Linguagens Formais e Autômatos

Gramáticas

(parte “a”)

Eduardo Furlan Miranda

Baseado em: GARCIA, A. de V.; HAEUSLER, E. H. Linguagens Formais e Autômatos. Londrina: EDA, 2017.

Aspecto sintático da linguagem

- A sintaxe define o conjunto de regras
  - Estrutura e formação de sentenças válidas na linguagem
    - Como os símbolos (letras, números, operadores, etc.)
      - que podem ser combinados para formar expressões, comandos, declarações e outros elementos sintáticos

- Se concentra na forma, e não no significado

sintaxe semântica

Gramática

- O objetivo da gramática é formar palavras (strings, cadeias)
- É um conjunto de regras formais
- Mostra como cada cadeia de uma linguagem é gerada através de regras
- Como as sequências de símbolos do alfabeto podem ser combinadas para formar cadeias válidas
- Existem notações como p. ex. a Backus-Naur Form (BNF) que é bastante usada em gramáticas de linguagens de programação

Exemplo 1

- Especificar a linguagem L_D dos números não negativos, inteiros ou ponto decimal, na base decimal

- Ex. de números nesta linguagem são
  - L_D = { 0, 1, 2, ... 9, 10, 11, ... , 99, 100, ... , 0.1, 0.2, ... }

- O alfabeto sobre o qual L_D está definida é
  - Σ = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, . }

ponto

- Tente escrever um programa que vai lendo o teclado, e se encontrar algum erro no número, ele avisa

(continua)

Exemplo 1

(continuação)

- Um jeito de começar é ir escrevendo frases:
  - Se apertar a tecla “0” então armazene e espere a próxima tecla ser apertada
  - Se apertar a tecla “.”, e não existir ainda o “.” no número, acrescente “.” ao que já foi lido
  - E assim por diante
- Será que podem ocorrer erros de interpretação? Para um problema maior, não ficaria muito complicado?
- É desejável uma maneira de descrever mais próxima da lógica de programação, e que não esteja sujeita a interpretações

Outro jeito de especificar

N → L (um número N pode ser uma lista de dígitos L)

N → L.L (N pode ser uma lista de dígitos L seguida de outra lista L) (especifica o “.”)

L → D (uma lista de dígitos L pode ser um dígito D)

L → LD (uma lista de dígitos L pode ser outra lista L seguida de um dígito D)

D → 0 (uma dígito D pode ser o dígito 0)

D → 1 (uma dígito D pode ser o dígito 1)

...

D → 9 (uma dígito D pode ser o dígito 9)

“→” : “regra”
“⇒” : “derivação”

Derivação

- Podemos usar as regras do slide anterior para gerar, p. ex., o nº 1.23
  - Neste caso dizemos que N gera (ou deriva) 1.23

- O processo de derivação é representado por $\Rightarrow$
  - $N \Rightarrow L.L \Rightarrow D.L \Rightarrow 1.L \Rightarrow 1.LD \Rightarrow 1.DD \Rightarrow 1.2D \Rightarrow 1.23$
    - Neste caso vai-se substituindo as regras até chegar no nº
    - Se chegou no nº que queremos, dizemos que reconheceu
  - A essa especificação damos o nome de gramática
  - Descreve uma derivação em uma gramática formal

"$\rightarrow$" : "regra"
"$\Rightarrow$" : "derivação"

Elementos da gramática

(outro jeito de especificar o slide 6)

- T = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, . }
  - Alfabeto sobre o qual a linguagem é definida
  - Chamamos os elementos deste alfabeto de símbolos terminais
- V = { N, L, D }
  - Conjunto de símbolos auxiliares ou variáveis
  - Chamamos de símbolos não terminais

(continua)

Elementos da gramática

- P = { N→L, N→L.L, L→D, L→LD, D→0, D→1,
  D→2, D→3, D→4, D→5, D→6, D→7, D→8, D→9 }
  - Conjunto de produções (ou regras de substituição)
    - Indicam como os símbolos são substituídos para gerar uma cadeia da linguagem
  - L → ω indica que podemos substituir qualquer ocorrência de L por ω na palavra (cadeia) que está sendo gerada

- Símbolo inicial
  - Neste caso: N
    - As derivações iniciam por ele

Tupla

- Tupla: $(x_1, x_2, \ldots, x_n)$
  - sequência ordenada de elementos
  - a ordem dos elementos é crucial
  - n é o nº de elementos da tupla = aridade
    - Ex.: a aridade de (1, 2, 3) é 3

"ên-upla"

- “n-upla” = tupla com “n” elementos
  - Ex.: (1, "texto", 3.14) = 3-upla, tripla, ou terna

"tri-upla"

  - "n-upla" e "3-upla" são aridades

Igualdade entre Tuplas

(a ordem é importante)

- $(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n)$ se, e somente se, para todo $i \in \{1, 2, \ldots, n\}$, $x_i = y_i$
  - **$(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n)$** : indica que as duas tuplas são iguais
  - **se, e somente se** : indica expressão lógica indicando equivalência
  - **para todo $i \in \{1, 2, \ldots, n\}$**: indica que i é um índice que percorre as posições dos elementos nas tuplas
  - **$x_i = y_i$** : indica que o elemento na posição i da primeira tupla (x_i) é igual ao elemento na mesma posição i da segunda tupla (y_i)

Gramática

sequência ordenada
de elementos

- Uma gramática G é uma tupla (V, T, P, S)
  - V : conjunto finito não vazio de variáveis (símbolos não-terminais)
    - Variáveis representam categorias sintáticas ou construções gramaticais (usa-se letras maiúsculas)
  - T : conjunto finito não vazio de símbolos terminais (minúsculas)
  - P : conjunto finito de regras de produção da forma:
    - $\alpha \rightarrow \beta$, onde:
      - $\alpha , \beta \in (V \cup T)^*$
      - α tem ao menos uma variável
  - S ($\in$ V) é o símbolo inicial (variável de partida)
    - Ponto de partida para a geração de cadeias da linguagem

Descrevem como as variáveis
podem ser substituídas por
outras sequências de símbolos

todas as combinações
possíveis, incluindo ε

G = (V, T, P, S) : definição
de uma gramática formal

$\alpha, \beta \in (V \cup T)^*$

todas as combinações possíveis, incluindo ε

- α e β : cadeias formadas por zero ou mais símbolos
  - Podem ser variáveis (V) ou terminais (T)
  - α tem ao menos uma variável (vide slide anterior)
- V = { S, A, B } variáveis

  variável = símbolo não-terminal
- T = { a, b } símbolos terminais
- (V ∪ T) = { S, A, B, a, b }
- Possíveis valores para α e β
  - Qualquer combinação finita de S, A, B, a, b
    - ε, a, b, S, A, B, ab, ba, aab, bba, SAS, ABA, Bsb, ...

Regra de produção ($\alpha \to \beta$)

- "A sequência α pode ser reescrita como a sequência β"
  - Como α e β pertencem a $(V \cup T)^*$, isso significa que
    - α e β podem ser qualquer sequência de símbolos
      - variáveis, terminais, cadeia vazia (ε)
    - α deve conter pelo menos uma variável (vide slide 12)
  - Ex. de regras de produção válidas
    - $S \to aSb$ ("α" = S, "β" = aSb)
    - $S \to \varepsilon$ ("α" = S, "β" = ε)
    - $A \to a$ ("α" = A, "β" = a)
    - $AB \to ba$ ("α" = AB, "β" = ba)
    - $SA \to aSB$ ("α" = SA, "β" = aSB)

$\to$ regra
$\Rightarrow$ derivação

Exemplo 1 - derivação canônica

refere-se a uma forma específica e padronizada de representar a derivação de uma cadeia a partir de uma gramática formal

- G = (V , T , P , S)
- V = { N , D }
- T = { 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 }
- P = { N → D (1)
  N → DN (2)
  D → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (3)
  }
- S = { N }

(continua)

Exemplo 1 - derivação canônica

$N \rightarrow D$ *(1)*
$N \rightarrow DN$ *(2)*
$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$ *(3)*

- Vamos derivar o nº 243
  - N — inicia-se pelo símbolo inicial
  - DN — aplica-se a regra 2, $N \rightarrow DN$
  - 2N — aplica-se a regra 3, $D \rightarrow 2$
  - 2DN — aplica-se a regra 2 de novo
  - 24N — aplica-se a regra 3, $D \rightarrow 4$
  - 24D — aplica-se a regra 1, $N \rightarrow D$
  - 243 — aplica-se a regra 3, $D \rightarrow 3$ — chegamos no nº desejado

Exemplo 2 - derivação canônica

inicial

formas sentenciais

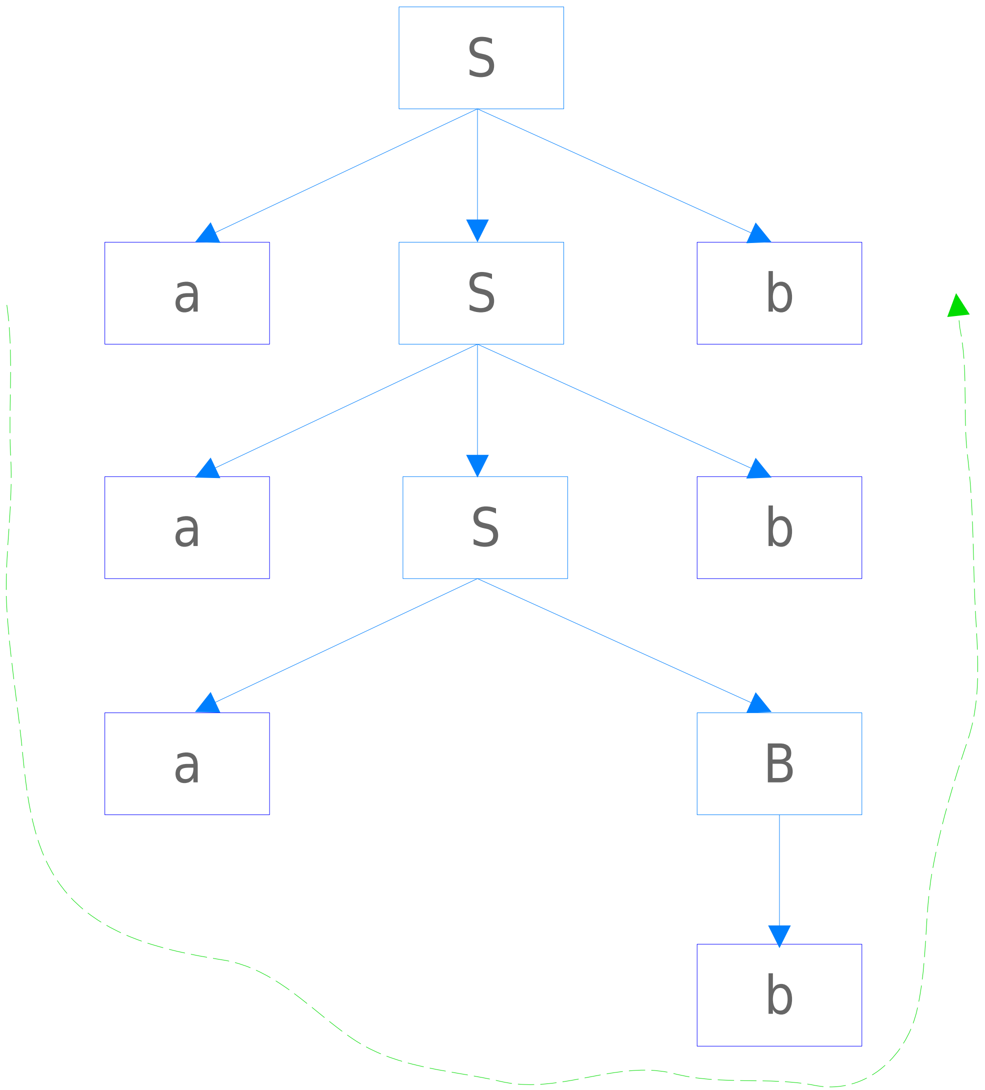
(V, T, P, S)
V : variáveis {S, B}
T : terminais {a, b}
P : produções
S : variável de partida

- $S \rightarrow a\ S\ b$ (1)
  $S \rightarrow a\ B$ (2)
  $B \rightarrow b$ (3)

- $S \Rightarrow a\ S\ b$ inicia por (1)
- $\Rightarrow a\ a\ S\ b\ b$ usa (1) de novo e substitui S por aSb
- $\Rightarrow a\ a\ a\ B\ b\ b$ usa (2) e substitui S por aB
- $\Rightarrow a\ a\ a\ b\ b\ b$ usa (3) e substitui B por b
- formou a palavra (ou cadeia): a a a b b b
  - com essa gramática é possível formar esse tipo de palavra
  - a gramática representa a linguagem $L = a^n b^n$

Árvore de derivação ou *parsing*

derivação
anterior



Derivação mais à esquerda ou à direita

- S → a S S b
  S → a B
  B → b

- esquerda
  S ⇒ a S S b
  S ⇒ a a B S b
  S ⇒ a a b S b
  S ⇒ a a b a B b
  S ⇒ a a b a b b

  direita
  S ⇒ a S S b
  S ⇒ a S a B b
  S ⇒ a S a b b
  S ⇒ a a B a b b
  S ⇒ a a b a b b

- nessa gramática todas as palavras começam com a e terminam com b

Relação de derivação

descreve como uma cadeia de símbolos pode ser transformada em outra, em um único passo, seguindo as regras de produção da gramática G

- $\Rightarrow_G$ ou $\Rightarrow$
- Gramática formal G definida por uma quádrupla (V, T, P, S)
  - Dizemos que $\alpha \Rightarrow_G \beta$ quando:

    relação de derivação direta (ou passo de derivação)
    - $\alpha = \delta_1 \alpha_1 \delta_2$ (α é formado por 3 partes)
      - α_1 : um trecho de símbolos no meio, onde ocorre a substituição
      - δ_1 e δ_2 : trechos de símbolos à esquerda e à direita
        - pode-se imaginar como “lugares para colocar algo”
          - podem ser usados em regras, ex.: “se existir, faça tal coisa”
    - $\beta = \delta_1 \beta_1 \delta_2$
    - e a regra $\alpha_1 \rightarrow \beta_1$ está em P

conjunto de regras de produção

- $\boldsymbol{\alpha, \beta, \delta_1, \delta_2, \alpha_1, \beta_1}$: sequências de símbolos, que podem conter tanto terminais quanto não-terminais (variáveis)

- $\boldsymbol{\alpha_1 \rightarrow \beta_1}$: uma regra de produção em P
  - Significa que a sequência α_1 pode ser substituída pela sequência β_1

lê-se "α deriva para β em um passo usando a gramática G"

- $\boldsymbol{\alpha \Rightarrow_G \beta}$: relação de derivação
  - Significa que a cadeia α deriva diretamente (em um único passo) na cadeia β usando as regras da gramática G

- $\boldsymbol{\alpha = \delta_1 \alpha_1 \delta_2}$: descreve a estrutura da cadeia α
  - $\boldsymbol{\delta_1}$ **e** $\boldsymbol{\delta_2}$: são sequências de símbolos em $(V \cup T)^*$
    - podendo ser vazias (ε)
  - $\boldsymbol{\alpha_1}$: sequência de símbolos em $(V \cup T)^*$ que corresponde ao
    - lado esquerdo de uma regra de produção ($\alpha 1 \rightarrow \beta 1$)
      - é crucial que α_1 contenha pelo menos um não-terminal (slide 12)

- $\boldsymbol{\beta = \delta_1 \beta_1 \delta_2}$: descreve a estrutura da cadeia β
  - mesma sequência descrita acima, com a diferença que β1 corresponde ao lado direito regra de produção aplicada ($\alpha 1 \rightarrow \beta 1$)

- **a regra $\alpha_1 \rightarrow \beta_1$ está em P** : esta é a condição fundamental para que a derivação $\alpha \Rightarrow_G \beta$ seja válida
  - Significa que existe uma regra de produção no conjunto P da gramática que permite substituir α_1 por β_1

- Para que α derive diretamente em β, devemos encontrar uma subcadeia α_1 dentro de α que corresponda ao lado esquerdo de uma regra de produção
  - Então, substituímos essa subcadeia α_1 pelo lado direito β_1 dessa regra, mantendo o restante da cadeia (δ_1 e δ_2) inalterado

Exemplo 3

- Considere a gramática G = (V, T, P, S), onde:
  - V = {S, A}, T = {a, b}, P = {$S \rightarrow aA$, $A \rightarrow b$}, S é o símbolo inicial

- Passo 1: começando com o símbolo inicial
  - Início: "α" = S
    - Aplicamos a regra $S \rightarrow aA$, que está em P
      - $S \Rightarrow_G aA$
  - Depois: $\beta = aA$
  - Substituição:
    - $\alpha_1 = S$ o símbolo que está sendo substituído
    - $\beta_1 = aA$ o resultado da substituição
    - δ_1 e $\delta_2 = \varepsilon$

$\alpha \Rightarrow \beta$
$\alpha = \delta_1\ \alpha_1\ \delta_2$
$\beta = \delta_1\ \beta_1\ \delta_2$

(continua)

Exemplo 3

(continuação)

- Passo 2: Substituindo A por b
  - Agora: α = aA
    - Aplicamos a regra A→b , que está em P
      - aA $\Rightarrow_G$ ab
  - Depois: β = ab
  - Substituição:
    - α_1 = A o símbolo que está sendo substituído
    - β_1 = b o resultado da substituição
    - δ_1 = a o trecho à esquerda de A, que não muda
    - δ_2 = ε vazio, pois não há nada à direita de A

$\alpha \Rightarrow \beta$
$\alpha = \delta_1\ \alpha_1\ \delta_2$
$\beta = \delta_1\ \beta_1\ \delta_2$

- Após os 2 passos, a palavra gerada é ab, que é formada apenas por símbolos terminais (a e b)

(continua)

Exemplo 3

(continuação)

- Resumo da derivação
  - $S \Rightarrow_G aA \Rightarrow_G ab$

- Cada flecha ($\Rightarrow_G$) representa uma derivação direta usando uma regra de produção da gramática P

- Em cada passo, substituímos uma única variável (α_1) por seu equivalente na regra (β_1)

- O restante da palavra (δ_1 e δ_2) permanece inalterado durante a substituição

- O processo termina quando não há mais variáveis na palavra derivada, ou seja, a palavra contém apenas símbolos terminais

Exemplo 4

- Considere a gramática G com 2 regras de produção P :
  - S → aSb (1)
  - S → ε (2)

S : Símbolo inicial
a, b : Símbolos terminais

- Derivação
  - Processo de aplicar as regras de produção para transformar o símbolo inicial S em uma cadeia na linguagem
  - S → aSb (aplicando a regra 1)
  - aSb → aaSbb (aplicando a regra 1 novamente)
  - aaSbb → aabb (aplicando a regra 2 para S)

ε

Aplicamos as regras até que não hajam mais variáveis

(continua)

Exemplo 4

(continuação)

$\alpha \Rightarrow \beta$
$\alpha = \delta_1\ \alpha_1\ \delta_2$
$\beta = \delta_1\ \beta_1\ \delta_2$

- Detalhamento dos passos da derivação $S \Rightarrow_G aSb$
  - $\alpha = S$: a derivação começa com o símbolo inicial **S**
  - $\beta = aSb$: a regra de produção (1) é aplicada a S
  - $\alpha 1 = S$: a parte da cadeia α que será substituída pela regra
  - $\beta 1 = aSb$: a parte que substituirá $\alpha 1$, de acordo com a regra (1)
  - $\delta 1 = \delta 2 = \varepsilon$: cadeia vazia
  - $\alpha 2 = aSb$
  - $\beta 2 = aaSbb$
  - (...)
  - $\alpha 4 = aaSbb$
  - $\beta 4 = aabb$

α, β, δ_1, δ_2, α_1, β_1 : representam sequências de símbolos, que podem conter tanto terminais quanto variáveis

Exemplo 5

- Seja uma gramática que só possui os dígitos 0 e 1, gerando os números de ponto flutuante na base binária
  - $G = (V, T, P, N)$, onde:
    - $T = \{ 0, 1 \}$
    - $V = \{ N, L, D \}$
    - $P = \{ N \rightarrow L \ , \ N \rightarrow L.L \ , \ L \rightarrow D \ , \ L \rightarrow LD \ , \ D \rightarrow 0 \ , \ D \rightarrow 1 \}$
- Para esta gramática temos os seguintes exemplos de $\Rightarrow_G$:
  - $L.L \Rightarrow_G L.LD$
  - $L.L \Rightarrow_G LD.L$
- Quando a gramática G está subentendida no contexto, nós a suprimimos da notação, escrevendo simplesmente $\Rightarrow$:
  - $L.LD \Rightarrow L.L0$