Computación y Estructuras Discretas III

Andrés A. Aristizábal P.
aaaristizabal@icesi.edu.co

Departamento de Computación y Sistemas Inteligentes

2023-2

Agenda del día

1. **Lenguajes y gramáticas independientes de contexto**
   - Presentación del tema
   - Ejercicios

1. **Lenguajes y gramáticas independientes de contexto**
   - Presentación del tema
   - Ejercicios

Lenguajes y gramáticas independientes de contexto

¿Cuándo está una GIC G en Forma Normal de Greibach (FNG)?

Lenguajes y gramáticas independientes de contexto

¿Cuándo está una GIC *G* en Forma Normal de Greibach (FNG)?

1. La variable inicial no es recursiva.

Lenguajes y gramáticas independientes de contexto

¿Cuándo está una GIC G en Forma Normal de Greibach (FNG)?

1. La variable inicial no es recursiva.
2. G no tiene variables inútiles.

¿Cuándo está una GIC G en Forma Normal de Greibach (FNG)?

1. La variable inicial no es recursiva.
2. G no tiene variables inútiles.
3. G no tiene producciones λ (excepto posiblemente $S \to \lambda$).

Lenguajes y gramáticas independientes de contexto

¿Cuándo está una GIC G en Forma Normal de Greibach (FNG)?

1. La variable inicial no es recursiva.
2. G no tiene variables inútiles.
3. G no tiene producciones λ (excepto posiblemente $S \to \lambda$).
4. Todas las producciones son de la forma: $A \to a$ (producciones simples) ó $A \to aB_1B_2...B_k$, donde las B_i son variables.

Lenguajes y gramáticas independientes de contexto

¿Cuándo está una GIC G en Forma Normal de Greibach (FNG)?

1. La variable inicial no es recursiva.
2. G no tiene variables inútiles.
3. G no tiene producciones λ (excepto posiblemente $S \to \lambda$).
4. Todas las producciones son de la forma: $A \to a$ (producciones simples) ó $A \to aB_1B_2...B_k$, donde las B_i son variables.

¿Cuáles son las características notables de una gramática en FNG?

Lenguajes y gramáticas independientes de contexto

¿Cuándo está una GIC G en Forma Normal de Greibach (FNG)?

1. La variable inicial no es recursiva.
2. G no tiene variables inútiles.
3. G no tiene producciones λ (excepto posiblemente $S \rightarrow \lambda$).
4. Todas las producciones son de la forma: $A \rightarrow a$ (producciones simples) ó $A \rightarrow aB_1B_2...B_k$, donde las B_i son variables.

¿Cuáles son las características notables de una gramática en FNG?

1. En cada paso aparece un único terminal

Lenguajes y gramáticas independientes de contexto

¿Cuándo está una GIC G en Forma Normal de Greibach (FNG)?

1. La variable inicial no es recursiva.
2. G no tiene variables inútiles.
3. G no tiene producciones λ (excepto posiblemente $S \to \lambda$).
4. Todas las producciones son de la forma: $A \to a$ (producciones simples) ó $A \to aB_1 B_2 ... B_k$, donde las B_i son variables.

¿Cuáles son las características notables de una gramática en FNG?

1. En cada paso aparece un único terminal
2. La derivación de una cadena de longitud n ($n \geq 1$) tiene exactamente n pasos.

Lenguajes y gramáticas independientes de contexto

¿Cuándo está una GIC G en Forma Normal de Greibach (FNG)?

1. La variable inicial no es recursiva.
2. G no tiene variables inútiles.
3. G no tiene producciones λ (excepto posiblemente $S \to \lambda$).
4. Todas las producciones son de la forma: $A \to a$ (producciones simples) ó $A \to aB_1 B_2 ... B_k$, donde las B_i son variables.

¿Cuáles son las características notables de una gramática en FNG?

1. En cada paso aparece un único terminal
2. La derivación de una cadena de longitud n ($n \geq 1$) tiene exactamente n pasos.

¿Se puede convertir una GIC G a una gramática en FNG equivalente?

Lenguajes y gramáticas independientes de contexto

¿Cuándo está una GIC G en Forma Normal de Greibach (FNG)?

1. La variable inicial no es recursiva.
2. G no tiene variables inútiles.
3. G no tiene producciones λ (excepto posiblemente $S \to \lambda$).
4. Todas las producciones son de la forma: $A \to a$ (producciones simples) ó $A \to aB_1 B_2 ... B_k$, donde las B_i son variables.

¿Cuáles son las características notables de una gramática en FNG?

1. En cada paso aparece un único terminal
2. La derivación de una cadena de longitud n ($n \geq 1$) tiene exactamente n pasos.

¿Se puede convertir una GIC G a una gramática en FNG equivalente?

Existe un procedimiento algorítmico para transformar una GIC dada en una gramática equivalente en FNG.

Antes de presentar el algoritmo debemos tener claros ciertos conceptos.

Antes de presentar el algoritmo debemos tener claros ciertos conceptos.
¿Qué es una variable recursiva a la izquierda?

Antes de presentar el algoritmo debemos tener claros ciertos conceptos.
¿Qué es una variable recursiva a la izquierda?

Definición

Una variable se llama recursiva a la izquierda si tiene una producción de la forma:

Antes de presentar el algoritmo debemos tener claros ciertos conceptos.
¿Qué es una variable recursiva a la izquierda?

Definición

Una variable se llama recursiva a la izquierda si tiene una producción de la forma:

$$A \to Aw,\ w \in (V \cup \Sigma)^*.$$

¿Cómo se puede eliminar la recursividad a la izquierda?

¿Cómo se puede eliminar la recursividad a la izquierda?

Teorema 1

Las producciones de una variable A cualquiera se pueden dividir en dos clases:

¿Cómo se puede eliminar la recursividad a la izquierda?

Teorema 1

Las producciones de una variable A cualquiera se pueden dividir en dos clases:

$$\begin{cases} A \to A\alpha_1 \mid A\alpha_2 \mid \cdots \mid A\alpha_n \\ A \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_m \end{cases}$$

¿Cómo se puede eliminar la recursividad a la izquierda?

Teorema 1

Las producciones de una variable A cualquiera se pueden dividir en dos clases:

$$\begin{cases} A \to A\alpha_1 \mid A\alpha_2 \mid \cdots \mid A\alpha_n \\ A \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_m \end{cases}$$

donde $\alpha_i,\ \beta_i \in (V \cup \Sigma)^*$ *y el primer símbolo de* β_i *es diferente de A. Sin alterar el lenguaje generado, las anteriores producciones se pueden simular, reemplazándolas por las siguientes:*

¿Cómo se puede eliminar la recursividad a la izquierda?

Teorema 1

Las producciones de una variable A cualquiera se pueden dividir en dos clases:

$$\begin{cases} A \to A\alpha_1 \mid A\alpha_2 \mid \cdots \mid A\alpha_n \\ A \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_m \end{cases}$$

donde $\alpha_i, \beta_i \in (V \cup \Sigma)^*$ *y el primer símbolo de* β_i *es diferente de A. Sin alterar el lenguaje generado, las anteriores producciones se pueden simular, reemplazándolas por las siguientes:*

$$\begin{cases} A \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_m \mid \beta_1 Z \mid \beta_2 Z \mid \cdots \mid \beta_m Z \\ Z \to \alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_n \mid \alpha_1 Z \mid \alpha_2 Z \mid \cdots \mid \alpha_n Z \end{cases}$$

¿Cómo se puede eliminar la recursividad a la izquierda?

Teorema 1

Las producciones de una variable A cualquiera se pueden dividir en dos clases:

$$\begin{cases} A \to A\alpha_1 \mid A\alpha_2 \mid \cdots \mid A\alpha_n \\ A \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_m \end{cases}$$

donde α_i, $\beta_i \in (V \cup \Sigma)^*$ *y el primer símbolo de* β_i *es diferente de A. Sin alterar el lenguaje generado, las anteriores producciones se pueden simular, reemplazándolas por las siguientes:*

$$\begin{cases} A \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_m \mid \beta_1 Z \mid \beta_2 Z \mid \cdots \mid \beta_m Z \\ Z \to \alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_n \mid \alpha_1 Z \mid \alpha_2 Z \mid \cdots \mid \alpha_n Z \end{cases}$$

donde Z es una variable completamente nueva.

¿Qué se desprende del anterior teorema?

¿Qué se desprende del anterior teorema?

Lema 1

En una GIC cualquiera, una producción $A \to uBv$ se puede reemplazar (simular) por:

$$A \to uw_1v \mid uw_2v \mid \cdots \mid uw_nv$$

siendo $B \to w_1 \mid w_2 \mid \cdots \mid w_n$ todas las producciones de B.

¿Cómo se convierte una GIC G a una gramática en FNG equivalente?

¿Cómo se convierte una GIC G a una gramática en FNG equivalente?

1. Se supone que la gramática está en FNC.

Lenguajes y gramáticas independientes de contexto

¿Cómo se convierte una GIC G a una gramática en FNG equivalente?

1. Se supone que la gramática está en FNC.
2. Se enumeran las variables en un orden arbitrario pero fijo durante el procedimiento. S debe ser la variable con orden 1.

¿Cómo se convierte una GIC G a una gramática en FNG equivalente?

1. Se supone que la gramática está en FNC.
2. Se enumeran las variables en un orden arbitrario pero fijo durante el procedimiento. S debe ser la variable con orden 1.
3. Para cada variable A de la gramática original, siguiendo el orden elegido, se debe modificar sus producciones de tal manera que el primer símbolo del cuerpo de cada producción sea un terminal o una variable con orden mayor que el de A. Para lograrlo se usa el teorema de eliminación de la recursividad a la izquierda (teorema 1) y el lema 1, todas las veces que sea necesario.

Lenguajes y gramáticas independientes de contexto

¿Cómo se convierte una GIC G a una gramática en FNG equivalente?

1. Se supone que la gramática está en FNC.
2. Se enumeran las variables en un orden arbitrario pero fijo durante el procedimiento. S debe ser la variable con orden 1.
3. Para cada variable A de la gramática original, siguiendo el orden elegido, se debe modificar sus producciones de tal manera que el primer símbolo del cuerpo de cada producción sea un terminal o una variable con orden mayor que el de A. Para lograrlo se usa el teorema de eliminación de la recursividad a la izquierda (teorema 1) y el lema 1, todas las veces que sea necesario.
4. Se utiliza el lema 1 para modificar las producciones de las variables originales de tal manera que el primer símbolo del cuerpo de cada producción sea un terminal. Esto se hace siguiendo el orden inverso de enumeración de las variables.

Lenguajes y gramáticas independientes de contexto

¿Cómo se convierte una GIC G a una gramática en FNG equivalente?

1. Se supone que la gramática está en FNC.
2. Se enumeran las variables en un orden arbitrario pero fijo durante el procedimiento. S debe ser la variable con orden 1.
3. Para cada variable A de la gramática original, siguiendo el orden elegido, se debe modificar sus producciones de tal manera que el primer símbolo del cuerpo de cada producción sea un terminal o una variable con orden mayor que el de A. Para lograrlo se usa el teorema de eliminación de la recursividad a la izquierda (teorema 1) y el lema 1, todas las veces que sea necesario.
4. Se utiliza el lema 1 para modificar las producciones de las variables originales de tal manera que el primer símbolo del cuerpo de cada producción sea un terminal. Esto se hace siguiendo el orden inverso de enumeración de las variables.
5. Se utiliza de nuevo el lema 1 para modificar las producciones de las variables nuevas de tal manera que el primer símbolo del cuerpo de cada producción sea un terminal.

Ejemplo

Encuentre una gramática en FNG equivalente a la siguiente gramática (que está en FNC):

Ejemplo

Encuentre una gramática en FNG equivalente a la siguiente gramática (que está en FNC):

$$G : \begin{cases} S \rightarrow AA \mid a \\ A \rightarrow AA \mid b \end{cases}$$

Ejemplo

Encuentre una gramática en FNG equivalente a la siguiente gramática (que está en FNC):

$$G : \begin{cases} S \to AA \mid a \\ A \to AA \mid b \end{cases}$$

Solución

Paso 1: *Solamente hay un orden posible para las variables:* S, A.

Ejemplo

Encuentre una gramática en FNG equivalente a la siguiente gramática (que está en FNC):

$$G : \begin{cases} S \to AA \mid a \\ A \to AA \mid b \end{cases}$$

Solución

Paso 1: *Solamente hay un orden posible para las variables: S, A.*

Paso 2: *En este paso sólo hay que eliminar la recursividad a izquierda de la variable A. Al hacerlo se obtiene la gramática:*

Ejemplo

Encuentre una gramática en FNG equivalente a la siguiente gramática (que está en FNC):

$$G : \begin{cases} S \to AA \mid a \\ A \to AA \mid b \end{cases}$$

Solución

Paso 1: *Solamente hay un orden posible para las variables: S, A.*

Paso 2: *En este paso sólo hay que eliminar la recursividad a izquierda de la variable A. Al hacerlo se obtiene la gramática:*

$$\begin{cases} S \to AA \mid a \\ A \to b \mid bZ \\ Z \to A \mid AZ \end{cases}$$

Solución

$$\left\{\begin{array}{l} S \to AA \mid a \\ A \to b \mid bZ \\ Z \to A \mid AZ \end{array}\right.$$

Solución

$$\begin{cases} S \to AA \mid a \\ A \to b \mid bZ \\ Z \to A \mid AZ \end{cases}$$

Paso 3: *Se modifican las producciones de las variables originales de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

Solución

$$\begin{cases} S \rightarrow AA \mid a \\ A \rightarrow b \mid bZ \\ Z \rightarrow A \mid AZ \end{cases}$$

Paso 3: *Se modifican las producciones de las variables originales de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

$$\begin{cases} S \rightarrow bA \mid bZA \mid a \\ A \rightarrow b \mid bZ \\ Z \rightarrow A \mid AZ \end{cases}$$

Solución

$$\left\{ \begin{array}{l} S \rightarrow AA \mid a \\ A \rightarrow b \mid bZ \\ Z \rightarrow A \mid AZ \end{array} \right.$$

Paso 3: *Se modifican las producciones de las variables originales de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

$$\left\{ \begin{array}{l} S \rightarrow bA \mid bZA \mid a \\ A \rightarrow b \mid bZ \\ Z \rightarrow A \mid AZ \end{array} \right.$$

Paso 4: *Se modifican las producciones de las variables nuevas de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

Solución

$$\begin{cases} S \to AA \mid a \\ A \to b \mid bZ \\ Z \to A \mid AZ \end{cases}$$

Paso 3: *Se modifican las producciones de las variables originales de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

$$\begin{cases} S \to bA \mid bZA \mid a \\ A \to b \mid bZ \\ Z \to A \mid AZ \end{cases}$$

Paso 4: *Se modifican las producciones de las variables nuevas de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

$$\begin{cases} S \to bA \mid bZA \mid a \\ A \to b \mid bZ \\ Z \to b \mid bZ \mid bZZ \end{cases}$$

Ejemplo

Encuentre una gramática en FNG equivalente a la siguiente gramática (que está en FNC):

Ejemplo

Encuentre una gramática en FNG equivalente a la siguiente gramática (que está en FNC):

$$G : \begin{cases} S \to AB \mid BC \\ A \to AB \mid a \\ B \to AA \mid CB \mid a \\ C \to a \mid b \end{cases}$$

Ejemplo

Encuentre una gramática en FNG equivalente a la siguiente gramática (que está en FNC):

$$G : \begin{cases} S \to AB \mid BC \\ A \to AB \mid a \\ B \to AA \mid CB \mid a \\ C \to a \mid b \end{cases}$$

Solución

Paso 1: *Orden de las variables:* S, B, A, C. *Este orden es muy adecuado porque el cuerpo de las producciones de* B *comienza con* A *o* C, *que son variables de orden mayor.*

Ejemplo

Encuentre una gramática en FNG equivalente a la siguiente gramática (que está en FNC):

$$G : \begin{cases} S \to AB \mid BC \\ A \to AB \mid a \\ B \to AA \mid CB \mid a \\ C \to a \mid b \end{cases}$$

Solución

Paso 1: *Orden de las variables: S, B, A, C. Este orden es muy adecuado porque el cuerpo de las producciones de B comienza con A o C, que son variables de orden mayor.*

$$\begin{cases} S \to AB \mid BC \\ B \to AA \mid CB \mid a \\ A \to AB \mid a \\ C \to a \mid b \end{cases}$$

Solución

$$\begin{cases} S \to AB \mid BC \\ B \to AA \mid CB \mid a \\ A \to AB \mid a \\ C \to a \mid b \end{cases}$$

Solución

$$\begin{cases} S \rightarrow AB \mid BC \\ B \rightarrow AA \mid CB \mid a \\ A \rightarrow AB \mid a \\ C \rightarrow a \mid b \end{cases}$$

Paso 2: *En este paso sólo hay que eliminar la recursividad a izquierda de la variable A. Al hacerlo se obtiene la gramática:*

Solución

$$\begin{cases} S \rightarrow AB \mid BC \\ B \rightarrow AA \mid CB \mid a \\ A \rightarrow AB \mid a \\ C \rightarrow a \mid b \end{cases}$$

Paso 2: *En este paso sólo hay que eliminar la recursividad a izquierda de la variable A. Al hacerlo se obtiene la gramática:*

$$\begin{cases} S \rightarrow AB \mid BC \\ B \rightarrow AA \mid CB \mid a \\ A \rightarrow a \mid aZ \\ C \rightarrow a \mid b \\ Z \rightarrow B \mid BZ \end{cases}$$

Lenguajes y gramáticas independientes de contexto

Solución

$$\begin{cases} S \rightarrow AB \mid BC \\ B \rightarrow AA \mid CB \mid a \\ A \rightarrow AB \mid a \\ C \rightarrow a \mid b \end{cases}$$

Paso 2: *En este paso sólo hay que eliminar la recursividad a izquierda de la variable A. Al hacerlo se obtiene la gramática:*

$$\begin{cases} S \rightarrow AB \mid BC \\ B \rightarrow AA \mid CB \mid a \\ A \rightarrow a \mid aZ \\ C \rightarrow a \mid b \\ Z \rightarrow B \mid BZ \end{cases}$$

Paso 3: *Se modifican las producciones de las variables originales de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

Lenguajes y gramáticas independientes de contexto

Solución

$$\begin{cases} S \to AB \mid BC \\ B \to AA \mid CB \mid a \\ A \to AB \mid a \\ C \to a \mid b \end{cases}$$

Paso 2: *En este paso sólo hay que eliminar la recursividad a izquierda de la variable A. Al hacerlo se obtiene la gramática:*

$$\begin{cases} S \to AB \mid BC \\ B \to AA \mid CB \mid a \\ A \to a \mid aZ \\ C \to a \mid b \\ Z \to B \mid BZ \end{cases}$$

Paso 3: *Se modifican las producciones de las variables originales de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

$$\begin{cases} S \to aB \mid aZB \mid aAC \mid aZAC \mid aBC \mid bBC \mid aC \\ B \to aA \mid aZA \mid aB \mid bB \mid a \\ A \to a \mid aZ \\ C \to a \mid b \\ Z \to B \mid BZ \end{cases}$$

Solución

$$\begin{cases} S \to aB \mid aZB \mid aAC \mid aZAC \mid aBC \mid bBC \mid aC \\ B \to aA \mid aZA \mid aB \mid bB \mid a \\ A \to a \mid aZ \\ C \to a \mid b \\ Z \to B \mid BZ \end{cases}$$

Solución

$$\begin{cases} S \to aB \mid aZB \mid aAC \mid aZAC \mid aBC \mid bBC \mid aC \\ B \to aA \mid aZA \mid aB \mid bB \mid a \\ A \to a \mid aZ \\ C \to a \mid b \\ Z \to B \mid BZ \end{cases}$$

Paso 4: *Se modifican las producciones de las variables nuevas de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

Solución

$$\begin{cases} S \to aB \mid aZB \mid aAC \mid aZAC \mid aBC \mid bBC \mid aC \\ B \to aA \mid aZA \mid aB \mid bB \mid a \\ A \to a \mid aZ \\ C \to a \mid b \\ Z \to B \mid BZ \end{cases}$$

Paso 4: *Se modifican las producciones de las variables nuevas de manera que el primer símbolo del cuerpo de cada producción sea un terminal:*

$$\begin{cases} S \to aB \mid aZB \mid aAC \mid aZAC \mid aBC \mid bBC \mid aC \\ B \to aA \mid aZA \mid aB \mid bB \mid a \\ A \to a \mid aZ \\ C \to a \mid b \\ Z \to aA \mid aZA \mid aB \mid bB \mid a \mid aAZ \mid aZAZ \mid aBZ \mid bBZ \mid aZ \end{cases}$$

¿En qué consiste un problema de decisión para GIC?

¿En qué consiste un problema de decisión para GIC?

Dada una propiedad $\mathcal{P}$, referente a gramáticas independientes del contexto, un problema de decisión para $\mathcal{P}$ consiste en buscar un algoritmo, aplicable a una GIC arbitraria G, que responda SI o NO a la pregunta: ¿satisface G la propiedad $\mathcal{P}$?

¿En qué consiste un problema de decisión para GIC?

Dada una propiedad $\mathcal{P}$, referente a gramáticas independientes del contexto, un problema de decisión para $\mathcal{P}$ consiste en buscar un algoritmo, aplicable a una GIC arbitraria G, que responda SI o NO a la pregunta: ¿satisface G la propiedad $\mathcal{P}$?

¿Qué puede ser útil para la resolución de algoritmos de decisión más complejos?

¿En qué consiste un problema de decisión para GIC?

Dada una propiedad $\mathcal{P}$, referente a gramáticas independientes del contexto, un problema de decisión para $\mathcal{P}$ consiste en buscar un algoritmo, aplicable a una GIC arbitraria G, que responda SI o NO a la pregunta: ¿satisface G la propiedad $\mathcal{P}$?

¿Qué puede ser útil para la resolución de algoritmos de decisión más complejos?

Los algoritmos vistos recientemente (para encontrar las variables terminables, alcanzables, anulables, etc) son frecuentemente útiles en el diseño de algoritmos de decisión más complejos.

¿Cómo se resuelve el problema de vacuidad para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$, ¿es $L(G) \neq \emptyset$?

¿Cómo se resuelve el problema de vacuidad para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$, ¿es $L(G) \neq \emptyset$?

El algoritmo de decisión sería: ejecutar el algoritmo para determinar el conjunto **TERM** de variables terminables. $L(G) \neq \emptyset$ si y sólo si $S \in$ **TERM**.

Lenguajes y gramáticas independientes de contexto

¿Cómo se resuelve el problema de vacuidad para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$, ¿es $L(G) \neq \emptyset$?

El algoritmo de decisión sería: ejecutar el algoritmo para determinar el conjunto **TERM** de variables terminables. $L(G) \neq \emptyset$ si y sólo si $S \in$ **TERM**.

¿Cómo se resuelve el problema de pertenencia para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$ y una cadena $w \in \Sigma^*$, ¿se tiene $w \in L(G)$?

¿Cómo se resuelve el problema de vacuidad para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$, ¿es $L(G) \neq \emptyset$?

El algoritmo de decisión sería: ejecutar el algoritmo para determinar el conjunto **TERM** de variables terminables. $L(G) \neq \emptyset$ si y sólo si $S \in$ **TERM**.

¿Cómo se resuelve el problema de pertenencia para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$ y una cadena $w \in \Sigma^*$, ¿se tiene $w \in L(G)$?

- Para resolver este problema primero se convierte G a la forma FNC, con variable inicial no recursiva.

Lenguajes y gramáticas independientes de contexto

¿Cómo se resuelve el problema de vacuidad para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$, ¿es $L(G) \neq \emptyset$?

El algoritmo de decisión sería: ejecutar el algoritmo para determinar el conjunto **TERM** de variables terminables. $L(G) \neq \emptyset$ si y sólo si $S \in$ **TERM**.

¿Cómo se resuelve el problema de pertenencia para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$ y una cadena $w \in \Sigma^*$, ¿se tiene $w \in L(G)$?

- Para resolver este problema primero se convierte G a la forma FNC, con variable inicial no recursiva.
- A partir de una GIC G en FNC se puede diseñar un algoritmo bastante ineficiente para decidir si $w \in L(G)$.

¿Cómo se resuelve el problema de vacuidad para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$, ¿es $L(G) \neq \emptyset$?

El algoritmo de decisión sería: ejecutar el algoritmo para determinar el conjunto **TERM** de variables terminables. $L(G) \neq \emptyset$ si y sólo si $S \in$ **TERM**.

¿Cómo se resuelve el problema de pertenencia para GIC? Es decir, determinar si dada una gramática $G = (V, \Sigma, S, P)$ y una cadena $w \in \Sigma^*$, ¿se tiene $w \in L(G)$?

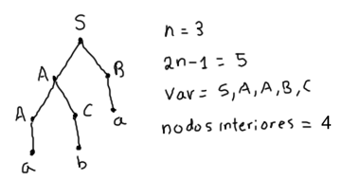
- Para resolver este problema primero se convierte G a la forma FNC, con variable inicial no recursiva.
- A partir de una GIC G en FNC se puede diseñar un algoritmo bastante ineficiente para decidir si $w \in L(G)$.
- Se encuentran todas las posibles derivaciones a izquierda (o los árboles de derivación) que generen cadenas de longitud $n = |w|$.

Lenguajes y gramáticas independientes de contexto

- Un árbol de derivación de una cadena de longitud n tiene exactamente $2n - 1$ nodos etiquetados con variables.

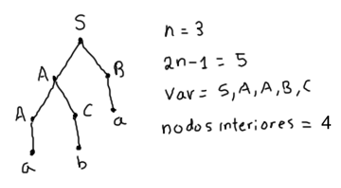
Lenguajes y gramáticas independientes de contexto

- Un árbol de derivación de una cadena de longitud n tiene exactamente $2n - 1$ nodos etiquetados con variables.



Lenguajes y gramáticas independientes de contexto

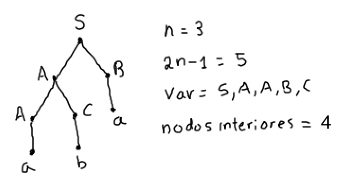
- Un árbol de derivación de una cadena de longitud n tiene exactamente $2n - 1$ nodos etiquetados con variables.



Ya que la raíz del árbol es S y S no es recursiva, en un árbol de derivación de una cadena de longitud n hay exactamente $2n - 2$ nodos interiores etiquetados con variables.

Lenguajes y gramáticas independientes de contexto

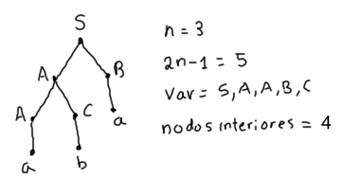
- Un árbol de derivación de una cadena de longitud n tiene exactamente $2n - 1$ nodos etiquetados con variables.



  Ya que la raíz del árbol es S y S no es recursiva, en un árbol de derivación de una cadena de longitud n hay exactamente $2n - 2$ nodos interiores etiquetados con variables.

- Para determinar si una cadena dada de longitud n es o no generada por G, se deben considerar todos los posibles árboles de derivación con $2n - 2$ variables interiores.

- Un árbol de derivación de una cadena de longitud n tiene exactamente $2n - 1$ nodos etiquetados con variables.

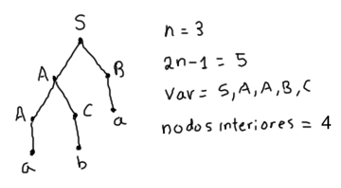


  Ya que la raíz del árbol es S y S no es recursiva, en un árbol de derivación de una cadena de longitud n hay exactamente $2n - 2$ nodos interiores etiquetados con variables.

- Para determinar si una cadena dada de longitud n es o no generada por G, se deben considerar todos los posibles árboles de derivación con $2n - 2$ variables interiores.
- Este algoritmo es ineficiente porque si G tiene k variables, hay que verificar no menos de k^{2n-2} árboles (esto sin contar las posibilidades para las hojas).

Lenguajes y gramáticas independientes de contexto

- Un árbol de derivación de una cadena de longitud n tiene exactamente $2n - 1$ nodos etiquetados con variables.



  Ya que la raíz del árbol es S y S no es recursiva, en un árbol de derivación de una cadena de longitud n hay exactamente $2n - 2$ nodos interiores etiquetados con variables.

- Para determinar si una cadena dada de longitud n es o no generada por G, se deben considerar todos los posibles árboles de derivación con $2n - 2$ variables interiores.
- Este algoritmo es ineficiente porque si G tiene k variables, hay que verificar no menos de k^{2n-2} árboles (esto sin contar las posibilidades para las hojas).
- Por consiguiente, el procedimiento tiene complejidad exponencial con respecto al tamaño de la entrada.

¿Existe algún algoritmo eficiente para resolver este problema?

¿Existe algún algoritmo eficiente para resolver este problema?

- Sí, para resolver el problema de la pertenencia hay un algoritmo muy eficiente (su complejidad es polinomial).

Lenguajes y gramáticas independientes de contexto

¿Existe algún algoritmo eficiente para resolver este problema?

- Sí, para resolver el problema de la pertenencia hay un algoritmo muy eficiente (su complejidad es polinomial).
- En él se usa la llamada programacioń dinámica (técnica para llenar tablas progresivamente, re-utilizando información previamente obtenida).

¿Existe algún algoritmo eficiente para resolver este problema?

- Sí, para resolver el problema de la pertenencia hay un algoritmo muy eficiente (su complejidad es polinomial).
- En él se usa la llamada programacioń dinámica (técnica para llenar tablas progresivamente, re-utilizando información previamente obtenida).
- Éste se denomina algoritmo CYK (nombre que corresponde a las iniciales de los investigadores Cocke, Younger y Kasami).

Lenguajes y gramáticas independientes de contexto

¿En qué consiste el algoritmo CYK?

¿En qué consiste el algoritmo CYK?

- Este algoritmo tiene como entrada una GIC G en FNC y una cadena de n terminales $w = a_1 a_2 \cdots a_n$.

¿En qué consiste el algoritmo CYK?

- Este algoritmo tiene como entrada una GIC G en FNC y una cadena de n terminales $w = a_1 a_2 \cdots a_n$.
- Se aplica llenando una tabla de n filas (una por cada terminal de la entrada w) y n columnas.

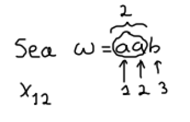
¿En qué consiste el algoritmo CYK?

- Este algoritmo tiene como entrada una GIC G en FNC y una cadena de n terminales $w = a_1 a_2 \cdots a_n$.
- Se aplica llenando una tabla de n filas (una por cada terminal de la entrada w) y n columnas.
- X_{ij} es el conjunto de variables de las que se puede derivar la subcadena de w cuyo primer símbolo está en la posición i y cuya longitud es j.

Lenguajes y gramáticas independientes de contexto

¿En qué consiste el algoritmo CYK?

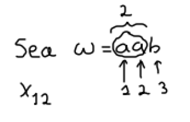
- Este algoritmo tiene como entrada una GIC G en FNC y una cadena de n terminales $w = a_1 a_2 \cdots a_n$.
- Se aplica llenando una tabla de n filas (una por cada terminal de la entrada w) y n columnas.
- X_{ij} es el conjunto de variables de las que se puede derivar la subcadena de w cuyo primer símbolo está en la posición i y cuya longitud es j.



Lenguajes y gramáticas independientes de contexto

¿En qué consiste el algoritmo CYK?

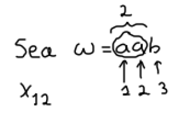
- Este algoritmo tiene como entrada una GIC G en FNC y una cadena de n terminales $w = a_1 a_2 \cdots a_n$.
- Se aplica llenando una tabla de n filas (una por cada terminal de la entrada w) y n columnas.
- X_{ij} es el conjunto de variables de las que se puede derivar la subcadena de w cuyo primer símbolo está en la posición i y cuya longitud es j.



- Es decir $X_{ij} =$ conjunto de variables A tales que $A \stackrel{+}{\Longrightarrow} a_i a_{i+1} \cdots a_{i+j-1}$.

¿En qué consiste el algoritmo CYK?

- Este algoritmo tiene como entrada una GIC G en FNC y una cadena de n terminales $w = a_1 a_2 \cdots a_n$.
- Se aplica llenando una tabla de n filas (una por cada terminal de la entrada w) y n columnas.
- X_{ij} es el conjunto de variables de las que se puede derivar la subcadena de w cuyo primer símbolo está en la posición i y cuya longitud es j.



- Es decir $X_{ij} =$ conjunto de variables A tales que $A \stackrel{+}{\Longrightarrow} a_i a_{i+1} \cdots a_{i+j-1}$.
- Al determinar los conjuntos X_{ij} se obtienen las posibles maneras de derivar subcadenas de w que permitan construir una derivación de la cadena completa w.

- La tabla se llena por columnas, de arriba hacia abajo; la primera columna ($j = 1$) corresponde a las subcadenas de longitud 1, la segunda columna ($j = 2$) corresponde a las subcadenas de longitud 2, y así sucesivamente. La última columna ($j = n$) corresponde a la única subcadena de longitud n que tiene w, que es la propia cadena w.

Lenguajes y gramáticas independientes de contexto

- La tabla se llena por columnas, de arriba hacia abajo; la primera columna ($j = 1$) corresponde a las subcadenas de longitud 1, la segunda columna ($j = 2$) corresponde a las subcadenas de longitud 2, y así sucesivamente. La última columna ($j = n$) corresponde a la única subcadena de longitud n que tiene w, que es la propia cadena w.
- Se encuentran las variables que derivan terminales en las distintas posiciones de la cadena (primera columna)

- La tabla se llena por columnas, de arriba hacia abajo; la primera columna ($j = 1$) corresponde a las subcadenas de longitud 1, la segunda columna ($j = 2$) corresponde a las subcadenas de longitud 2, y así sucesivamente. La última columna ($j = n$) corresponde a la única subcadena de longitud n que tiene w, que es la propia cadena w.
- Se encuentran las variables que derivan terminales en las distintas posiciones de la cadena (primera columna)
- Como j es la longitud cuando $j = 1$ se encuentran las variables que generan cadenas de tamaño 1 y esas son terminales.

Lenguajes y gramáticas independientes de contexto

- La tabla se llena por columnas, de arriba hacia abajo; la primera columna ($j = 1$) corresponde a las subcadenas de longitud 1, la segunda columna ($j = 2$) corresponde a las subcadenas de longitud 2, y así sucesivamente. La última columna ($j = n$) corresponde a la única subcadena de longitud n que tiene w, que es la propia cadena w.
- Se encuentran las variables que derivan terminales en las distintas posiciones de la cadena (primera columna)
- Como j es la longitud cuando $j = 1$ se encuentran las variables que generan cadenas de tamaño 1 y esas son terminales.
- Se tendrá que $w \in L(G)$ si y sólo si $S \in X_{1n}$.

Ejemplo

$$G: \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases} \qquad w = a\,a\,b$$

Ejemplo

$$G: \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases} \qquad w = a\,a\,b$$

$$X_{11} = \{A\}$$

$$w = \overset{1}{\overbrace{\underset{\substack{\uparrow \\ 1}}{a}}}\ \underset{\substack{\uparrow \\ 2}}{a}\ \underset{\substack{\uparrow \\ 3}}{b}$$

Lenguajes y gramáticas independientes de contexto

Ejemplo

$$G : \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases} \qquad \omega = a\,a\,b$$

$$X_{11} = \{A\}$$

$$\omega = \begin{array}{ccc} \overbrace{a}^{1} & a & b \\ \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

$$X_{21} = \{A\}$$

$$\omega = \begin{array}{ccc} a & \overbrace{a}^{1} & b \\ \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

Lenguajes y gramáticas independientes de contexto

Ejemplo

$$G : \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases} \qquad w = a\,a\,b$$

$$X_{11} = \{A\}$$

$$w = \overbrace{\underset{\overset{\uparrow}{1}}{a}}^{1}\ \underset{\overset{\uparrow}{2}}{a}\ \underset{\overset{\uparrow}{3}}{b}$$

$$X_{21} = \{A\}$$

$$w = \underset{\overset{\uparrow}{1}}{a}\ \overbrace{\underset{\overset{\uparrow}{2}}{a}}^{1}\ \underset{\overset{\uparrow}{3}}{b}$$

$$X_{31} = \{B\}$$

$$w = \underset{\overset{\uparrow}{1}}{a}\ \underset{\overset{\uparrow}{2}}{a}\ \overbrace{\underset{\overset{\uparrow}{3}}{b}}^{1}$$

Ejemplo

$$G: \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases} \qquad w = a\,a\,b$$

$$X_{11} = \{A\}$$

$$w = \begin{array}{ccc} \overbrace{a}^{1} & a & b \\ \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

AA

$$X_{12} = \{A\}$$

$$w = \begin{array}{ccc} \multicolumn{2}{c}{\overbrace{a \quad a}^{2}} & b \\ \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

$$X_{21} = \{A\}$$

$$w = \begin{array}{ccc} a & \overbrace{a}^{1} & b \\ \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

$$X_{31} = \{B\}$$

$$w = \begin{array}{ccc} a & a & \overbrace{b}^{1} \\ \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

Ejemplo

$$G: \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases} \qquad w = a\,a\,b$$

$X_{11} = \{A\}$

$$w = \overset{1}{\overbrace{a}}\; a\; b \qquad \begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

$X_{12} = \{A\}$

$$w = \overset{2}{\overbrace{a\; a}}\; b \qquad \begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

AA

$X_{21} = \{A\}$

$$w = a\; \overset{1}{\overbrace{a}}\; b \qquad \begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

$X_{22} = \{S\}$

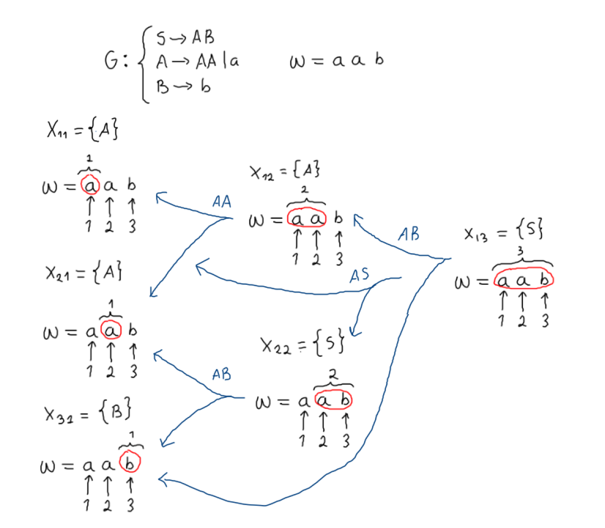
$$w = a\; \overset{2}{\overbrace{a\; b}} \qquad \begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

AB

$X_{31} = \{B\}$

$$w = a\; a\; \overset{1}{\overbrace{b}} \qquad \begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 \end{array}$$

Ejemplo



$G: \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases}$
$w = a\,a\,b$
$X_{11} = \{A\}$
$X_{12} = \{A\}$
$X_{13} = \{S\}$
$X_{21} = \{A\}$
$X_{22} = \{S\}$
$X_{31} = \{B\}$
AA
AB
AS
AB
$w = a\,a\,b$
1 2 3

¿Cuál el pseudocódigo para el algoritmo CYK?

¿Cuál el pseudocódigo para el algoritmo CYK?

Algoritmo CYK

ENTRADA:
Gramática G en FNC y cadena de n terminales $w = a_1 a_2 \cdots a_n$.

INICIALIZAR:
$j = 1$. Para cada i, $1 \leq i \leq n$,
$X_{ij} = X_{i1}$:= conjunto de variables A tales que $A \to a_i$

REPETIR:
$j := j + 1$. Para cada i, $1 \leq i \leq n - j + 1$,
$X_{ij} :=$ conjunto de variables A tales que $A \to BC$ es una producción de G, con $B \in X_{ik}$ y $C \in X_{i+k,j-k}$, considerando todos los k tales que $1 \leq k \leq j - 1$.

HASTA: $j = n$.

SALIDA: $w \in L(G)$ si y sólo si $S \in X_{1n}$.

Ejemplo

Sea G la gramática

$$G : \begin{cases} S \rightarrow AB \\ A \rightarrow AA \mid a \\ B \rightarrow b \end{cases}$$

Ejemplo

Sea G la gramática

$$G : \begin{cases} S \to AB \\ A \to AA \,|\, a \\ B \to b \end{cases}$$

y la cadena $w = aab$*, utilice el algoritmo CYK para determinar si* $w \in L(G)$ *o no.*

Solución

		$j=1$	$j=2$	$j=3$
a	$i=1$			
a	$i=2$			
b	$i=3$			

$$G: \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases}$$

Solución

$$G: \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases}$$

		$j=1$	$j=2$	$j=3$
a	$i=1$	$\{A\}$		
a	$i=2$	$\{A\}$		
b	$i=3$	$\{B\}$		

$X_{11} = \{A\}$

$X_{21} = \{A\}$

$X_{31} = \{B\}$

Solución

$$G : \begin{cases} S \rightarrow AB \\ A \rightarrow AA \mid a \\ B \rightarrow b \end{cases}$$

		$j=1$	$j=2$	$j=3$
a	$i=1$	$\{A\}$	$\{A\}$	
a	$i=2$	$\{A\}$	$\{S\}$	
b	$i=3$	$\{B\}$		

$X_{11} = \{A\}$

$X_{21} = \{A\}$

$X_{31} = \{B\}$

$$X_{ij} \longrightarrow X_{ik} X_{i+k\, j-k}$$

$j=2 \quad 1 \leq k \leq 1$

$i=1$

$X_{12} \rightarrow X_{11} X_{21}$

$X_{12} \rightarrow \{A\}\{A\} = \{AA\}$

$X_{12} = \{A\}$

$i=2$

$X_{22} \rightarrow X_{21} X_{31}$

$X_{22} \rightarrow \{A\}\{B\} = \{AB\}$

$X_{22} = \{S\}$

Solución

$$G: \begin{cases} S \to AB \\ A \to AA \mid a \\ B \to b \end{cases}$$

		$j=1$	$j=2$	$j=3$
a	$i=1$	$\{A\}$	$\{A\}$	$\{S\}$
a	$i=2$	$\{A\}$	$\{S\}$	
b	$i=3$	$\{B\}$		

$X_{11} = \{A\}$

$X_{21} = \{A\}$

$X_{31} = \{B\}$

$$X_{ij} \longrightarrow X_{ik}\, X_{i+k\; j-k}$$

$j=2 \quad 1 \leq k \leq 1$

$i=1$

$X_{12} \rightarrow X_{11} X_{21}$

$X_{12} \rightarrow \{A\}\{A\} = \{AA\}$

$X_{12} = \{A\}$

$i=2$

$X_{22} \rightarrow X_{21} X_{31}$

$X_{22} \rightarrow \{A\}\{B\} = \{AB\}$

$X_{22} = \{S\}$

$j=3 \quad 1 \leq k \leq 2$

$i=1$

$X_{13} \rightarrow X_{11} X_{22} \cup X_{12} X_{31}$

$X_{13} \rightarrow \{A\}\{S\} \cup \{A\}\{B\} = \{AS, AB\}$

$X_{13} = \{S\}$

$$S \in X_{13}$$

Ejemplo

Sea G la gramática

$$G : \begin{cases} S \rightarrow BA \,|\, AC \\ A \rightarrow CC \,|\, b \\ B \rightarrow AB \,|\, a \\ C \rightarrow BA \,|\, a \end{cases}$$

Ejemplo

Sea G la gramática

$$G : \begin{cases} S \to BA \mid AC \\ A \to CC \mid b \\ B \to AB \mid a \\ C \to BA \mid a \end{cases}$$

y la cadena $w = bbab$, utilice el algoritmo CYK para determinar si $w \in L(G)$ o no.

Lenguajes y gramáticas independientes de contexto

Solución

		$j=1$	$j=2$	$j=3$	$j=4$
b	$i=1$				
b	$i=2$				
a	$i=3$				
b	$i=4$				

$$G : \begin{cases} S \to BA \mid AC \\ A \to CC \mid b \\ B \to AB \mid a \\ C \to BA \mid a \end{cases}$$

Solución

		$j=1$	$j=2$	$j=3$	$j=4$
b	$i=1$	$\{A\}$			
b	$i=2$	$\{A\}$			
a	$i=3$	$\{B,C\}$			
b	$i=4$	$\{A\}$			

$$G: \begin{cases} S \to BA \mid AC \\ A \to CC \mid b \\ B \to AB \mid a \\ C \to BA \mid a \end{cases}$$

$X_{11} = \{A\}$

$X_{21} = \{A\}$

$X_{31} = \{B,C\}$

$X_{41} = \{A\}$

Solución

		$J=1$	$J=2$	$J=3$	$J=4$
b	$i=1$	$\{A\}$	$\{\ \}$		
b	$i=2$	$\{A\}$	$\{B,S\}$		
a	$i=3$	$\{B,C\}$	$\{S,C\}$		
b	$i=4$	$\{A\}$			

$$G:\begin{cases} S \rightarrow BA \mid AC \\ A \rightarrow CC \mid b \\ B \rightarrow AB \mid a \\ C \rightarrow BA \mid a \end{cases}$$

$X_{11} = \{A\}$

$X_{21} = \{A\}$

$X_{31} = \{B,C\}$

$X_{41} = \{A\}$

$$X_{ij} \longrightarrow X_{ik}\, X_{i+k\ j-k}$$

$j=2 \quad 1 \leq k \leq 1$

$i=1$

$X_{12} \rightarrow X_{11} X_{21}$

$X_{12} \rightarrow \{A\}\{A\} = \{AA\}$

$X_{12} = \{\ \}$

$i=2$

$X_{22} \rightarrow X_{21} X_{31}$

$X_{22} \rightarrow \{A\}\{B,C\} = \{AB, AC\}$

$X_{22} = \{B,S\}$

$i=3$

$X_{32} \rightarrow X_{31} X_{41}$

$X_{32} \rightarrow \{B,C\}\{A\} = \{BA, CA\}$

$X_{32} = \{S,C\}$

Solución

$$G: \begin{cases} S \to BA \mid AC \\ A \to CC \mid b \\ B \to AB \mid a \\ C \to BA \mid a \end{cases}$$

		$j=1$	$j=2$	$j=3$	$j=4$
b	$i=1$	$\{A\}$	$\{\}$	$\{B\}$	
b	$i=2$	$\{A\}$	$\{B,S\}$	$\{S,C\}$	
a	$i=3$	$\{B,C\}$	$\{S,C\}$		
b	$i=4$	$\{A\}$			

$X_{11} = \{A\}$

$X_{21} = \{A\}$

$X_{31} = \{B,C\}$

$X_{41} = \{A\}$

$$X_{ij} \longrightarrow X_{ik}\, X_{i+k\; j-k}$$

$j=2 \quad 1 \leq k \leq 1$

$i=1$

$X_{12} \to X_{11} X_{21}$

$X_{12} \to \{A\}\{A\} = \{AA\}$

$X_{12} = \{\}$

$i=2$

$X_{22} \to X_{21} X_{31}$

$X_{22} \to \{A\}\{B,C\} = \{AB, AC\}$

$X_{22} = \{B,S\}$

$i=3$

$X_{32} \to X_{31} X_{41}$

$X_{32} \to \{B,C\}\{A\} = \{BA, CA\}$

$X_{32} = \{S,C\}$

$j=3 \quad 1 \leq k \leq 2$

$i=1$

$X_{13} \to X_{11} X_{22} \cup X_{12} X_{31}$

$X_{13} \to \{A\}\{B,S\} \cup \{\}\{B,C\} = \{AB, AS\}$

$X_{13} = \{B\}$

$i=2$

$X_{23} \longrightarrow X_{21} X_{32} \cup X_{22} X_{41}$

$X_{23} \to \{A\}\{S,C\} \cup \{B,S\}\{A\} = \{AS, AC, BA, SA\}$

$X_{23} = \{S,C\}$

Solución

		$J=1$	$J=2$	$J=3$	$J=4$
b	$i=1$	$\{A\}$	$\{\}$	$\{B\}$	$\{S,C\}$
b	$i=2$	$\{A\}$	$\{B,S\}$	$\{S,C\}$	
a	$i=3$	$\{B,C\}$	$\{S,C\}$		
b	$i=4$	$\{A\}$			

$$G:\begin{cases} S \to BA \mid AC \\ A \to CC \mid b \\ B \to AB \mid a \\ C \to BA \mid a \end{cases}$$

$X_{11} = \{A\}$

$X_{21} = \{A\}$

$X_{31} = \{B,C\}$

$X_{41} = \{A\}$

$$X_{ij} \longrightarrow X_{ik}\, X_{i+k\; j-k}$$

$j=2 \quad 1 \leq k \leq 1$

$i=1$

$X_{12} \to X_{11} X_{21}$

$X_{12} \to \{A\}\{A\} = \{AA\}$

$X_{12} = \{\}$

$i=2$

$X_{22} \to X_{21} X_{31}$

$X_{22} \to \{A\}\{B,C\} = \{AB,AC\}$

$X_{22} = \{B,S\}$

$i=3$

$X_{32} \to X_{31} X_{41}$

$X_{32} \to \{B,C\}\{A\} = \{BA,CA\}$

$X_{32} = \{S,C\}$

$J=3 \quad 1 \leq k \leq 2$

$i=1$

$X_{13} \to X_{11} X_{22} \cup X_{12} X_{31}$

$X_{13} \to \{A\}\{B,S\} \cup \{\}\{B,C\} = \{AB,AS\}$

$X_{13} = \{B\}$

$i=2$

$X_{23} \to X_{21} X_{32} \cup X_{22} X_{41}$

$X_{23} \to \{A\}\{S,C\} \cup \{B,S\}\{A\} = \{AS,AC,BA,SA\}$

$X_{23} = \{S,C\}$

$J=4 \quad 1 \leq K \leq 3$

$i=1$

$X_{14} \to X_{11} X_{23} \cup X_{12} X_{32} \cup X_{13} X_{41}$

$X_{14} \to \{A\}\{S,C\} \cup \{\}\{S,C\} \cup \{B\}\{A\} = \{AS,AC,BA\}$

$X_{14} = \{S,C\}$

$S \in X_{14}$

1. **Lenguajes y gramáticas independientes de contexto**
   - Presentación del tema
   - Ejercicios