IV – Gramáticas Livres de Contexto

Introdução

Definições de GLC

1 – G = (Vn, Vt, P, S) onde

$P = \{A \rightarrow \alpha \mid A \in Vn \wedge \alpha \in (Vn \cup Vt)^{+}\}$

2 – GLC ε - LIVRE :

$S \rightarrow \varepsilon$ pode pertencer a P, desde que:

S seja o símbolo inicial de G

S não apareça no lado direito de nenhuma produção de P

3 – G = (Vn, Vt, P, S) onde

$\mathbf{P = \{A \rightarrow \alpha \mid A \in Vn \wedge \alpha \in (Vn \cup Vt)^{*}\}}$

Ou seja α pode ser ε !!!

IV.1 - Árvore de Derivação

- Representação estruturada das derivações de G

Exemplo: $S \rightarrow a\ S\ c \mid B$

$B \rightarrow b\ B \mid \varepsilon$

IV.2 - Limite de uma AD

- Concatenação das folhas da AD ≡ Forma sentencial

IV.3 - Formas de Derivação

- Derivação + a Esquerda
- Derivação + a Direita

Exemplo: $S \rightarrow A\ B \mid S\ c$

$A \rightarrow a\ A \mid \boldsymbol{\varepsilon}$

$B \rightarrow b\ B \mid \boldsymbol{\varepsilon}$

IV.4 - Gramáticas Ambíguas

- G é ambígua se

$$\exists\ \mathbf{x \in L(G) \mid x} \textbf{ possua mais de uma AD}$$

- Exemplos

1- $S \rightarrow S\ b\ S \mid a$

2- $E \rightarrow E + E \mid E * E \mid (E) \mid id$

3- A gramática do “if”

Linguagens inerentemente ambíguas

- Linguagens que só possuem representações ambíguas

$$\mathbf{L(G) = \{ a^n\ b^m\ c^k \mid n = m \vee m = k \}}$$

IV.5 – Transformações (Simplificações) em G.L.C.

IV.5.1 – Eliminação de Símbolos Inúteis

- **Inalcançáveis e/ou inférteis**

$$\mathbf{S} \overset{*}{\Rightarrow} \mathbf{w\,X\,y} \overset{*}{\Rightarrow} \mathbf{w\,x\,y}$$

- **Algoritmo IV.1**

Objetivo – Encontrar o conjunto de Não Terminais Férteis.
Entrada – Uma G.L.C. G = (Vn, Vt, P, S).
Saída – NF – Conjunto de Não-Terminais Férteis.
Método:

Construa conjuntos N0, N1, ..., como segue:
i ← 0
Ni ← ϕ
repita
  i ← i +1
  Ni ← Ni-1 $\cup$ {A | A → $\alpha \in$ P $\wedge$ $\alpha \in$ (Ni-1 $\cup$ Vt)*}
até Ni = Ni-1
NF ← Ni

Fim

Exemplo:

$$\begin{aligned}
\mathbf{S} &\rightarrow \mathbf{a\,S \mid B\,C \mid B\,D} \\
\mathbf{A} &\rightarrow \mathbf{c\,C \mid A\,B} \\
\mathbf{B} &\rightarrow \mathbf{b\,B} \mid \varepsilon \\
\mathbf{C} &\rightarrow \mathbf{a\,A \mid B\,C} \\
\mathbf{D} &\rightarrow \mathbf{d\,D\,d \mid c}
\end{aligned}$$

• Algoritmo IV.2:

Objetivo: Eliminar símbolos **Inalcançáveis**.
Entrada: Uma G.L.C. G = (Vn, Vt, P, S).
Saída: Uma G.L.C. G' = (Vn', Vt', P', S') na qual todos os símbolos $\in$ (Vn' $\cup$ Vt') sejam alcançáveis.
Método:

Construa conjuntos V0, V1, ..., como segue:
i $\leftarrow$ 0 ; Vi $\leftarrow$ {S}
repita
  i $\leftarrow$ i +1
  Vi $\leftarrow$ Vi-1 $\cup$ {X | A $\rightarrow$ α X β $\in$ P, A $\in$ Vi-1 $\wedge$ α e β $\in$ (Vn $\cup$ Vt)*}
até Vi = Vi-1
Construa G' = (Vn', Vt', P', S'), como segue:

a) Vn' $\leftarrow$ Vi $\cap$ Vn
b) Vt' $\leftarrow$ Vi $\cap$ Vt
c) P' $\leftarrow$ conjunto de produções de P, que envolvam apenas símbolos de Vi
d) S' $\leftarrow$ S

Fim

Exemplo:

S $\rightarrow$ a S a | d D d
A $\rightarrow$ a B | C c | a
B $\rightarrow$ d D | b B | b
C $\rightarrow$ A a | d D | c
D $\rightarrow$ b b B | d

• Algoritmo IV.3

Objetivo: Eliminar **símbolos inúteis**.
Entrada: Uma G.L.C. G = (Vn, Vt, P, S).
Saída: Uma G.L.C. G’ = (Vn’, Vt’, P’, S’) | L(G’) = L(G) e nenhum símbolo de G’ seja inútil.
Método:

1 – Aplique o algoritmo IV.1 para obter NF;
2 – Construa G1 = (Vn $\cap$ NF, Vt, P1, S), onde P1 contém apenas produções envolvendo NF $\cup$ Vt;
3 – Aplique o ALG IV.2 em G1, para obter G’ = (Vn’, Vt’, P’, S’);

Fim

Exemplo:

$S \rightarrow a\ F\ G \mid b\ F\ d \mid S\ a$
$A \rightarrow a\ A \mid \varepsilon$
$B \rightarrow c\ G \mid a\ C\ G$
$C \rightarrow c\ B\ a \mid c\ a \mid \varepsilon$
$D \rightarrow d\ C\ c \mid \varepsilon$
$F \rightarrow b\ F\ d \mid a\ C \mid A\ b \mid G\ A$
$G \rightarrow B\ c \mid B\ C\ a$

IV.5.2 – Transformações de GLC em GLC ε-Livre

• Eliminação de ε-produções

Algoritmo IV.4:

Objetivo: Transformar uma G.L.C. G em uma G.L.C. ε - LIVRE G' equivalente.
Entrada: Uma G.L.C. G = (Vn, Vt, P, S).
Saída: **Uma G.L.C. ε - Livre G' = (Vn', Vt, P', S') | L(G') = L(G).**
Método:

1 – Construa Ne = $\{A \mid A \in Vn \wedge A \overset{*}{\Rightarrow} \varepsilon\}$.

2 – Construa P' como segue:

a) **Inclua em P' todas as produções de P, com exceção daquelas da forma A → ε.**

b) Para cada produção de P da forma:

$$A \rightarrow \alpha B \beta \mid B \in Ne \wedge \alpha, \beta \in V^*$$

inclua em P' a produção $A \rightarrow \alpha\beta$

c) Se S ∈ Ne, adicione a P' as seguintes produções:

$$S' \rightarrow S \mid \varepsilon$$

Exemplos:

1) S → AB
A → aA | ε
B → bB | ε

2) **S → c S c | B A**
A → a A | A B C | ε
B → b B | C A | ε
C → c C c | A S

IV.5.3 – Eliminação de Produções Simples

Definição: **$A \rightarrow B$, onde A e $B \in Vn$.**

Algoritmo IV.5:

Entrada: Uma G.L.C. ε - **Livre** $G = (Vn, Vt, P, S)$.
Saída: Uma G.L.C. ε - **Livre** $G' = (Vn', Vt', P', S')$ sem produções simples | $L(G') = L(G)$.
Método:

1 – Para todo $A \in Vn$, construa $NA = \{B \mid A \overset{*}{\Rightarrow} B\}$

2 – Construa P' como segue:

se $B \rightarrow \alpha \in P$ e não é uma produção simples, então adicione a P' as produções da forma:

$$A \rightarrow \alpha, \text{ para todo } A \mid B \in NA$$

3 – Faça $G' = (Vn, Vt, P', S)$.

Fim.

Exemplos:

1) $S \rightarrow FGH$
$F \rightarrow G \mid a$
$G \rightarrow dG \mid H \mid b$
$H \rightarrow c$

2) $S \rightarrow a\,B\,c\,D\,e$
$B \rightarrow b\,B \mid E \mid F$
$D \rightarrow d\,D \mid F \mid d$
$E \rightarrow e\,E \mid e$
$F \rightarrow f\,F \mid f$

IV.5.4 – Fatoração de GLC

- Uma GLC G é dita FATORADA, se ela NÃO possui A $\in$ Vn | A derive seqüências que iniciam com o mesmo símbolo por mais de um caminho

- **Processo de Fatoração**

  - **Não-Determinismo Direto**

    Substituir produções da forma:

    $$A \rightarrow \alpha\,\beta \mid \alpha\,\gamma$$

    Pelo seguinte conjunto de produções:

    $$A \rightarrow \alpha\, A'$$

    $$A' \rightarrow \beta \mid \gamma$$

  - **Não-Determinismo Indireto**

    Transformar em Direto via derivações sucessivas

- Exemplos:

  1) S → a S | a B | d S
  B → b B | b

  2) S → AB | BC
  A → a A | ε
  B → b B | d
  C → c C | c

  3) S → a S | A
  A → a A c | ε

IV.5.5 – Eliminação de Recursão à Esquerda

- **Não Terminal Recursivo**

  $A \overset{+}{\Rightarrow} \alpha A \beta$, para $\alpha \wedge \beta \in V^*$.

  se $\alpha \overset{*}{\Rightarrow} \varepsilon$ então A é Recursivo à Esquerda

- **G é Rec. à Esquerda se possui NT Rec. à Esquerda**

Processo de eliminação

- **R.E. Direta**

Substituir produções da forma:

$$A \rightarrow A\alpha 1 \mid A\alpha 2 \mid \ldots \mid A\alpha n \mid \beta 1 \mid \beta 2 \mid \ldots \mid \beta m$$

Por produções da forma:

$$A \rightarrow \beta 1A' \mid \beta 2A' \mid \ldots \mid \beta mA'$$
$$A' \rightarrow \alpha 1A' \mid \alpha 2A' \mid \ldots \mid \alpha nA' \mid \varepsilon$$

Exemplos:

1) S → S a | b

2) E → E + T | T
   T → T * F | F
   F → (E) | id

• Eliminação de R.E. Indireta

Algoritmo IV.6:

Entrada: **Uma G.L.C. Própria G = (Vn, Vt, P, S).**

Saída: **Uma GLC G' = (Vn', Vt, P', S) | L(G') = L(G) $\wedge$ G' não possui Recursão a Esquerda.**

Método:

1 – Ordene os não-terminais de G em uma ordem qualquer (digamos: A1, A2, ..., An);

2 – Para i = 1, n faça

Para j = 1, i – 1 faça

Substitua as produções da forma

Ai → Aj γ

por produções da forma

Ai → δ1 γ | δ2 γ | ... | δk γ

onde δ1, δ2, ..., δk são os lados direitos das Aj – produções (Aj → δ1 | δ2 | ... | δk)

fim para

Elimine as rec. esq. Diretas das Ai – produções

fim para

3 – Fim**.**

Exemplos:

1) S → Aa | S b
   A → S c | d

2) **S → a S | A b**
   A → A b | B c | a
   B → B d | S a | e

IV.6 – Tipos Especiais de GLC

Gramática Própria:

- **Não possui** Ciclos**;**
- **É ε -** Livre**;**
- **Não possui** Símbolos Inúteis**.**

Gramática Sem Ciclos:

Não possui derivação da forma $A \overset{+}{\Rightarrow} A$

Gramática Reduzida:

- **L(G) não é vazia;**
- **Se $A \rightarrow \alpha \in P$, $A \neq \alpha$;**
- **G não possui Símbolos Inúteis.**

Gramática de Operadores:

Não possui produções cujo lado direito contenha NT consecutivos.

Gramática Unicamente Inversível:

não possui produções com lados direitos iguais.

Gramática Linear:

$A \rightarrow x\ B\ w \mid x$, onde $A, B \in Vn \wedge x, w \in Vt^*$.

Forma Normal de Chomsky (F.N.C.):

Uma G.L.C. está na F.N.G. se ela é ε - LIVRE e todas as suas produções são da forma:

- **$A \rightarrow BC$, com A, B e $C \in Vn$ ou**
- **$A \rightarrow a$, com $A \in Vn \wedge a \in Vt$.**

Forma Normal de Greibach (F.N.G.):

Uma G.L.C. está na F.N.G. se ela é ε - LIVRE e todas as suas produções são da forma:

$$A \rightarrow a\alpha \mid a \in Vt,\ \alpha \in Vn^{*} \wedge A \in Vn.$$

Notações de GLC

- BNF – Backus-Naur Form

Exemplos: 1) <S> :: = a <S> | ε
2) <E> :: = <E> + id | id

- BNF Estendida (notação de Wirth)

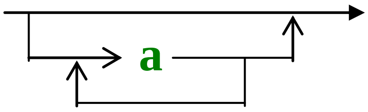
Exemplos: 1) <S> :: = {a}
2) <E> :: = id {+ id}

- RRP (ER estendidas com NT)

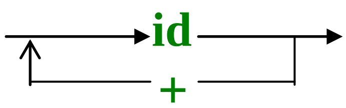
Exemplos: 1) <S> :: = a*
2) <E> :: = id $^{£}$+

- Diagramas Sintáticos (Conway)

Exemplos: 1) <S> :



2) <E> :



Principais Aplicações de GLC

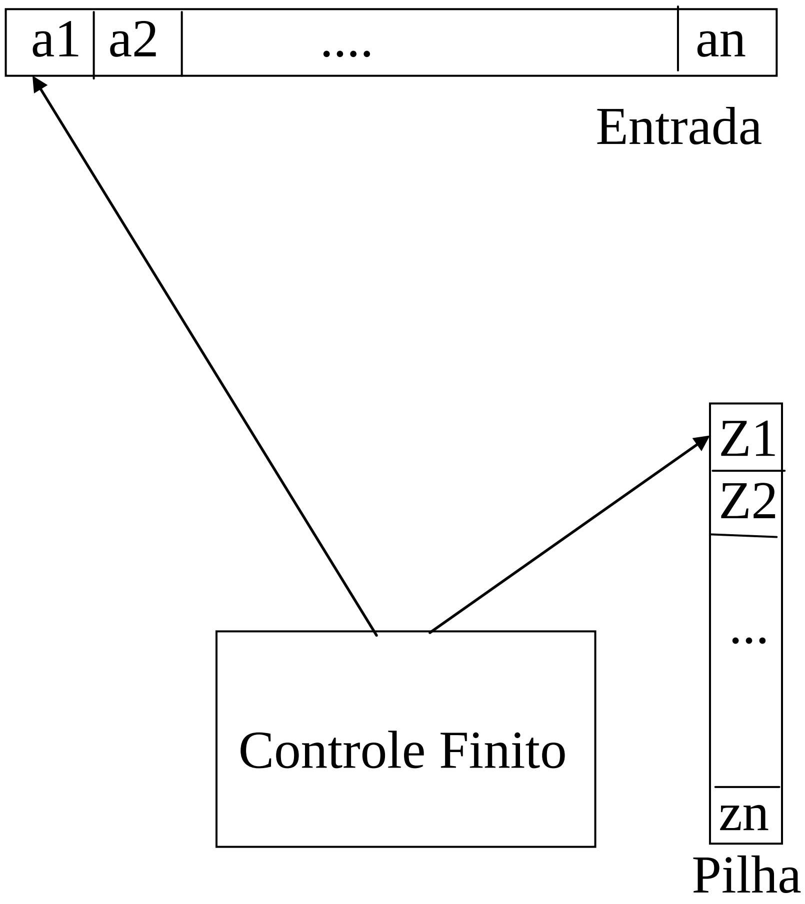
1 – Especificação de linguagens de programação;
2 – Formalização de parsing / implementação de parser's;
3 – Esquemas de tradução dirigidos pela sintaxe
4 – Processamento de string's, de modo geral.

IV.7 – Autômatos de Pilha (PDA)

(Push Down Automata)

- **Um PDA é um dispositivo não-determinístico reconhecedor de Ling. Livres de Contexto (LLC).**

- **Estrutura Geral:**



- **Definição Formal**

  P = (K, ∑, Γ, δ, qo, Zo, F), onde

  - **K é um conjunto finito de** Estados
  - **∑ é o <u>alfabeto</u> finito de** Entrada
  - **Γ é o <u>alfabeto</u> finito de** Pilha
  - **δ é uma** Função De Mapeamento**, definida por:**

$$(K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma) \rightarrow \{K \times \Gamma^*\}$$

  - **q0** $\in$ **K, é o** Estado Inicial **de P**
  - **Z0** $\in$ **Γ , é o** Símbolo Inicial da Pilha
  - **F** $\subseteq$ **K, é o conjunto de** Estados Finais**.**

- **Exemplo: P = (K, ∑, Γ, δ, qo, Zo, F), onde**

  - **K** = **{q0, q1, q2}**
  - **∑** = **{ 0, 1 }**
  - **Γ** = **{ Z, B }**
  - **δ** = **{ δ(q0, 0, Z) = (q1, BZ), δ(q1, 0,B) = (q1, BB),**
    δ(q1, 1, B) = (q2, ε), δ(q2, 1, B) = (q2, ε),
    δ(q2, ε ,Z) = (q0, ε)}
  - **q0** = **q0**
  - **Z0** = **Z**
  - **F** = **{q0]**

- **Representação gráfica:**

  Os rótulos das arestas devem ser da forma:

  (a, Z) → γ, onde a $\in$ **∑, Z** $\in$ **Γ, γ** $\in$ **Γ***

- **Movimentos de um PDA**

  1 – Movimento dependente da entrada (a-move)

  $\delta(q, a, Z) = \{(p1, \gamma 1), (p2, \gamma 2), ..., (pm, \gamma m)\}$

  onde: $a \in \Sigma$, $Z \in \Gamma$
  $q, p1, p2, ..., pm \in K$
  $\gamma i \in \Gamma^*$, para $i \leq i \leq m$

  2 – Movimento independente da entrada (ε -move)

  $\delta(q, \varepsilon, Z) = \{(p1, \gamma 1), (p2, \gamma 2), ..., (pm, \gamma m)\}$

  onde: ε é a sentença vazia, $Z \in \Gamma$
  $q, p1, p2, ..., pm \in k$
  $\gamma i \in \Gamma^*$, para $i \leq i \leq m$

- **Descrição Instantânea (ou Configuração) de um PDA**

  A configuração de um PDA é dada por

  (q, w, γ)

  onde: $q \in K$, é o estado atual (corrente);
  $w \in \Sigma^*$, é a parte da entrada não analisada;
  $\gamma \in \Gamma^*$, é o conteúdo (corrente) da pilha.

- **Notação dos movimentos de um PDA**

  Se $(q, aw, Z\gamma)$ **é uma configuração e**
  se P contém a transição $\delta (q, a, Z) \rightarrow (q', a)$

  então $(q, aw, Z\gamma) \vdash (q', w, \alpha\gamma)$

- **Linguagem Aceita por um PDA**

  1 – Linguagem Aceita por Estado Final (T(P))

  $T(P)=\{w \mid (q0, w, Z0) \vdash^{*} (p, \varepsilon, \gamma),\ p \in F \wedge \gamma \in \Gamma^{*}\}$

  2 – Linguagem Aceita Por Pilha Vazia (N(P))

  $N(P) = \{w \mid (q0, w, Z0) \vdash^{*} (p, \varepsilon, \varepsilon), \text{ para } \forall\ p \in K\}$

- **Autômato de Pilha Determinístico**

  Um PDA $P=(K, \Sigma, \Gamma, \delta, q0, Z0, F)$ é determinístico se:

  1 – se $\delta(q, \varepsilon, Z) \neq \phi$
  então $\delta(q, a, Z) = \phi$ para todo $a \in \Sigma$.

  2 – Para $q \in K$, $Z \in \Gamma$ e $a \in \Sigma$, existe no máximo uma transição envolvendo $\delta(q, a, Z)$

- **Equivalência Entre PDA e G.L.C.**

  Teorema: **"Se L é uma L.L.C., então existe um PDA P | P aceita L".**

  1 – É sempre possível construir a partir de uma G.L.C. G um PDA P que aceite L(G) por pilha vazia

  2 – É sempre possível construir uma G.L.C. G a partir de um PDA P de forma que ambos aceitam a mesma linguagem.

  A prova deste teorema pode ser encontrada na bibliografia indicada.

PDA's não-determinísticos x PDA's determinísticos

- **DPDA's não são equivalentes a NDPDA's**
- **Exemplo:**
  Não existe DPDA's para representar

$$T(P)=\{ww^r \mid w \in (0,1)^+\}$$

- **T(P) pode ser representada pelo seguinte NDPDA:**
  $P = (K, \Sigma, \Gamma, \delta, q_o, Z_o, F)$, onde

$K = \{q_0, q_1, q_2\}$, $\Sigma = \{ 0, 1 \}$, $\Gamma = \{ Z, A, B \}$,
$q_0 = q_0$, $Z_0 = Z$, $F = \{q_2\}$
$\delta = \{ \delta(q_0,0,Z) = (q_0,AZ)$, $\delta(q_0,1,Z) = (q_0,BZ)$,
$\delta(q_0,0,A) = (q_0,AA)$, $\delta(q_0,0,A) = (q_1, \varepsilon)$,
$\delta(q_0,0,B) = (q_0, AB)$, $\delta(q_0,0,B) = (q_1, \varepsilon)$,
$\delta(q_0,1,A) = (q_0, BA)$, $\delta(q_0,1,B) = (q_0, BB)$,
$\delta(q_1,0,A) = (q_1,\varepsilon)$, $\delta(q_1,1,B) = (q_1,\varepsilon)$, $\delta(q1,\varepsilon,Z) = (q_2, Z)\}$

- **Exercício:**

○ **O PDA abaixo é Determinístico?**

$P = (K, \Sigma, \Gamma, \delta, qo, Zo, F)$, onde
$K = \{q_0, q_1, q_2\}$, $\Sigma = \{ a,b \}$, $\Gamma = \{ Z, B \}$, $q0 = q0$, $Z0 = Z$, $F = \{q_2\}$
$\delta = \{ \delta(q_0,a,Z) = (q_0,BZ)$, $\delta(q_0,a,B) = (q_0,BB)$, $\delta(q_0,b,B) = (q_1, B)$,
$\delta(q_0,b,Z) = (q_1,Z)$, $\delta(q_1,b,B) = (q_1, \varepsilon)$, $\delta(q_1,b,B) = (q_1, B)$,
$\delta(q_1,b,Z) = (q_1, Z)$, $\delta(q_1, \varepsilon,Z) = (q_2, \varepsilon) \}$

○ **Existe DPDA $P' \equiv P$?**

Equivalência Entre PDA e G.L.C.

Teorema: "Se L é uma L.L.C. gerada por uma GLC G, então existe um PDA P | P aceita L".

Prova: Para provar este teorema, basta mostrar que:

1 – É sempre possível construir um PDA P (pilha vazia) a partir de uma GLC G de forma que N(P) = L(G).

- **Algoritmo (idéia geral): Seja G uma GLC na FNG**

  Se $A \rightarrow a\,\alpha \in P$ **entao** $\delta\,(q, a, A) = (q, \alpha) \in \delta$

- **Exemplo:**

  G: $S \rightarrow a\,B \mid b\,A$ **P: ?**

  $A \rightarrow a \mid a\,S \mid a\,B\,B$

  $B \rightarrow b \mid b\,S \mid b\,A\,A$

2 – É sempre possível construir uma GLC G a partir de um PDA P de forma que L(G) = N(P)

- **Algoritmo (idéia geral): Seja P um PDA que aceita por pilha vazia (com 1 estado)**

  Se $\delta\,(q, a, A) = (q, \alpha) \in \delta$ **entao** $A \rightarrow a\,\alpha \in P$

  Se $\delta\,(q, \varepsilon, A) = (q, \alpha) \in \delta$ **entao** $A \rightarrow \alpha \in P$

- **Exemplo:** $P = (K, \Sigma, \Gamma, \delta, qo, Zo, F)$, **onde**

  $K = \{q_0\}$, $\Sigma = \{a,b\}$, $\Gamma = \{Z, B\}$, $q_0 = q_0$, $Z_0 = Z$, $F = \phi$

  $\delta = \{\ \delta(q_0, \varepsilon, Z) = (q_0, \varepsilon),\ \ \delta(q_0, a, Z) = (q_0, ZB),$

  $\delta(q_0, b, B) = (q_0, \varepsilon)\}$

  G: $Z \rightarrow a\,Z\,B \mid \varepsilon$

  $B \rightarrow b$