UNIVERSIDAD DE GRANADA

DECSAI

Modelos de Computación

Grado en Ingeniería Informática

Tema 4 – Gramáticas independientes del contexto

Manuel Pegalajar Cuéllar
manupc@ugr.es
Departamento de Ciencias de la Computación e Inteligencia Artificial
http://decsai.ugr.es

Objetivos del tema

- Conocer el concepto de árbol de derivación y ambigüedad en lenguajes independientes del contexto.
- Conocer algoritmos para simplificación de gramáticas.
- Conocer el problema de la pertenencia y la aplicación de formas normales para resolverlo.
- Conocer las formas normales de Chomsky y de Greibach y cómo pasar una gramática a estas formas normales.

Anotación sobre estas diapositivas:

El contenido de estas diapositivas es esquemático y representa un apoyo para las clases presenciales teóricas. No se considera un sustituto para apuntes de la asignatura.

Se recomienda al alumno completar estas diapositivas con notas/apuntes propios, tomados en clase y/o desde la bibliografía principal de la asignatura.

DECSAI

Gramáticas independientes del contexto

Una gramática $G = (V,T,P,S)$ se dice que es independiente del contexto o de tipo 2 si y solo si todas las producciones tienen la forma:

$$A \rightarrow \alpha$$

donde $A \in V, \alpha \in (V \cup T)^*$.

Los lenguajes generados por estas gramáticas se llaman también independiente del contexto o de tipo 2.

Árbol de derivación

Sea la gramática dada por las producciones:

$$S \rightarrow a|b|c$$
$$S \rightarrow (S+S)$$
$$S \rightarrow (S*S)$$

A esta gramática, por ejemplo, pertenece la palabra ((a+b)*c). Se obtiene aplicando la siguiente cadena de producciones:

S=>(S*S)=>((S+S)*S)=>((a+S)*S)=>((a+b)*S)=>((a+b)*c)

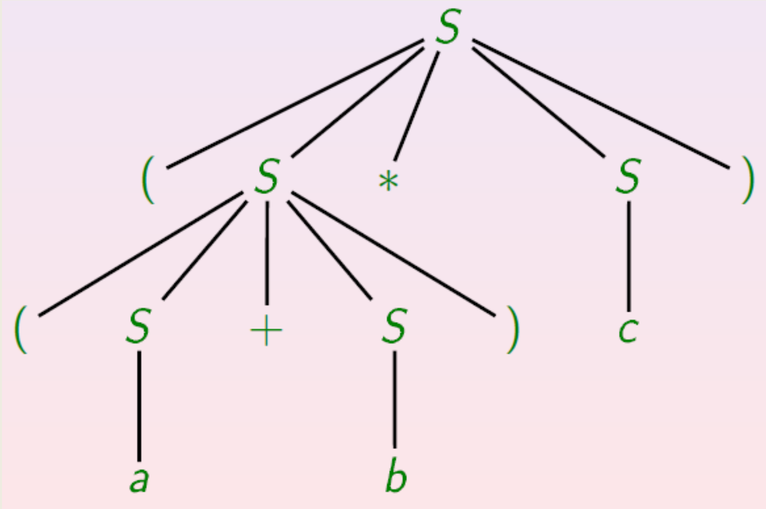
Árbol de derivación

Sea la gramática dada por las producciones:

$$S \rightarrow a|b|c$$
$$S \rightarrow (S+S)$$
$$S \rightarrow (S*S)$$

A esta gramática, por ejemplo, pertenece la palabra ((a+b)*c

Árbol de derivación:

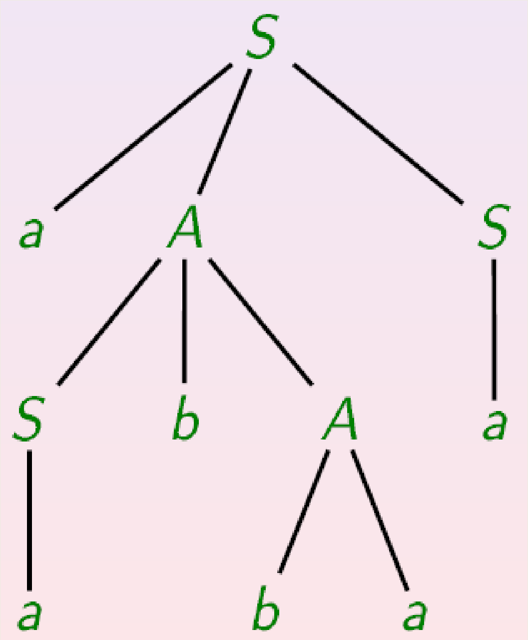


Construcción del árbol de derivación

- Cada nodo del árbol va a contener un símbolo.
- En el nodo raíz se pone el símbolo inicial *S*.
- Se efectúa una ramicación del árbol por cada producción que se aplique: Si a la variable de un nodo, *A*, se le aplica una determinada regla $A \rightarrow \alpha$, entonces para cada símbolo que aparezca en α se añade un hijo con el símbolo correspondiente, situados en el orden de izquierda a derecha.
- Este proceso se repite para todo paso de la derivación.
- Si la parte derecha es una cadena vacía, entonces se añade un solo hijo, etiquetado con ε.
- En cada momento, leyendo los nodos de izquierda a derecha se lee la palabra generada.

Un ejemplo

Con la gramática S→aAS, S→a, A→SbA, A→SS, A→ba, la palabra **aabbaa** tiene el árbol asociado:



Derivaciones de árboles

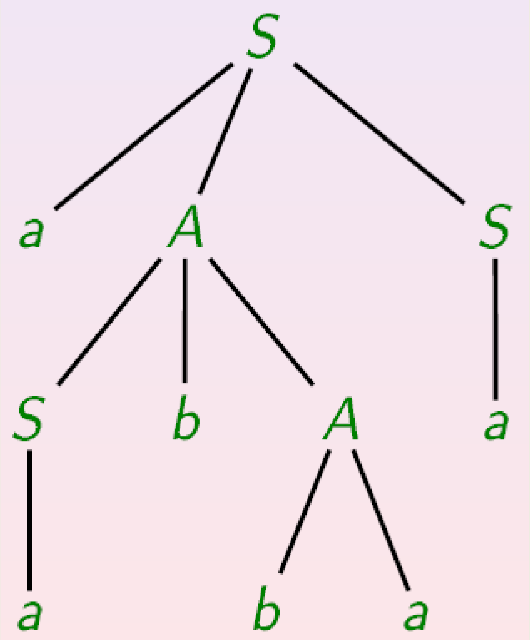
Un árbol de derivación puede proceder de dos cadenas de derivación distintas:

- Se llama derivación **por la izquierda** asociada a un árbol a aquella en la que siempre se deriva primero la primera variable (más a la izquierda) que aparece en la palabra.

- Se llama derivación **por la derecha** asociada a un árbol a aquella en la que siempre se deriva primero la última variable (más a la derecha) que aparece en la palabra.

Un ejemplo

Con la gramática S→aAS, S→a, A→SbA, A→SS, A→ba, la palabra **aabbaa** tiene el árbol asociado:



Derivación por la izquierda:
S=>aAS=>aSbAS=>aabAS=>aabbaS =>aabbaa

Derivación por la derecha:
S=>aAS=>aAa=>aSbAa=>aSbbaa=> =>aabbaa

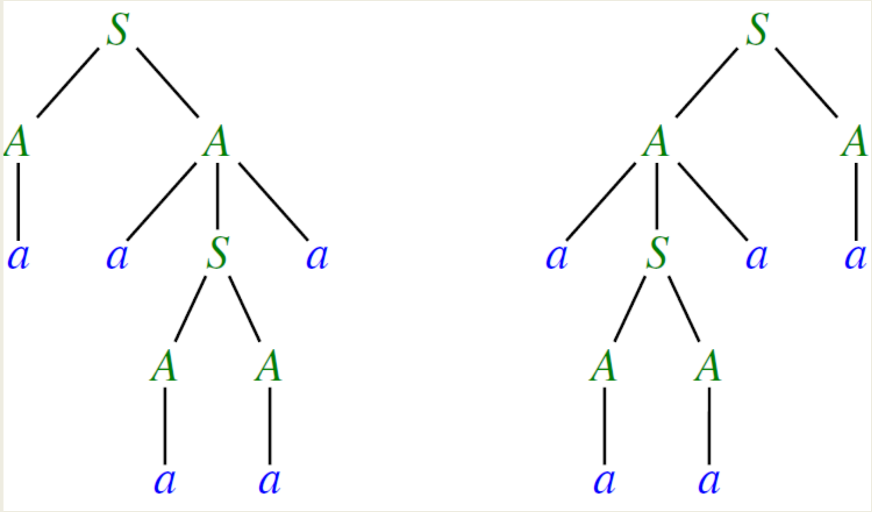
1. Árbol de derivación
2. Ambigüedad
3. Algoritmos de simplificación de gramáticas
    1. Símbolos y producciones inútiles
    2. Producciones nulas
    3. Producciones unitarias
4. Formas normales
    1. Forma normal de Chomsky
    2. Forma normal de Greibach

Gramáticas ambiguas

Una gramática se dice **ambigua** si existe una palabra con dos árboles de derivación distintos.

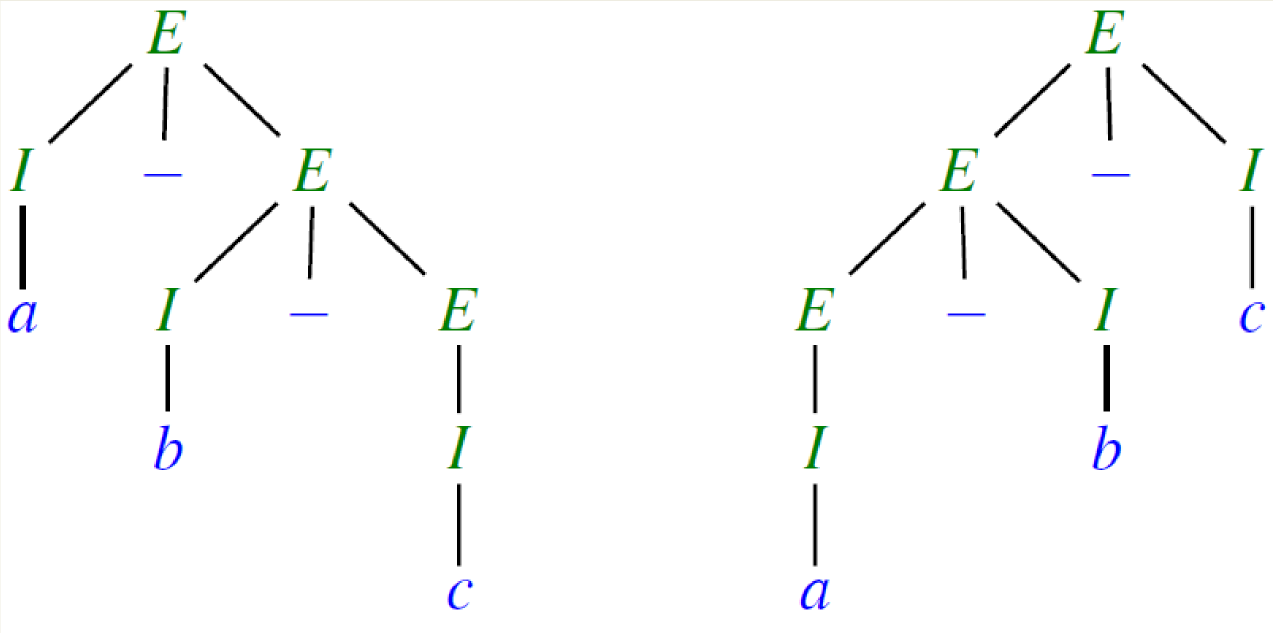
Un ejemplo

La gramática: $S \rightarrow AA$; $A \rightarrow aSa$; $A \rightarrow a$ es ambigua, ya que la palabra a^5 tiene los dos árboles siguientes:



Un ejemplo

La gramática $E \rightarrow I$; $E \rightarrow I\text{-}E$; $E \rightarrow E\text{-}I$; $I \rightarrow a|b|c|d$; es ambigua, ya que la palabra *a-b-c* admite dos árboles de derivación distintos:



Eliminando la producción E→I-E la gramática deja de ser ambigua.

Lenguajes inherentemente ambiguos

Un lenguaje de tipo 2 es **inherentemente ambiguo** si toda gramática que lo genera es ambigua.

Un ejemplo

Ejemplo: El lenguaje generado por la gramática
$S \rightarrow AA$; $A \rightarrow aSa$; $A \rightarrow a$, no es inherentemente ambiguo.

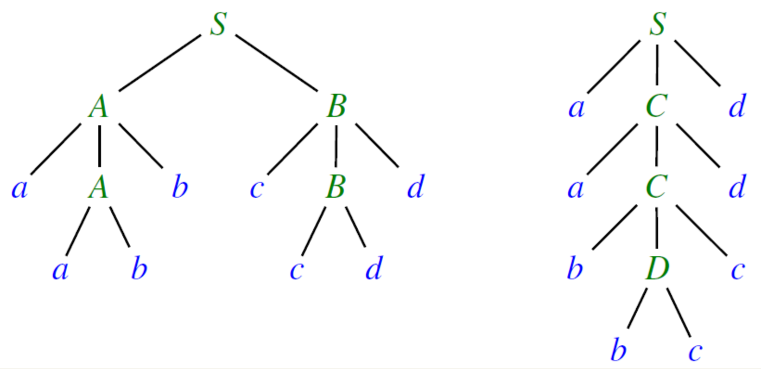
Este lenguaje es $L = \{a^{2+3*i} : i \geq 0\}$ y puede ser generado por la gramática: $S \rightarrow aa$; $S \rightarrow aaU$; $U \rightarrow aaaU$; $U \rightarrow aaa$, que no es ambigua. Ninguna palabra de esta gramática tiene dos árboles de derivación diferentes.

Un ejemplo

Ejemplo: Lenguaje $L = \{a^n b^n c^m d^m : n \geq 1, m \geq 1\} \cup \{a^n b^m c^m d^n : n \geq 1, m \geq 1\}$. Este lenguaje es inherentemente ambiguo. No hay una gramática de tipo 2 que lo genere sin ambigüedad.

Ejemplo de gramática:
$S \rightarrow AB$; $A \rightarrow ab$; $A \rightarrow aAb$; $B \rightarrow cd$; $B \rightarrow cBd$;
$S \rightarrow aCd$; $C \rightarrow aCd$; $C \rightarrow bDc$; $C \rightarrow bc$; $D \rightarrow bDc$; $D \rightarrow bc$

Por ejemplo, la palabra *aabbccdd* tiene dos árboles de derivación



Lenguajes inherentemente ambiguos

Un lenguaje de tipo 2 es **inherentemente ambiguo** si toda gramática que lo genera es ambigua.

Un ejemplo

Ejemplo: El lenguaje generado por la gramática
$S \rightarrow AA$; $A \rightarrow aSa$; $A \rightarrow a$, no es inherentemente ambiguo.

Este lenguaje es $L = \{a^{2+3*i} : i \geq 0\}$ y puede ser generado por la gramática: $S \rightarrow aa$; $S \rightarrow aaU$; $U \rightarrow aaaU$; $U \rightarrow aaa$, que no es ambigua. Ninguna palabra de esta gramática tiene dos árboles de derivación diferentes.

UNIVERSIDAD DE GRANADA

Gramáticas independientes del contexto

1. Árbol de derivación
2. Ambigüedad
3. Algoritmos de simplificación de gramáticas
    1. Símbolos y producciones inútiles
    2. Producciones nulas
    3. Producciones unitarias
4. Formas normales
    1. Forma normal de Chomsky
    2. Forma normal de Greibach

DECSAI

Símbolos útiles

Un símbolo $X \in (V \cup T)$ se dice **útil** si y solo si existe una cadena de derivaciones en G tal que

$$S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w \in T^*$$

Producciones útiles

Una producción se dice **útil** si y solo si todos sus símbolos son útiles. Esto es equivalente a que pueda usarse en la derivación de alguna palabra del lenguaje asociado a la gramática.

Eliminando todos los símbolos y producciones inútiles el lenguaje generado por la gramática no cambia.

Algoritmo para eliminar símbolos y producciones inútiles

El algoritmo para eliminar los símbolos y producciones inútiles consta de dos pasos:

1. Eliminar las variables desde las que no se puede llegar a una palabra de T^* y las producciones en las que aparezcan.

2. Eliminar aquellos símbolos que no sean alcanzables desde el símbolo inicial, *S*, y las producciones en las que estos aparezcan.

Un ejemplo

Ejemplo: S→AB, S→a, A→a.

Paso 1. Se elimina B y la producción S→AB
Paso 2. Entonces en el segundo se elimina la variable *A* y la producción *A*→*a*.

Algoritmo para el primer paso

Calculando V_t , conjunto de variables que se pueden sustituir por símbolos terminales.

Condición Básica: Si $A \rightarrow u$; u pertenece a T^*, entonces A pertenece a V_t

Condición Recursiva: Si $A \rightarrow \beta_1 \ldots \beta_k$ y cada β_j está en $T \cup V_t$, entonces A pertenece a V_t .

1. $V_t = \emptyset$
2. Para cada producción de la forma $A \rightarrow w$, A se introduce en V_t .
3. Mientras V_t cambie
   4. Para cada producción $B \rightarrow \alpha$
   5. Si todas las variables de α pertenecen a V_t ,
      B se introduce en V_t
6. Eliminar las variables que estén en V y no en V_t
7. Eliminar todas las producciones donde aparezca una variable de las eliminadas en el paso anterior.

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \rightarrow gAe, & S \rightarrow aYB, & S \rightarrow cY, & A \rightarrow bBY, \\
A \rightarrow ooC, & B \rightarrow dd, & B \rightarrow D, & C \rightarrow jVB, \\
C \rightarrow gi, & D \rightarrow n, & U \rightarrow kW, & V \rightarrow baXXX, \\
V \rightarrow oV, & W \rightarrow c, & X \rightarrow fV, & Y \rightarrow Yhm
\end{array}
$$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{B$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{B$

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \to gAe, & S \to aYB, & S \to cY, & A \to bBY, \\
A \to ooC, & B \to dd, & B \to D, & C \to jVB, \\
C \to gi, & D \to n, & U \to kW, & V \to baXXX, \\
V \to oV, & W \to c, & X \to fV, & Y \to Yhm
\end{array}
$$

$V_t = \{B, C$

Un ejemplo del algoritmo para el paso 1

$$S \longrightarrow gAe, \quad S \longrightarrow aYB, \quad S \longrightarrow cY, \quad A \longrightarrow bBY,$$
$$A \longrightarrow ooC, \quad B \longrightarrow dd, \quad B \longrightarrow D, \quad C \longrightarrow jVB,$$
$$C \longrightarrow gi, \quad D \longrightarrow n, \quad U \longrightarrow kW, \quad V \longrightarrow baXXX,$$
$$V \longrightarrow oV, \quad W \longrightarrow c, \quad X \longrightarrow fV, \quad Y \longrightarrow Yhm$$

$V_t = \{B, C$

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \to gAe, & S \to aYB, & S \to cY, & A \to bBY, \\
A \to ooC, & B \to dd, & B \to D, & C \to jVB, \\
C \to gi, & D \to n, & U \to kW, & V \to baXXX, \\
V \to oV, & W \to c, & X \to fV, & Y \to Yhm
\end{array}
$$

$V_t = \{B, C, D$

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \to gAe, & S \to aYB, & S \to cY, & A \to bBY, \\
A \to ooC, & B \to dd, & B \to D, & C \to jVB, \\
C \to gi, & D \to n, & U \to kW, & V \to baXXX, \\
V \to oV, & W \to c, & X \to fV, & Y \to Yhm
\end{array}
$$

$V_t = \{B, C, D$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe,\quad S \to aYB,\quad S \to cY,\quad A \to bBY,$$
$$A \to ooC,\quad B \to dd,\quad B \to D,\quad C \to jVB,$$
$$C \to gi,\quad D \to n,\quad U \to kW,\quad V \to baXXX,$$
$$V \to oV,\quad W \to c,\quad X \to fV,\quad Y \to Yhm$$

$V_t = \{B, C, D, W$

Un ejemplo del algoritmo para el paso 1

$$S \rightarrow gAe, \quad S \rightarrow aYB, \quad S \rightarrow cY, \quad A \rightarrow bBY,$$
$$A \rightarrow ooC, \quad B \rightarrow dd, \quad B \rightarrow D, \quad C \rightarrow jVB,$$
$$C \rightarrow gi, \quad D \rightarrow n, \quad U \rightarrow kW, \quad V \rightarrow baXXX,$$
$$V \rightarrow oV, \quad W \rightarrow c, \quad X \rightarrow fV, \quad Y \rightarrow Yhm$$

$V_t = \{B, C, D, W$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{B, C, D, W$

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \rightarrow gAe, & S \rightarrow aYB, & S \rightarrow cY, & A \rightarrow bBY, \\
A \rightarrow ooC, & B \rightarrow dd, & B \rightarrow D, & C \rightarrow jVB, \\
C \rightarrow gi, & D \rightarrow n, & U \rightarrow kW, & V \rightarrow baXXX, \\
V \rightarrow oV, & W \rightarrow c, & X \rightarrow fV, & Y \rightarrow Yhm
\end{array}
$$

$V_t = \{B, C, D, W$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{B, C, D, W$

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \to gAe, & S \to aYB, & S \to cY, & A \to bBY, \\
A \to ooC, & B \to dd, & B \to D, & C \to jVB, \\
C \to gi, & D \to n, & U \to kW, & V \to baXXX, \\
V \to oV, & W \to c, & X \to fV, & Y \to Yhm
\end{array}
$$

$V_t = \{B, C, D, W$

Un ejemplo del algoritmo para el paso 1

$$S \rightarrow gAe, \quad S \rightarrow aYB, \quad S \rightarrow cY, \quad A \rightarrow bBY,$$
$$A \rightarrow ooC, \quad B \rightarrow dd, \quad B \rightarrow D, \quad C \rightarrow jVB,$$
$$C \rightarrow gi, \quad D \rightarrow n, \quad U \rightarrow kW, \quad V \rightarrow baXXX,$$
$$V \rightarrow oV, \quad W \rightarrow c, \quad X \rightarrow fV, \quad Y \rightarrow Yhm$$

$V_t = \{B, C, D, W, A$

Un ejemplo del algoritmo para el paso 1

$S \rightarrow gAe,$	$S \rightarrow aYB,$	$S \rightarrow cY,$	$A \rightarrow bBY,$
$A \rightarrow ooC,$	$B \rightarrow dd,$	$B \rightarrow D,$	$C \rightarrow jVB,$
$C \rightarrow gi,$	$D \rightarrow n,$	$U \rightarrow kW,$	$V \rightarrow baXXX,$
$V \rightarrow oV,$	$W \rightarrow c,$	$X \rightarrow fV,$	$Y \rightarrow Yhm$

$V_t = \{B, C, D, W, A$

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \to gAe, & S \to aYB, & S \to cY, & A \to bBY, \\
A \to ooC, & B \to dd, & B \to D, & C \to jVB, \\
C \to gi, & D \to n, & U \to kW, & V \to baXXX, \\
V \to oV, & W \to c, & X \to fV, & Y \to Yhm
\end{array}
$$

$V_t = \{B, C, D, W, A, U$

Un ejemplo del algoritmo para el paso 1

$$\begin{array}{llll}
S \to gAe, & S \to aYB, & S \to cY, & A \to bBY, \\
A \to ooC, & B \to dd, & B \to D, & C \to jVB, \\
C \to gi, & D \to n, & U \to kW, & V \to baXXX, \\
V \to oV, & W \to c, & X \to fV, & Y \to Yhm
\end{array}$$

$V_t = \{B, C, D, W, A, U$

Un ejemplo del algoritmo para el paso 1

$$S \rightarrow gAe, \quad S \rightarrow aYB, \quad S \rightarrow cY, \quad A \rightarrow bBY,$$
$$A \rightarrow ooC, \quad B \rightarrow dd, \quad B \rightarrow D, \quad C \rightarrow jVB,$$
$$C \rightarrow gi, \quad D \rightarrow n, \quad U \rightarrow kW, \quad V \rightarrow baXXX,$$
$$V \rightarrow oV, \quad W \rightarrow c, \quad X \rightarrow fV, \quad Y \rightarrow Yhm$$

$V_t = \{B, C, D, W, A, U$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{B, C, D, W, A, U$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{B, C, D, W, A, U$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$$V_t = \{B, C, D, W, A, U$$

Un ejemplo del algoritmo para el paso 1

$$S \rightarrow gAe, \quad S \rightarrow aYB, \quad S \rightarrow cY, \quad A \rightarrow bBY,$$
$$A \rightarrow ooC, \quad B \rightarrow dd, \quad B \rightarrow D, \quad C \rightarrow jVB,$$
$$C \rightarrow gi, \quad D \rightarrow n, \quad U \rightarrow kW, \quad V \rightarrow baXXX,$$
$$V \rightarrow oV, \quad W \rightarrow c, \quad X \rightarrow fV, \quad Y \rightarrow Yhm$$

$V_t = \{B, C, D, W, A, U, S$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{B, C, D, W, A, U, S$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$

$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$

$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$

$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$$V_t = \{B, C, D, W, A, U, S$$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{B, C, D, W, A, U, S$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe,\quad S \to aYB,\quad S \to cY,\quad A \to bBY,$$
$$A \to ooC,\quad B \to dd,\quad B \to D,\quad C \to jVB,$$
$$C \to gi,\quad D \to n,\quad U \to kW,\quad V \to baXXX,$$
$$V \to oV,\quad W \to c,\quad X \to fV,\quad Y \to Yhm$$

$$V_t = \{B, C, D, W, A, U, S$$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$V_t = \{B, C, D, W, A, U, S$

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \to gAe, & S \to aYB, & S \to cY, & A \to bBY, \\
A \to ooC, & B \to dd, & B \to D, & C \to jVB, \\
C \to gi, & D \to n, & U \to kW, & V \to baXXX, \\
V \to oV, & W \to c, & X \to fV, & Y \to Yhm
\end{array}
$$

$V_t = \{B, C, D, W, A, U, S$

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \to gAe, & S \to aYB, & S \to cY, & A \to bBY, \\
A \to ooC, & B \to dd, & B \to D, & C \to jVB, \\
C \to gi, & D \to n, & U \to kW, & V \to baXXX, \\
V \to oV, & W \to c, & X \to fV, & Y \to Yhm
\end{array}
$$

$V_t = \{B, C, D, W, A, U, S$

Un ejemplo del algoritmo para el paso 1

$$
\begin{array}{llll}
S \to gAe, & S \to aYB, & S \to cY, & A \to bBY, \\
A \to ooC, & B \to dd, & B \to D, & C \to jVB, \\
C \to gi, & D \to n, & U \to kW, & V \to baXXX, \\
V \to oV, & W \to c, & X \to fV, & Y \to Yhm
\end{array}
$$

$V_t = \{B, C, D, W, A, U, S$

Un ejemplo del algoritmo para el paso 1

$$S \to gAe, \quad S \to aYB, \quad S \to cY, \quad A \to bBY,$$
$$A \to ooC, \quad B \to dd, \quad B \to D, \quad C \to jVB,$$
$$C \to gi, \quad D \to n, \quad U \to kW, \quad V \to baXXX,$$
$$V \to oV, \quad W \to c, \quad X \to fV, \quad Y \to Yhm$$

$$V_t = \{B, C, D, W, A, U, S\}, \quad V - V_t = \{V, X, Y\}$$

$$S \to gAe,$$
$$A \to ooC, \quad B \to dd, \quad B \to D,$$
$$C \to gi, \quad D \to n, \quad U \to kW,$$
$$W \to c,$$

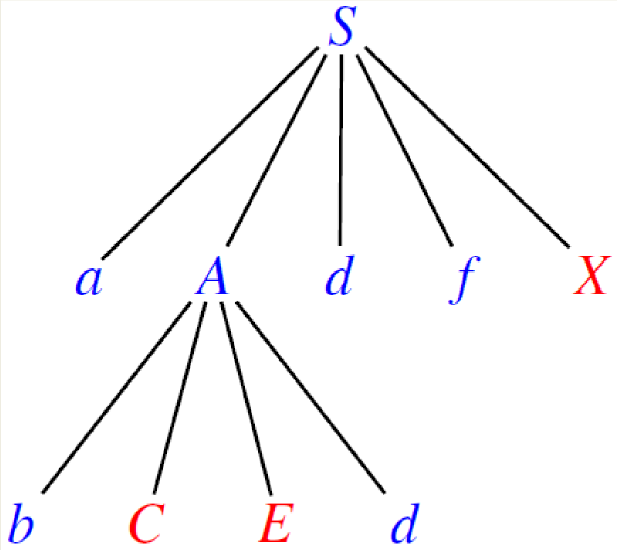
Idea del algoritmo para el segundo paso

Realizamos una búsqueda recursiva a partir del símbolo inicial S de todos los símbolos que se pueden alcanzar a partir de el.

Ejemplo

$$S \rightarrow aAd;\ S \rightarrow fX;\ A \rightarrow b;\ A \rightarrow CEd$$

$\boldsymbol{V_S}$ variables obtenidas
$\boldsymbol{T_S}$ símbolos terminales obtenidos
$\boldsymbol{J}$ variables por analizar



Algoritmo para el segundo paso

V_S y J son conjuntos de variables.
T_S es un conjunto de símbolos terminales
1. $J = \{S\}$, $V_S = \{S\}$, $T_S = \emptyset$
2. Mientras J != Ø
    3. Extraer un elemento de J : A, ($J = J$-{A}).
    4. Para cada producción de la forma $A \rightarrow \alpha$
    5.Para cada variable B en α
        6. Si B no está en V_S añadir B a J y a V_S
    7. Poner todos los símbolos terminales de α en T_S
8. Eliminar todas las variables que no estén en V_S y todos los símbolos terminales que no estén en T_S.
9. Eliminar todas las producciones donde aparezca un símbolo o variable de los eliminados en el paso anterior.

Un ejemplo del algoritmo para el paso 2

$$
\begin{array}{lll}
S \rightarrow gAe, & A \rightarrow ooC, & \\
B \rightarrow dd, & B \rightarrow D, & C \rightarrow gi, \\
D \rightarrow n, & U \rightarrow kW, & W \rightarrow c
\end{array}
$$

Un ejemplo del algoritmo para el paso 2

$$S \to gAe, \quad A \to ooC,$$
$$B \to dd, \quad B \to D, \quad C \to gi,$$
$$D \to n, \quad U \to kW, \quad W \to c$$

$V_S = \{S$

$J = \{S$

$T_S = \{$

Variable analizada:

Un ejemplo del algoritmo para el paso 2

$$S \to gAe, \quad A \to ooC,$$
$$B \to dd, \quad B \to D, \quad C \to gi,$$
$$D \to n, \quad U \to kW, \quad W \to c$$

$$V_S = \{S$$

$$J = \{$$

$$T_S = \{$$

Variable analizada: S

Un ejemplo del algoritmo para el paso 2

$$S \rightarrow gAe, \quad A \rightarrow ooC,$$
$$B \rightarrow dd, \quad B \rightarrow D, \quad C \rightarrow gi,$$
$$D \rightarrow n, \quad U \rightarrow kW, \quad W \rightarrow c$$

$$V_S = \{S, A$$

$$J = \{A$$

$$T_S = \{g, e$$

Variable analizada:

Un ejemplo del algoritmo para el paso 2

$$S \to gAe, \quad A \to ooC,$$
$$B \to dd, \quad B \to D, \quad C \to gi,$$
$$D \to n, \quad U \to kW, \quad W \to c$$

$$V_S = \{S, A$$

$$J = \{$$

$$T_S = \{g, e$$

Variable analizada: A

Un ejemplo del algoritmo para el paso 2

$$S \to gAe, \quad A \to ooC,$$
$$B \to dd, \quad B \to D, \quad C \to gi,$$
$$D \to n, \quad U \to kW, \quad W \to c$$

$$V_S = \{S, A, C$$

$$J = \{C$$

$$T_S = \{g, e, o$$

Variable analizada:

Un ejemplo del algoritmo para el paso 2

$$S \to gAe, \quad A \to ooC,$$
$$B \to dd, \quad B \to D, \quad C \to gi,$$
$$D \to n, \quad U \to kW, \quad W \to c$$

$$V_S = \{S, A, C$$

$$J = \{$$

$$T_S = \{g, e, o$$

Variable analizada: C

Un ejemplo del algoritmo para el paso 2

$$S \to gAe, \quad A \to ooC,$$
$$B \to dd, \quad B \to D, \quad C \to gi,$$
$$D \to n, \quad U \to kW, \quad W \to c$$

$$V_S = \{S, A, C\}$$

$$J = \{\}$$

$$T_S = \{g, e, o, i\}$$

Variable analizada:

Un ejemplo del algoritmo para el paso 2

$S \rightarrow gAe,\quad A \rightarrow ooC,$

$B \cancel{\rightarrow} dd,\quad B \cancel{\rightarrow} D,\quad C \rightarrow gi,$

$D \cancel{\rightarrow} n,\quad U \cancel{\rightarrow} kW,\quad W \cancel{\rightarrow} c$

$$V_S = \{S, A, C\}$$

$$J = \{\}$$

$$T_S = \{g, e, o, i\}$$

Variable analizada:

Un ejemplo del algoritmo para el paso 2

$$S \rightarrow gAe, \quad A \rightarrow ooC,$$
$$B \cancel{\rightarrow} dd, \quad B \cancel{\rightarrow} D, \quad C \rightarrow gi,$$
$$D \cancel{\rightarrow} n, \quad U \cancel{\rightarrow} kW, \quad W \cancel{\rightarrow} c$$

$$V_S = \{S, A, C\}$$

$$J = \{\}$$

$$T_S = \{g, e, o, i\}$$

Única derivación posible: $S \Rightarrow gAe \Rightarrow gooCe \Rightarrow googie$

Lenguaje generado: $\{googie\}$.

Lenguaje vacío

Si el lenguaje generado por una gramática es vacío, esto se detecta en que la variable S resulta inútil en el primer algoritmo. En ese caso se pueden eliminar directamente todas las producciones, pero no el símbolo S.

Un ejemplo

$$S \to aSb, \quad S \to bcD, \quad S \to cSE,$$
$$E \to aDb, \quad F \to abc, \quad E \to abF$$

$$V_t = \{F, E\}, \qquad L(G) = \emptyset$$

Algoritmo para eliminar producciones nulas

Algoritmo que dada una gramática G, construye una gramática G_n sin producciones nulas y tal que $L(G_n) = L(G) \setminus \{\varepsilon\}$.

Un ejemplo

Si tenemos, $A \rightarrow \varepsilon$; $D \rightarrow aABc$; entonces:

$$D \Rightarrow aABc \Rightarrow aBc$$

Insertamos D→aBc

Si también hubiese B→ ε, entonces también habría que añadir:

D→aBc; D→aAc; D→ac

Variables anulables

Si tenemos $C \rightarrow AB$; $A \rightarrow \varepsilon$; $B \rightarrow \varepsilon$, habría que añadir, al quitar las producciones nulas: $C \rightarrow A$; $C \rightarrow B$; $C \rightarrow \varepsilon$
Y después habría que eliminar $C \rightarrow \varepsilon$

Nosotros vamos a calcular desde el principio todas las variables, E, tales que en algún momento aparece $E \rightarrow \varepsilon$. Estas variables se dicen **anulables**. Son variables tales que $E \overset{*}{\Rightarrow} \varepsilon$

Después vamos a eliminar todas las producciones nulas y a añadir las producciones necesarias para compensar esta eliminación.

Algoritmo para cálculo de variables anulables

H es el conjunto de las variables anulables.

Cálculo de Variables Anulables:

1. $H = \emptyset$
2. Para cada producción $A \rightarrow \varepsilon$, se hace $H = H \cup \{A\}$
3. Mientras H cambie
   4. Para cada producción $B \rightarrow A_1 A_2 \ldots A_n$, donde $A_i \in H$ para todo $i = 1 \ldots n$, se hace $H = H \cup \{B\}$

Eliminar y añadir:

5. Se eliminan todas las producciones nulas de la gramática
6. Para cada producción de la gramática de la forma $A \rightarrow \alpha_1\ \alpha_2 \ldots \alpha_n$; donde $\alpha_i \in (V \cup T)$
   7. Se añaden todas las producciones de la forma $A \rightarrow \beta_1 \ldots \beta_n$ donde $\beta_i = \alpha_i$ si $\alpha_i \notin H$, y $(\beta_i = \alpha_i) \vee (\beta_i = \varepsilon)$ *si* $\alpha_i \in H$ y no todos los β_i puedan ser nulos al mismo tiempo.

Consecuencias

- Si G generaba inicialmente la palabra nula, entonces la nueva gramática no la genera.
- Se puede saber si se pierde la palabra vacía comprobando si $S \in H$.
- Si tenemos una gramática G, podemos construir una gramática G_v con una sola producción nula y que genera el mismo lenguaje que G más la palabra vacía. Para ello se añade una nueva variable, S_v, que pasa a ser el símbolo inicial de la nueva gramática, G_v. También se añaden dos producciones: $S_v \rightarrow S$; $S_v \rightarrow \varepsilon$

Un ejemplo: Paso 1

Eliminar las producciones nulas de la siguiente gramática:

$$S \longrightarrow ABb, \quad S \longrightarrow ABC, \quad C \longrightarrow abC,$$
$$B \longrightarrow bB, \quad B \longrightarrow \varepsilon, \quad A \longrightarrow aA,$$
$$A \longrightarrow \varepsilon, \quad C \longrightarrow AB$$

Cálculo de $H = \{$

Un ejemplo: Paso 1

Eliminar las producciones nulas de la siguiente gramática:

$$S \rightarrow ABb, \quad S \rightarrow ABC, \quad C \rightarrow abC,$$
$$B \rightarrow bB, \quad B \rightarrow \varepsilon, \quad A \rightarrow aA,$$
$$A \rightarrow \varepsilon, \quad C \rightarrow AB$$

Cálculo de $H = \{A, B$

Un ejemplo: Paso 1

Eliminar las producciones nulas de la siguiente gramática:

$$S \to ABb, \quad S \to ABC, \quad C \to abC,$$
$$B \to bB, \quad B \to \varepsilon, \quad A \to aA,$$
$$A \to \varepsilon, \quad C \to AB$$

Cálculo de $H = \{A, B, C$

Un ejemplo: Paso 1

Eliminar las producciones nulas de la siguiente gramática:

$$S \to ABb,\quad S \to ABC,\quad C \to abC,$$
$$B \to bB,\quad B \to \varepsilon,\quad A \to aA,$$
$$A \to \varepsilon,\quad C \to AB$$

Cálculo de $H = \{A, B\ , C\ , S$

Un ejemplo: Paso 1

Eliminar las producciones nulas de la siguiente gramática:

$$S \to ABb, \quad S \to ABC, \quad C \to abC,$$
$$B \to bB, \quad \cancel{B \to \varepsilon}, \quad A \to aA,$$
$$\cancel{A \to \varepsilon}, \quad C \to AB$$

Cálculo de $H = \{A, B, C, S\}$.

Al ser *S* anulable, la palabra vacía puede generarse a través de esta gramática.

Un ejemplo: Paso 2

$$S \to ABb, \quad S \to ABC, \quad C \to abC, \quad B \to bB,$$
$$A \to aA, \quad C \to AB,$$

$$H = \{A, B, C, S\}$$

Un ejemplo: Paso 2

$S \rightarrow ABb$, $S \rightarrow ABC$, $C \rightarrow abC$, $B \rightarrow bB$,
$A \rightarrow aA$, $C \rightarrow AB$, $S \rightarrow Ab$, $S \rightarrow Bb$,
$S \rightarrow b$,

$$H = \{A, B, C, S\}$$

Un ejemplo: Paso 2

$$S \rightarrow ABb, \quad S \rightarrow ABC, \quad C \rightarrow abC, \quad B \rightarrow bB,$$
$$A \rightarrow aA, \quad C \rightarrow AB, \quad S \rightarrow Ab, \quad S \rightarrow Bb,$$
$$S \rightarrow b, \quad S \rightarrow AB, \quad S \rightarrow AC, \quad S \rightarrow BC,$$
$$S \rightarrow A, \quad S \rightarrow B, \quad S \rightarrow C,$$

$$H = \{A, B, C, S\}$$

Un ejemplo: Paso 2

$$S \to ABb, \quad S \to ABC, \quad C \to abC, \quad B \to bB,$$
$$A \to aA, \quad C \to AB, \quad S \to Ab, \quad S \to Bb,$$
$$S \to b, \quad S \to AB, \quad S \to AC, \quad S \to BC,$$
$$S \to A, \quad S \to B, \quad S \to C, \quad C \to ab,$$

$$H = \{A, B, C, S\}$$

Un ejemplo: Paso 2

$$S \to ABb, \quad S \to ABC, \quad C \to abC, \quad B \to bB,$$
$$A \to aA, \quad C \to AB, \quad S \to Ab, \quad S \to Bb,$$
$$S \to b, \quad S \to AB, \quad S \to AC, \quad S \to BC,$$
$$S \to A, \quad S \to B, \quad S \to C, \quad C \to ab,$$
$$B \to b,$$

$$H = \{A, B, C, S\}$$

Un ejemplo: Paso 2

$S \to ABb,$	$S \to ABC,$	$C \to abC,$	$B \to bB,$
$A \to aA,$	$C \to AB,$	$S \to Ab,$	$S \to Bb,$
$S \to b,$	$S \to AB,$	$S \to AC,$	$S \to BC,$
$S \to A,$	$S \to B,$	$S \to C,$	$C \to ab,$
$B \to b,$	$A \to a,$		

$$H = \{A, B, C, S\}$$

Un ejemplo: Paso 2

$S \rightarrow ABb$, $S \rightarrow ABC$, $C \rightarrow abC$, $B \rightarrow bB$,

$A \rightarrow aA$, $C \rightarrow AB$, $S \rightarrow Ab$, $S \rightarrow Bb$,

$S \rightarrow b$, $S \rightarrow AB$, $S \rightarrow AC$, $S \rightarrow BC$,

$S \rightarrow A$, $S \rightarrow B$, $S \rightarrow C$, $C \rightarrow ab$,

$B \rightarrow b$, $A \rightarrow a$, $C \rightarrow B$, $C \rightarrow A$,

$$H = \{A, B, C, S\}$$

Un ejemplo: Paso 2

$$S \to ABb, \quad S \to ABC, \quad C \to abC, \quad B \to bB,$$
$$A \to aA, \quad C \to AB, \quad S \to Ab, \quad S \to Bb,$$
$$S \to b, \quad S \to AB, \quad S \to AC, \quad S \to BC,$$
$$S \to A, \quad S \to B, \quad S \to C, \quad C \to ab,$$
$$B \to b, \quad A \to a, \quad C \to B, \quad C \to A,$$

$$H = \{A, B, C, S\}$$

UNIVERSIDAD DE GRANADA

Gramáticas independientes del contexto

DECSAI

Producciones unitarias

Son producciones del tipo:

$$A \rightarrow B;\ A, B \text{ pertenecen a } V$$

Si queremos eliminar $A \rightarrow B$, perdemos la posibilidad de:

$$A \Rightarrow B \Rightarrow \alpha$$

Para eliminar $A \rightarrow B$, añadimos todas las producciones $A \rightarrow \alpha$ donde $B \rightarrow \alpha$ es una producción.

Si también tenemos $B \rightarrow C$ introduciríamos una unitaria que habría que eliminar después. Para poder eliminar todas de una vez calculamos:

***H*: conjunto de parejas (*A*,*B*) tales que *B* derivable a partir de *A*.**

Algoritmo para eliminar producciones unitarias

Se supone que no hay transiciones nulas

H conjunto de parejas (*A*;*B*) tales que *B* derivable a partir de *A*.

1. $H = \emptyset$
2. Para toda producción de la forma $A \rightarrow B$, la pareja (*A*;*B*) se introduce en *H*.
3. Mientras *H* cambie
    4. Para cada dos parejas (*A*,*B*); (*B*,*C*) en *H*, si la pareja (*A*,*C*) no esta en *H* entonces (*A*,*C*) se introduce en *H*
5. Se eliminan las producciones unitarias
6. Para cada pareja (*B*,*A*) en *H*
    7. Para cada producción $A \rightarrow \alpha$
        9. Se añade una producción $B \rightarrow \alpha$

Un ejemplo

$$
\begin{array}{lll}
S \longrightarrow aBc & S \longrightarrow A & A \longrightarrow aAb \\
A \longrightarrow B & A \longrightarrow cd & B \longrightarrow ccBS \\
B \longrightarrow dc & &
\end{array}
$$

Un ejemplo

$S \longrightarrow aBc \quad S \longrightarrow A \quad A \longrightarrow aAb$

$A \longrightarrow B \quad A \longrightarrow cd \quad B \longrightarrow ccBS$

$B \longrightarrow dc$

Cálculo de $H = \{(S,A),(A,B)$

Un ejemplo

$$S \longrightarrow aBc \quad S \longrightarrow A \quad A \longrightarrow aAb$$
$$A \longrightarrow B \quad A \longrightarrow cd \quad B \longrightarrow ccBS$$
$$B \longrightarrow dc$$

Cálculo de $H = \{(S,A),(A,B)$

Un ejemplo

$$S \to aBc \quad S \to A \quad A \to aAb$$
$$A \to B \quad A \to cd \quad B \to ccBS$$
$$B \to dc$$

Cálculo de $H = \{(S,A),(A,B),(S,B)$

Un ejemplo

$$S \to aBc \quad S \cancel{\to} A \quad A \to aAb$$
$$A \cancel{\to} B \quad A \to cd \quad B \to ccBS$$
$$B \to dc$$

Cálculo de $H = \{(S,A),(A,B),(S,B)\}$

Un ejemplo

$$S \longrightarrow aBc \quad A \longrightarrow aAb \quad A \longrightarrow cd$$
$$B \longrightarrow ccBS \quad B \longrightarrow dc$$

$$H = \{(S,A),(A,B),(S,B)\}$$

Un ejemplo

$$S \longrightarrow aBc \quad A \longrightarrow aAb \quad A \longrightarrow cd$$

$$B \longrightarrow ccBS \quad B \longrightarrow dc \quad S \longrightarrow aAb$$

$$S \longrightarrow cd$$

$$H = \{(S,A),(A,B),(S,B)\}$$

Un ejemplo

$$S \longrightarrow aBc \quad A \longrightarrow aAb \quad A \longrightarrow cd$$
$$B \longrightarrow ccBS \quad B \longrightarrow dc \quad S \longrightarrow aAb$$
$$S \longrightarrow cd \quad A \longrightarrow ccBS \quad A \longrightarrow dc$$

$$H = \{(S,A),(A,B),(S,B)\}$$

Un ejemplo

Gramática resultante:

$$
\begin{array}{lll}
S \to aBc & A \to aAb & A \to cd \\
B \to ccBS & B \to dc & S \to aAb \\
S \to cd & A \to ccBS & A \to dc \\
S \to ccBS & S \to dc &
\end{array}
$$

$$H = \{(S,A),(A,B),(S,B)\}$$

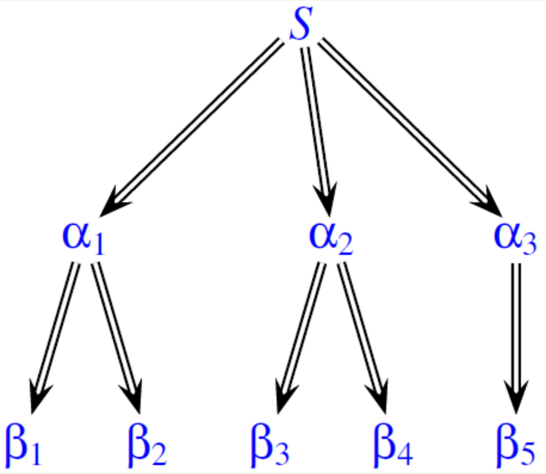
Gramáticas independientes del contexto

1. Árbol de derivación
2. Ambigüedad
3. Algoritmos de simplificación de gramáticas
    1. Símbolos y producciones inútiles
    2. Producciones nulas
    3. Producciones unitarias
4. Formas normales
    1. Forma normal de Chomsky
    2. Forma normal de Greibach

DECSAI

El problema de la pertenencia

Dada una gramática independiente del contexto G y una palabra u, ¿pertenece u a $L(G)$? Basta con generar el árbol de derivación (o árboles, en caso de ambigüedad) para la palabra.



Si la palabra es generada, nos sale en esta búsqueda. Si la palabra no es generada, ¿hasta **qué profundidad** tenemos que generar para convencernos de que no se puede?

Respuesta: $2n$-1 (en cada paso o sacamos al menos un nuevo símbolo terminal (n) o aumenta al menos en 1 la longitud (n-1))

DECSAI

La forma normal de Chomsky

Todas las producciones tienen la forma

$$A \rightarrow BC;\ A \rightarrow a$$

donde A,B,C son símbolos no terminales en V, y a es símbolo terminal en T.

El algoritmo se aplica a gramáticas sin producciones nulas ni unitarias.

Hay dos operaciones básicas:
Paso 1. Eliminar terminales en producciones que no sean $A \rightarrow a$
Paso 2. Eliminar producciones con una longitud en la parte derecha mayor de 2

Un ejemplo: Paso 1

Producciones iniciales:

$$B \rightarrow aBc$$
$$E \rightarrow aBEd$$

Paso del algoritmo:

$$\boxed{\begin{array}{l} B \cancel{\rightarrow} aBc \\ E \cancel{\rightarrow} aBEd \end{array}} \Rightarrow \boxed{\begin{array}{ll} C_a \rightarrow a & B \rightarrow C_aBC_c \\ C_c \rightarrow c & E \rightarrow C_aBEC_d \\ C_d \rightarrow d & \end{array}}$$

Un ejemplo: Paso 2

Producción inicial: $A \longrightarrow BEFG$

Paso del algoritmo:

$$\begin{tikzcd}[arrows=dash]
 & A \longrightarrow BEFG \arrow[dl] \arrow[dr] & & \\
A \longrightarrow BD_1 & & D_1 \longrightarrow EFG \arrow[dl] \arrow[dr] & \\
 & D_1 \longrightarrow ED_2 & & D_2 \longrightarrow FG
\end{tikzcd}$$

Algoritmo para pasar a forma normal de Chomsky

1. Para cada producción $A \rightarrow \alpha_1 \ldots \alpha_n$; $\alpha_i \in (V \cup T), n \geq 2$
    2. Para cada α_i , si $\alpha_i = a$ es terminal
        3. Se añade la produccion $C_a \rightarrow a$
        4. Se cambia α_i por C_a en $A \rightarrow \alpha_1 \ldots \alpha_n$
5. Para cada producción de la forma $A \rightarrow B_1 \ldots B_m$, $m>=3$
    6. Se añaden (m-2) variables $D_1 \ldots D_{m-2}$
        (distintas para cada producción)
    7. La produccion $A \rightarrow B_1 \ldots B_m$ se reemplaza por:
        $A \rightarrow B_1 D_1$; $D_1 \rightarrow B_2 D_2$;… ; $D_{m-2} \rightarrow B_{m-1} B_m$

Un ejemplo

Gramática inicial:

$$
\begin{array}{llll}
S \longrightarrow bA & S \longrightarrow aB & A \longrightarrow bAA & A \longrightarrow AS \\
A \longrightarrow a & B \longrightarrow aBB & B \longrightarrow bS & B \longrightarrow b
\end{array}
$$

Paso 1:

$$\Big\downarrow$$

$$
\begin{array}{llll}
S \longrightarrow C_bA & S \longrightarrow C_aB & A \longrightarrow C_bAA & A \longrightarrow AS \\
A \longrightarrow a & B \longrightarrow C_aBB & B \longrightarrow C_bS & B \longrightarrow b \\
C_a \longrightarrow a & C_b \longrightarrow b & &
\end{array}
$$

Un ejemplo (paso 2)

$$S \rightarrow C_b A \quad S \rightarrow C_a B \quad A \rightarrow C_b A A \quad A \rightarrow A S$$
$$A \rightarrow a \quad B \rightarrow C_a B B \quad B \rightarrow C_b S \quad B \rightarrow b$$
$$C_a \rightarrow a \quad C_b \rightarrow b$$

Un ejemplo (paso 2)

$S \longrightarrow C_bA$	$S \longrightarrow C_aB$	$A \longrightarrow C_bAA$	$A \longrightarrow AS$
$A \longrightarrow a$	$B \longrightarrow C_aBB$	$B \longrightarrow C_bS$	$B \longrightarrow b$
$C_a \longrightarrow a$	$C_b \longrightarrow b$	$A \longrightarrow C_bD_1$	$D_1 \longrightarrow AA$

$S \longrightarrow C_bA$	$S \longrightarrow C_aB$		$A \longrightarrow AS$
$A \longrightarrow a$	$B \longrightarrow C_aBB$	$B \longrightarrow C_bS$	$B \longrightarrow b$
$C_a \longrightarrow a$	$C_b \longrightarrow b$	$A \longrightarrow C_bD_1$	$D_1 \longrightarrow AA$

Un ejemplo (paso 2)

$S \rightarrow C_bA$	$S \rightarrow C_aB$		$A \rightarrow AS$
$A \rightarrow a$	$B \rightarrow C_aBB$	$B \rightarrow C_bS$	$B \rightarrow b$
$C_a \rightarrow a$	$C_b \rightarrow b$	$A \rightarrow C_bD_1$	$D_1 \rightarrow AA$
$B \rightarrow C_aE_1$	$E_1 \rightarrow BB$		

$S \rightarrow C_bA$	$S \rightarrow C_aB$		$A \rightarrow AS$
$A \rightarrow a$		$B \rightarrow C_bS$	$B \rightarrow b$
$C_a \rightarrow a$	$C_b \rightarrow b$	$A \rightarrow C_bD_1$	$D_1 \rightarrow AA$
$B \rightarrow C_aE_1$	$E_1 \rightarrow BB$		

DECSAI

La forma normal de Greibach

Todas las producciones tienen la forma

$A \rightarrow a\alpha$; *donde a es terminal y* α *está en V**

donde *A*,*B*,*C* son símbolos no terminales en *V*, y *a* es símbolo terminal en *T*.

Condiciones que se deben cumplir para aplicar el algoritmo para pasar a forma normal de Greibach:

$$A \rightarrow a\alpha, a \in T, \alpha \in V^*$$
$$A \rightarrow \alpha, \alpha \in V^*, |\alpha| \geq 2$$

(se cumplen si la gramática ya está en forma normal de Chomsky)

El algoritmo contempla básicamente dos operaciones (I)

Suponiendo: $A \rightarrow B\alpha \quad B \rightarrow \beta_1 \quad B \rightarrow \beta_2 \quad B \rightarrow \beta_3$

La idea es concentrar las reglas en una:

$$A \Rightarrow B\alpha \Rightarrow \beta_2\alpha$$

ELIMINA$_1$($A \rightarrow B\alpha$)

1. Eliminar $A \rightarrow B\alpha$
2. Para cada produccion $B \rightarrow \beta$
   3. Añadir $A \rightarrow \beta\alpha$

Nos queda:

$$A \rightarrow \beta_1\alpha \quad A \rightarrow \beta_2\alpha \quad A \rightarrow \beta_3\alpha$$
$$B \rightarrow \beta_1 \quad B \rightarrow \beta_2 \quad B \rightarrow \beta_3$$

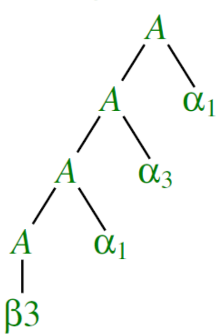
El algoritmo contempla básicamente dos operaciones (II)

ELIMINA$_2(A)$

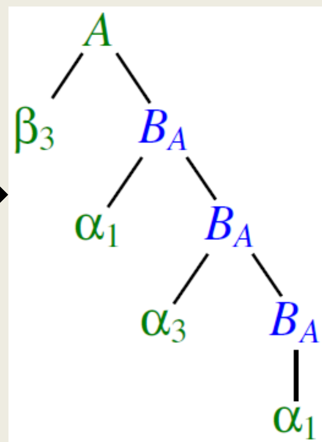
1. Añadir una nueva variable B_A
2. Para cada produccion $A \longrightarrow A\alpha$
    3. Añadir $B_A \longrightarrow \alpha$ y $B_A \longrightarrow \alpha B_A$
    4. Eliminar $A \longrightarrow A\alpha$
5. Para cada produccion $A \longrightarrow \beta$, β no empieza por A
    6. Añadir $A \longrightarrow \beta B_A$.

$A \longrightarrow A\alpha_1 \quad A \longrightarrow A\alpha_2 \quad A \longrightarrow A\alpha_3$

$A \longrightarrow \beta_1 \quad A \longrightarrow \beta_2 \quad A \longrightarrow \beta_3$



→



Queda:

$A \longrightarrow \beta_1$	$A \longrightarrow \beta_2$	$A \longrightarrow \beta_3$
$A \longrightarrow \beta_1 B_A$	$A \longrightarrow \beta_2 B_A$	$A \longrightarrow \beta_3 B_A$
$B_A \longrightarrow \alpha_1$	$B_A \longrightarrow \alpha_2$	$B_A \longrightarrow \alpha_3$
$B_A \longrightarrow \alpha_1 B_A$	$B_A \longrightarrow \alpha_2 B_A$	$B_A \longrightarrow \alpha_3 B_A$

Algoritmo para pasar a forma normal de Greibach

Objetivo: que todas las producciones sean de la forma:

$$A \rightarrow a\alpha, \quad a \in T, \alpha \in V^*.$$
$$A_i \rightarrow A_j\alpha, \quad j > i, \alpha \in V^*.$$
$$B_j \rightarrow A_i\alpha, \quad \alpha \in V^*$$

Donde B_k sería la variable resultante al eliminar A_k con la operación ELIMINA$_2$(A_k).

Algoritmo para pasar a forma normal de Greibach

PASO 1:

1. Para cada $k = 1\ldots m$
    2. Para cada j = 1...k-1
        3. Para cada producción $A_k \rightarrow A_j\alpha$
            4. ELIMINA$_1$($A_k \rightarrow A_j\alpha$)
    5. Si existe alguna producción de la forma $A_k \rightarrow A_k\alpha$
        6. ELIMINA$_2$(A_k)

PASO 2:

1. Para cada $i = m$-1...1
    2. Para cada producción de la forma $A_i \rightarrow A_j\alpha$; $j > i$
        3. ELIMINA$_1$($A_i \rightarrow A_j\alpha$)
4. Para cada i = 1...m
    5. Para cada producción de la forma $B_j \rightarrow A_i\alpha$,
        6. ELIMINA$_1$($B_j \rightarrow A_i\alpha$)

Un ejemplo

$$\underline{A_1 \rightarrow A_2A_3}, \quad A_2 \rightarrow A_3A_1, \quad A_2 \rightarrow b,$$

$$\boxed{A_3 \rightarrow A_1A_2}, \quad A_3 \rightarrow a$$

Aplicamos ELIMINA$_1$ a $A_3 \rightarrow A_1A_2$
Se elimina esta producción y se añade: $A_3 \rightarrow A_2A_3A_2$
Queda:

$$A_1 \rightarrow A_2A_3, \quad A_2 \rightarrow A_3A_1, \quad A_2 \rightarrow b,$$

$$A_3 \rightarrow a, \quad A_3 \rightarrow A_2A_3A_2$$

Un ejemplo

$$A_1 \rightarrow A_2A_3, \quad \underline{A_2 \rightarrow A_3A_1}, \quad \underline{A_2 \rightarrow b},$$

$$A_3 \rightarrow a, \quad \boxed{A_3 \rightarrow A_2A_3A_2}$$

Aplicamos ELIMINA$_1$ a $A_3 \rightarrow A_2A_3A_2$
Se elimina esta producción y se añaden:
$A_3 \rightarrow A_3A_1A_3A_2, \quad A_3 \rightarrow bA_3A_2$
Queda:

$$A_1 \rightarrow A_2A_3, \quad A_2 \rightarrow A_3A_1, \quad A_2 \rightarrow b,$$

$$A_3 \rightarrow a, \quad A_3 \rightarrow A_3A_1A_3A_2, \quad A_3 \rightarrow bA_3A_2$$

Un ejemplo

$$A_1 \to A_2A_3, \quad A_2 \to A_3A_1, \quad A_2 \to b, \quad A_3 \to a,$$
$$\boxed{A_3 \to A_3A_1A_3A_2}, \quad A_3 \to bA_3A_2$$

Aplicamos ELIMINA$_2$ a A_3

Se añade B_3 y las producciones $B_3 \to A_1A_3A_2, \quad B_3 \to A_1A_3A_2B_3$

Se elimina $A_3 \to A_3A_1A_3A_2$.

Se añaden las producciones: $A_3 \to aB_3, \quad A_3 \to bA_3A_2B_3$

Queda:

$$A_1 \to A_2A_3, \quad A_2 \to A_3A_1, \quad A_2 \to b, \quad A_3 \to a,$$
$$A_3 \to bA_3A_2 \quad B_3 \to A_1A_3A_2, \quad B_3 \to A_1A_3A_2B_3 \quad A_3 \to aB_3,$$
$$A_3 \to bA_3A_2B_3$$

Un ejemplo

$$A_1 \to A_2A_3, \quad \boxed{A_2 \to A_3A_1}, \quad A_2 \to b, \quad \underline{A_3 \to a}, \quad \underline{A_3 \to bA_3A_2} \quad B_3 \to A_1A_3A_2, \quad B_3 \to A_1A_3A_2B_3 \quad \underline{A_3 \to aB_3}, \quad \underline{A_3 \to bA_3A_2B_3}$$

Se aplica ELIMINA$_1$ a $A_2 \to A_3A_1$
Se elimina esta producción y se añaden:

$$A_2 \to aA_1, \quad A_2 \to aB_3A_1, \quad A_2 \to bA_3A_2B_3A_1, \quad A_2 \to bA_3A_2A_1$$

Un ejemplo

$$\boxed{A_1 \to A_2A_3}, \quad \underline{A_2 \to b}, \quad \underline{A_2 \to aA_1}, \quad \underline{A_2 \to aB_3A_1},$$
$$\underline{A_2 \to bA_3A_2B_3A_1}, \quad \underline{A_2 \to bA_3A_2A_1}, \quad A_3 \to a, \quad A_3 \to bA_3A_2,$$
$$B_3 \to A_1A_3A_2, \quad B_3 \to A_1A_3A_2B_3 \quad A_3 \to aB_3, \quad A_3 \to bA_3A_2B_3$$

Se aplica ELIMINA$_1$ a $A_1 \to A_2A_3$:

Se elimina esta producción y se añaden:

$$A_1 \to bA_3, \quad A_1 \to aA_1A_3, \quad A_1 \to aB_3A_1A_3,$$

$$A_1 \to bA_3A_2B_3A_1A_3, \quad A_1 \to bA_3A_2A_1A_3$$

Un ejemplo

$$A_2 \to b, \quad A_2 \to aA_1 \quad A_2 \to aB_3A_1,$$
$$A_2 \to bA_3A_2B_3A_1, \quad A_2 \to bA_3A_2A_1, \quad A_3 \to a,$$
$$A_3 \to bA_3A_2, \quad \boxed{B_3 \to A_1A_3A_2}, \quad B_3 \to A_1A_3A_2B_3,$$
$$A_3 \to aB_3, \quad A_3 \to bA_3A_2B_3, \quad \underline{A_1 \to bA_3},$$
$$\underline{A_1 \to aA_1A_3}, \quad \underline{A_1 \to aB_3A_1A_3}, \quad \underline{A_1 \to bA_3A_2B_3A_1A_3},$$
$$\underline{A_1 \to bA_3A_2A_1A_3}$$

Se aplica ELIMINA$_1$ a $B_3 \to A_1A_3A_2$ Se elimina esta producción y se añaden:

$$B_3 \to bA_3A_3A_2, \quad B_3 \to aA_1A_3A_3A_2, \quad B_3 \to aB_3A_1A_3A_3A_2,$$
$$B_3 \to bA_3A_2B_3A_1A_3A_3A_2, \quad B_3 \to aB_3A_1A_3A_3A_2,$$

Un ejemplo

$$A_2 \rightarrow b, \quad A_2 \rightarrow aA_1 \quad A_2 \rightarrow aB_3A_1,$$
$$A_2 \rightarrow bA_3A_2B_3A_1, \quad A_2 \rightarrow bA_3A_2A_1 \quad A_3 \rightarrow a,$$
$$A_3 \rightarrow bA_3A_2, \quad \boxed{B_3 \rightarrow A_1A_3A_2B_3}, \quad A_3 \rightarrow aB_3,$$
$$A_3 \rightarrow bA_3A_2B_3, \quad \underline{A_1 \rightarrow bA_3}, \quad \underline{A_1 \rightarrow aA_1A_3},$$
$$\underline{A_1 \rightarrow aB_3A_1A_3}, \quad \underline{A_1 \rightarrow bA_3A_2B_3A_1A_3}, \quad \underline{A_1 \rightarrow bA_3A_2A_1A_3}$$
$$B_3 \rightarrow bA_3A_3A_2, \quad B_3 \rightarrow aA_1A_3A_3A_2, \quad B_3 \rightarrow aB_3A_1A_3A_3A_2,$$
$$B_3 \rightarrow bA_3A_2B_3A_1A_3A_3A_2, \quad B_3 \rightarrow aB_3A_1A_3A_3A_2,$$

Se aplica ELIMINA$_1$ a $B_3 \rightarrow A_1A_3A_2B_3$. Se elimina esta producción y se añaden:

$$B_3 \rightarrow bA_3A_3A_2B_3, \quad B_3 \rightarrow aA_1A_3A_3A_2B_3, \quad B_3 \rightarrow aB_3A_1A_3A_3A_2B_3,$$
$$B_3 \rightarrow bA_3A_2B_3A_1A_3A_3A_2B_3, \quad B_3 \rightarrow aB_3A_1A_3A_3A_2B_3,$$

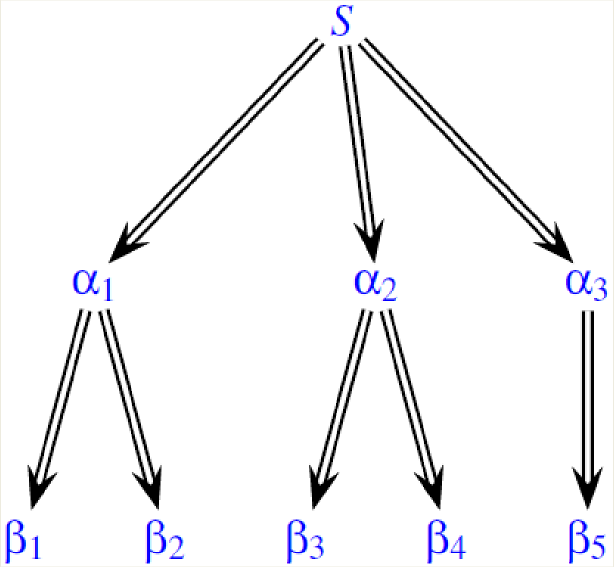
Un ejemplo

Resultado:

$$
\begin{array}{lll}
A_2 \rightarrow b, & A_2 \rightarrow aA_1, & A_2 \rightarrow aB_3A_1, \\
A_2 \rightarrow bA_3A_2B_3A_1, & A_2 \rightarrow bA_3A_2A_1 & A_3 \rightarrow a, \\
A_3 \rightarrow bA_3A_2, & A_3 \rightarrow aB_3, & A_3 \rightarrow bA_3A_2B_3, \\
A_1 \rightarrow bA_3, & A_1 \rightarrow aA_1A_3, & A_1 \rightarrow aB_3A_1A_3, \\
A_1 \rightarrow bA_3A_2B_3A_1A_3, & A_1 \rightarrow bA_3A_2A_1A_3, & B_3 \rightarrow bA_3A_3A_2, \\
B_3 \rightarrow aA_1A_3A_3A_2, & B_3 \rightarrow aB_3A_1A_3A_3A_2, & B_3 \rightarrow bA_3A_2B_3A_1A_3A_3A_2, \\
B_3 \rightarrow aB_3A_1A_3A_3A_2, & B_3 \rightarrow bA_3A_3A_2B_3, & B_3 \rightarrow aA_1A_3A_3A_2B_3, \\
B_3 \rightarrow aB_3A_1A_3A_3A_2B_3, & B_3 \rightarrow bA_3A_2B_3A_1A_3A_3A_2B_3, & B_3 \rightarrow aB_3A_1A_3A_3A_2B_3
\end{array}
$$

Método para el problema de la pertenencia

Dada G en forma normal de Greibach y u de longitud n, ¿pertenece u a $L(G)$?



Si la palabra no es generada, ¿Hasta qué profundidad tenemos que generar para convencernos de que no se puede? n (en cada paso sacamos al menos un nuevo símbolo terminal (n)).

UNIVERSIDAD DE GRANADA

DECSAI

Modelos de Computación

Grado en Ingeniería Informática

Tema 4 – Gramáticas independientes del contexto

Manuel Pegalajar Cuéllar
manupc@ugr.es

Departamento de Ciencias de la Computación e Inteligencia Artificial
http://decsai.ugr.es