

LINGUAGENS FORMAIS E AUTÔMATOS

LINGUAGENS LIVRES DO CONTEXTO

Autor(a): Me. Pedro Henrique Chagas Freitas

Parecerista: Edilson José Rodrigues

Tempo de leitura do conteúdo estimado em 60 minutos.

Introdução

Olá, caro(a) estudante! Com grande entusiasmo, convidamos você para a leitura deste material. Você sabia que o estudo das linguagens livres de contexto é muito importante? Isso porque compreende a sintaxe de inúmeras linguagens de programação com intenções gerais, dentre elas, Pascal, C e Java.

Neste material, vamos tratar de dois formalismos livres do contexto: gramática livre do contexto e autômato com pilha. Na aplicação, os estudos e as soluções sobre linguagens livres do contexto são realizados com o objetivo de gerar reconhecimento em linguagens e desenvolvimento de propriedades, como árvores de derivação, simplificação de gramática e de formas normais de gramática.

Os assuntos deste material são, extremamente, interessantes e serão apresentados da forma mais simples e didática possível.

Boa leitura!

Definição de Linguagens Livres do Contexto

O estudo da classe das linguagens livres do contexto, ou tipo 2, é de grande importância na teoria da computação, pois, a partir delas, é possível entender um universo extenso de linguagens (equiparando-se com o das regulares), tendo origem, por exemplo, de demandas como: parênteses balanceados, produções de bloco estruturado, dentre outras estruturas, tradicionalmente, vistas em linguagens de programação como Pascal, C, Java, Python etc.

Como exemplos específicos de aplicações nos conceitos e resultados referentes às linguagens livres do contexto, há as aplicações centradas em linguagens artificiais e, exclusivamente, em linguagens de programação. De maneira exclusiva, destacam-se analisadores sintáticos, tradutores de linguagens e processadores de texto de modo geral.

É importante destacar que os algoritmos reconhecedores e geradores que implementam as linguagens livres do contexto são relativamente simples e possuem uma eficiência razoável. De acordo com a hierarquia de Chomsky, a classe das linguagens livres

do contexto contém propriamente a classe das linguagens regulares (MENEZES, 2015, p. 19).

Há, no entanto, uma classe de linguagens parcialmente reduzida, sendo simples estabelecer linguagens que não afetem essa classe. O estudo das linguagens livres do contexto é questionado, utilizando-se os formalismos expostos a seguir.

FORMALISMOS EMPREGADOS NAS LINGUAGENS LIVRES DECONTEXTO



#PraCegoVer : o infográfico tem o título “Formalismos empregados nas linguagens livres de contexto”, escrito em azul-escuro sobre um fundo azul-claro, seguido de uma linha vermelha logo abaixo. Abaixo da linha, há abas para interagir, que, com um clique, abrem e/ou fecham. O conteúdo de cada aba é: 1ª

aba: “Gramática livre do contexto: porta-se de um formalismo inquestionável ou causador, o qual, como o nome mostra, é uma gramática, mas com limitações no aspecto das normas de produção. Tais limitações são determinadas de modo mais livre que na gramática regular”; 2ª aba: “Autômato com pilha: trata-se de um convencionalismo operacional ou identificador, cuja ordenação essencial é análoga à do Autômato Finito não Definido (AFN), incorporada de uma memória secundária tipo pilha (a qual pode ser lida ou gravada). Relativamente às gramáticas livres do contexto, os tópicos seguintes são importantes tanto para o desenvolvimento e a otimização de algoritmos reconhecedores como para a prova de teoremas (MENEZES, 2015)”; 3ª aba: “Árvore de derivação: configura a derivação de uma palavra no aspecto de árvore, partindo do atributo inicial, como a raiz, e finalizando em atributos terminais, como as folhas”; 4ª aba: “Gramática ambígua: há pelo menos uma palavra que desfrute de duas ou mais árvores de derivação nessa gramática”; 5ª aba: “Simplificação de gramática: facilita as produções sem abater o poder de geração da gramática”; 6ª aba: “Forma normal: determina retenções firmes na forma de produção, sem diminuir o poder de geração da gramática”.

A elaboração de um autômato com pilha, a partir de uma gramática livre do contexto, proporciona a produção de um reconhecedor para uma linguagem livre do contexto, por meio de sua gramática básica e instantânea. Qualquer linguagem livre do contexto pode ser identificada por um autômato com pilha, apresentando apenas um estado de controle lógico. Consequentemente, a estrutura de pilha é suficiente com única memória, logo, não é indispensável usar os estados da pilha para “memorizar” estados anteriores.

As linguagens livres do contexto, ou tipo 2, são desenvolvidas a partir das gramáticas livres do contexto (G). Portanto, trata-se de uma gramática:

$$G = (V, T, P, S)$$

Há a limitação de que qualquer norma de produção de P é da forma:

$$A \rightarrow \alpha$$

Nesse caso, A é uma variável de V , e α é uma palavra de $(V \cup T)^*$. Logo, uma gramática livre do contexto é uma gramática na qual o lado esquerdo das criações inclui, exatamente, uma variável.

Quanto à produção ou reprodução da linguagem livre do contexto, a mesma nasce a partir de uma gramática livre de contexto (G). A linguagem constituída pela gramática G será:

$$\text{GERA}(G) = \{w \in T^* \mid S \Rightarrow^+ w\}$$

É importante ressaltar que toda linguagem regular é livre do contexto, conforme demonstra a figura a seguir.



© VIG Educacional

Figura 3.1 - Visão formal do pertencimento das linguagens regulares

Fonte: Elaborada pelo autor.

#PraCegoVer : a imagem está dividida em três blocos, um dentro do outro, do menor para o maior. Seguindo a ordem de cima para baixo, no primeiro bloco, há o título do quadro “Universo de todas as linguagens”. No segundo bloco, há “Linguagens livres do contexto”. No terceiro bloco, há “Linguagens regulares”.

A expressão “livre do contexto” refere-se à caracterização da classe geral de linguagens, cuja construção é da forma $A \rightarrow \alpha$. Por exemplo, a partir de uma derivação, a variável A pode derivar α , sem depender, isto é, sendo livre de qualquer pesquisa dos ícones que possam anteceder ou suceder A , que é o “contexto” da palavra que será derivada. Para facilitar o entendimento, a seguir, há um exemplo no qual são apresentados dois balanceamentos. A partir da linguagem: $L1 = \{anbn \mid n \geq 0\}$, temos a gramática livre do contexto:

$$G1 = (\{S\}, \{a, b\}, P1, S), \text{ em que:}$$

$$P1 = \{S \rightarrow aSb \mid S \rightarrow \epsilon\}$$

Logo, temos $GERA(G1) = L1$. Como exemplo, a palavra “aabb” pode ser gerada pelo segmento de derivação:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aa\epsilon bb = aabb$$

Esse é um caso relevante para o estudo da computação, pois torna possível uma analogia com estruturas de duplo balanceamento em linguagens de programação como Pascal, Java, C, dentre outras. Como exemplo, há as linguagens blocos estruturadas ou linguagens com parênteses equilibrados em expressões na forma (n) elevado a n . A seguir, serão apresentados os parênteses equilibrados em expressões aritméticas.

Tendo em vista uma gramática livre do contexto com uma expressão aritmética, no caso de uma linguagem ($L2$), gerada a partir de uma gramática livre do contexto e constituída de expressões aritméticas, incluindo colchetes balanceados, dois operadores e um operando, teremos:

$$G2 = (\{E\}, \{+, *, [,], x\}, P2, E), \text{ em que:}$$

$$P2 = \{E \rightarrow E+E \mid E * E \mid [E] \mid x\}$$

A título de exemplo, a expressão $[x + x] * x$ pode ser gerada pela sequência de derivação a seguir:

$$E \Rightarrow E * E \Rightarrow [E] * E \Rightarrow [E + E] * E \Rightarrow [x + E] * E \Rightarrow [x + x] * E \Rightarrow [x + x] * x$$

SAIBA MAIS

Em computação, uma maneira frequente de caracterizar uma gramática livre do contexto é o uso da forma de Backus Naur ou BNF (do inglês, Backus Naur form). Em uma BNF:

- as variáveis são palavras determinadas pelos ícones $\langle \text{ e } \rangle$;
- as palavras que não são determinadas são terminais;
- uma regra de produção $A \rightarrow \alpha$ é representada por: $A::=\alpha$.

Para explorar, com maior profundidade, as gramáticas livres do contexto, assista ao vídeo disponível no *link* a seguir.

ASSISTIR

Agora, apresentaremos mais um exemplo, mas aproveitando o conceito de BNF na linguagem de programação Pascal. Considere que se pretende definir uma BNF preparada para gerar qualquer identificador válido na linguagem de programação Pascal, isto é, toda letra é um diferenciador. Sendo S um

identificador, a construção à direita de S , com qualquer letra ou número, também é um identificador.

Em uma BNF, a variável $\langle \text{identificador} \rangle$ é o ícone inicial, logo, temos:

$$\langle \text{identificador} \rangle ::= \langle \text{letra} \rangle \mid \langle \text{identificador} \rangle \langle \text{letra} \rangle \mid \langle \text{identificador} \rangle \langle \text{número} \rangle$$
$$\langle \text{letra} \rangle ::= a \mid b \mid \dots \mid z$$
$$\langle \text{número} \rangle ::= 0 \mid 1 \mid \dots \mid 9$$

Tendo em vista as informações apresentadas neste material, acerca das linguagens livres de contexto, analisaremos as árvores de derivação. Antes, porém, há uma interessante atividade.

Conhecimento

Teste seus Conhecimentos

(Atividade não pontuada)

As linguagens livres de contexto são bastante utilizadas na conceituação de gramáticas, empregadas nas linguagens de programação que, por sua vez, podem derivar vários tipos de aplicações com o uso dos autômatos e dos formalismos presentes nas linguagens.

DIVÉRIO. **A teoria da computação:** máquinas universais e computabilidade.

10. ed. Porto Alegre: Sagah, 2015.

Diante das definições apresentadas e dos estudos sobre autômatos e linguagens livres do contexto, quais formalismos segmentam os estudos das respectivas linguagens?

- ☐ a) Alfabeto livre de contexto e árvore de derivação.
- ☐ b) Gramática livre de contexto e alfabeto com pilha.
- ☐ c) Alfabeto livre de contexto e autômato com fila.
- ☐ d) Gramática livre de contexto e autômato com pilha.
- ☐ e) Árvore de derivação e forma normal de Chomsky.

Árvore de Derivação

Em aplicações como compiladores e processadores de textos, é comum caracterizar a derivação das palavras em estrutura de dados de árvore, partindo-se da raiz até as folhas. Sucintamente, como definição, para uma estipulada gramática livre do contexto, há a simbolização da derivação de palavras na forma de árvore, denominada árvore de derivação, como ilustra a

Figura 3.2.

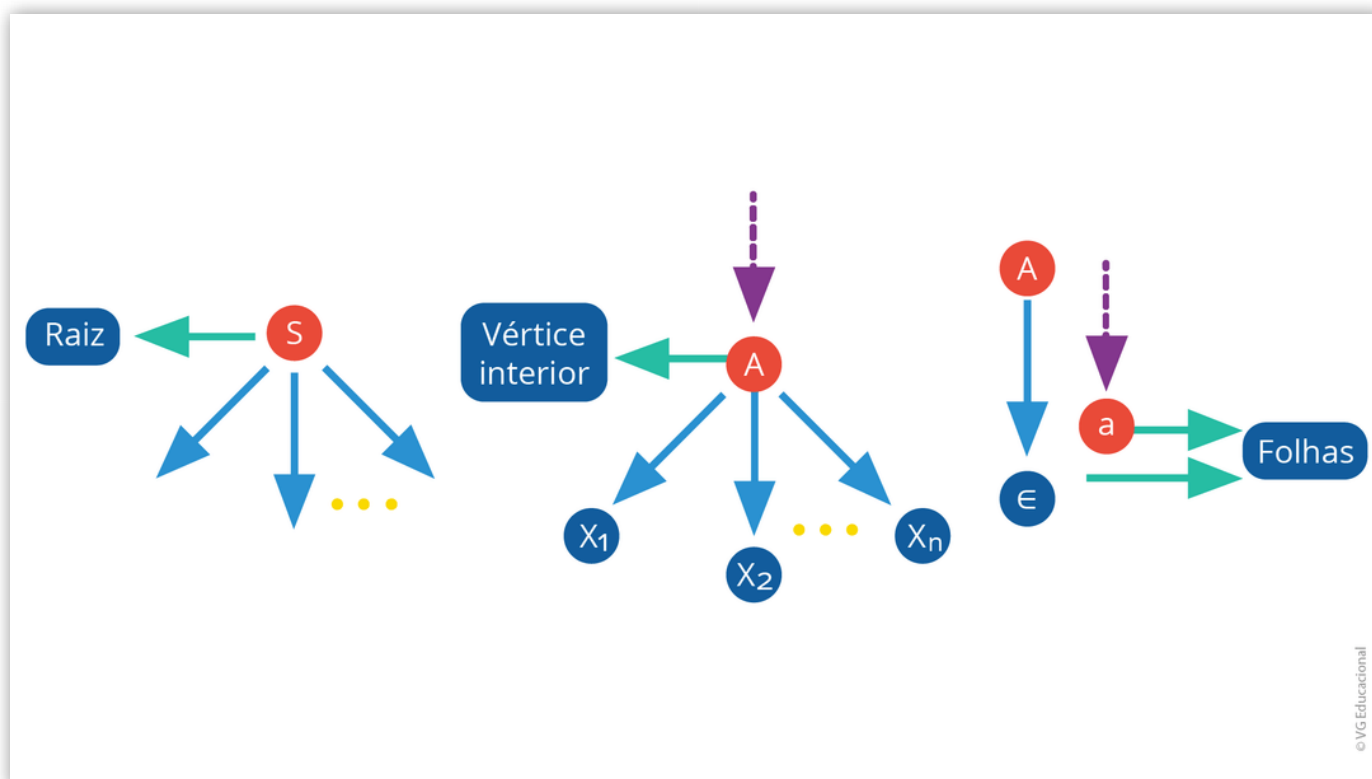


Figura 3.2 - Esquematização de uma árvore de derivação

Fonte: Elaborada pelo autor.

#PraCegoVer : na figura, há a esquematização de uma árvore de derivação, demonstrando as localizações da raiz, do vértice interior e das folhas à direita e à esquerda.

Na figura, há a estrutura exposta a seguir.

- 1) A raiz é o ícone que inicializa a gramática.
- 2) Os vértices interiores, que determinam as variáveis.

Exemplificando, sendo A um vértice interior e X_1, X_2, \dots, X_n os “filhos” de A, então, $A \rightarrow X_1 X_2 \dots X_n$ será a produção da gramática. Já os vértices X_1, X_2, \dots, X_n são organizados da esquerda para a direita.

- 3) Por fim, o vértice folha é um ícone terminal.

Agora, analisaremos um exemplo de árvore de derivação. Para isso, caro(a) aluno(a), observe a palavra aabb e a expressão $[x + x] * x$ dos exemplos a seguir, na devida ordem.

As gramáticas livres do contexto são reproduzidas por meio das árvores de derivação demonstradas na figura a seguir, à esquerda e à direita, na devida ordem.

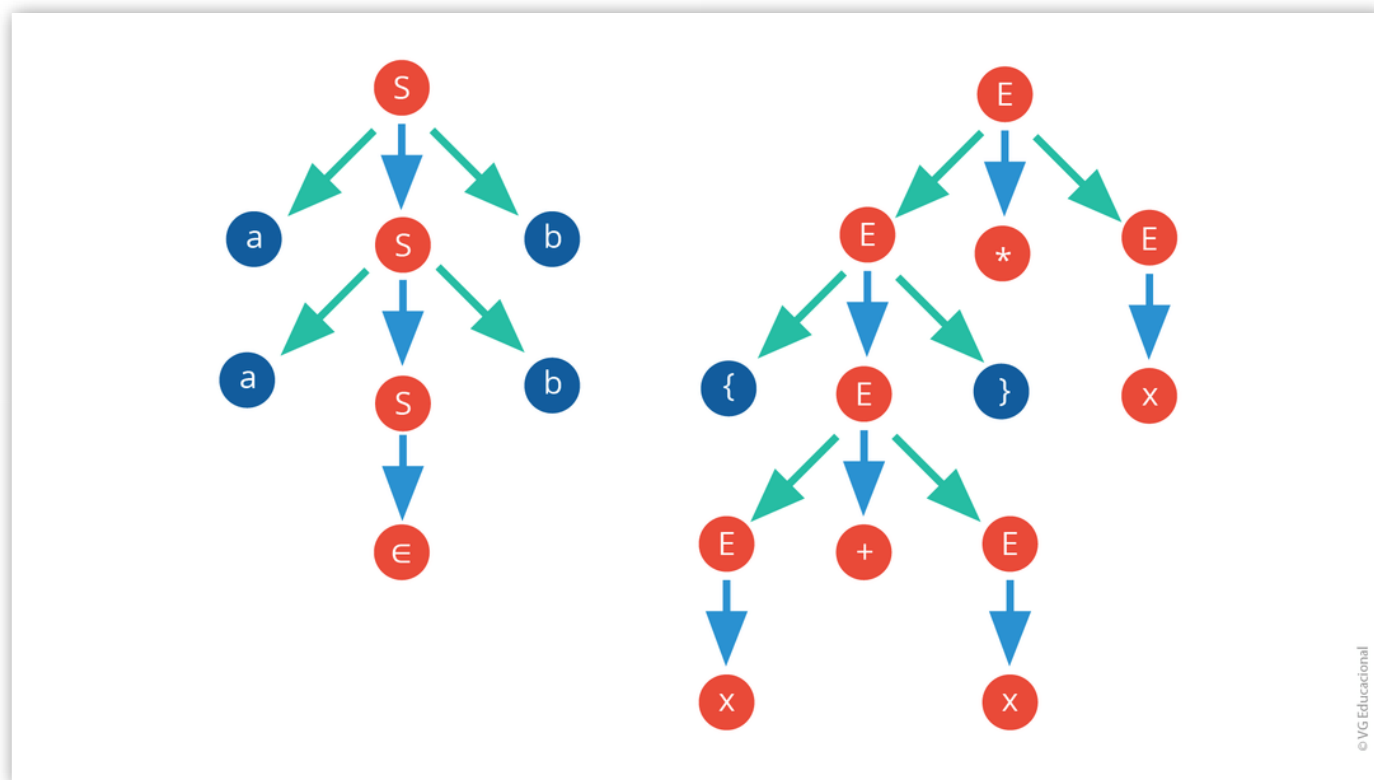
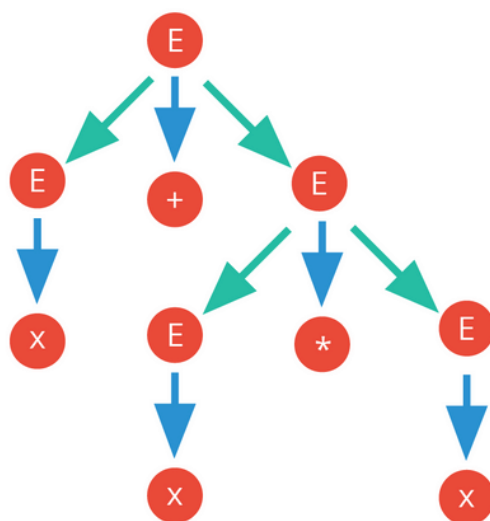


Figura 3.3 - Esquematização de uma árvore de derivação: aabb e $[xx+x]*x$

Fonte: Elaborada pelo autor.

#PraCegoVer : na figura, há a esquematização de uma árvore de derivação, com raiz e vértices folhas. Nesse caso, há uma árvore de derivação aabb.

Uma única árvore de derivação pode caracterizar derivações diferentes de uma mesma palavra, como podemos analisar na árvore ilustrada na figura a seguir.



© VG Educacional

Figura 3.4 - Esquemática de uma árvore de derivação: $x+x*x$

Fonte: Elaborada pelo autor.

#PraCegoVer : na figura, há a esquematização de uma árvore de derivação, com raiz e vértices folhas. Nesse caso, há uma árvore de derivação $x+x*x$.

A palavra “ $x + x * x$ ” pode ser reproduzida por derivações diversas, como segue:

$$(1) E \Rightarrow E+E \Rightarrow x+E \Rightarrow x+E * E \Rightarrow x+x * E \Rightarrow x+x * x$$

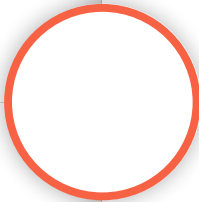
$$(2) E \Rightarrow E+E \Rightarrow E+E * E \Rightarrow E+E * x \Rightarrow E+x * x \Rightarrow x+x * x$$

$$(3) E \Rightarrow E+E \Rightarrow E+E * E \Rightarrow x+E * E \Rightarrow x+x * E \Rightarrow x+x * x \text{ etc.}$$

Caro(a) aluno(a), observe que, no primeiro passo (1), em cada passo de derivação, sempre deriva-se a variável mais à esquerda. Analogamente, para a variável mais à direita, também derivam-se as variáveis, conforme mostrado

no item (2).

REFLITA



A estrutura de dados de árvore é muito comum em diversos tipos de aplicações, sendo basilar na teoria da computação, assim como as estruturas de dados de filas, listas, grafos, dentre outros. A árvore em questão é, comumente, empregada na elaboração de gramáticas livres de contexto, tendo em vista seu caráter estrutural, conveniente para se criar estruturas encadeadas. Nesse sentido, é possível questionar: quais são as outras estruturas de dados que poderiam ser empregadas com o objetivo de estruturar uma árvore de derivação?

Fonte: Adaptado de Menezes (2015).

Em uma árvore de derivação, quando há derivações à esquerda ou à direita de uma palavra, há a sequência de produções empregadas sempre na variável mais à esquerda ou à direita da palavra em que se inicia a derivação. Em outras palavras, se iniciamos à esquerda, as produções se desencadeiam mais à esquerda e vice-versa. Um exemplo está presente na figura a seguir:

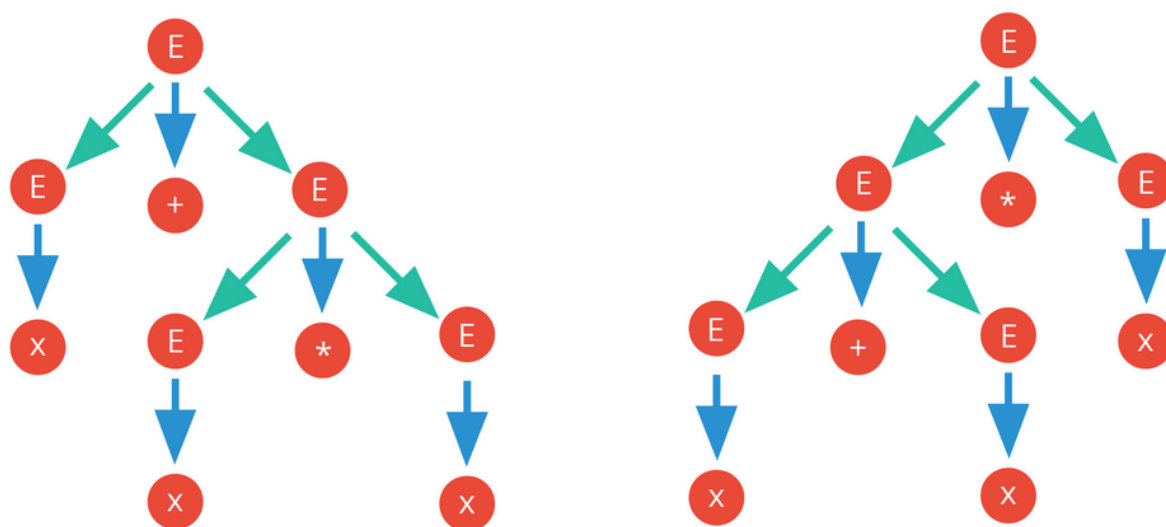


Figura 3.5 - Gramática ambígua: árvores diferentes para a palavra $x+x*x$

Fonte: Menezes (2015, p. 158).

#PraCegoVer : a figura contém uma gramática ambígua, com árvores diferentes para a palavra $x+x*x$; há uma derivação à esquerda e uma derivação à direita.

Após esse exemplo, definiremos as gramáticas livres do contexto indeterminado. Porventura, a mesma palavra pode ser relacionada a duas ou a mais árvores de derivação, tipificando uma **gramática ambígua**.

Em muitas aplicações, como no funcionamento de alguns tipos de algoritmos de reconhecimento, é oportuno que a gramática utilizada não seja ambígua, mas nem sempre é viável excluir ambiguidades. De fato, é simples determinar linguagens para as quais qualquer gramática livre do contexto pode ser ambígua. Por determinação, uma gramática livre do contexto é considerada uma gramática livre do contexto ambígua ou, basicamente, uma gramática ambígua, quando se encontra, pelo menos, uma palavra que contenha duas ou mais árvores de derivação nessa gramática.

Na Figura 3.5, a palavra " $x + x*x$ " tem uma derivação à esquerda e, na devida

ordem, à direita.

- Derivação à esquerda:

$$E \Rightarrow E + E \Rightarrow x + E \Rightarrow x + E * E \Rightarrow x + x * E \Rightarrow x + x * x$$

$$E \Rightarrow E * E \Rightarrow E + E * E \Rightarrow x + E * E \Rightarrow x + x * E \Rightarrow x + x *$$

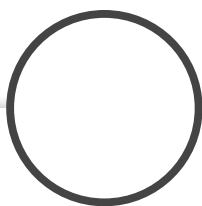
- Derivação à direita:

$$E \Rightarrow E + E \Rightarrow E + E * E \Rightarrow E + E * x \Rightarrow E + x * x \Rightarrow x + x *$$

$$E \Rightarrow E * E \Rightarrow E * x \Rightarrow E + E * x \Rightarrow E + x * x \Rightarrow x + x * x$$

Uma das formas de se definir gramáticas ambíguas é analisar a existência de, pelo menos, uma palavra com duas ou mais derivações à esquerda ou à direita.

É importante destacar que uma linguagem é considerada essencialmente ambígua somente se qualquer gramática livre do contexto que a define for ambígua.



SAIBA MAIS

“É possível facilitar algumas classes de produções sem diminuir o poder de produção das gramáticas livres do contexto. Geralmente, as reduções de gramáticas são utilizadas na produção e na melhoria de algoritmos e na exibição

de teoremas”.

É possível compreender mais a respeito das árvores de derivação no texto de Gláucia Carreiro Boechat, disponível no *link* a seguir:

https://www.cin.ufpe.br/~gcb/tc/tc_gramaticas_livres_contexto.pdf

Fonte: Diverio (2015, p. 159).

Por fim, é válido salientar que as linguagens livres de contexto são desenvolvidas a partir das gramáticas livres de contexto, logo, existe a complementação de uma em relação à outra, e elas estruturam os algoritmos e as próprias linguagens de programação que conhecemos hoje.

Conhecimento

Teste seus Conhecimentos

(Atividade não pontuada)

Aplicações como compiladores e processadores de textos, frequentemente, apresentam representação por meio da derivação de palavras na forma de árvore, seja para a direita, seja para a esquerda, partindo-se do ícone inicial, seja ele qual for, e terminando em ícones verticais à esquerda ou à direita.

DIVERIO, T. **A teoria da computação:** máquinas universais e computabilidade. 10. ed. Porto Alegre: Sagah, 2015.

Dada uma árvore de derivação com a palavra " $x+x \star x$ ", com diversas derivações distintas, em cada passo de derivação, sempre será derivada a variável mais à esquerda, contendo, pelo menos:

- ☐ a) Folhas e vértices.
- ☐ b) Vértices e caules.
- ☐ c) Raiz e folhas.
- ☐ d) Raiz e caules.
- ☐ e) Folhas e galhos.

Forma normal de Chomsky

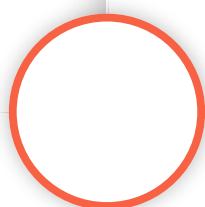
Agora, apresentaremos a concepção das formas normais, em especial, a forma normal de Chomsky. As formas normais determinam moderações firmes na forma de produção, sem diminuir a capacidade de criação das gramáticas livres do contexto (excetuando-se a criação da palavra vazia). Essas formas são utilizadas, especialmente, no desenvolvimento de algoritmos (com destaque para reconhecedores de linguagens) e na prova de teoremas. As formas são as apresentadas a seguir.

Em cada situação, é apresentado um algoritmo de conversão de uma gramática livre do contexto para uma equivalente na forma normal. A partir disso, exclusivamente na forma normal de Chomsky, por definição, é apresentada uma gramática livre do contexto, $G = (V, T, P, S)$, se todas as suas produções apresentarem a forma:

$$A \rightarrow BC \text{ ou } A \rightarrow a$$

Assim, a palavra vazia não é adequada à linguagem produzida por uma gramática na forma normal de Chomsky.

REFLITA



“O algoritmo apresentado altera uma gramática livre do contexto qualquer, cuja linguagem produzida não guarda a palavra vazia, em uma gramática na forma normal de Chomsky. O algoritmo é dividido em três fases”.

Fonte: Menezes (2015, p. 178).

Portanto, por definição, o algoritmo da forma normal de Chomsky é: $G = (V, T, P, S)$, uma gramática livre do contexto, tanto que $\varepsilon \notin \text{GERA}(G)$. O algoritmo para modificar na forma normal de Chomsky está exposto a seguir.

- **Fase 1:** modificação do lado direito das produções de comprimento maior ou igual a dois. A gramática resultante dessa fase é: $G_2 = (V_2, T_1, P_2, S)$, na qual V_2 e P_2 são produzidos de acordo com o algoritmo ilustrado no código a seguir (para cada variável a , considere $Ca \notin V_2$).

$$V_2 = V_1$$

$$P_2 = P_1$$

Para toda $A \rightarrow X_1X_2X_n \in P_2$, tal que $n \geq 2$. Se, para $r \in \{1, \dots, n\}$, X_r é um símbolo terminal, então, considere $X_r = a$. Assim, há: $V_2 = V_2 \cup \{Ca\}$. Substituindo a por Ca , obtemos:

$$A \rightarrow X_1X_2X_n \in P_2; P_2 = P_2 \cup \{Ca \rightarrow a\}.$$

Fase 2 : modificação do lado direito das produções, de comprimento maior ou igual a três em produções, mais precisamente, por duas variáveis. A gramática resultante dessa etapa é: $G_3 = (V_3, T_1, P_3, S)$, na qual V_3 e P_3 são produzidos de acordo com o algoritmo ilustrado na equação a seguir (a cada ciclo, considere $D_1 \notin V_3, \dots, D_{n-2} \notin V_3$).

$$V_3 = V_2$$

$$P_3 = P_2$$

Para toda $A \rightarrow B_1B_2B_n \in P_3$, tal que $n \geq 3$. Assim:

$$P_3 = P_3 - \{A \rightarrow B_1B_2\dots B_n\};$$

$$V_3 = V_3 \cup \{D_1, \dots, D_{n-2}\};$$

$$P_3 = P_2 \cup \{ A \rightarrow B_1 D_1, D_1 \rightarrow B_2 D_2, \dots, \\ D_{n-3} \rightarrow B_{n-2} D_{n-2}, D_{n-2} \rightarrow B_{n-1} B_n \};$$

Como exemplo, considere a seguinte gramática livre do contexto:

$G = (\{E\}, \{+, *, [,], x\}, P, E)$, na qual:

$$P = \{E \rightarrow E+E \mid E * E \mid [E] \mid x\}$$

• **Fase 3:** modificação do lado direito das produções de comprimento maior ou igual a dois. Excetuando-se a produção $E \rightarrow x$, as demais devem ser mudadas, como apresentamos a seguir.

$$E \rightarrow EC + E \mid EC * E \mid C[EC]$$

$$C + \rightarrow +$$

$$C * \rightarrow *$$

$$C [\rightarrow [$$

$$C] \rightarrow]$$

• **Fase 4:** modificação do lado direito das produções de comprimento maior ou igual a três, em produções precisas de duas variáveis. As produções “ $E \rightarrow EC + E \mid EC * E \mid C[EC]$ ” precisam ser mudadas da seguinte forma:

$$E \rightarrow ED_1 \mid ED_2 \mid C[D_3]$$

$$D_1 \rightarrow C + E$$

$$D_2 \rightarrow C * E$$

$$D3 \rightarrow E \ C]$$

A gramática resultante (aspecto normal de Chomsky) é a seguinte:

$$\text{GFNC} = (\{E, C+, C*, C \ [, C], D1, D2, D3\}, \{+, *, [,], x\}, \\ \text{PFNC}, E)$$

Nesse caso:

$$\text{PFNC} = \{E \rightarrow E \ D1 \mid E \ D2 \mid C \ [D3 \mid x,$$

$$D1 \rightarrow C+ \ E, D2 \rightarrow C* \ E, D3 \rightarrow EC],$$

$$C+ \rightarrow +, C* \rightarrow *, C \ [\rightarrow [, C] \rightarrow]\}$$

A forma normal de Chomsky foi fundamental no desenvolvimento da teoria da computação, abarcando, a partir da teoria das formas normais, a implementação matemática que viabilizou a estruturação das linguagens formais.

praticar
Vamos Praticar

Quando o assunto refere-se às formas normais, encontramos a forma normal de Chomsky e a forma normal de Greibach. As formas normais estabelecem restrições rígidas na forma das produções, sem reduzir o poder de geração das gramáticas livres do contexto. Essas formas estão expostas a seguir:

- **Forma normal de Chomsky** – as produções são da forma: $A \rightarrow BC$ ou $A \rightarrow a$.
- **Forma normal de Greibach** – as produções são da forma: $A \rightarrow a\alpha$.

Essas formas são empregadas, em especial, no desenvolvimento de algoritmos (principalmente, nos reconhecedores de linguagens) e na prova de teoremas.

Ao analisar a forma normal de Greibach e as variáveis empregadas na explicação do funcionamento matemático da forma normal de Chomsky, explique, demonstrando ou não matematicamente, o funcionamento da forma normal de Greibach.

Autômato com Pilha

As classes das linguagens livres do contexto podem ser correlacionadas a um formalismo do tipo autômato ou autômato com pilha. O **autômato com pilha** é similar ao autômato finito e, nele, é inserida uma pilha como memória adicional. A pilha é o autômato da fita de entrada e não há limite máximo de tamanho, o que provoca uma ideia de “tão grande quanto se queira”.

Nesse sentido, a ideia é constituir um grupo de autômatos por meio da estrutura de pilha infinitamente computável. Estruturalmente, seu principal atributo é que o último ícone memorizado é o primeiro a ser lido, como exemplificado na Figura 3.6. A base de uma pilha é firme e define o seu início. O topo é variável e define a posição do último ícone memorizado.

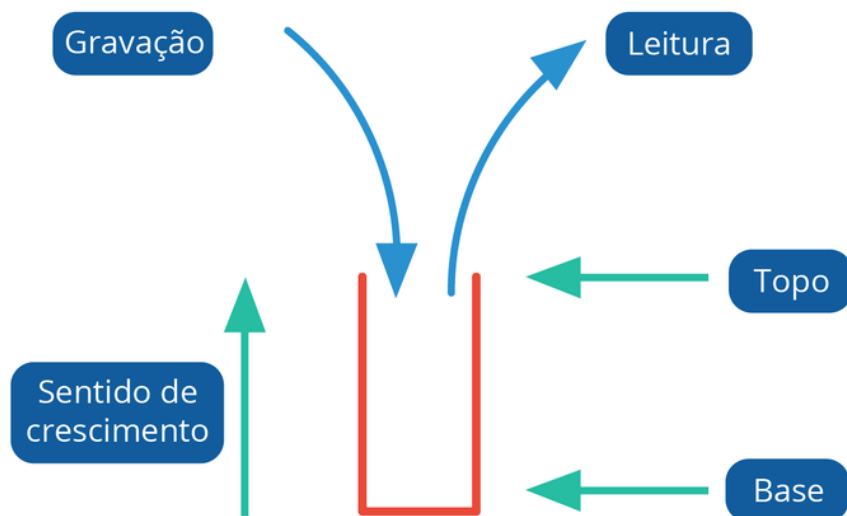


Figura 3.6 - Estrutura do tipo pilha
Fonte: Adaptada de Menezes (2015).

#PraCegoVer : na figura, há a ilustração do funcionamento da estrutura de dados em pilha, com o sentido superior (de crescimento de cima para baixo) e os estados de gravação e leitura.

A clareza dos não escolhidos é **significativa e indispensável**, pois amplia o controle computacional dos autômatos com pilha, possibilitando identificar, formalmente, a classe das linguagens livres do contexto.

Um efeito notável exibido adiante é que qualquer linguagem livre do contexto pode ser diferenciada por um autômato com pilha, somente com um estado (ou três estados, dependendo da definição). Em outras palavras, a estrutura de pilha é satisfatória como única memória, não sendo fundamental utilizar certas condições para “memorizar” conhecimentos passados, isto é, a estrutura de condições no autômato com pilha seria capaz de ser eliminada, sem se limitar ao poder computacional.

Como efeito, considerando que a pilha não tem tamanho máximo, o autômato

com pilha pode alcançar quantos estados desejar. O protótipo do autômato com pilha tem dois conceitos, universalmente aprovados, o que é diferente no processo de parada do autômato, como exposto a seguir.

- 1) O valor inicial da pilha é vazio, e o autômato é permitido ao atingir um estado final.
- 2) A pilha envolve, no início, um ícone especial, denominado ícone inicial da pilha.

Não há situações finais, e o autômato é permitido quando a pilha ficar vazia. Esses dois conceitos são proporcionais (têm o mesmo poder computacional), tornando simples alterar um autômato com pilha para atender a outro conceito. Neste material, empregamos o conceito utilizado nos estados finais.

Ademais, um autômato com pilha não estabelecido ou autômato com pilha é gerado, especialmente, por quatro partes, como apresentado a seguir.

Fita**Pilha****Unidade de controle****Programa**

Função programa ou função de transição; orienta a leitura da fita, a leitura e a gravação da pilha e conceitua o estado da máquina.

A pilha é fragmentada em células, armazenando, cada uma, um ícone do alfabeto auxiliar (pode ser idêntico ao princípio de entrada). O estudo, ou a memorização, são sempre no topo. Não tem tamanho fixo nem máximo, assim, o tamanho é trivial e semelhante ao tamanho da palavra armazenada. Seu número inicial é vazio (envolve a palavra vazia). A unidade de comando tem um algoritmo finito e predeterminado de condições. Há uma cabeça de

fita e uma cabeça de pilha, conforme ensina Medeiros (2015, p. 221):

1) Cabeça da fita. Unidade de identificação que conecta uma célula da fita de cada vez e se mobiliza apenas para a direita. Se a entrada foi completamente lida, é possível testar.

2) Cabeça da pilha. Unidade de identificação e memorização a qual se mexe para a esquerda (ou “para cima”), ao memorizar, e para a direita (ou “para baixo”), ao ler um ícone. Conecta um ícone de cada vez, posicionada sempre no topo. A identificação elimina o ícone reconhecido. É viável testar se a pilha está vazia. Em uma mesma memória, há a possibilidade de armazenar uma palavra composta por mais de um ícone. Diante disso, o ícone do topo é o mais à esquerda da palavra memorizada.

Considere que, apesar de a unidade de controle ter um algarismo finito e predeterminado de condições, tal unidade não é considerada de controle finito (contrário dos autômatos finitos), pois a capacidade da pilha também representa o estado geral do sistema.

O programa é uma atribuição limitada que, dependendo do estado corrente, ícone captado da fita e ícone captado da pilha, designa um novo estado e a palavra a ser gravada (na pilha).

O programa apresenta movimentos vazios simples (proporcional ao do autômato finito), autorizando modificar o estado, sem ler a fita.

praticar

Vamos Praticar

Um autômato com pilhas não precisas, ou apenas autômato com pilha M (constantemente, resumido como AP), é uma 6-upla:

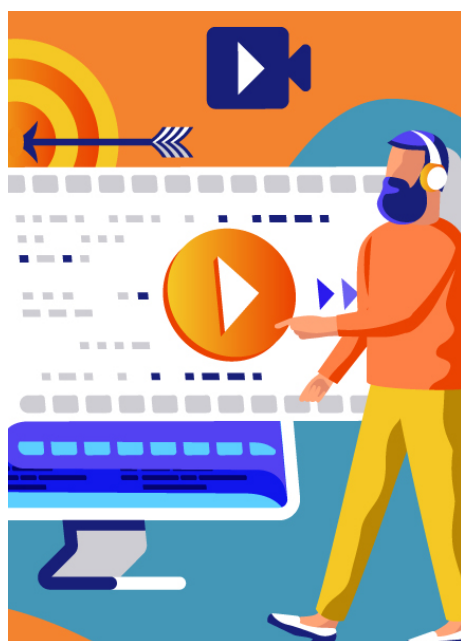
$$M = (\Sigma, Q, \delta, q_0, F, V)$$

Sabe-se que:

- Σ é um alfabeto de ícones de entrada ou, simplesmente, alfabeto de entrada;
- Q é um conjunto de estados possíveis do autômato, o qual é finito;
- δ é uma função programa ou, simplesmente, programa ou, ainda, função de transição.

Nesse contexto, o que são os elementos q_0 , F e V ?

Material Complementar



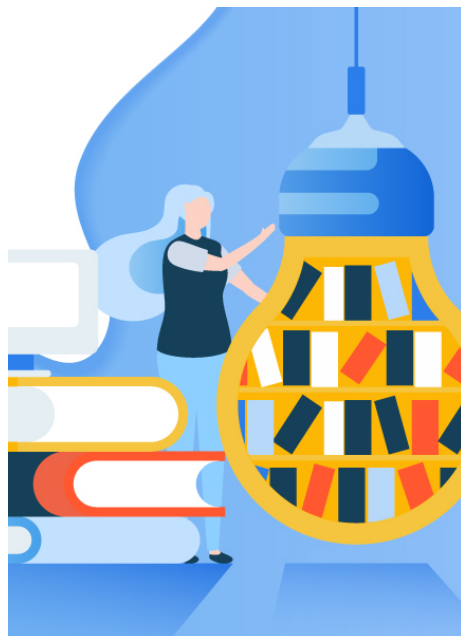
FILME

Upgrade

Ano: 2018

Comentário: O filme retrata, em um futuro distante, o emprego de tecnologias diversas para realizar o controle de quase todos os aspectos da vida. No filme, é possível abstrair o progresso do desenvolvimento das linguagens livres do contexto e o funcionamento dos autômatos na ficção científica, porém é possível aguçar a imaginação quanto ao assunto apresentado neste material. Esse filme ajuda a compreender o desenvolvimento das linguagens livres do contexto e dos autômatos, sob o ponto de vista da ficção científica.

Para conhecer mais sobre o filme, assista ao trailer disponível a seguir.

[TRAILER](#)

LIVRO

Linguagens formais e autômatos

Paulo Menezes

Editora: Grupo A

ISBN: 8577807657

Comentário: O livro é a principal referência sobre autômatos aplicados à teoria da computação. Em especial, trata da definição das linguagens livres do contexto, das árvores de decisão, da forma normal de Chomsky e abarca, de forma precisa, o funcionamento dos autômatos com pilha. O livro apresenta vários exemplos práticos e é fundamental, porque, como exposto, é a principal referência sobre o funcionamento teórico e prático dos autômatos.

Caro(a) estudante, chegamos ao fim deste material e podemos confirmar a importância dos tópicos apresentados: a definição de linguagens livres do contexto, as árvores de decisão e seus funcionamentos, a forma normal de Chomsky e a definição dos autômatos que adotam a estrutura de dados de pilha. Esses tópicos demonstram, claramente, a relevância do assunto para a teoria da computação e o desenvolvimento de novas linguagens de programação e, ainda, para o desenvolvimento de novos paradigmas de programação.

A cada novo ano, novos paradigmas e modelos arquiteturais surgem, fomentando, assim, novas aplicações, a partir dos conceitos apresentados ao longo da unidade. Desse modo, é fundamental aplicar esses estudos, a fim de compreender o futuro das linguagens de programação.

Referências

BOECHAT, G. C. Linguagem livre de contexto. *In: Teoria da Computação*. [2020]. Disponível em:

https://www.cin.ufpe.br/~gcb/tc/tc_gramaticas_livres_contexto.pdf.

Acesso em: 30 maio 2021.

DIVERIO, T. A. **Teoria da computação:** máquinas universais e computabilidade.





10. ed. Porto Alegre: Grupo A, 2015.

MENEZES, P. F. B. **Linguagens formais e autômatos** . Porto Alegre: Grupo A, 2015.

MENEZES, P. F. B. Linguagens formais e autômatos. *In*: GOSUEN, G. M. **Edoc** , 2021.

Disponível em: <https://qdoc.tips/linguagem-formais-e-automatos-pdf-free.html>.

Acesso em: 30 maio 2021.

UPGRADE [S. l.: s. n .], 2018. 1 vídeo (2 min 29 s). Publicado pelo canal Movieclips

Trailers. Disponível em: <https://www.youtube.com/watch?v=36PDeN9NRZ0>.

Acesso em: 30 maio 2021.