GLC: propriedades e problemas

IBM1088 Linguagens Formais e Teoria da Computação

Evandro Eduardo Seron Ruiz
evandro@usp.br

Universidade de São Paulo

Frase do dia

The secret of genius is to carry the spirit of the child into old age, which mean never losing your enthusiasm.

Aldous Huxley[a]

[a] Escritor inglês (1894–1963).

FNC e GLC

Teorema

Qualquer linguagem livre do contexto (LLC) é gerada por uma GLC na Forma Normal de Chomsky.

Forma Normal de Chomsky

FNC

Uma GLC $G = (V, \Sigma, P, S)$ é dita estar na Forma Normal de Chomsky se todas as regras de produção estão escritas num dos seguintes modos:

$$A \to BC \quad \text{ou} \quad A \to a$$

para $A, B, C \in V$, $a \in \Sigma$, sendo que B e C não podem ser a variável inicial. Permite-se $S \to \epsilon$ para S como a variável inicial.

Transformando uma GLC qualquer em FNC, (1/2)

Etapa 1 Simplificação
Exclua as produções vazias, tipo $A \rightarrow \epsilon$. Exclua também as produções unitárias tipo $A \rightarrow B$, ou seja, de alteração de variável; como também os símbolos inúteis, ou não utilizados.

Etapa 2 Transformação do lado direito das produções de comprimento maior ou igual a dois
Objetivo: Garantir que o lado direito das produções de comprimento maior ou igual a dois seja composto exclusivamente por variáveis. A exclusão de um terminal pode acontecer desde que seja substituído por uma variável intermediária C_a e desde que incluída a produção $C_a \rightarrow a$.

Transformando uma GLC qualquer em FNC, (2/2)

Etapa 3 Transformação do lado direito das produções de comprimento maior ou igual a três em produções com exatamente duas variáveis

O objetivo desta etapa é garantir que o lado direito das produções de comprimento maior ou igual a um seja composto exatamente por duas variáveis.

Exemplo

Considere a GLC G abaixo e simplifique suas produções para descrevê-la na Forma Normal de Chomsky. Seja

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

para as produções P, como seguem:

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

Aplicando as regras em G para G_{FNC}, veja:

Aplicando as regras

Etapa 1 A gramática já esta de acordo.

Etapa 2 As 3 primeiras produções precisam ser modificadas usando-se variáveis intermediárias desta maneira:

Regras originais

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

Etapa 2

$$E \rightarrow E C_{+} E$$
$$E \rightarrow E C_{*} E$$
$$E \rightarrow C_{[} E C_{]}$$
$$C_{+} \rightarrow +$$
$$C_{*} \rightarrow *$$
$$C_{[} \rightarrow [$$
$$C_{]} \rightarrow]$$

continuação...

Etapa 3 As três primeiras produções precisam ser substituídas para:

Etapa 3

$$
\begin{aligned}
E &\rightarrow E D_1 \\
E &\rightarrow E D_2 \\
E &\rightarrow C_{[} D3 \\
D_1 &\rightarrow C_{+} E \\
D_2 &\rightarrow C_{*} E \\
D_3 &\rightarrow E C_{]}
\end{aligned}
$$

Por último...

Finalizando, a gramática livre do contexto G fica assim:

$$G_{FNC} = (\{E, C_+, C_*, C_[, C_], D_1, D_2, D_3\}, \{+, *, [,], x\}, P', E)$$

para as novas regras de produção P', como seguem:

Novas regras de produção, P'

$$E \rightarrow ED_1$$
$$E \rightarrow ED_2$$
$$E \rightarrow C_{[}D3$$
$$E \rightarrow x$$
$$D_1 \rightarrow C_{+}E$$
$$D_2 \rightarrow C_{*}E$$
$$D_3 \rightarrow EC_{]}$$

$$C_{+} \rightarrow +$$
$$C_{*} \rightarrow *$$
$$C_{[} \rightarrow [$$
$$C_{]} \rightarrow]$$

Conteúdo

1. Recordando
2. Conversão para FNC: mais um exemplo
3. Propriedades
4. Problemas decidíveis: 3
5. Problemas indecidíveis
6. Exercícios

1 Recordando

2 Conversão para FNC: mais um exemplo

3 Propriedades

4 Problemas decidíveis: 3

5 Problemas indecidíveis

6 Exercícios

Recordando as regras de conversão para FNC

Uma variação pequena do algoritmo de conversão é a proposta por Sipser. Veja:

Etapa 1 Adicionamos uma variável inicial S_0 e a regra $S_0 \to S$ para garantir que a variável inicial não ocorra do lado direito.

Etapa 2 Exclua as produções vazias, tipo $A \to \epsilon$. Exclua também as produções unitárias tipo $A \to B$.

Etapa 3 Transformação do lado direito das produções de comprimento maior ou igual a três em produções com exatamente duas variáveis
Objetivo: garantir que o lado direito das produções de comprimento maior ou igual a um seja composto exatamente por duas variáveis.

Primeiro passo: aplicando S_0

A GLC original esta à esquerda. O resultado à direita.

$$S \to ASA|aB$$
$$A \to B|S$$
$$B \to b|\epsilon$$

$$S_0 \to S$$
$$S \to ASA|aB$$
$$A \to B|S$$
$$B \to b|\epsilon$$

Remoção das regras ϵ, ou seja, produções vazias

A regra mostrada em cinza (ϵ) indica a remoção recente.
Analisemos as conseqüências da remoção de $B \to \epsilon$.
e.g. $A \to B$ e $B \to \epsilon$, então $A \to \epsilon$.

$$
\begin{aligned}
S_0 &\to S \\
S &\to ASA|aB \\
A &\to B|S \\
B &\to b|\epsilon
\end{aligned}
$$

$$
\begin{aligned}
S_0 &\to S \\
S &\to ASA|aB|a \\
\bullet\ A &\to B|S|\epsilon \\
B &\to b
\end{aligned}
$$

Ainda com as regras ϵ, produções vazias

Sendo que $A \to \epsilon$, temos como resultado à direita

$$
\begin{aligned}
S_0 &\to S \\
S &\to ASA|aB|a \\
A &\to B|S|\epsilon \\
B &\to b
\end{aligned}
\qquad\qquad
\begin{aligned}
S_0 &\to S \\
S &\to ASA|aB|a|SA|AS|S \\
A &\to B|S|\epsilon \\
B &\to b
\end{aligned}
$$

Removendo regras unitárias

Usando $S_0 \to S$.

$$
\begin{aligned}
S_0 &\to S \\
S &\to ASA|aB|a|SA|AS|S \\
A &\to B|S \\
B &\to b
\end{aligned}
\qquad
\begin{aligned}
S_0 &\to S|ASA|aB|a|SA|AS \\
S &\to ASA|aB|a|SA|AS \\
A &\to B|S \\
B &\to b
\end{aligned}
$$

Removendo as unitárias $A \to B$ e $A \to S$

$$
\begin{aligned}
S_0 &\to ASA|aB|a|SA|AS \\
S &\to ASA|aB|a|SA|AS \\
A &\to B|S|b \\
B &\to b
\end{aligned}
\qquad
\begin{aligned}
S_0 &\to ASA|aB|a|SA|AS \\
S &\to ASA|aB|a|SA|AS \\
A &\to S|b|ASA|aB|a|SA|AS \\
B &\to b
\end{aligned}
$$

Ajustando para não termos mais que 2 variáveis ao lado direito

$$
\begin{aligned}
S_0 &\rightarrow ASA|aB|a|SA|AS \\
S &\rightarrow ASA|aB|a|SA|AS \\
A &\rightarrow b|ASA|aB|a|SA|AS \\
B &\rightarrow b
\end{aligned}
$$

$$
\begin{aligned}
S_0 &\rightarrow AA_1|UB|a|SA|AS \\
S &\rightarrow AA_1|UB|a|SA|AS \\
A &\rightarrow b|AA_1|UB|a|SA|AS \\
A_1 &\rightarrow SA \\
U &\rightarrow a \\
B &\rightarrow b
\end{aligned}
$$

E pronto!

1 Recordando

2 Conversão para FNC: mais um exemplo

3 Propriedades

4 Problemas decidíveis: 3

5 Problemas indecidíveis

6 Exercícios

Questões importantes em Linguagens Formais

1. Como determinar se uma linguagem é LLC?

Questões importantes em Linguagens Formais

1. Como determinar se uma linguagem é LLC?
2. A união de duas LLC produz uma LLC? E as operações de intersecção, concatenação e complemento?

Questões importantes em Linguagens Formais

1. Como determinar se uma linguagem é LLC?
2. A união de duas LLC produz uma LLC? E as operações de intersecção, concatenação e complemento?
3. Como verificar se uma LLC é finita?

Questões importantes em Linguagens Formais

1. Como determinar se uma linguagem é LLC?
2. A união de duas LLC produz uma LLC? E as operações de intersecção, concatenação e complemento?
3. Como verificar se uma LLC é finita?

Vejamos alguns exemplos de linguagens a seguir:

Estas não são LLC

Palavras duplicadas

$$L_1 = \{ww \mid w \text{ é palavra sobre } \Sigma = \{a, b\}\}$$

Este é um exemplo de linguagem que não é uma LLC.

Estas não são LLC

Palavras duplicadas

$$L_1 = \{ww \mid w \text{ é palavra sobre } \Sigma = \{a, b\}\}$$

Este é um exemplo de linguagem que não é uma LLC.

Triplo balanceamento

$$L_2 = \{a^n b^n c^n \mid n \geq 0\}$$

Sorte nossa que a maioria das LP não possui estruturas deste tipo.

LLC e problemas sem solução

Algumas questões relativas às LLC não possuem solução, como:

- Não existe um algoritmo genérico capaz de analisar duas LLC quaisquer e concluir se elas são iguais ou diferentes

LLC e problemas sem solução

Algumas questões relativas às LLC não possuem solução, como:

- Não existe um algoritmo genérico capaz de analisar duas LLC quaisquer e concluir se elas são iguais ou diferentes

Outras questões possuem soluções, tais como:

- Verificar se uma dada palavra pertence ou não a uma linguagem, através de um algoritmo genérico.

Estas são questões muito relevantes para a CC.

Fechamento das LLC

Fechamento

Das operações já estudadas, as linguagens livre do contexto são fechadas sob a união, concatenação e operação de Kleene.
Elas ainda são fechadas sob a substituição (se $f(\epsilon) = \epsilon$, então $f(sa) = f(s)f(a)$), homomorfismo inverso e intersecção com uma linguagem regular.

1 Recordando

2 Conversão para FNC: mais um exemplo

3 Propriedades

4 Problemas decidíveis: 3

5 Problemas indecidíveis

6 Exercícios

Problemas decidíveis: O que são

- Na Teoria da Computabilidade, um problema de decisão é uma questão feita a um sistema formal que exige ume resposta sim ou não
- Exe: Dados dois números x e y, podemos afirmar que x é múltiplo de y?
- Algoritmos: são métodos usados para resolver os problemas de decisão
- Se um problema de decisão pode ser resolvido através de um algoritmo, este é um problema decidível
- A área de Complexidade Computacional classifica problemas de decisão decidíveis pelo esforço computacional em resolvê-los

Dedidíveis

Vacuidade

Dada uma gramática livre de contexto A, $L(A) = \emptyset$?

Finitude

Dada uma GLC A, $L(A)$ é finita?

Composição

Dada uma GLC G, e uma palavra w, então $w \in L(G)$?

Algoritmos eficientes em tempo polinomial para o problema de composição são o Algoritmo CYK e Algoritmo Earley.

Parsers: analisadores sintáticos

Composição

Dada uma GLC G, e uma palavra w, então $w \in L(G)$?

Algoritmos eficientes em tempo polinomial para o problema de composição são o Algoritmo CYK e Algoritmo Earley.

- CYK e Early são, de fato, analisadores sintáticos para GLC
- CYK (Cocke–Younger–Kasami) algoritmo necessita de GLC na FNC
- A maioria da LP usam analisadores sintáticos baseados em GLC

Nota

- Muitos distinguem duas fases no processo de transformação em LP de código-fonte para códido objeto
  1. Verificação da sintaxe, análise sintática
  2. Análise semântica. Gerar programas que compilam.
- Para a fase 1, as GLC são suficientes
- Fase 2: nomes das var. existem, definição de tipos, compatibilidade de tipos, macros, desambiguação de operadores, etc. São operações que não são possíveis de definir numa GLC.

1 Recordando

2 Conversão para FNC: mais um exemplo

3 Propriedades

4 Problemas decidíveis: 3

5 Problemas indecidíveis

6 Exercícios

Universalidade

Universalidade

Dada GLC A, $L(A) = \Sigma^*$.
(Dada uma GLC podemos afirmar que ela gera a linguagem formada por todas as cadeias sobre os símbolos terminais usados nas regras de produção?)

Universalidade

Universalidade

Dada GLC A, $L(A) = \Sigma^*$.
(Dada uma GLC podemos afirmar que ela gera a linguagem formada por todas as cadeias sobre os símbolos terminais usados nas regras de produção?)

Pode-se mostrar que este problema é indecidível a partir do problema da parada, provado Alan Turing.

Universalidade

Universalidade

Dada GLC A, $L(A) = \Sigma^*$.
(Dada uma GLC podemos afirmar que ela gera a linguagem formada por todas as cadeias sobre os símbolos terminais usados nas regras de produção?)

Pode-se mostrar que este problema é indecidível a partir do problema da parada, provado Alan Turing.

Problema da Parada

Dadas uma descrição de um programa e uma entrada finita, decida se o programa termina de rodar ou rodará indefinidamente.

Alan Turing demonstrou, em 1936, que não existe um algoritmo genérico para resolver este problema.

Igualdade de linguagens

Igualdade de linguagens

Dadas duas GLC, A e B, $L(A) = L(B)$?
Dadas duas GLC, pode-se afirmar que elas geram a mesma linguagem?

Igualdade de linguagens

Igualdade de linguagens

Dadas duas GLC, A e B, $L(A) = L(B)$?
Dadas duas GLC, pode-se afirmar que elas geram a mesma linguagem?

A indecibilidade deste problema é conseqüencia direta do problema anterior, ou seja, é imposível decidir se uma GLC é equivalente a uma GLC trivial que produza todas as cadeias com os símbolos terminais usados nas regras de produção.

Ainda sobre os indedidíveis

Intersecção vazia

Dadas duas GLC A e B, $L(A) \cap L(B) = \emptyset$?

Sabe-se que intersecção entre uma LLC e uma LR é uma LLC. A variante do problema em que B é uma gramática regular, é decidível.

Contenção

Dadas duas GLC A e B $L(A) \subseteq L(B)$?

Se B for uma GR é decidível.

Nas bordas da definição de Chomsky

Usando o Teorema de Greibach[1] pode-se demonstrar que este três problemas são indecidíveis:

Mais três

1. Dada uma GLC pode-se afirmar que ela descreve uma LLC?
2. Dada uma GLC, pode-se afirmar que ela descreve uma gramática regular?
3. Dada uma GLC, pode-se afirmar que ela é inerentemente ambígua?

[1] Sheila Greibach mostrou em 1963 que algumas propriedades das LF são indecidíveis.

1. Recordando
2. Conversão para FNC: mais um exemplo
3. Propriedades
4. Problemas decidíveis: 3
5. Problemas indecidíveis
6. Exercícios

GLC 1

$$
\begin{aligned}
R &\rightarrow XRX|S \\
S &\rightarrow aTb|bTa \\
T &\rightarrow XTX|X|\epsilon \\
X &\rightarrow a|b
\end{aligned}
$$

Dada a GLC G definida pelas regras de produção acima, responda:

1. Caracterize os terminais, as variáveis e o símbolo inicial de G.
2. Dê 3 cadeias que estão em $L(G)$ e 3 que não estão.
3. $T \rightarrow^* aba$?
4. $T \rightarrow^* XX$?
5. Descreva $L(G)$ em português.

GLC 2

Descreva a linguagem gerada pela seguinte GLC:

$$G_2 = (\{S\}, \{a, b\}, P_2, S)$$

para P_2

$$S \rightarrow aSa$$
$$S \rightarrow bSb$$
$$S \rightarrow \epsilon$$

GLC 2

Descreva a linguagem gerada pela seguinte GLC:

$$G_2 = (\{S\}, \{a, b\}, P_2, S)$$

para P_2

$$S \to aSa$$
$$S \to bSb$$
$$S \to \epsilon$$

Resposta $G = \{ww^R \mid w \in \{a, b\}^*\}$

GLC 3

Descreva uma GLC que gere a seguinte linguagem:

$$L_3 = \{w \in \{0,1\}^* \mid w \text{ contém ao menos dois símbolos '1'}\}$$

GLC 3

Descreva uma GLC que gere a seguinte linguagem:

$$L_3 = \{w \in \{0,1\}^* \mid w \text{ contém ao menos dois símbolos '1'}\}$$

Resposta $G = \{V, \Sigma, P_3, S\}$ tal que $V = \{S, X\}$, $\Sigma = \{0,1\}$ e P_3

$$S \rightarrow X1X1X$$
$$X \rightarrow 0X \mid 1X \mid \epsilon$$

GLC 4

Descreva uma GLC que gere a seguinte linguagem:

$$L_4 = \{w \in \{0,1\}^* \mid w = w^R \text{ e } w \text{ é ímpar}\}$$

GLC 4

Descreva uma GLC que gere a seguinte linguagem:

$$L_4 = \{w \in \{0,1\}^* \mid w = w^R \text{ e } w \text{ é ímpar}\}$$

Resposta $G = \{V, \Sigma, P_4, S\}$ tal que $V = \{S\}$, $\Sigma = \{0,1\}$ e P_4

$$S \rightarrow 0S0 \mid 1S1 \mid \epsilon$$

GLC 5

Descreva uma GLC que gere a seguinte linguagem:

$$L_5 = \{w \in \{0,1\}^* \text{ t.q. } |w| = \text{ ímpar e o símbolo do meio é } 0\}$$

GLC 5

Descreva uma GLC que gere a seguinte linguagem:

$$L_5 = \{w \in \{0,1\}^* \text{ t.q. } |w| = \text{ ímpar e o símbolo do meio é } 0\}$$

Resposta $G = \{V, \Sigma, P_5, S\}$ tal que $V = \{S\}$, $\Sigma = \{0,1\}$ e P_5

$$S \to 0S0 \mid 0S1 \mid 1S0 \mid 1S1 \mid 0$$

GLC 6

Descreva uma GLC que gere a seguinte linguagem:

$$L_6 = \varnothing$$

GLC 6

Descreva uma GLC que gere a seguinte linguagem:

$$L_6 = \varnothing$$

Resposta $G = \{V, \Sigma, P_6, S\} \mid V = \{S\}$, $\Sigma = \{0, 1\}$ e P_6

$$S \rightarrow S$$

Caros,

Procurem fixar o conteúdo lendo os livros texto e fazendo exercícios.