Gramáticas

Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Sistemas e Computação – DSC
Professor: Andrey Brito

Período: 2023.2

Árvore de derivação

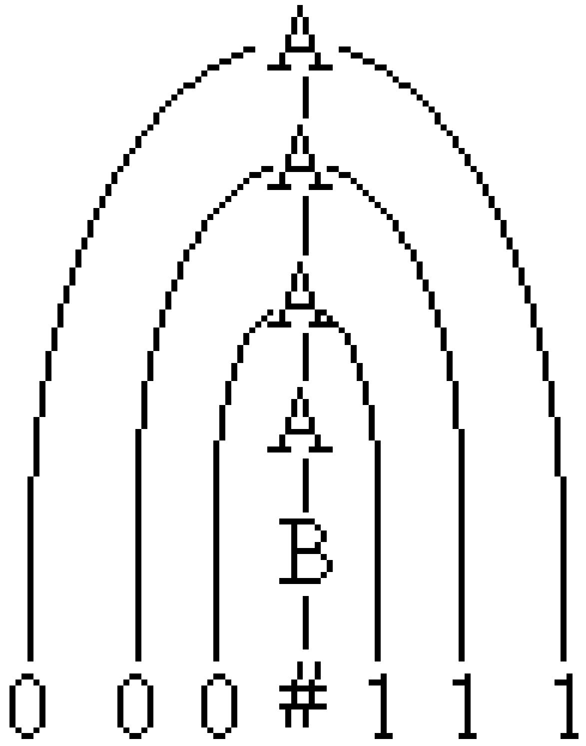
- Árvore de sintática ou árvore de parsing: representação gráfica da aplicação de regras

- Seja a gramática

  $A \rightarrow 0A1$

  $A \rightarrow B$

  $B \rightarrow \#$



Definição formal

- Uma gramática é definida por G = <V, Σ, R, S>
    - V: conjunto das variáveis
    - Σ: conjunto dos terminais (símbolos do alfabeto da linguagem)
    - R: conjunto das regras que definem a gramática
    - S: variável inicial

Que tipo de linguagens são geradas por gramáticas?

Que tipo de linguagens são geradas por gramáticas?

- Considere agora um tipo especial de gramática...

- Uma gramática $G = \langle V, \Sigma, R, S \rangle$ é linear (à direita) se suas produções forem da forma $A \rightarrow xB$, onde
  - $A \in V$
  - $B \in (V \cup \lambda)$
  - $x \in (\Sigma \cup \lambda)$

Exemplo 1:

$A \rightarrow 0B$
$B \rightarrow 0C$
$C \rightarrow 1$

Exemplo 2:

$A \rightarrow 0B$
$B \rightarrow 0C$
$C \rightarrow 1C \mid 0$

O que eu consigo descrever com uma gramática linear?

- Linguagens regulares!
  - Note: aplicar regras faz com que um símbolo (só um) seja inserido na frente da palavra, como só um símbolo pode ser inserido por vez isso não cria um "vínculo" entre duas partes da palavra (ou seja, não permite uma simetria ou contagem)

- Como gerar o autômato
  - Se existe uma regra A → xB, crie dois estados A e B e coloque uma transição com o símbolo *x* entre eles
  - Se existe C → λ, marque C como um estado final (a palavra acabou aqui!)
  - Se existe D → x, crie um estado final novo que é alcançável de D com um símbolo x (a palavra vai acabar aqui)

Exemplos

$S \rightarrow 0A$
$A \rightarrow 0A \mid 1B$
$B \rightarrow 1B \mid \lambda$

$S \rightarrow 0S \mid R$
$R \rightarrow 1R \mid \lambda$

- Que tipos de palavras são geradas?
- Quais seriam os autômatos?

GLD ← AFND?

GLD ← AFND

- Crie uma variável para cada estado
- Adicione uma regra na forma A → xB para cada transição de A para B que consome x
- Adicione uma regra na forma C → λ para cada estado C que seja final
- A variável inicial é a variável criada para corresponder ao estado inicial

Gramáticas Sensíveis ao Contexto

- Mais de um símbolo do lado esquerdo da regra

- Ou seja, certas trocas podem só estar habilitadas em determinados contextos

- Exemplo: $G = \langle\{S,B,C\},\{a,b,c\},R,S\rangle$

  R: $S \rightarrow aBC \mid aSBC \mid \lambda$
  $CB \rightarrow BC$
  $aB \rightarrow ab$
  $bB \rightarrow b$
  $bC \rightarrow bc$
  $cC \rightarrow cc$
  ...

$L = ?$

$S \rightarrow LDABCR$
$LDA \rightarrow LAAD$
$ADA \rightarrow AAD$
$ADB \rightarrow ABBD$
$BDB \rightarrow BBD$
$BDC \rightarrow BCCD$
$CDC \rightarrow CCD$
$DR \rightarrow ER$

$CE \rightarrow EC$
$BE \rightarrow EB$
$AE \rightarrow EA$
$LE \rightarrow LD$
$A \rightarrow 0$
$B \rightarrow 1$
$C \rightarrow 2$
$R \rightarrow \lambda$
$LD \rightarrow \lambda$

$L = 0^n 1^n 2^n$

$S \rightarrow LDABCR$
$LDA \rightarrow LAAD$
$ADA \rightarrow AAD$
$ADB \rightarrow ABBD$
$BDB \rightarrow BBD$
$BDC \rightarrow BCCD$
$CDC \rightarrow CCD$
$DR \rightarrow ER$

$CE \rightarrow EC$
$BE \rightarrow EB$
$AE \rightarrow EA$
$LE \rightarrow LD$
$A \rightarrow 0$
$B \rightarrow 1$
$C \rightarrow 2$
$R \rightarrow \lambda$
$LD \rightarrow \lambda$

Note que você não conseguiria gerar algo diferente!

Mais restrições, mais poder...
Mais poder → mais linguagens

Gramáticas livre-de-contexto

Gramática “livre de contexto” - Definição

- $G = (V, \Sigma, R, S)$
  - V é o conjunto de variáveis
  - Σ é o conjunto de terminais
  - R é um conjunto de regras do tipo $\alpha \to \beta$ onde $\alpha \in V$ e $\beta \in (V \cup \Sigma)^*$
  - $S \in V$ é o símbolo inicial

Regras do tipo $S \to 0S1$ são possíveis

- Toda vez que a regra for usada duas partes da palavra vão crescer de forma proporcional
- A seguinte gramática gera um linguagem que não é regular

$$S \to 0S1 \mid \lambda$$

Para gramaticas lineares à direita: $A \to xB$, onde

$A \in V$

$B \in (V \cup \lambda)$

$x \in (\Sigma \cup \lambda)$

Linguagem “livre de contexto”

- $G=\langle V, \Sigma, R, S\rangle$ uma gramática livre de contexto

- Então:

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

- L é uma linguagem livre de contexto se, e somente se, existir uma gramática G livre-de-contexto tal que L = L(G)

Mais um exemplo de GLC

- Para simplificar, consideramos, como sempre, um alfabeto pequeno: {0,1}

- Queremos gerar **<u>palíndromos</u>** (strings que podem ser lidas da mesma forma de trás para frente)

- Base: 0, 1 e λ são palíndromos

- Indução: se w é um palíndromo 0w0 e 1w1 são palíndromos

Mais um exemplo de GLC

- Base: 0, 1 e λ são palíndromos
  - P → 0 | 1 | λ

- Indução: se w é um palíndromo 0w0 e 1w1 são palíndromos
  - P → 0P0 | 1P1

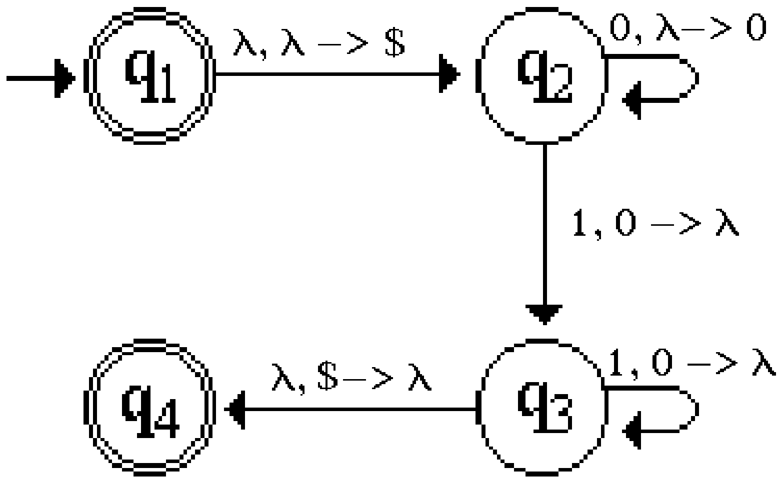
- A gramática será:

P → 0 | 1 | λ

P → 0P0 | 1P1

Nossos exemplos com APs, como seriam as gramáticas?

- $0^n\#1^n$, n >= 0
- $0^n\#1^{n+2}$, n >= 0
- $0^n\#111\#1^n$, n >= 0
- 0^n1^n, n >= 0 (outra forma)
- $0^n1^n2^m3^m$ n >= 0



0^n1^n, n >= 0

Outro exemplo

- Considere a gramática G_5 = (V, Σ, R, <EXPR>)

V = {<EXPR>, <TERMO>, <FATOR>}

Σ = {a, +, ×, (,)}

R: <EXPR> → <EXPR> + <TERMO> | <TERMO>
<TERMO> → <TERMO> × <FATOR> | <FATOR>
<FATOR> → (<EXPR>) | a

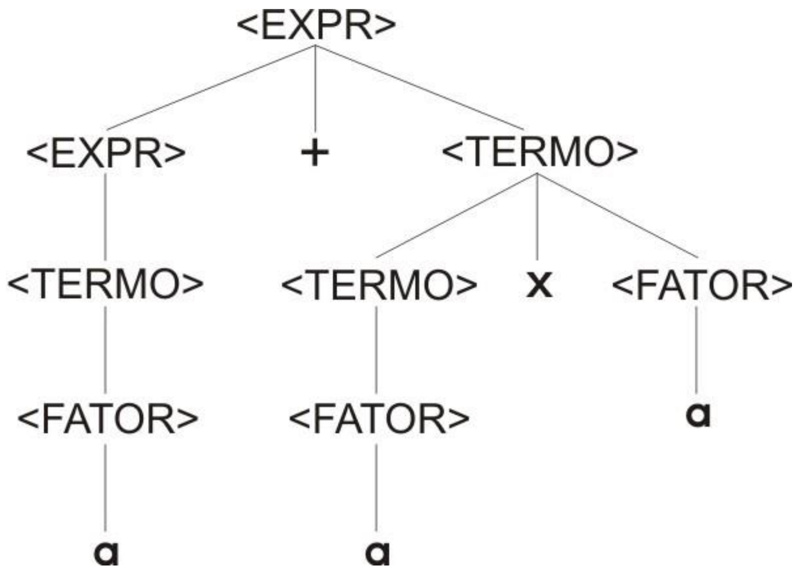
S = <EXPR>

Que tipo de cadeia ela gera?

Exemplo de geração de palavra

- $a + a \times a$

<EXPR> $\rightarrow$ <EXPR> + <TERMO> | <TERMO>
<TERMO> $\rightarrow$ <TERMO> $\times$ <FATOR> |
<FATOR>
<FATOR> $\rightarrow$ (<EXPR>) | a



Exemplo de geração de palavra

- *(a + a) × a*

<EXPR> → <EXPR> + <TERMO> | <TERMO>
<TERMO> → <TERMO> × <FATOR> |
<FATOR>
<FATOR> → (<EXPR>) | a