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

Thales L. A. Valente

Disciplina: Linguagens Formais e Autômatos
Código: EECP0020

16 de março de 2024

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
- Gramáticas
- Linguagens, gramáticas e conjuntos
- Reconhecedores
- 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 ($a, b, c, \ldots$).
  - Cadeias: letras minúsculas do final do alfabeto romano ($r, s, x, w, \ldots$) ou letras minúsculas do alfabeto grego ($\alpha, \beta, \gamma, \ldots$).
  - Alfabetos: letras maiúsculas do alfabeto grego ($\Sigma, \Gamma, \Delta, \ldots$).

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.

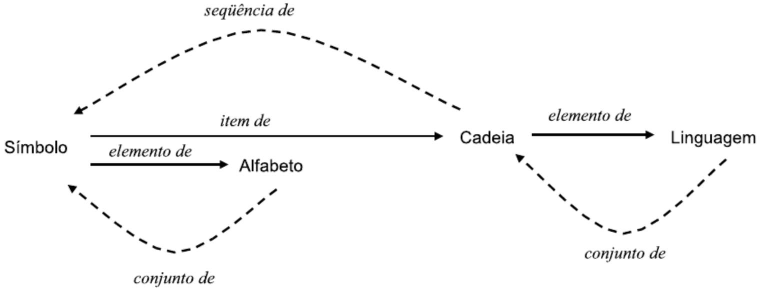

Figura: Símbolo, alfabeto, cadeia e linguagem

Definição

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

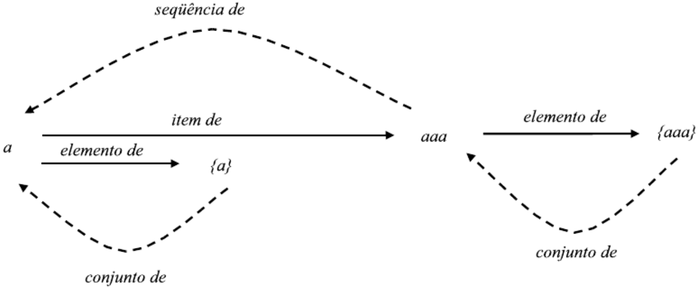

Figura: Exemplo de Símbolo, alfabeto, cadeia e linguagem

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.

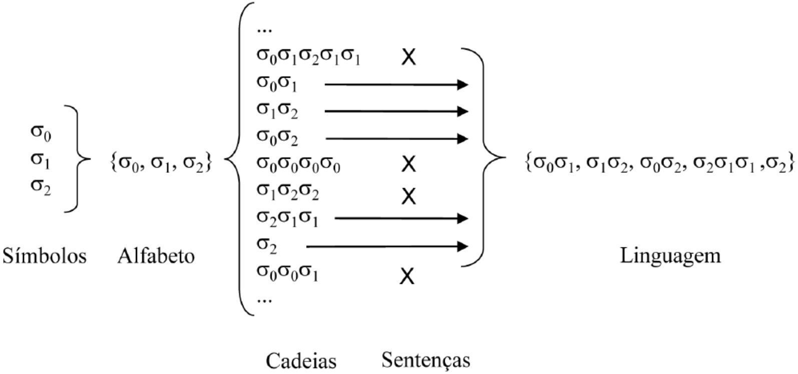

Figura: Símbolo, alfabeto, cadeia e linguagem

Definição

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

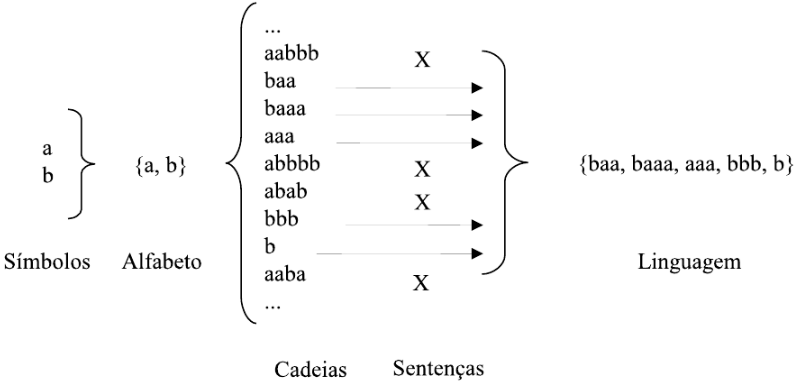

Figura: Exemplo de Símbolo, alfabeto, cadeia e linguagem

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 $\boldsymbol{L \subseteq \Sigma^*}$.

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.

Definição

- **Gramáticas**: Também conhecidas como **dispositivos generativos**, **dispositivos de síntese** ou **dispositivos de geração de cadeias**, as gramáticas constituem sistemas formais baseados em regras de substituição, através dos quais é possível sintetizar, de forma exaustiva, o conjunto das cadeias que compõem uma determinada linguagem.

Definição

- Por exemplo, em português, tem-se, dentre outras, a seguinte regra de formação de senteças: “uma sentença pode consistir de uma frase nominal seguida de um predicado”. Concisamente, tem-se:

$$\langle \textit{sentenca} \rangle \to \langle \textit{frase\_nominal} \rangle + \langle \textit{predicado} \rangle$$

- A definição acima pode ser mais precisa, fazendo-se:

$$\langle \textit{frase\_nominal} \rangle \to \langle \textit{artigo} \rangle \langle \textit{substantivo} \rangle,$$

$$\langle \textit{predicado} \rangle \to \langle \textit{verbo} \rangle$$

- Ao associar palavras às classes sintáticas, podem-se formar sentenças.
    - $\langle \textit{artigo} \rangle$: a, o, um, uma
    - $\langle \textit{substantivo} \rangle$: menino, cachorro
    - $\langle \textit{verbo} \rangle$: anda, corre

Definição

- Formalmente, uma gramática G pode ser definida como sendo uma quádrupla:

$$G = (V, \Sigma, P, S)$$

onde:

  - $\boldsymbol{V}$: é o vocabulário da gramática; corresponde a um conjunto finito e não vazio de símbolos;
  - $\boldsymbol{\Sigma}$: é o conjunto finito e não vazio dos símbolos terminais da gramática. É o alfabeto.
  - $\boldsymbol{P}$: é o conjunto finito e não vazio de produções ou regras de substituição da gramática;
  - $\boldsymbol{S}$: é a raiz da gramática, $S \in V$.
  - $N = V - \Sigma$: é o conjunto de símbolos não terminais da gramática. São as classes sintáticas.

Definição

- O conjunto P de produções gramaticais obedecem à forma geral:

$$\alpha \rightarrow \beta,\ \mathit{com}\ \alpha \in V^*NV^*\ \mathit{e}\ \beta \in V^*$$

- Portanto, α é uma cadeia constituída de quaisquer combinações de símbolos de V, contendo pelo menos um símbolo não-terminal, e β é uma cadeia qualquer, eventualmente vazia, de elementos de V.
- Sendo assim, o conjunto P pode ser expressao como uma relação:

$$P = \{(\alpha, \beta) | (\alpha, \beta) \in V^*NV^* \times V^*\}$$

Definição

- Como exemplo de especificação de gramáticas, tem-se $G_1 = (V_1, \Sigma_1, P_1, S)$, com:
  - $V_1 = \{0, 1, 2, 3, S, A\}$
  - $\Sigma_1 = \{0, 1, 2, 3\}$
  - $N_1 = \{S, A\}$
  - $P_1 = \{S \rightarrow 0S33, S \rightarrow A, A \rightarrow 12, A \rightarrow \epsilon\}$

Definição

- **Forma sentencial**: é qualquer cadeia obtida pela aplicação recorrente das seguintes regras de sustituição:
  1. A raiz S da gramática é por definição uma forma sentencial.
  2. Seja $\alpha\rho\beta$ uma forma sentencial, com α e β cadeias quaisquer de terminais e/ou não-terminais, e seja $\rho \to \gamma$ uma produção da gramática. Então, dessa regra àquela forma sentencial, substituindo a ocorrência de ρ por γ, produz uma nova forma sentencial $\alpha\gamma\beta$.
- Esta forma de substituição é chamada de **derivação direta** e é denotada por:

$$\alpha\rho\beta \underset{G}{\Longrightarrow} \alpha\gamma\beta$$

Definição

- **Derivação**: é a sequência de zero ou mais derivações diretas $\alpha \Longrightarrow \beta ... \Longrightarrow \mu$. É denotada por $\alpha \overset{*}{\Longrightarrow} \mu$.
- **Derivação não-trivial**: é aquela em que ocorre a aplicação de pelo menos uma produção. É denotada por $\alpha \overset{+}{\Longrightarrow} \mu$.
- **Sentença**: Se, pela aplicação de uma derivação não-trivial à raiz S de uma gramática, for possível obter uma cadeia w formada exclusivamente de símbolos terminais, diz-se que w, além de ser uma forma sentencial, é também uma sentença, e denota-se a sua derivação por $S \overset{+}{\Longrightarrow} w$.

Definição

- Por exemplo, considere a gramática G_1, definida anteriormente. Então, tem-se que:
    - S é uma forma sentecial.
    - $0S33$ é uma forma sentencial, pois $S \implies 0S33$.
    - $S \implies 0S33$ é uma derivação direta.
    - $00S3333$ e $00A3333$ são formas sentenciais, pois $0S33 \implies 00S3333 \implies 00A3333$ através das produções $S \rightarrow 0S33$ e $S \rightarrow A$, aplicadas nesta ordem.

Definição

- Por exemplo, considere a gramática G_1, definida anteriormente. Então, tem-se que:
  - $S \overset{+}{\Longrightarrow} 00A3333$ e $S \overset{+}{\Longrightarrow} 0S33$ são exemplos de derivações não-triviais.
  - $00S3333 \overset{*}{\Longrightarrow} 00S3333$ e $0S33 \overset{*}{\Longrightarrow} 00A3333$ são exemplos de derivações;
  - 12 e 00123333 são exemplos de sentenças, pois ambas são formadas exclusivamente por símbolos terminais e $S \Longrightarrow A \Longrightarrow 12$, ou seja, $S \overset{+}{\Longrightarrow} 12$, e $S \Longrightarrow 0S33 \Longrightarrow 00S3333 \Longrightarrow 00A3333 \Longrightarrow 00123333$, ou seja, $S \overset{+}{\Longrightarrow} 00123333$.

Definição

- **Linguagem definida pela gramática G**: é o conjunto de todas as sentenças w geradas por uma gramática G. Formalmente, a linguagem $L(G)$ é definida como:

$$L(G) = \{w \in \Sigma^* | S \stackrel{+}{\Longrightarrow} w\}$$

- Por exemplo, considerando-se a gramática G_1 definida a anteriormente, pode-se concluir que:

$$L_1(G_1) = \{0^m 1^n 2^n 3^{2m} | m \geq 0 \ e \ (n = 0 \ ou \ n = 1)\}$$

- São exemplos de sentenças pertencentes a $L1$: $\epsilon, 12, 033, 01233, 003333, 00123333$, etc.

Definição

- Por exemplo, considere a gramática $G_2 = (V_2, \Sigma_2, P_2, S)$, com:
  - $V_2 = \{a, b, c, S, B, C\}$
  - $\Sigma_2 = \{a, b, c\}$
  - $P_2 = \{S \rightarrow aSBC, S \rightarrow abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$
- A linguagem definida por essa gramática é:

$$L_2(G_2) = \{a^n b^n c^n | n \geq 1\}$$

- Pergunta-se: como a sentença *aabbcc* foi gerada utilizando G_2?

Definição

- Por exemplo, considere a gramática $G_2 = (V_2, \Sigma_2, P_2, S)$, com:
  - $V_2 = \{a, b, c, S, B, C\}$
  - $\Sigma_2 = \{a, b, c\}$
  - $P_2 = \{S \rightarrow aSBC, S \rightarrow abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\}$
- A linguagem definida por essa gramática é:

$$L_2(G_2) = \{a^n b^n c^n | n \geq 1\}$$

- Pergunta-se: como a sentença *aabbcc* foi gerada utilizando G_2?
  $S \Longrightarrow aSBC \Longrightarrow aabCBC \Longrightarrow aabBCC \Longrightarrow aabbCC \Longrightarrow aabbbcC \Longrightarrow aabbcc$
  pela aplicação das produções:
  $S \rightarrow aSBC, S \rightarrow abC, CB \rightarrow BC, bB \rightarrow bb, bC \rightarrow bc$ e $cC \rightarrow cc$

Definição

- **Gramáticas equivalentes**: quando uma mesma linguagem pode ser definida por duas ou mais gramáticas, diz-se que as gramáticas são sintaticamente equivalentes, ou simplesmente, equivalentes.
- Por exemplo, as gramáticas G_3 e G_4 são equivalentes:
  - $G_3 = (\{a, b, S\}, \{a, b\}, \{S \to aS, S \to a, S \to bS, S \to b, S \to aSb\}, S)$
  - $G_4 = (\{a, b, S, X\}, \{a, b\}, \{S \to XS, S \to X, X \to a, X \to b\}, S)$
- Pergunta-se: que linguagem G_3 e G_4 definem?

Definição

- **Gramáticas equivalentes**: quando uma mesma linguagem pode ser definida por duas ou mais gramáticas, diz-se que as gramáticas são sintaticamente equivalentes, ou simplesmente, equivalentes.
- Por exemplo, as gramáticas G_3 e G_4 são equivalentes:
  - $G_3 = (\{a, b, S\}, \{a, b\}, \{S \rightarrow aS, S \rightarrow a, S \rightarrow bS, S \rightarrow b, S \rightarrow aSb\}, S)$
  - $G_4 = (\{a, b, S, X\}, \{a, b\}, \{S \rightarrow XS, S \rightarrow X, X \rightarrow a, X \rightarrow b\}, S)$
- Pergunta-se: que linguagem G_3 e G_4 definem? $L_3(G_3) = L_4(G_4) = \{a, b\}^+$.

Contextualização

- A fim de prover um entendimento pleno de linguagens, é importante desenvolver a percepção de que linguagens são conjuntos. Isso facilitará o estudo de novas operações sobre linguagens, e também de suas propriedades.

Exemplos

- Por exemplo, considere as seguintes gramáticas e as respectivas linguagens por elas definidas:

  $\boldsymbol{G}_0 = (\{a, b, S\}, \{a, b\}, \{S \rightarrow aS, S \rightarrow bS, S \rightarrow \epsilon\}, S)$
  $\boldsymbol{L}_0(G_0) = \Sigma^*$

  $\boldsymbol{G}_1 = (\{a, b, S\}, \{a, b\}, \{S \rightarrow aS, S \rightarrow bS, S \rightarrow a, S \rightarrow b\}, S)$
  $\boldsymbol{L}_1(G_1) = \Sigma^+$

  $\boldsymbol{G}_2 = (\{a, b, S, X\}, \{a, b\}, \{S \rightarrow aX, X \rightarrow aX, X \rightarrow bX, X \rightarrow \epsilon\}, S)$
  $\boldsymbol{L}_2(G_2)$: linguagens de todas as cadeias sobre $\Sigma = \{a, b\}$ que começam com a.

  $\boldsymbol{G}_3 = (\{a, b, S, X\}, \{a, b\}, \{S \rightarrow aX, X \rightarrow aX, X \rightarrow bX, X \rightarrow b\}, S)$
  $\boldsymbol{L}_3(G_3)$: linguagens de todas as cadeias sobre $\Sigma = \{a, b\}$ que começam com a e terminam com b.

  $\boldsymbol{G}_4 = (\{a, b, S, X\}, \{a, b\}, \{S \rightarrow XbXbX, X \rightarrow aX, X \rightarrow \epsilon\}, S)$ $\boldsymbol{L}_4(G_4)$: linguagens de todas as cadeias sobre $\Sigma = \{a, b\}$ que possuem exatamente dois b.

  $\boldsymbol{G}_5 = (\{a, b, S, X\}, \{a, b\}, \{S \rightarrow bX, X \rightarrow aX, X \rightarrow \epsilon\}, S)$
  $\boldsymbol{L}_5(G_5)$: linguagens de todas as cadeias sobre $\Sigma = \{a, b\}$ que possuem apenas um b, no início da cadeia.

Visualização

- Esquematicamente, a relação entre as linguagens L_0, L_1, L_2, L_3, L_4 e L_5 pode ser observada abaixo:

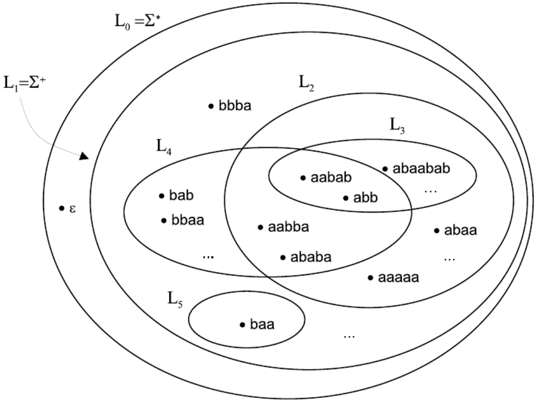

Figura: Relação entre as linguagens L_0, L_1, L_2, L_3, L_4 e L_5

Definição

- **Reconhecedores**: conhecidos também como **dispositivos cognitivos**, **dispositivos de aceitação**, **aceitadores sintáticos** ou simplesmente **autômatos**, os reconhecedores são sistemas formais capazes de aceitar todas as sentenças que pertençam a uma determinada linguagem, rejeitando todas as demais. Por esse motivo, constituem uma forma alternativa às gramáticas para a representação finita de linguagens.

Definição

- Um reconhecedor pode ser visto como uma abstração de um computador digital. A figura abaixo ilustra tal máquina.

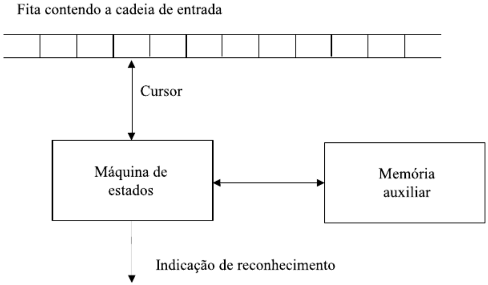

Figura: Organização de um reconhecedor genérico

Definição

- Um reconhecedor apresentam quatro componentes fundamentais:
  - Fita de entrada
  - Cursor
  - Máquina de estados
  - Memória auxiliar

Definição

- Um reconhecedor apresentam quatro componentes fundamentais:
  - **Fita de entrada**: contém a cadeia a ser analisada pelo reconhecedor. Ela é dividida em células e cada célula pode conter um único símbolo da cadeia de entrada, pertencente a um dado alfabeto. A cadeia é disposta da esquerda para a direita, sendo o primeiro símbolo colocado na célula mais à esquerda da fita.
  - Pode apresentar comprimento finito ou infinito.

Definição

- Um reconhecedor apresentam quatro componentes fundamentais:
  - **Cursor**: A leitura dos símbolos gravados na fita de entrada é feita através de um cabeçote de acesso, normalmente denominado cursor, o qual sempre aponta o próximo símbolo da cadeia a ser processado.
  - Os movimentos do cursor são controlados pela máquina de estados, e podem, dependendo do tipo de reconhecedor, ser unidirecionais ou bidirecionais.
  - Determinados tipos de reconhecedores utilizam o cursor não apenas para lerem os símbolos da fita de entrada, mas também para escreverem sobre a fita.

Definição

- Um reconhecedor apresentam quatro componentes fundamentais:
  - **Máquina de estados**: A máquina de estados funciona como um controlador central do reconhecedor, e contém uma coleção finita de estados.
  - Os estados são responsáveis pelas informações colhidas no passado e consideradas relevantes para decisões futuras e transições.
  - As transições promovem as mudanças de um estado a outro, em sincronismo com as operações efetuadas através do cursor sobre a fita de entrada.
  - Pode ler e escrever em uma memória auxiliar.

Definição

- Um reconhecedor apresentam quatro componentes fundamentais:
    - **Memória auxiliar**: A memória auxiliar é opcional, e torna-se necessária apenas em reconhecedores de linguagens que apresentam uma certa complexidade.
    - Normalmente, ela assume a forma de uma estrutura de dados de baixa complexidade, como, por exemplo, uma pilha.
    - As informações registradas na memória auxiliar são codificadas com base em um alfabeto de memória, e todas as operações de manipulação da memória auxiliar (leitura e escrita) fazem referência apenas aos símbolos que compõem esse alfabeto.
    - Os elementos dessa memória são referenciados através de um cursor auxiliar.

Definição

- Um reconhecedor genérico pode se apresentar de diversas formas, a saber:

$$
\text{Reconhecedor} \begin{cases} \text{Máquina de Estados} \begin{cases} \text{Finita} \end{cases} \\[2ex] \text{Fita de entrada} + \text{Cursor} \begin{cases} \text{Limitada / Não limitada} \\ \text{Leitura apenas / Leitura e escrita} \\ \text{Direita apenas / Direita e esquerda} \end{cases} \\[2ex] \text{Memória auxiliar} + \text{Cursor} \begin{cases} \text{Não limitada} \\ \text{Leitura e escrita} \end{cases} \end{cases}
$$

Figura: Diversas formas de apresentação dos reconhecedores

Definição

- A **operação** de um reconhecedor baseia-se em uma seqüência de movimentos que o conduzem de uma configuração inicial única para alguma configuração de parada, indicativa do sucesso ou do fracasso da tentativa de reconhecimento da cadeia de entrada.
- Cada configuração de um autômato é caracterizada pela quádrupla:
  1. Estado
  2. Conteúdo da fita de entrada
  3. Posição do cursor
  4. Conteúdo da memória auxiliar

Definição

- A **configuração inicial** de um autômato é definida como sendo aquela em que as seguintes condições são verificadas:
  1. Estado: inicial, único para cada reconhecedor;
  2. Conteúdo da fita de entrada: com a cadeia completa a ser analisada;
  3. Posição do cursor: apontando para o símbolo mais à esquerda da cadeia;
  4. Conteúdo da memória auxiliar: inicial, pré-definido e único.

Definição

- A **configuração final** de um autômato é aquela na qual as seguintes condições são obedecidas:
  1. Estado: algum dos estados finais, que não são necessariamente únicos no reconhecedor;
  2. Conteúdo da fita de entrada: inalterado ou alterado, em relação à configuração inicial, conforme o tipo de reconhecedor;
  3. Posição do cursor: apontando para a direita do último símbolo da cadeia de entrada ou apontando para qualquer posição da fita, conforme o tipo de reconhecedor;
  4. Conteúdo da memória auxiliar: final e pré-definido, não necessariamente único ou idêntico ao da configuração inicial, ou apenas indefinido.

Definição

- A operação de um reconhecedor observa os seguinte itens:
  - A especificação de uma possibilidade de **movimentação** entre uma configuração e outra é denominada **transição**.
  - Quando há apenas uma possibilidade movimentação de uma configuração a outra, dado o estado atual da máquina, o símbolo atualmente apontado pelo cursor e o símbolo atualmente apontado pelo cursor da memória auxiliar, então diz-se que o autômato é **determinístico**; caso contrário, ele é **não-determinítico**.
  - Se a partir da configuração inicial o reconhecedor alcança uma configuração final, então diz-se que ele **aceita** a cadeia de entrada; caso contrário, ele **rejeita** a cadeia de entrada.

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 $\alpha \rightarrow \beta$ das gramáticas, Chomsky definiu a seguinte hierarquia para as linguagens geradas:

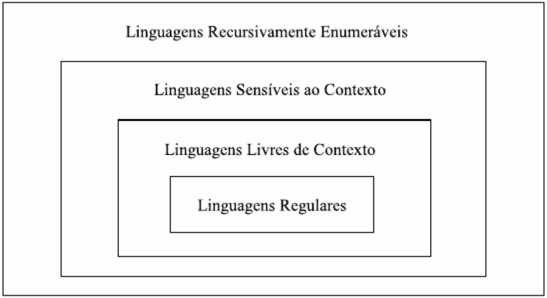

Figura: Hierarquia de Chomsky

Linguagens regulares ou do tipo 3

- As **linguagens regulares** ou **do tipo 3** são aquelas geradas por gramáticas lineares à direita ou por gramáticas lineares à esquerda.
- Uma gramática é dita **linear à direita** caso todas suas regras de produção obedeçam às seguintes condições:
    - $\alpha \in N$
    - $\beta \in \Sigma, \beta \in N, \beta \in \Sigma N$ ou $\beta = \epsilon,$ de forma não exclusiva.
- Por exemplo, a gramática $G_1 = (\{0, 1, 2, 3, S, A\}, \{0, 1, 2, 3\}, \{S \to 0S, S \to 1S, S \to A, A \to 2, A \to 3\}, S)$ é linear à direita.

Linguagens regulares ou do tipo 3

- As **linguagens regulares** ou **do tipo 3** são aquelas geradas por gramáticas lineares à direita ou por gramáticas lineares à esquerda.
- Uma gramática é dita **linear à esquerda** caso todas suas regras de produção obedeçam às seguintes condições:
  - $\alpha \in N$
  - $\beta \in \Sigma, \beta \in N, \beta \in N\Sigma$ ou $\beta = \epsilon,$ de forma não exclusiva.
- Por exemplo, a gramática $G_2 = (\{0, 1, 2, 3, S, A\}, \{0, 1, 2, 3\}, \{S \to S2, S \to S3, S \to A, A \to 1, A \to 0\}, S)$ é linear à esquerda.

Linguagens livres de contexto ou do tipo 2

- As linguagens **livres de contexto** ou **do tipo 2** são aquelas geradas por gramáticas cujas produções possuem apenas um símbolo não-terminal em seu lado esquerdo e uma combinação qualquer de símbolos terminais e não-terminais no lado direito. Gramáticas desse tipo são chamadas livres de contexto ou do tipo 2. Formalmente:
  - $\alpha \in N$
  - $\beta \in V^*$
- Por exemplo, a gramática $G_3 = (\{0, 1, S\}, \{0, 1\}, \{S \rightarrow 0S1, S \rightarrow \epsilon\}, S)$ é livre de contexto.

Linguagens livres de contexto ou do tipo 2

- Deve-se notar que toda gramática do tipo 3 também se enquadra na definição de gramática do tipo 2, constituindo caso particular deste último. Logo, é correto dizer que toda gramática do tipo 3 é também uma gramática do tipo 2. Por outro lado, nem toda gramática do tipo 2 pode ser caracterizada também como gramática do tipo 3.
- Por exemplo, as gramáticas G_1 e G_2 são simultaneamente lineares e livres de contexto. A gramática G_3 é livre de contexto porém não é regular.

Linguagens sensíveis ao contexto ou do tipo 1

- As linguagens **sensíveis ao contexto** ou **do tipo 1** são aquelas geradas por gramáticas cujas produções apresentam o comprimento da cadeia do lado direito igual ou maior do que o comprimento da cadeia do lado esquerdo. Gramáticas desse tipo são chamadas livres sensíveis ao contexto ou do tipo 1. Formalmente:
  - $\alpha \in V^*NV^*$
  - $\beta \in V^*$
  - $|\beta| \geq |\alpha|$
- Por exemplo, a gramática $G_4 = (\{a, b, c, S, X, Y\}, \{a, b, c\}, \{S \rightarrow aXb, S \rightarrow aXa, Xa \rightarrow bc, Xb \rightarrow cb\}, S)$ é sensível ao contexto.

Linguagens sensíveis ao contexto ou do tipo 1

- Deve-se notar que nem toda gramática do tipo 2 pode ser considerada uma gramática do tipo 1. De fato, gramáticas do tipo 2 permitem a geração da cadeia vazia, ao passo que gramáticas do tipo 1 não prevêem essa possibilidade. Tem-se também que nem toda gramática do tipo 1 pode ser considerada uma gramática do tipo 2.
- Por exemplo, as gramáticas lineares G_1 e G_2 são também sensíveis ao contexto. A gramática livre de contexto G_3, no entanto, não é sensível ao contexto, devido à presença da produção $S \to \epsilon$.

Linguagens sensíveis ao contexto ou do tipo 1

- Deve-se notar ainda que a definição de gramáticas sensíveis ao contexto não permite, a priori, que as respectivas linguagens por elas geradas incluam a cadeia vazia, justamente devido ao fato de que $|\beta| \leq |\alpha|$. No entanto, é comum se considerar L uma linguagem sensível ao contexto, ou simplesmente do tipo 1, mesmo que $\epsilon \in L$, se $L - \epsilon$ puder ser gerada por uma gramática sensível ao contexto. Em outras palavras, linguagens sensíveis ao contexto são aquelas que são geradas por gramáticas sensíveis ao contexto, com a eventual incorporação da cadeia vazia.

Linguagens irrestritas ou do tipo 0

- As linguagens **irrestritas**, **recursivamente enumeráveis** ou **do tipo 0** são aquelas geradas por gramáticas cujas produções não apresentam nenhuma restrição quanto a seu formato, exceto pelo fato de que o lado esquerdo das mesmas deva sempre conter pelo menos um símbolo não-terminal. Gramáticas desse tipo são chamadas livres irrestritas ou do tipo 0. Formalmente:
  - $\alpha \in V^* N V^*$
  - $\beta \in V^*$
  - Não se exige a validade de qualquer relação restritiva entre $|\beta|$ e $|\alpha|$
- Por exemplo, a gramática $G_5 = (\{a, b, c, S, X, Y\}, \{a, b, c\}, \{S \rightarrow aXb, S \rightarrow aXa, Xa \rightarrow c, Xb \rightarrow c, X \rightarrow \epsilon\}, S)$ é irrestrita, porém não é sensível ao contexto, devido à presença das produções $Xa \rightarrow c$, $Xb \rightarrow c$ e $S \rightarrow \epsilon$. As gramáticas G_1, G_2, G_3 e G_4 são todas irrestritas.

Resumo

- Toda linguagem do tipo i, $0 \leq i \leq 3$ é gerada por uma gramática do tipo i;
- A classe das linguagens do tipo i, $1 \leq i \leq 3$ está incluída propriamente na classe das linguagens $i - 1$. Consequentemente, toda linguagem do tipo i, $1 \leq i \leq 3$ é também uma linguagem do tipo $i - 1$.

Resumo

- Toda gramática do tipo 3 é também do tipo 2.
- Nem toda gramática do tipo 2 é também do tipo 1. São do tipo 1 apenas aquelas que não possuem produções $\alpha \to \beta$ em que $\beta = \epsilon$.
- Toda gramática do tipo 1 é do tipo 0.

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?