UNIVERSIDAD DE GRANADA

DECSAI

Modelos de Computación

Grado en Ingeniería Informática

Tema 1 – Introducción a la computación

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

Objetivos del tema

- Conocer los fundamentos de los modelos de computación.
- Conocer los conceptos de alfabeto, lenguajes y gramáticas.
- Conocer la jerarquía de los diferentes lenguajes.

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.

1. ¿Por qué estudiar computación?
2. Operaciones con palabras y lenguajes
3. Gramáticas
4. Jerarquía de Chomsky

DECSAI

La computación

- ¿Qué puede ser resuelto de forma automática?
- ¿Qué puede ser resuelto de forma eficiente?
- ¿Existen subclases de problemas que tengan una eficiencia garantizada?

Solución:

Estudio de **Modelos de computación**.

Ejemplo: El problema de la parada

¿Existe un programa (llamémoslo **Stops(P,x)**) que tenga como entrada CUALQUIER programa **P** junto con sus datos de entrada **x** y nos diga si ese programa termina o cicla indefinidamente?

No, no existe

Por ejemplo, supongamos el programa **Turing(P)**:

```
L            If Stops(P,P) GOTO L
```

¿Cuál es la salida de **Turing(Turing)** ? Se entra a una contradicción que demuestra que tal programa **Stops(P,x)** no puede existir.

Esto no quita que exista algún caso o algunos casos para los que se pueda hacer.

1. ¿Por qué estudiar computación?
2. Operaciones con palabras y lenguajes
3. Gramáticas
4. Jerarquía de Chomsky

Definición: Alfabeto

Un alfabeto es un conjunto de símbolos o letras.

Notación: Alfabeto A, B, ...
Notación: Símbolos a, b, c, a_1, b_1, ...

Ejemplos de alfabetos

- A= {0,1}
- B= {a, b, c, …, z}
- C= { <0,1>, <0,0>, <1,1>, <1,0>}

Definición: Palabra

Una sucesión finita $\mathbf{a_1 a_2 a_3 a_4 \ldots a_n}$ de símbolos $\mathbf{a_i}$ de un alfabeto.

Al conjunto de **todas las palabras** de A se le llama **A***.

Notación: Palabras u, v, x, y

Ejemplos de palabras

- A= {0,1} → 0110 es palabra
- A= {a, b, c, …, z} → MC **NO ES PALABRA**
- A= {a, b, c, …, z} → qwerty es palabra
- A= { <0,1>, <0,0>, <1,1>, <1,0>} →<1,1><0,1> es palabra

Definición: Longitud de palabra

Dada una palabra **u**, **la longitud** |u| es el número de símbolos que contiene.

Ejemplos de palabras

- A= {0,1} → 0110 es palabra. |0110|= 4
- A= {a, b, c, …, z} → qwerty es palabra. |qwerty|= 6
- A= { <0,1>, <0,0>, <1,1>, <1,0>} →<1,1><0,1> es palabra. Su longitud |<1,1><0,1> |=2

Definición: Palabra vacía

La palabra vacía **ε** es la palabra de longitud 0.

Definición: Conjunto de palabras de un alfabeto

Notaremos como $\mathbf{A}^{+}$ al conjunto de palabras que se pueden formar del alfabeto **A, excluyendo la palabra vacía.**

$$\boldsymbol{A}^{*} = \boldsymbol{A}^{+} \cup \{\boldsymbol{\varepsilon}\}$$

Definición: Concatenación

Sean $u, v \in A$ dos palabras del alfabeto A, $u = a_1 a_2 \ldots a_n$, y $v = b_1 b_2 \ldots b_m$. La **concatenación de u y v, notada como u.v**, es:

$$u.v = a_1 a_2 \ldots a_n b_1 b_2 \ldots b_m$$

Ejemplos de concatenación

- A= {0,1}, u=0110,v=10 → u.v=011010
- A= {a, b, c, …, z}, u=qwerty, v=asd → u.v= qwertyasd.
- A= { a,b, ε},u=aaba, v= ε → u.v=aaba

Iteración n-ésima

Notamos $\mathbf{u}^n$ a la iteración n-ésima de la palabra **u** de A* como a la concatenación de la palabra **u** con sí misma **n** veces. Su definición es recursiva:

- $\mathbf{u}^0 = \varepsilon$
- $\mathbf{u}^{i+1} = \mathbf{u}^i.\mathbf{u}$, para todo i>=0.

Propiedades de la concatenación

- |u.v|= |u|+|v| para todo u,v en A*
- **Propiedad asociativa:** u.(v.w)= (u.v).w
- **Elemento neutro:** u. ε = ε.u= u

Cadena inversa

Sea **u=** $a_1 \ldots a_n$ una palabra de A*. Entonces la cadena inversa $\mathbf{u^{-1}}$ se define como:

$$\mathbf{u^{-1}=}\ a_n \ldots a_1$$

Ejemplo de iteración y de cadena inversa

- A= {0,1}, u= 0101
  - $\mathbf{u^{-1}= 1010;}$
  - $\mathbf{u^{2}=01010101}$
  - $\mathbf{(u^{2})^{-1}=10101010}$

Definición: Lenguaje

Un **lenguaje** L sobre el alfabeto A es un subconjunto del conjunto de las cadenas sobre A^*. Es decir:

$$L \subseteq A^*.$$

Ejemplos de lenguajes

$L1 = \{a,b, \varepsilon\}$ → Lenguaje formado por 3 palabras

$L2 = \{a^i b^i \mid i = 0,1,2,\dots\}$ → Lenguaje formado por un número dado de a's seguido del mismo números de b's

$L3 = \{uu^{-1} \mid u \in A^*\}$ → Lenguaje formado por palíndromos

$L4 = \{a^{n^2} \mid n = 1,2,3,\dots\}$ → Lenguaje formado por palabras formadas por a's cuya longitud (número de a's) es cuadrado perfecto.

Conjuntos numerables

Un conjunto se dice numerable si existe una aplicación inyectiva de este conjunto en el conjunto de los números naturales, o lo que es lo mismo, se le puede asignar un número natural a cada elemento del conjunto de tal manera que dos elementos distintos tengan números distintos.

Ejemplos de lenguajes numerables

A^* es siempre numerable. Si $A = \{a_1, \ldots , a_n\}$, por ejemplo, a cada símbolo a_i le puedo asignar un número binario diferente. A cada palabra $\{u_1, \ldots , u_n\}$ le puedo asignar otro número binario, resultado de sustituir cada símbolo por su binario.

¿Es el lenguaje C numerable?

Sí. A cada programa se le asigna un número binario (la contatenación de todos los bytes del código compilado).

Operaciones con lenguajes

- **Unión:** L $= L1 \cup L2$= Las palabras del lenguaje L son las palabras del lenguaje L1 junto con las del lenguaje L2.
- **Intersección:** L $= L1 \cap L2$= Las palabras del lenguaje L son las palabras communes del lenguaje L1 y del lenguaje L2.
- **Concatenación:** $L = L1.L2 = \{u_1.u_2 | u_1 \in L1, u_2 \in L2\}$ →Nota: Puede verse como producto cartesiano.

Propiedades de los lenguajes

- $L.\emptyset = \emptyset.L = \emptyset$
- **Elemento neutro:** L$.\{\varepsilon\} = \{\varepsilon\}.L = L$
- **Propiedad Asociativa:** $(L1\ L2)L3 = L1(L2\ L3)$

Ejemplo de concatenación de lenguajes

Si $L_1 = \{0^i1^i : i \geq 0\}$, $L_2 = \{1^i0^i : i \geq 0\}$ entonces,

$$L_1L_2 = \{0^i1^i1^j0^j : i, j \geq 0\}$$

Ejemplo: $u = 0011 \in L_1, v = 10 \in L_2 \rightarrow u.v = 001110 \in L_1L_2$

Definición: Iteración de un lenguaje

Notamos $\mathbf{L^n}$ a la iteración n-ésima del lenguaje L como a la concatenación de las palabras de L con las palabras de L **n** veces. Su definición es recursiva:

- $\mathbf{L^0} = \varepsilon$
- $\mathbf{L^{i+1}} = \mathbf{L^i}.\mathbf{L}$, para todo i>=0.

Definición: Cláusula de Kleene

Si L es un lenguaje, entonces la cláusula de Kleene sobre L se define como:

$$L^* = \bigcup_{i \geq 0} L^i$$

$$L^+ = \bigcup_{i > 0} L^i$$

Propiedades de la cláusula de Kleen

$$L^+ = L^* \text{ si } \varepsilon \in L$$

$$L^+ = L^* - \{\varepsilon\} \text{ si } \varepsilon \notin L$$

Ejemplo de cláusula de Kleene

$L=\{0,01\}$

- L^*= Conjunto de palabras formadas por 0's y 1's donde un 1 siempre va precedido por un 0.
- L^+= Conjunto de palabras formadas por 0's y 1's donde un 1 siempre va precedido por un 0, excluyendo la palabra vacía.

Definición: Lenguaje inverso

El lenguaje inverso $\mathbf{L^{-1}}$ del lenguaje L es aquel tal que:

$$L^{-1} = \{u | u^{-1} \in L\}$$

Definición: Cabecera de un lenguaje.

La cabecera de un lenguaje **CAB(L)** se define como el conjunto de palabras del alfabeto (no necesariamente del lenguaje) para las que, añadiendo otra palabra más, da una del lenguaje:

$$CAB(L) = \{u \in A^* | \exists v \in A^* \; u.v \in L\}$$

Ejemplo de cabecera de un lenguaje

Sea $L=\{a^i b^i \mid i>=0\}$

Entonces $CAB(L)= \{a^i b^j \mid i>=j>=0\}$

Definición: Homomorfismo

Supongamos $\mathbf{A_1}$, $\mathbf{A_2}$ **dos alfabetos**, y una aplicación $\mathbf{h: A_1 \rightarrow A_2}$.

Diremos que *h* es un homomorfismo si se cumple:

$$h(uv) = h(u)h(v)$$

Para cualquier par de palabras $u \in A_1, v \in A_2$

Consecuencias

- $h(\varepsilon) = \varepsilon$
- $h(a_1 \ldots a_n) = h(a_1)\, h(a_2) \ldots h(a_n)$

Ejemplo de homomorfismo

- Sean A_1={0,1,2,3,4,5,6,7,8,9}, A_2={0,1}
- Sea h la aplicación siguiente:
  h(0) = 0000, h(1) = 0001, h(2) = 0010, h(3) = 0011
  h(4) = 0100, h(5) = 0101, h(6) = 0110, h(7) = 0111
  h(8)= 1000, h(9)= 1001
  - **h** es un homomorfismo porque h(uv)=h(u)h(v) para cualquier par de palabras $u \in A_1, v \in A_2$
- Ejemplo: h(034)= h(0)h(3)h(4)= 000000110100

Pregunta 1: ¿Verdadero o falso?

Si A es un alfabeto cualquiera, la aplicación que transforma cada palabra $u \in A^*$ en su inversa es un homomorfismo de A^* en A^*

Pregunta 1: ¿Verdadero o falso?

Si A es un alfabeto cualquiera, la aplicación que transforma cada palabra $u \in A^*$ en su inversa es un homomorfismo de A^* en A^*

Respuesta:

FALSO. Por ejemplo, si A={0, 1}, entonces la palabra de dos símbolos u=01 tiene como inversa h(01)= 10, pero no se cumple h(01)!= h(0)h(1)=01

Pregunta 2: ¿Verdadero o falso?

La transformación que a cada palabra sobre $\{0, 1\}^*$ le añade 00 al principio y 11 al final es un homomorfismo.

Pregunta 2: ¿Verdadero o falso?

La transformación que a cada palabra sobre $\{0, 1\}^*$ le añade 00 al principio y 11 al final es un homomorfismo.

Respuesta:

FALSO. Por ejemplo, si A={0, 1}, entonces la palabra de dos símbolos u=01 tiene como valor h(01)= 001011, pero no se cumple h(01)!= h(0)h(1)=00011 00111

Definición: Gramáticas generativas

Una gramática generativa es un cuadrupla (V, T, P, S) donde:

- V es un alfabeto, llamado de variables o símbolos no terminales. Sus elementos se suelen representar con letras mayúsculas.
- T es un alfabeto, llamado de símbolos terminales. Sus elementos se suelen representar con letras minúsculas.
- P es un conjunto de pares (α,β), llamados reglas de producción, donde $\alpha, \beta \in (V \cup T)^*$ y α contiene, al menos un símbolo de V. El par (α,β) se suele representar como α→β
- S es un elemento de V , llamado símbolo de partida.

Ejemplo de gramática generadora de expresiones

- V= E → Símbolos no terminales
- T= {+,*,(,),a,b,c} → Símbolos terminales
- P = Reglas de producción:

  E→E+E E→E*E E→(E) E→a E→b E→c

- S= E → Símbolo inicial de partida

Gramáticas para generar palabras de un lenguaje

Una gramática sirve para determinar un lenguaje. Las palabras son las de T* que se obtienen a partir del símbolo inicial efectuando pasos de derivación. Cada paso consiste en elegir una parte de la palabra que coincide con la parte izquierda de una producción y sustituir esa parte por la derecha de la misma producción.

Ejemplo de gramática generadora de expresiones

- V= E → Símbolos no terminales
- T= {+,*,(,),a,b,c} → Símbolos terminales
- P = Reglas de producción:

  E→E+E E→E*E E→(E) E→a E→b E→c

- S= E → Símbolo inicial de partida

Ejemplo de generación de una palabra del lenguaje asociado

- S= E => E*E

Ejemplo de gramática generadora de expresiones

- V= E → Símbolos no terminales
- T= {+,*,(,),a,b,c} → Símbolos terminales
- P = Reglas de producción:

  E→E+E E→E*E E→(E) E→a E→b E→c

- S= E → Símbolo inicial de partida

Ejemplo de generación de una palabra del lenguaje asociado

- S= E => E*E => (E)*E

Ejemplo de gramática generadora de expresiones

- V= E → Símbolos no terminales
- T= {+,*,(,),a,b,c} → Símbolos terminales
- P = Reglas de producción:

  E→E+E E→E*E E→(E) E→a E→b E→c

- S= E → Símbolo inicial de partida

Ejemplo de generación de una palabra del lenguaje asociado

- S= E => E*E => (E)*E => (E+E)*E

Ejemplo de gramática generadora de expresiones

- V= E → Símbolos no terminales
- T= {+,*,(,),a,b,c} → Símbolos terminales
- P = Reglas de producción:

  E→E+E E→E*E E→(E) E→a E→b E→c

- S= E → Símbolo inicial de partida

Ejemplo de generación de una palabra del lenguaje asociado

- S= E => E*E => (E)*E => (E+E)*E => (a+E)*E

Ejemplo de gramática generadora de expresiones

- V= E → Símbolos no terminales
- T= {+,*,(,),a,b,c} → Símbolos terminales
- P = Reglas de producción:

  E→E+E   E→E*E   E→(E)   E→a   E→b   E→c

- S= E → Símbolo inicial de partida

Ejemplo de generación de una palabra del lenguaje asociado

- S= E => E*E => (E)*E => (E+E)*E => (a+E)*E => (a+b)*E

Ejemplo de gramática generadora de expresiones

- V= E → Símbolos no terminales
- T= {+,*,(,),a,b,c} → Símbolos terminales
- P = Reglas de derivación:

  E→E+E E→E*E E→(E) E→a E→b E→c
- S= E → Símbolo inicial de partida

Ejemplo de generación de una palabra del lenguaje asociado

- S= E => E*E => (E)*E => (E+E)*E => (a+E)*E => (a+b)*E => **(a+b)*b**
- Todos los símbolos resultantes son terminales (conjunto T). No se pueden aplicar más reglas de derivación.

Ejemplo de gramática generadora de expresiones

- V= E → Símbolos no terminales
- T= {+,*,(,),a,b,c} → Símbolos terminales
- P = Reglas de derivación:

  E→E+E E→E*E E→(E) E→a E→b E→c
- S= E → Símbolo inicial de partida

Ejemplo de generación de una palabra del lenguaje asociado

- S= E => E*E => (E)*E => (E+E)*E => (a+E)*E => (a+b)*E => **(a+b)*b → palabra generada**
- Todos los símbolos resultantes son terminales (conjunto T). No se pueden aplicar más reglas de derivación.

Definición: Paso de derivación

Sea una gramática G = (V, T, P, S) y dos palabras $\alpha,\beta \in (V \cup T)^*$
Se dice que β es derivable a partir de α en un paso ($\alpha \Rightarrow \beta$) si, y solo si:

existe una producción $Y \rightarrow \Phi$ tal que α contiene a Y como subcadena y β se obtiene sustituyendo Y por Φ en α

Ejemplo de derivación en un paso

- Reglas de derivación:

  $E \rightarrow E+E \quad E \rightarrow E*E \quad E \rightarrow (E) \quad E \rightarrow a \quad E \rightarrow b \quad E \rightarrow c$

- Una posible derivación en un paso: (E)*E => (E+E)*E

Definición: Secuencia de Derivación

β es derivable de α , y lo notamos $\alpha \overset{*}{\Rightarrow} \beta$, si y solo si existe una sucesión de palabras $Y_{1, \ldots,} Y_n$ (n>=1) tales que:

$$\alpha = Y_1 \Rightarrow Y_2 \Rightarrow \ldots Y_n \Rightarrow \beta$$

Ejemplo de derivación

- **(a+b)*b se deriva de E** , mediante la siguiente secuencia de derivación:

E => E*E => (E)*E => (E+E)*E => (a+E)*E => (a+b)*E => **(a+b)*b**

Definición: Lenguaje generado por una gramática

Sea una gramática generativa G=(V, T, P, S). Denominamos al lenguaje L generado por la gramática L(G) como:

$$L(G) = \{u \in T^* | S \overset{*}{\Rightarrow} u\}$$

Ejemplo

Sea G=(V,T,P,S), donde V ={S,A,B}, T={a,b}, el símbolo de partida es S y las reglas en P son:

S→aB S→bA A→a A→aS A→bAA B→b B→bS B→aBB

¿Qué lenguaje genera esta gramática?

Ejemplo

Sea G=(V,T,P,S), donde V ={S,A,B}, T={a,b}, el símbolo de partida es S y las reglas en P son:

S→aB S→bA A→a A→aS A→bAA B→b B→bS B→aBB

¿Qué lenguaje genera esta gramática?

Solución

Genera el lenguaje $L = \{u \in \{a,b\}^+ | N_a(u) = N_b(u)\}$,

donde $N_x(u)$ es el número de veces que aparece el símbolo x en la palabra u.

Ejemplo

Sea G=(V,T,P,S), donde V ={S,X,Y}, T={a,b,c}, el símbolo de partida es S y las reglas en P son:

S→abc S→aXbc Xb→bX Xc→Ybcc bY→Yb aY→aaX aY→ aa

¿Qué lenguaje genera esta gramática?

Solución

Genera el lenguaje $L = \{a^i b^i c^i, i > 0\}$,

donde $N_x(u)$ es el número de veces que aparece el símbolo x en la palabra u.

DECSAI

Jerarquía de lenguajes de Chomsky

- **Lenguajes de Tipo 0:** Cualquier gramática. Sin restricciones. Da como resultado lenguajes recursivamente enumerables.

- **Lenguajes de Tipo 1: Dependientes del contexto.** Todas las reglas de producción tienen la forma $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$, donde $\alpha_1, \alpha_2, \beta \in (V \cup T)^*, A \in V, \beta \neq \varepsilon$ excepto para S→ε, donde S no aparece a la derecha de las reglas.

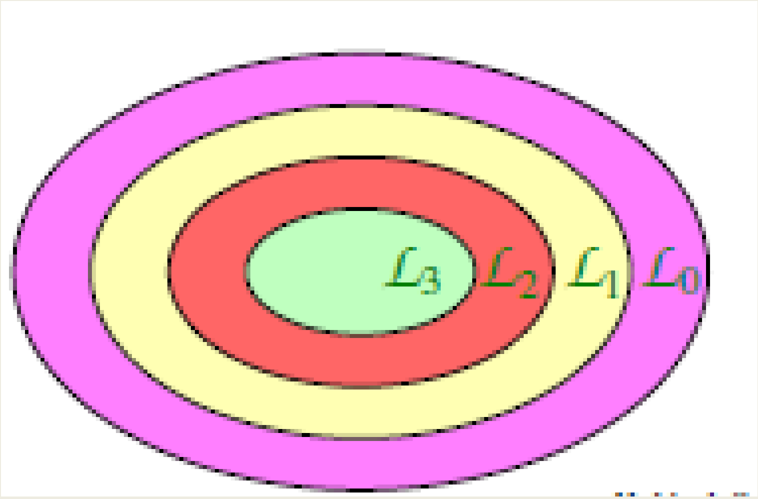
- **Lenguajes de Tipo 2: Independientes del contexto.** Toda producción tiene la forma A→α, donde $\alpha \in (V \cup T)^*, A \in V$

- **Lenguajes de Tipo 3: Regulares.** Todas las reglas tienen la forma A→ αB ó A→ α , donde $\alpha \in (V \cup T)^*\ y\ A, B \in V$

Clases de lenguajes

- Un lenguaje se dice de tipo *i (i=0,1,2,3)* si es generado por una gramática de tipo *i*.
- La clase o familias de lenguajes generados por gramáticas de tipo *i* se denomina L_i

$$L_3 \subseteq L_2 \subseteq L_1 \subseteq L_0$$



Ejercicio 1

Demostrar que la gramática G=({S}, {a,b} {S→Ɛ, S→aSb}, S) genera el lenguaje $L=\{a^i b^i, i>=0\}$

Ejercicio 2

¿Cuál es el lenguaje generado por la siguiente gramática?

G= ({A,B,C}, {a,b}, P, S), con P= {S→aAB, bB→a, Ab→SBb, Aa→SaB, B→SA, B→ab}

Ejercicio 1

Demostrar que la gramática $G=(\{S\}, \{a,b\} \{S \rightarrow \varepsilon, S \rightarrow aSb\}, S)$ genera el lenguaje $L=\{a^i b^i, i>0\}$

Ejercicio 2

¿Cuál es el lenguaje generado por la siguiente gramática?

$G= (\{A,B,C\}, \{a,b\}, P, S)$, con $P= \{S \rightarrow aAB, bB \rightarrow a, Ab \rightarrow SBb, Aa \rightarrow SaB, B \rightarrow SA, B \rightarrow ab\}$

Solución

El lenguaje vacío. No existe una derivación que lleve a contener sólo símbolos de T ({a,b})

Ejercicios

Encontrar gramáticas libres del contexto que generen cada uno de los siguientes lenguajes:

1. $L = \{a^i b^j \mid i, j \in N, i \leq j\}$
2. $L = \{a^i b^j a^j b^i \mid i, j \in N\}$
3. $L = \{a^i b^i a^j b^j \mid i, j \in N\}$
4. $L = \{a^i b^i \mid i \in N\} \cup \{b^i a^i \mid i \in N\}$
5. $L = \{uu{-}1 \mid u \in \{a, b\}^*\}$
6. $L = \{a^i b^j c^{i+j} \mid i, j \in N\}$

Donde N denota al conjunto de los naturales (incluyendo el 0)

7. Determinar si el lenguaje generado por la siguiente gramática es de tipo 3: $S \rightarrow AB$, $A \rightarrow Ab$, $A \rightarrow a$, $B \rightarrow cB$, $B \rightarrow d$

Ejercicio 1: Solución

$L = \{a^i b^j \mid i, j \in N, i \leq j\}$

$S \rightarrow \varepsilon$, $S \rightarrow aSb$, $S \rightarrow Sb$

Ejercicio 2: Solución

$L = \{a^i b^j a^j b^i \mid i, j \in N\}$

$$S \rightarrow aSb,\ S \rightarrow B,\ B \rightarrow bBa,\ B \rightarrow \varepsilon$$

Ejercicio 3: Solución

$L=\{a^i b^i a^j b^j \mid i, j \in N\}$

$$S_1 \rightarrow aS_1b,\ S_1 \rightarrow \varepsilon$$

$$S \rightarrow S_1S_1$$

Ejercicio 4: Solución

$L=\{a^i b^i \mid i \in N\} \cup \{b^i a^i \mid i \in N\}$

$$S_1 \rightarrow aS_1b,\ S_1 \rightarrow \varepsilon$$

$$S_2 \rightarrow bS_2a,\ S_2 \rightarrow \varepsilon$$

$$S \rightarrow S_1$$

$$S \rightarrow S_2$$

Ejercicio 5: Solución

$L=\{uu^{-1} \mid u \in \{a,b\}^{+}\}$

$S \rightarrow aSa$, $S \rightarrow bSb$ $S \rightarrow \varepsilon$

Ejercicio 6: Solución

$L=\{a^i b^j c^{i+j} \mid i, j \in N\}$

$$S \rightarrow aSc,\ S \rightarrow B,\ B \rightarrow bBc,\ B \rightarrow \varepsilon$$

Ejercicio 7: Solución

Determinar si el lenguaje generado por la siguiente gramática es de tipo 3:

S→AB, A→Ab, A→a, B→cB, B→d

El lenguaje generado es $L=\{ab^ic^jd \mid i, j \in N\}$, que también puede ser generado por la siguiente gramática

S→aB, B→bB, B→C, C→cC C→d

Como la gramática es de tipo 3, el lenguaje también es de tipo 3.

UNIVERSIDAD DE GRANADA

DECSAI

Modelos de Computación

Grado en Ingeniería Informática

Tema 1 – Introducción a la computación

Manuel Pegalajar Cuéllar
manupc@ugr.es

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