4.7. Eliminación de las variables inútiles

En una GIC puede haber dos tipos de variables inútiles: aquéllas que nunca aparecen en el curso de una derivación y aquéllas que no se pueden convertir en cadenas de terminales. A continuación se precisan estos conceptos.

4.7.1. Definiciones.

(i) Una variable A es **alcanzable** o **accesible** si existen $u, v \in (V \cup \Sigma)^*$ tales que $S \stackrel{*}{\Longrightarrow} uAv$. La variable inicial S es alcanzable por definición.

(ii) Una variable A es **terminable** si existe $w \in \Sigma^*$ tal que $A \stackrel{*}{\Longrightarrow} w$. En particular, si $A \to \lambda$ es una producción entonces A es terminable.

(iii) A es una variable **inútil** si no es alcanzable o no es terminable.

Existen algoritmos para detectar todas las variables inútiles de una GIC que permiten construir una gramática G' equivalente a una gramática G dada, de tal manera que G' no tenga variables inútiles.

4.7.2. Algoritmo para encontrar las variables terminables.

El siguiente algoritmo sirve para encontrar todas las variables terminables de una GIC.

$\mathbf{TERM}_1 := \{A \in V : \text{Existe una producción de la forma } A \to w, w \in \Sigma^*\}$.
$\mathbf{TERM}_{i+1} := \mathbf{TERM}_i \cup \{A \in V : \exists \text{ producción } A \to w,\ w \in (\Sigma \cup\ \mathbf{TERM}_i)^*\}$.

Obtenemos una sucesión creciente de conjuntos de variables:

$$\mathbf{TERM}_1 \subseteq \mathbf{TERM}_2 \subseteq \mathbf{TERM}_3 \subseteq \cdots$$

Como el conjunto de variables es finito, existe k tal que

$$\mathbf{TERM}_k = \mathbf{TERM}_{k+1} = \mathbf{TERM}_{k+2} = \cdots$$

El conjunto **TERM** de variables terminables es entonces

$$\mathbf{TERM} := \bigcup_{i \geq 1} \mathbf{TERM}_i$$

El anterior algoritmo se puede presentar de la siguiente forma:

INICIALIZAR:

$\mathbf{TERM} := \left\{A \in V : \exists \text{ producción } A \to w, w \in \Sigma^*\right\}$

REPETIR:

$\mathbf{TERM} := \mathbf{TERM} \cup \left\{A \in V : \exists \text{ producción } A \to w, w \in (\Sigma \cup \mathbf{TERM})^*\right\}$

HASTA:

No se añaden nuevas variables a **TERM**

Ejemplo Encontrar las variables terminables de la siguiente gramática.

$$G : \begin{cases} S \to ACD \mid bBd \mid ab \\ A \to aB \mid aA \mid C \\ B \to aDS \mid aB \\ C \to aCS \mid CB \mid CC \\ D \to bD \mid ba \\ E \to AB \mid aDb \end{cases}$$

<u>Solución</u>:

$$\begin{aligned} &\mathbf{TERM}_1 = \{S, D\}. \\ &\mathbf{TERM}_2 = \{S, D\} \cup \{B, E\} = \{S, D, B, E\}. \\ &\mathbf{TERM}_3 = \{S, D, B, E\} \cup \{A\} = \{S, D, B, E, A\}. \\ &\mathbf{TERM}_4 = \{S, D, B, E, A\} \cup \{\ \} = \{S, D, B, E, A\}. \\ &\mathbf{TERM} = \{S, A, B, D, E\}. \end{aligned}$$

Conjunto de variables no terminables: $\{C\}$.

4.7.3. Algoritmo para encontrar las variables alcanzables.

El siguiente algoritmo sirve para encontrar todas las variables alcanzables de una GIC.

$\mathbf{ALC}_1 := \{S\}$.

$\mathbf{ALC}_{i+1} := \mathbf{ALC}_i \cup \{A \in V : \exists \text{ producc. } B \to uAv, B \in \mathbf{ALC}_i\ u, v \in (V \cup \Sigma)^*\}$.

Obtenemos una sucesión creciente de conjuntos de variables:

$$\mathbf{ALC}_1 \subseteq \mathbf{ALC}_2 \subseteq \mathbf{ALC}_3 \subseteq \cdots$$

Como el conjunto de variables es finito, existe k tal que

$$\mathbf{ALC}_k = \mathbf{ALC}_{k+1} = \mathbf{ALC}_{k+2} = \cdots$$

El conjunto **ALC** de variables alcanzables es entonces

$$\mathbf{ALC} = \bigcup_{i \geq 1} \mathbf{ALC}_i$$

El anterior algoritmo se puede presentar de la siguiente forma:

INICIALIZAR:

$\mathbf{ALC} := \{S\}$

REPETIR:

$\mathbf{ALC} := \mathbf{ALC} \cup \big\{ A \in V : \exists$ producción $B \to uAv$, $B \in \mathbf{ALC}$ y $u, v \in (V \cup \Sigma)^* \big\}$

HASTA:

No se añaden nuevas variables a **ALC**

Ejemplo Encontrar las variables alcanzables de la siguiente gramática.

$$G: \quad \begin{cases} S \to aS \mid AaB \mid ACS \\ A \to aS \mid AaB \mid AC \\ B \to bB \mid DB \mid BB \\ C \to aDa \mid ABD \mid ab \\ D \to aD \mid DD \mid ab \\ E \to FF \mid aa \\ F \to aE \mid EF \end{cases}$$

Solución:

$\mathbf{ALC}_1 = \{S\}$.
$\mathbf{ALC}_2 = \{S\} \cup \{A, B, C\} = \{S, A, B, C\}$.
$\mathbf{ALC}_3 = \{S, A, B, C\} \cup \{D\} = \{S, A, B, C, D\}$.
$\mathbf{ALC}_4 = \{S, A, B, C, D\} \cup \{\ \} = \{S, A, B, C, D\}$
$\mathbf{ALC} = \{S, A, B, C, D\}$.

Conjunto de variables no alcanzables: $\{E, F\}$.

Dada una GIC G, los dos algoritmos anteriores (algoritmo 4.7.2 y algoritmo 4.7.3) se pueden llevar a cabo en dos órdenes diferentes:

(I) $G \xrightarrow{Algoritmo}$ Eliminar variables no-terminables $G_1 \xrightarrow{Algoritmo}$ Eliminar variables no-alcanzables G_2

(II) $G \xrightarrow{Algoritmo}$ Eliminar variables no-alcanzables $G_3 \xrightarrow{Algoritmo}$ Eliminar variables no-terminables G_4

Esto da lugar a las siguientes preguntas:

(1) ¿Es $G_2 = G_4$?

(2) ¿G_2 tiene variables inútiles?

(3) ¿G_4 tiene variables inútiles?

El siguiente ejemplo muestra que la respuesta a la pregunta (1) es *no* y que al realizar los algoritmos en el orden (II) pueden permanecer en G_4 algunas variables inútiles.

Ejemplo Considérese la siguiente gramática G.

$$G: \quad \begin{cases} S \to a \mid AB \\ A \to aA \mid \lambda \end{cases}$$

Aplicación de los algoritmos en el orden (I):

Variables terminables de G: **TERM**$= \{S, A\}$.

$$G_1: \quad \begin{cases} S \to a \\ A \to aA \mid \lambda \end{cases}$$

Variables alcanzables de G_1: **ALC**$= \{S\}$.

$$G_2: \quad \{ S \to a$$

Se tiene que $L(G_2) = \{a\}$.

Aplicación de los algoritmos en el orden (II):

Variables alcanzables de G: **ALC**$=\{S, A, B\}$.

$$G_3: \quad \begin{cases} S \to a \mid AB \\ A \to aA \mid \lambda \end{cases}$$

Variables terminables de G_3: **TERM**$=\{S, A\}$.

$$G_4: \quad \begin{cases} S \to a \\ A \to aA \mid \lambda \end{cases}$$

Se observa que en G_4, A no es alcanzable. Además, $G_2 \neq G_4$.

No obstante, si los algoritmos se llevan a cabo en el orden (I) la gramática G_2 ya no tiene variables inútiles. Nos podemos convencer de eso observando que las variables alcanzables obtenidas al finalizar el procedimiento siguen siendo terminables. Más precisamente, si una variable A permanece al finalizar el procedimiento completo, será alcanzable, y si la derivación $A \stackrel{*}{\Rightarrow} w$, con $w \in \Sigma^*$, se podía hacer antes de eliminar las variables no alcanzables, también se podrá realizar en la gramática final ya que todas las variables que aparezcan en esa derivación serán alcanzables.

Ejemplo Eliminar las variables inútiles de la siguiente gramática G.

$$G: \quad \begin{cases} S \to SBS \mid BC \mid Bb \\ A \to AA \mid aA \\ B \to aBCa \mid b \\ C \to aC \mid ACC \mid abb \\ D \to aAB \mid ab \\ E \to aS \mid bAA \\ F \to aDb \mid aF \end{cases}$$

Solución: Ejecutamos los algoritmos en el orden (I):

$$\begin{aligned} &\mathbf{TERM}_1 = \{B, C, D\} \\ &\mathbf{TERM}_2 = \{B, C, D\} \cup \{S, F\} \\ &\mathbf{TERM}_3 = \{B, C, D, S, F\} \cup \{E\} = \{B, C, D, S, F, E\} \\ &\mathbf{TERM}_4 = \{B, C, D, S, F, E\} \cup \{\ \} \end{aligned}$$

La única variable no-terminable de G es A. Por lo tanto, G es equivalente a la siguiente gramática G_1:

$$G_1: \quad \begin{cases} S \to SBS \mid BC \mid Bb \\ B \to aBCa \mid b \\ C \to aC \mid abb \\ D \to ab \\ E \to aS \\ F \to aDb \mid aF \end{cases}$$

Variables alcanzables de G_1:

$$\begin{aligned}
&\mathbf{ALC}_1 = \{S\}\\
&\mathbf{ALC}_2 = \{S\} \cup \{B, C\}\\
&\mathbf{ALC}_3 = \{S, B, C\} \cup \{\ \}
\end{aligned}$$

Las variables D, E, F son no alcanzables. Por lo tanto, G es equivalente a la siguiente gramática G_2, que no tiene variables inútiles.

$$G_2 : \quad \begin{cases} S \to SBS \mid BC \mid Bb \\ B \to aBCa \mid b \\ C \to aC \mid abb \end{cases}$$

Ejercicios de la sección 4.7

Eliminar las variables inútiles de las siguientes gramáticas:

1.
$$G : \quad \begin{cases} S \to SS \mid SBB \mid CCE \\ A \to aE \mid bE \\ B \to bB \mid Db \\ C \to aC \mid bB \\ D \to aDb \mid ab \mid \lambda \\ E \to aA \mid bB \end{cases}$$

2.
$$G : \quad \begin{cases} S \to EA \mid SaBb \mid aEb \\ A \to DaD \mid bD \\ B \to bB \mid Ab \mid \lambda \\ C \to aC \mid bBC \\ D \to aEb \mid ab \\ E \to aA \mid bB \mid \lambda \\ F \to Fb \mid Fa \mid a \end{cases}$$