Forma Normal de Chomsky

Quando trabalhamos com GLCs, é conveniente simplificá-las por diversas razões. Por exemplo, podemos identificar variáveis que não participam da geração de sentença alguma, as quais poderiam ser eliminadas da gramática. Ou ainda, podemos querer simplificá-las para tornar o processo de geração de palavra mais eficiente. Além disso, podemos adequar as GLCs a formatos específicos, facilitando a construção automática de analisadores sintáticos para elas.

Em geral, esses objetivos são alcançados colocando uma gramática em uma forma normal. Existem algumas formas normais para GLC. Nessa aula, estudaremos a forma normal de Chomsky (FNC). As regras na FNC possuem a seguinte forma:

- $A \rightarrow BC$ (só duas variáveis no lado direito das regras, não incluindo a variável inicial)
- $A \rightarrow a$ (uma variável produz um terminal)
- $P \rightarrow \lambda$ (lambda pode ser gerado somente pela variável de partida, caso ela pertença à linguagem)

Intuitivamente, a FNC faz com que as árvores de derivação sejam árvores binárias. Logo, são necessários no máximo 2n-1 passos para geração de qualquer palavra. Observe que as regras em GLCs na FNC ou introduzem uma variável à forma sentencial, ou eliminam uma variável, substituindo-a por um terminal.

O método abaixo descreve o procedimento para obtenção de uma GLC na FNC. Ele consiste dos seguintes passos:

1. Crie uma nova variável de partida P' e adicione a regra $P' \rightarrow P$ ao conjunto de regras
2. Elimine regras λ
3. Elimine regras unitárias
4. Elimine variáveis inúteis
5. Altere regras $X \rightarrow w$ tal que $w \in \Sigma^*$ e $|w| \geq 2$ por regras em que cada terminal é substituído por uma variável e acrescentar regras do tipo $Y \rightarrow a$ para $a \in \Sigma$
6. Substitua regras $X \rightarrow Y_1 Y_2 \ldots Y_n$ em que $n \geq 3$ e Y_i é uma variável por um encadeamento de regras com lado direito de tamanho 2

Vamos discutir como cada passo é executado agora.

- <u>Eliminação de regras λ:</u>

★ Como vimos, a palavra vazia só pode ser produzida pela variável de partida, caso ela pertença à linguagem. Sendo assim, precisamos eliminar todas as outras regras que produzam λ. Para isso, precisamos introduzir o conceito de **variáveis anuláveis**. Uma variável X é anulável se:

- $X \Rightarrow^* \lambda$

Denotamos o conjunto de variáveis anuláveis de uma GLC $G = (V, \Sigma, R, P)$ por $VA_G = \{X \in V \mid X \Rightarrow^* \lambda\}$. Esse conjunto também pode ser definido recursivamente por:

- Se $X \rightarrow \lambda$, então $X \in VA_G$
- Se $X \rightarrow w \wedge w \in VA_G^*$, então $X \in VA_G$

Então, para eliminar as regras λ, substituímos cada regra $Y \rightarrow x_o X_1 x_1 X_2 x_2 \dots X_n x_n \in R$, em que $X_i \in VA_G$, por duas novas regras:

1. Uma em que X_i continua ocorrendo (já que pode produzir outras sentenças);
2. Outra em que X_i é apagado da regra (caso em que X_i produziu λ).

Dessa forma, o novo conjunto de regras R' sem regras λ será composto por:

1. Todas as regras $Y \rightarrow z \in R$ tais que $z \cap VA_G = \emptyset$;
2. As regras construídas conforme o método acima.

Exemplo: Considere a gramática a seguir. Obter uma gramática equivalente sem regras λ.

$P \rightarrow APB \mid C$
$A \rightarrow A\mathtt{aa}A \mid \lambda$
$B \rightarrow BB\mathtt{b} \mid C$
$C \rightarrow \mathtt{c}C \mid \lambda$

$VA_G = \{A, C, P, B\}$

$P \rightarrow APB \mid PB \mid AB \mid AP \mid P \mid A \mid B \mid C$

$A \rightarrow AaaA \mid aaA \mid Aaa \mid aa$

$B \rightarrow BBb \mid Bb \mid b \mid C$

$C \rightarrow cC \mid c$

- <u>Eliminação de regras unitárias</u>:

O segundo passo é a eliminação de regras unitárias. Pela definição, cada regra ou acrescenta uma nova variável à forma sentencial ou remove, substituindo-a por um terminal.

A eliminação de regras unitárias envolve o conceito de **variáveis encadeadas**. Dizemos que duas variáveis $X, Z \in V$ estão encadeadas se existem regras $X \rightarrow Y_1; Y_1 \rightarrow Y_2; Y_2 \rightarrow Y_3 \dots Y_n \rightarrow Z$. Nesse caso, dizemos que $Z \in enc(X)$, ou seja, Z pertence ao conjunto de variáveis encadeadas a X. Recursivamente:

- $X \in enc(X)$
- Se $Y \in enc(X) \wedge Y \rightarrow Z \in R$, então $Z \in enc(X)$

Dessa forma, para eliminar regras unitárias, basta substituí-las por produções não

unitárias das variáveis encadeadas. Ou seja, o novo conjunto de regras $R' = \{X \to w \mid w \notin V \wedge \exists Y \in enc(X)\ Y \to w\}$.

- Exemplo: Elimine regras unitárias da gramática a seguir:

$enc(E) = \{E, T, F\}$

$enc(T) = \{T, F\}$

$enc(F) = \{F\}$

$E \to E+T \mid T*F \mid (E) \mid a$

$T \to T*F \mid (E) \mid a$

$F \to (E) \mid a$

- <u>Eliminar variáveis inúteis</u>:

Pode ser que, por um erro de projeto, existam variáveis que não geram qualquer palavra da linguagem. Regras envolvendo essas variáveis podem ser eliminadas do conjunto de regras, já que não interferem na linguagem gerada.

Uma variável é **útil** se ela participa da geração de pelo menos uma palavra da linguagem. Ou seja, uma variável $X \in V$ é útil se, e somente se, existirem $u, v \in (V \cup \Sigma)^*$ e $w \in \Sigma^*$ tais que:

- $P \Rightarrow^* uXv \Rightarrow^* w$

Considere a gramática a seguir:

As variáveis A e C são inúteis, visto que não produzem qualquer palavra; A não pode ser expandida, enquanto C é inalcançável a partir da variável de partida. A variável B produz terminais, contudo ela se torna também inútil, uma vez que nenhuma palavra pode ser gerada com a regra em que ela ocorre.

Dessa forma, precisamos determinar as variáveis que produzem sentenças que eventualmente podem produzir palavras, e as variáveis que são 'alcançáveis' a partir da variável inicial; ou seja, participam de alguma derivação.

O conjunto S_G de variáveis que eventualmente produzem palavras é dado pela seguinte definição recursiva:

- $X \in S_G$, se existe $X \to w \in R \wedge w \in \Sigma^*$
- Se $X \to w \in R$ e $vars(w) \subseteq S_G$, então $X \in S_G$; em que $vars(w)$ é o conjunto de variáveis em w.

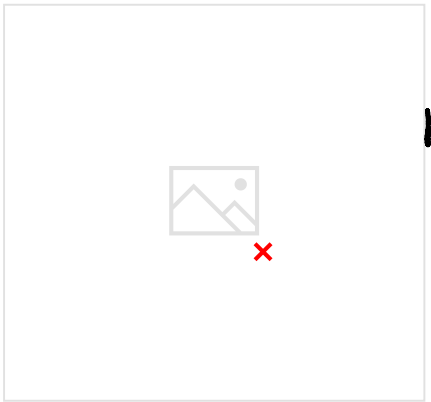
O conjunto A_G de variáveis que participam de alguma derivação é dado por:

- $P \in A_G$;
- Se $X \to w \in R \wedge X \in A_G$ então $vars(w) \subseteq A_G$

Finalmente, o processo de obtenção de variáveis úteis ocorre em duas etapas:

1. Obtém-se S_G e filtram-se somente as regras evolvendo essas variáveis;
2. Do conjunto resultante de 1, obtém-se A_G e filtram-se as regras com essas variáveis.

Exemplo: Remova as variáveis inúteis da gramática abaixo:



$S_G = \{B, D, F, A\}$

R':

$A \to BD$

$B \to B0 \mid 0$

$D \to 1D \mid 1$

$F \to 1F1 \mid 1$

$A_G = \{A, B, D\}$ R'':

$A_G = \{A, B, D\}$ R'':

$A \to BD$

$B \to BO \mid O$

$D \to 1D \mid 1$

- <u>Modificar regras que produzem palavras com pelo menos 2 terminais</u>:

Nesse caso, basta criar regras do tipo $Y \to a$ para cada $a \in \Sigma$

- <u>Particionar regras com várias variáveis</u>:

Nesse caso, basta criar variáveis adicionais para encadear as regras. Assim, uma regra $Y \to X_1X_2 \dots X_n$ é substituída pelas regras $Y \to X_1Z_1; Z_1 \to X_2Z_2; \dots Z_{n-2} \to X_{n-1}X_n$.

Embora sejam relativamente intuitivos, as demonstrações de que esses procedimentos resultam em gramáticas equivalentes podem ser verificadas no livro texto.

Segue um exemplo de obtenção de uma gramática na FNC.

Exemplo: Obtenha uma gramática na FNC equivalente à GLC abaixo:

$P' \to L$

$L \to (S)$

$S \to SE \mid \lambda$

$E \to a \mid L$

$VA_G = \{S\}$

$P' \to L$

$L \to (S) \mid ()$

$S \to SE \mid E$

$E \to a \mid L$

$enc(P') = \{P, L\}$

$enc(L) = \{L\}$

$enc(S) = \{S, E, L\}$

$enc(E) = \{E, L\}$

$P' \to (S) \mid ()$

$L \to (S) \mid ()$

$S \to SE \mid a \mid (S) \mid ()$

$E \to a \mid (S) \mid ()$

$S_G = \{P', L, S, E\}$

$A_G = \{P', S, E\}$

$P' \to (S) \mid ()$

$S \to SE \mid a \mid (S) \mid ()$

$E \to a \mid (S) \mid ()$

$P' \to XSY \mid XY$

$S \to SE \mid a \mid XSY \mid XY$

$E \to a \mid XSY \mid XY$

$X \to ($

$Y \to)$

$P' \to XZ_1 \mid XY$

$Z_1 \to SY$

$S \to SE \mid a \mid XZ_1 \mid XY$

$Z_1 \rightarrow$?

$S \rightarrow SE \mid a \mid XZ_1 \mid XY$

$E \rightarrow a \mid XZ_1 \mid XY$

$X \rightarrow ($

$Y \rightarrow)$