticulados de que se serve o homem para exprimir esses pensamentos e sentimentos.
  - Faculdade inata de todo indivíduo de aprender e usar uma língua.
- Tais definições, embora nos deem uma noção intuitiva do conceito, não são precisas o suficiente para o estudo de linguagens formais.

Definições básicas

- **Símbolos**, ou **átomos**, são as entidades básicas do estudo de linguagens. São consideradas unidades atômicas e indivisíveis, não importando sua representação visual particular.
    - São símbolos (dependendo do contexto): a, abc, if, 5, 32.
- **Alfabetos** são conjuntos finitos não-vazios de símbolos.
    - São alfabetos: $\{a, b, c, ..., z\}$, $\{abc, def, ghi\}$, $\{while, for, if, else\}$, $\{0, 1, 2, ..., 9\}$, $\{2, 4, 8, 16, 32, 64\}$
- **Cadeias**, ou **palavras**, são sequências finitas de símbolos de um alfabeto justapostos.
    - São cadeias: abc, $abcdef$, $ifelse$, 012, 16322.

Definições básicas

- Geralmente, utilizam-se as sequintes convenções para representação:
  - Símbolos: letras minúsculas do início do alfabeto romano $(a, b, c, ...)$.
  - Cadeias: letras minúsculas do final do alfabeto romano $(r, s, x, w, ...)$ ou letras minúsculas do alfabeto grego $(\alpha, \beta, \gamma, ...)$.
  - Alfabetos: letras maiúsculas do alfabeto grego $(\Sigma, \Gamma, \Delta, ...)$.

Definições sobre cadeias

- **Comprimento**: é a quantidade $|\alpha|$ de símbolos da cadeia α.
  - Sobre o alfabeto binário $\Sigma = \{0, 1\}$, tem-se que $|0| = 1$, $|01| = 2$ e $|101| = 3$.
- **Cadeia elementar**: é toda cadeia de comprimento 1.
  - Sobre o alfabeto binário $\Sigma = \{0, 1\}$, são cadeias unitárias 0 e 1.
- **Cadeia vazia**: é a cadeia ϵ tal que $|\epsilon| = 0$.

Definições sobre cadeias

- **Concatenação**: é a operação binária realizada sobre duas cadeias α e β (elementares ou não) que resulta em uma nova cadeia $\alpha\beta$ formada pela justaposição ordenada dos símbolos que compõem os seus operandos separadamente.
  - Sobre o alfabeto $\Sigma = \{a, b, c, d\}$, as seguintes contenações são válidas para as cadeias $\alpha = abc$, $\beta = dbaca$ e $\sigma = a$: $\alpha\beta = abcdbaca$, $\beta\alpha = dbacaabc$ e $(\alpha\beta)\sigma = \alpha(\beta\sigma) = abcdbacaa$.
  - A concatenação é uma operação associativa, mas não comutativa.
  - A cadeia vazia ϵ é o elemento neutro em relação à operação de concatenação. Assim, tem-se que $\alpha\epsilon = \epsilon\alpha = \alpha$ e $|\alpha\epsilon| = |\epsilon\alpha| = |\alpha|$.

Definições sobre cadeias

- **Concatenação sucessiva**: dada uma cadeia w, a concatenação sucessiva de w é definida indutivamente a partir da operação de concatenação binária, como segue:
  $w^0 = \epsilon$
  $w^n = ww^{n-1}$
  onde n é o número de concatenações sucessivas.
  - Por exemplo, seja a cadeia $w = a$, então:
    - $w^0 = \epsilon$
    - $w^1 = a$
    - $w^2 = aa$
    - $w^3 = aaa$
    - $w^4 = aaaa$
    - $w^5 = aaaaa$

Definições sobre cadeias

- **Conjunto de todas as cadeias**: seja Σ um alfabeto. O conjunto de todas as possíveis cadeias sobre Σ é chamado de Σ^*. Formalmente, tal conjunto é definido por

  $$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots = \bigcup_{i=0}^{\infty} \Sigma^i,$$

  onde
  Σ^0 é a concatenação de todas as cadeias sobre Σ com comprimento 0
  Σ^1 é a concatenação de todas as cadeias sobre Σ com comprimento 1
  Σ^2 é a concatenação de todas as cadeias sobre Σ com comprimento 2
  Σ^3 é a concatenação de todas as cadeias sobre Σ com comprimento 3
  ...

Definições sobre cadeias

- Como exemplo do **conjunto de todas as cadeias**, considere $\Sigma = \{a, b, c\}$. Então Σ^* é obtido pela união dos seguintes conjuntos:
  $\Sigma^0 = \{\epsilon\}$
  $\Sigma^1 = \{a, b, c\}$
  $\Sigma^2 = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$
  $\Sigma^3 = \{aaa, aab, aac, aba, abb, abc, ..., ccc\}$
  ...

Definição

- Uma **linguagem formal** é um conjunto, finito ou infinito, de cadeias de comprimento finito, formadas pela concatenação de elementos de um alfabeto finito e não-vazio.

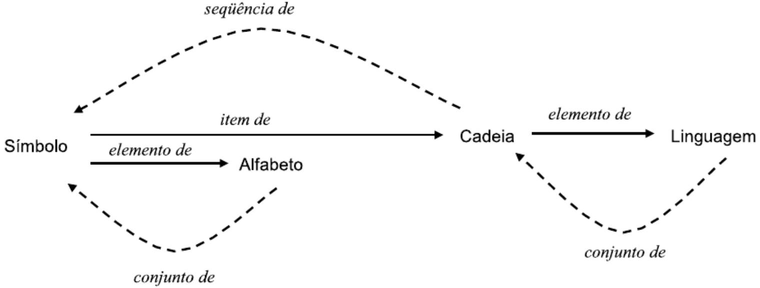

Figura: Símbolo, alfabeto, cadeia e linguagem

Definição

- Como exemplo do esquema anterior, tem-se o seguinte diagrama:

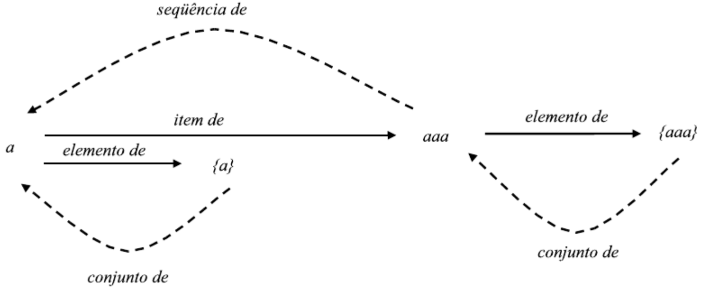

Figura: Exemplo de Símbolo, alfabeto, cadeia e linguagem

Definição

- Uma **linguagem formal** é um conjunto, finito ou infinito, de cadeias de comprimento finito, formadas pela concatenação de elementos de um alfabeto finito e não-vazio.

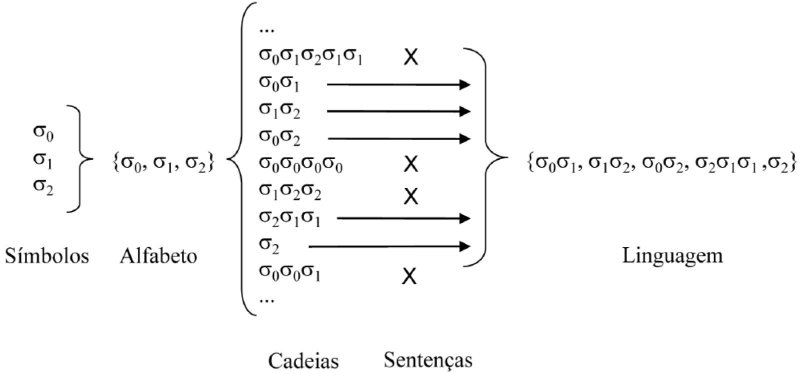

Figura: Símbolo, alfabeto, cadeia e linguagem

Definição

- Como exemplo do esquema anterior, tem-se o seguinte diagrama:

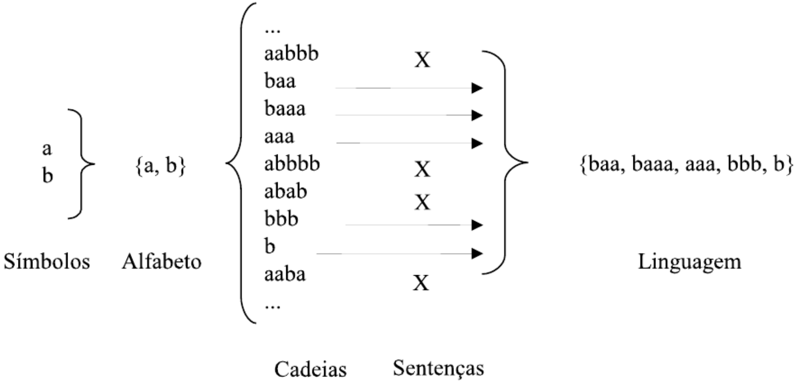

Figura: Exemplo de Símbolo, alfabeto, cadeia e linguagem

Definição

- Como ilustração, tem-se as seguintes linguagens:
  - A linguagem de todas as cadeias que consistem em n valores 0 seguidos por n valores 1: $\{\epsilon, 01, 0011, 000111, ...\}$, $n \geq 0$.
  - O conjunto de cadeias de valores 0 e 1 com um número igual de cada um deles: $\{\epsilon, 01, 10, 0011, 0101, 1001, ...\}$.
  - O conjunto de números binários cujo valor é um número primo: $\{10, 11, 101, 111, 1011, ...\}$.
  - $\emptyset$, a linguagem vazia, é uma linguagem sobre qualquer alfabeto.
  - $\{\epsilon\}$, a linguagem que consiste apenas na cadeia vazia, também é uma linguagem sobre qualquer alfabeto.
  - A língua portuguesa.
  - Qualquer linguagem de programação.

Operações e propriedades

- **Concatenação**: a concatenação de duas linguagens X e Y, denotada por XY, corresponde ao conjunto de todas as cadeias obtidas pela concatenação de qualquer elemento de X com qualquer elemento de Y, ou seja:

$$XY = \{xy | x \in X, y \in Y\}$$

.

- Como casos particulares, tem-se que:

$$L^0 = \{\epsilon\}$$

$$L^1 = L$$

Operações e propriedades

- Como exemplo de **concatenação**, considere $L = \{001, 10, 111\}$ e $M = \{\epsilon, 001\}$. Então,

$$LM = \{001, 10, 111, 001001, 10001, 111001\}$$

Operações e propriedades

- **Fechamento reflexivo e transitivo**: o fechamento reflexivo e transitivo de uma linguagem L é denotado por L^* e representa o conjunto de cadeias que podem ser formadas tomando-se qualquer número de cadeias de L, possivelmente com repetições, e concatenando-se todas elas. Formalmente, tem-se que:

$$L^* = L^0 \cup L^1 \cup L^2 \cup \cdots = \bigcup_{i=0}^{\infty} L^i$$

Operações e propriedades

- Como exemplo de **fechamento reflexivo e transitivo**, considere $L = \{0, 11\}$. Então, deve-se realizar a união dos seguintes conjuntos:
  $L^0 = \{\epsilon\}$
  $L^1 = \{0, 11\}$
  $L^2 = \{00, 011, 110, 1111\}$
  $L^3 = \{000, 0011, 0110, 1100, 01111, 11011, 11110, 111111\}$
  ...

Operações e propriedades

- Como exercício de **fechamento reflexivo e transitivo**, seja L o conjunto de todas as cadeias de 0. Pede-se então que se calcule L^*.

Operações e propriedades

- A operação de **fechamento reflexivo e transitivo** pode ser aplicada a um alfabeto Σ. Nesse caso, Σ^* segue a mesma definição do conjunto de todas as cadeias visto anteriormente.
- Como uma linguagem qualquer L é um conjunto de cadeias sobre um alfabeto Σ e Σ^* designa o conjunto de todas as cadeias sobre Σ, então tem-se que $\boldsymbol{L} \subseteq \boldsymbol{\Sigma^*}$.

Operações e propriedades

- Em relação à operação de **fechamento reflexivo e transitivo** pode-se fazer as seguintes observações:
  - $\varnothing$ é o conjunto constituído por zero cadeias e corresponde à **menor** linguagem que se pode definir sobre um alfabeto Σ qualquer.
  - Σ^* é o conjunto de todas as cadeias possíveis de serem construídas sobre Σ e corresponde à **maior** de todas as linguagens que pode ser definida sobre Σ.
  - 2^{Σ^*} é o conjunto de todos os subconjuntos possíveis de serem obtidos a partir de Σ^*, e corresponde ao conjunto formado por todas as possíveis linguagens que podem ser definidas sobre Σ. Note-se que $\varnothing \in 2^{\Sigma^*}$, e também que $\Sigma^* \in 2^{\Sigma^*}$.

Operações e propriedades

- Em relação à operação de **fechamento reflexivo e transitivo**, tem-se exemplo que segue. Seja $\Sigma = \{a, b, c\}$ e P o conjunto formado pela única propriedade "todas as cadeias são iniciadas com o símbolo a". Então:
  - A linguagem $L_0 = \varnothing$ é a menor linguagem que pode ser definida sobre Σ.
  - A linguagem $L_1 = \{a, ab, ac, abc, acb\}$ é finita e observa P.
  - A linguagem $L_2 = \{a\}\{a\}^*\{b\}^*\{c\}^*$ é infinita e observa P.
  - A linguagem $L_3 = \{a\}\{a, b, c\}^*$ é infinita, observa P e, dentre todas as que observam P, trata-se da maior linguagem, pois não existe nenhuma outra cadeia sem Σ^* que satisfaça a P e não pertença a L_3.
  - $L_0 \subseteq \Sigma^*,\ L_1 \subseteq \Sigma^*,\ L_2 \subseteq \Sigma^*,\ L_3 \subseteq \Sigma^*$.
  - $L_0 \in 2^{\Sigma^*},\ L_1 \in 2^{\Sigma^*},\ L_2 \in 2^{\Sigma^*},\ L_3 \in 2^{\Sigma^*}$
  - Além de $L_0,\ L_1,\ L_2$ e L_3, existem inúmeras outras linguagens que podem ser definidas sobre Σ.

Operações e propriedades

- **Fechamento transitivo**: o fechamento transitivo de um alfabeto Σ, denotado por Σ^+, é definido de maneira análoga ao fechamento reflexivo e transitivo, diferindo deste apenas por não incluir o conjunto Σ^0:

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \cdots = \bigcup_{i=1}^{\infty} \Sigma^i$$

Operações e propriedades

- Como exemplo de **fechamento transitivo**, considere $\Sigma = \{n, (,), +, *, -, /\}$. Neste caso:
  - $\Sigma^* = \{\epsilon, n, n+n, -n), */, n()), n-(n*n), ...\}$
  - $\Sigma^+ = \{n, n+n, -n), */, n()), n-(n*n), ...\}$
  - $\Sigma^+ = \Sigma^* - \{\epsilon\}$, pois $\epsilon \notin \Sigma$.

Operações e propriedades

- **Complementação**: a complementação de uma linguagem X sobre um alfabeto Σ é definida como:

$$\overline{X} = \Sigma^* - X$$

- Por exemplo, considere a linguagem X de cadeias de valores 0 e 1 com um número igual de cada um deles: $\{\epsilon, 01, 10, 0011, 0101, 1001, ...\}$. Então:

$$\overline{X} = \{001, 110, 01011, 00101, 11001, ...\},$$

representando todas as cadeias de valores 0 e 1 com número diferente de cada um deles.

Operações e propriedades

- **Reversão**: diz-se que uma linguagem L_1 é o reverso de uma linguagem L_2, denotando-se o fato por $L_1 = L_2^R$ (ou $L_2 = L_1^R$), quando as sentenças de L_1 corresponderem ao reverso das sentenças de L_2. Formalmente:

$$L_1 = L_2^R = \{x^R | x \in L_2\}$$

- Por exemplo, seja $L_2 = \{\epsilon, a, ab, abc\}$. Então, $L_1 = L_2^R = \{\epsilon, a, ba, cba\}$.

Operações e propriedades

- **Prefixo (sufixo) próprio**: diz-se que uma linguagem exibe a propriedade do prefixo (sufixo) próprio sempre que não houver nenhuma cadeia a ela pertencente que seja prefixo (sufixo) próprio de outra cadeia dessa mesma linguagem. Formalmente:

$$\text{Prefixo próprio: não existe } \alpha \in L | \beta \neq \epsilon, \alpha\beta \in L$$

$$\text{Sufixo próprio: não existe } \alpha \in L | \beta \neq \epsilon, \beta\alpha \in L$$

Operações e propriedades

- Como exemplo de **prefixo (sufixo) próprio**, considere as seguintes linguagens:

$$L_1 = \{a^i b^i | i \geq 1\} = \{ab, aabb, aaabbb, ...\}$$

$$L_2 = \{ab^i | i \geq 1\} = \{ab, abb, abbb, abbbb...\}$$

- Neste exemplo, a linguagem L_1 exibe a propriedade do prefixo próprio, ao passo que a linguagem L_2 não a exibe. A propriedade do sufixo próprio é exibida por ambas as linguagens.

Operações e propriedades

- **Quociente**: o quociente de uma linguagem L_1 por uma outra linguagem L_2, denotado por L_1/L_2, como sendo a linguagem:

$$L_1/L_2 = \{x|xy \in L_1, y \in L_2\}$$

- Por exemplo, considere as linguagens $L = \{a, aab, baa\}$ e $A = \{a\}$. Então, $L/A = \{\epsilon, ba\}$.

Operações e propriedades

- Como exercício de **quociente**, considere as linguagens seguintes:
  $L_1 = \{a^i b | i \geq 0\}$
  $L_2 = \{a^i bc^i | i \geq 0\}$
  $L_3 = \{b\}$
  $L_4 = \{a^i b | i \geq 1\}$
  $L_5 = \{bc^i | i \geq 0\}$
  $L_6 = \{c^i b | i \geq 0\}$
  $L_7 = \{a^i | i \geq 0\}$
- Responda os itens abaixo:
  - $L_1/L_3 =?$
  - $L_1/L_4 =?$
  - $L_5/L_7 =?$
  - $L_2/L_6 =?$

Operações e propriedades

- **Substituição**: uma substituição s é uma função que mapeia cada **elemento** de um alfabeto Σ_1 em **linguagens** sobre um alfabeto Σ_2. Formalmente, tem-se que:
$$s : \Sigma_1 \rightarrow 2^{\Sigma_2^*}$$
- Por exemplo, considerando os alfabetos $\Sigma_1 = \{a, b, c\}$ e $\Sigma_2 = \{x, y, z\}$, tem-se a seguinte substituição:

$s(a) = \{x\}$
$s(b) = \{y, yy\}$
$s(c) = \{z, zz, zzz\}$

Operações e propriedades

- Uma **substituição** s pode ser aplicada também sobre uma cadeia w. Neste caso, a operação $s(w)$ é definida indutivamente:

$$s(\epsilon) = \epsilon$$
$$s(a\alpha) = s(a)s(\alpha), a \in \Sigma_1, \alpha \in \Sigma_1^*$$

- Por exemplo, supondo a cadeia $w = abc$, tem-se que:

$$s(abc) = s(a)s(bc) = s(a)s(b)s(b) = \{xyz, xyzz, xyzzz, xyyz, xyyzz, xyyzzz\}$$

Operações e propriedades

- A definição da **substituição** s pode ainda ser estendida para aplicá-la a uma linguagem L da seguinte forma:

$$s(L) = \{y | y = s(x) \ para \ x \in L\},$$

- Por exemplo, para a linguagem $L = \{a^i b^i c^i | i \geq 1\}$, definida sobre o alfabeto $\Sigma_1 = \{a, b, c\}$, tem-se que:

$$s(L) = \{x^i y^j z^k \ | i \geq 1, i \leq j \leq 2i, i \leq k \leq 3i\}$$

Implementação

- Na implementação de uma linguagem de programação, devem-se observar duas questões importantes:
  - Como especificar de forma finita linguagens (eventualmente) infinitas?
  - Como identificar as sentenças de uma linguagem, descartando as demais cadeias?

Implementação

- Há três métodos mais empregados para a representação finita de linguagens:
  - **Gramáticas**: correspondem a especificações finitas de dispositivos de geração de cadeias. Um dispositivo desse tipo deve ser capaz de gerar toda e qualquer cadeia pertencente à linguagem definida pela gramática, e nada mais.
  - **Reconhecedores**: correspondem a especificações finitas de dispositivos de aceitação de cadeias. Um dispositivo desse tipo deverá aceitar toda e qualquer cadeia pertencente à linguagem por ele definido, e rejeitar todas as cadeias nãopertencentes à linguagem.
  - **Enumerações**: relacionam, de forma explícita e exaustiva, todas as cadeias pertencentes à particular linguagem a ser especificada.

Contextualização

- O estudo sistemático das linguagens formais teve um forte impulso no final da década de 1950, quando o linguista Noam Chomsky publicou dois artigos apresentando o resultado de suas pesquisas relativas à classificação hierárquica das linguagens.
- Como teórico e estudioso das linguagens naturais, Chomsky se dedicava à pesquisa de modelos que permitissem a formalização de tais linguagens. Porém, seu trabalho chamou a atenção de especialistas de outras áreas, em particular os da área de computação, que viam, para suas teorias, grande aplicabilidade para a formalização e o estudo sistemático de linguagens artificiais, especialmente as de programação.

Definição

- A classificação das linguagens proposta por Chomsky é conhecida como **Hierarquia de Chomsky**.
- Seu principal mérito é agrupar as linguagens em classesde complexidade relativa, de tal forma que seja possível antecipar as propriedades fundamentais exibidas por uma determinada linguagem e vislumbrar os modelos de implementação mais adequados a sua realização.

Definição

- De acordo com restrições aplicadas ao formato das produções $\alpha \to \beta$ das gramáticas, Chomsky definiu a seguinte hierarquia para as linguagens geradas:

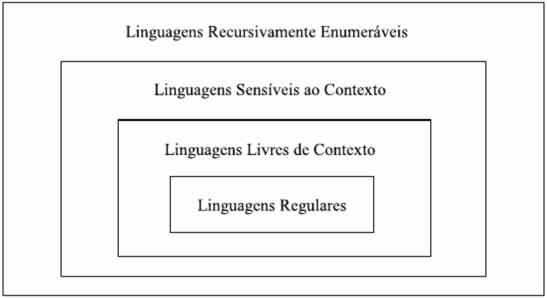

Figura: Hierarquia de Chomsky

Conclusão

- A associação entre linguagens, gramáticas e reconhecedores é destacada na tabela abaixo:

Tipo	Classe de linguagens	Modelo de gramática	Modelo de reconhecedor
0	Recursivamente enumeráveis	Irrestrita	Máquina de Turing
1	Sensíveis ao contexto	Sensível ao contexto	Máquina de Turing com fita limitada
2	Livres de contexto	Livre de contexto	Autômato de pilha
3	Regulares	Linear (direita ou esquerda)	Autômato finito

Figura: Linguagens, gramáticas e reconhecedores

Bibliografia

1. RAMOS, Marcus V. M. Linguagens formais: teoria, modelagem e implementação. 1ª ed. Porto Alegre: Bookman, 2009.
    - **Capítulo 2**.

Dúvidas?