Tema 1: Introducción a la Computación

Serafín Moral

Universidad de Granada

Septiembre, 2016

Tema 1: Introducción a la Computación

- Breve introducción histórica a la computación
- Definiciones: palabras y lenguajes
- Operaciones con palabras y lenguajes
- Gramáticas
- Jerarquía de Chomsky

La Computación

Vamos a tratar de responder varias preguntas:

- ¿Qué puede ser resuelto de forma automática?
- ¿Qué puede ser resuelto de forma eficiente?
- ¿Qué estructuras son comunes en la computación con palabras y símbolos y cómo se pueden procesar en un ordernador?

Consecuencia: *Estudio de Modelos de Computación*

Historia

- *Russell, Hilbert y Boole:* Los precursores
- *Turing y Church:* Los primeros años (30 y 40)
- FORTRAN, COBOL, LISP: Los primeros lenguajes
- *Rabin, Scott, Chomsky:* Autómatas y Lenguajes Formales (los 50).
- *Hartmanis, Lewis, Cook:* Complejidad Algorítmica (los 60)
- Complejidad (circuitos, tiempo-espacio), semántica y estructuras de datos (los 70)
- Complejidad paralela, computación distribuida y criptografía (los 80 y 90)

El problema de la parada

¿Existe un programa que lea un programa y unos datos y nos diga si ese programa termina o cicla indefinidamente?
No existe, ya que si existiera (programa `Stops(P,x)`) podríamos construir el algoritmo `Turing(P)` con entrada `P`.

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

¿Cual es el resultado de `Turing(Turing)`?

Alfabetos

Un **alfabeto** es un conjunto finito A. Sus elementos se llamarán símbolos o letras.

Notación - *Alfabetos:* $A, B, C, \ldots$.
Símbolos: $a, b, c, \ldots$ o números.

Ejemplos

- $A = \{0,1\}$
- $B = \{<0,0>,<0,1>,<1,0>,<1,1>\}$

Palabras

Una **palabra** sobre el alfabeto A es una sucesión finita de elementos de A.

$$u = a_1 \dots a_n$$

donde $a_i \in A$, $\forall i = 1, \dots, n$.

Ejemplo

Si $A = \{0, 1\}$ entonces 0111 es una palabra sobre este alfabeto.

Palabras

El conjunto de todas las palabras sobre un alfabeto A se nota como A^*.

Notación - *Palabras:* $u, v, x, y, z, \ldots$.
Si $u \in A^*$, entonces la *longitud* de la palabra u es el número de símbolos de A que contiene.
Notación: $|u|$
Si $u = a_1 \ldots a_n$, entonces $|u| = n$.
La *palabra vacía* es la palabra de longitud cero.
Notación: ε

Notación: El conjunto de cadenas sobre un alfabeto A excluyendo la cadena vacía se nota como A^+.

Operaciones: Concatenación

Si $u, v \in A^*$, $u = a_1 \dots a_n$, $v = b_1 \dots b_m$, se llama **concatenación** de u y v a la cadena $u.v$ (o simplemente uv) dada por $a_1 \dots a_n b_1 \dots b_m$.

Ejemplo

Si $u = 011, v = 1010$, entonces $uv = 0111010$

Propiedades

1. $|u.v| = |u| + |v|, \quad \forall u, v \in A^*$
2. *Asociativa.-* $u.(v.w) = (u.v).w, \quad \forall u, v, w \in A^*$
3. *Elemento Neutro.-* $u.\varepsilon = \varepsilon.u = u, \quad \forall u \in A^*$

Estructura de monoide

Prefijos y Sufijos

Prefijo

Si $u \in A^*$ entonces v es un prefijo de u si $\exists z \in A^*$ tal que $vz = u$
Un prefijo v de u se dice propio si $v \neq \varepsilon$ y $v \neq u$.

Sufijo

Si $u \in A^*$ entonces v es un sufijo de u si $\exists z \in A^*$ tal que $zv = u$
Un sufijo v de u se dice propio si $v \neq \varepsilon$ y $v \neq u$.

Subcadena

Si $u \in A^*$ v es una subcadena de u si $\exists z_1, z_2 \in A^*$ tal que $z_1 v z_2 = u$. Una subcadena se dice propia si $v \neq \varepsilon, u$

Iteración y Cadena Inversa

Iteración n-ésima de una cadena (u^n) como la concatenación con ella misma n veces.

Si $u \in A^*$ entonces

- $u^0 = \varepsilon$
- $u^{i+1} = u^i.u, \quad \forall i \geq 0$

Ejemplo

Si $u = 010$, entonces $u^3 = 010010010$.

Si $u = a_1 \dots a_n \in A^*$, entonces la *cadena inversa* de u es la cadena $u^{-1} = a_n \dots a_1 \in A^*$.

Ejemplo

Si $u = 011$, entonces $u^{-1} = 110$.

Lenguajes

Un **lenguaje** sobre el alfabeto A es un subconjunto del conjunto de las cadenas sobre A: $L \subseteq A^*$.

Notación - *Lenguajes:* $L, M, N, \ldots$.

Ejemplos:

- $L_1 = \{a, b, \varepsilon\}$ *Tres palabras*
- $L_2 = \{a^i b^i \mid i = 0, 1, 2, \ldots\}$ Una sucesión de a seguida de una de b de la misma longitud
- $L_3 = \{uu^{-1} \mid u \in A^*\}$ *Palíndromos de longitud par*
- $L_4 = \{a^{n^2} \mid n = 1, 2, 3, \ldots\}$ Sucesiones de a de longitud un 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

A^* es siempre numerable. Si $A = \{a_1, \ldots, a_n\}$ entonces puedo asignar un número binario distinto de 0 y de la misma longitud a cada a_i de tal manera que símbolos distintos reciben números distintos y a cada palabra $b_1 \ldots b_k$ se le asigna el número cuya representación en binario es el que se obtiene sustituyendo cada b_i por su número binario.

Ejemplo: El conjunto de programas bien escritos en C es numerable.

Un conjunto no numerable

Ejemplo: El conjunto de lenguajes sobre A^* (si A no es vacío) nunca es numerable.
Haremos la demostración por reducción al absurdo.
Si lo fuese, se podría asignar un número natural distinto $f(L)$ a cada lenguaje L.
Sea $a \in A$.
Definamos el lenguaje L formado por palabras de la forma a^i de acuerdo a lo siguiente: para cada i número natural:

- Si este número no es de un lenguaje, entonces $a^i \in L$.
- Si este número es del lenguaje, M, $(i = f(M))$
  - Si $a^i \notin M$ entonces $a^i \in L$.
  - Si $a^i \in M$ entonces $a^i \notin L$.

L no puede tener ningún número asociado. Si fuese $j = f(L)$, entonces la pertenencia de a^j a L es contradictoria:

- Si $a^j \in L$ como $j = f(L)$, entonces $a^j \notin L$
- Si $a^j \notin L$ y $j = f(L)$, entonces $a^j \in L$

Operaciones

Aparte de las operaciones de unión e intersección de lenguajes, dada su condición de conjuntos existe la operación de concatenación.

Si L_1, L_2 son dos lenguajes sobre el alfabeto A, la **concatenación** de estos dos lenguajes se define como,

$$L_1 L_2 = \{u_1 u_2 \mid u_1 \in L_1, u_2 \in L_2\}$$

Propiedades

- $L\emptyset = \emptyset L = \emptyset$
- *Elemento Neutro.-* $\{\varepsilon\}L = L\{\varepsilon\} = L$
- *Asociativa.-* $L_1(L_2 L_3) = (L_1 L_2)L_3$

Ejemplo

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

$L_1 L_2 = \{0^i 1^i 1^j 0^j : i, j \geq 0\}$

Operaciones

La **iteración** de lenguajes se define de forma recursiva:

$$L^0 = \{\varepsilon\}, \qquad L^{i+1} = L^i L$$

Si L es un lenguaje sobre el alfabeto A, la *clausura de Kleene* de L es

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

$$L^+ = \bigcup_{i \geq 1} L^i$$

Operaciones

Propiedades:

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

Ejemplo

Si $L = \{0, 01\}$, entonces:
$L^* =$ Conjunto de palabras sobre $\{0, 1\}$ en las que un uno va siempre precedido de un cero.

$L^+ =$ Conjunto de palabras sobre $\{0, 1\}$ en las que un uno va siempre precedido de un cero y distintas de la palabra vacía.

Lenguaje Inverso

El **lenguaje inverso** de L es el lenguaje dado por:

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

La **cabecera** de L es el lenguaje dado por

$$\mathrm{CAB}(L) = \{u \mid u \in A^* \text{ y } \exists v \in A^* \text{ tal que } uv \in L\}$$

Es el conjunto de prefijos de todas las palabras de L.

Ejemlo

Si $L = \{0^i 1^i \, : \, i \geq 0\}$, entonces $\mathrm{CAB}(L) = \{0^i 1^j \, : \, i \geq j \geq 0\}$.

Operaciones

Si A_1 y A_2 son dos alfabetos, una aplicación

$$h : A_1^* \to A_2^*$$

se dice que es un **homomorfismo** si y solo si

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

Consecuencias

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

Si $A_1 = \{0,1,2,3,4,5,6,7,8,9\}, A_2 = \{0,1\}$

$$
\begin{array}{llll}
h(0) = 0000, & h(1) = 0001, & h(2) = 0010, & h(3) = 0011 \\
h(4) = 0100, & h(5) = 0101, & h(6) = 0110, & h(7) = 0111 \\
h(8) = 1000 & h(9) = 1001 & &
\end{array}
$$

$$h(034) = 000000110100, \qquad h(\varepsilon) = \varepsilon$$

Preguntas

¿Verdadero of falso?

- Si A es un afabeto, la aplicación que transforma cada palabra $u \in A^*$ en su inversa es un homomorfismo de A^* en A^*.
  FALSO $f(u.v) = v_n v_{n-1} \cdots v_1 u_n \cdots u_1 = v^{-1} u^{-1} \neq f(u) f(v)$
- La transformación que a cada palabra sobre $\{0,1\}^*$ le añade 00 al principio y 11 al final es un homomorfismo.
  FALSO

Gramáticas Generativas

Una **gramática generativa** es un cuadrupla (V,T,P,S) en la que

- 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 finito 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 $\alpha \to \beta$. ("Trocito" α se puede cambiar por uno β).
- S es un elemento de V, llamado símbolo de partida.

Gramática

$G = (V, T, P, S)$ dada por,

- $V = \{E\}$
- $T = \{+, *, (,), a, b, c\}$
- P está compuesto por las siguientes reglas de producción

$$\begin{array}{lll} E \rightarrow E + E, & E \rightarrow E * E, & E \rightarrow (E), \\ E \rightarrow a, & E \rightarrow b, & E \rightarrow c \end{array}$$

- $S = E$

Lenguaje Generado: idea intuitiva

Una gramática sirve para determinar un lenguaje.

$$\begin{array}{lll} E \rightarrow E+E, & E \rightarrow E*E, & E \rightarrow (E), \\ E \rightarrow a, & E \rightarrow b, & E \rightarrow c \end{array}$$

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

$EE \Rightarrow E*E \Rightarrow (E)*E \Rightarrow (E+E)*E \Rightarrow (a+E)*E \Rightarrow$

$(a+b)*E \Rightarrow (a+b)*b$ Palabra Generada

Paso de Derivación

Gramática $G = (V, T, P, S)$ y dos palabras $\alpha, \beta \in (V \cup T)^*$
β es **derivable** a partir de α **en un paso** $(\alpha \Longrightarrow \beta)$

si y solo si

existe una producción $\gamma \to \varphi$ tal que

- α contiene a γ como subcadena.
- β se obtiene sustituyento γ por φ en α .

Secuencia de Derivación

β es **derivable** de α $(\alpha \overset{*}{\Longrightarrow} \beta)$,
si y solo si existe una sucesión de palabras $\gamma_1, \ldots, \gamma_n$ $(n \geq 1)$ tales que

$$\alpha = \gamma_1 \Longrightarrow \gamma_2 \Longrightarrow \ldots \Longrightarrow \gamma_n = \beta$$

Lenguaje Generado

Lenguaje generado por una gramática $G = (V, T, P, S)$ al conjunto de cadenas formadas por símbolos terminales y que son derivables a partir del símbolo de partida.
Es decir,

$$L(G) = \{u \in T^* \mid S \stackrel{*}{\Longrightarrow} u\}$$

Ejemplo

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

$$\begin{array}{llll} S \to aB, & S \to bA, & A \to a, & A \to aS, \\ A \to bAA, & B \to b, & B \to bS, & B \to aBB \end{array}$$

Esta gramática genera el lenguaje

$$L(G) = \{u \mid u \in \{a, b\}^+ \text{ y } N_a(u) = N_b(u)\}$$

donde $N_a(u)$ y $N_b(u)$ son el número de apariciones de símbolos a y b, en u, respectivamente.

Demostración

Esto es fá cil de ver interpretando,

- A palabra con una a de más
- B palabra con una b de más
- S palabra con igual número de a que de b.

Hay que demostrar dos cosas:

- Todas las palabras generadas por la gramática tienen el mismo número de a que de b.
- Cualquier palabra con el mismo número de a que de b es generada.

Demostración

Para lo primero, podemos considerar $N_{a,A}(\alpha)$ (número de a + número de A) y $N_{b,B}(\alpha)$ (número de b + número de B) y tener en cuenta lo siguiente para una generación $S \stackrel{*}{\Longrightarrow} u$:

- Al principio de la generación tenemos: $N_{a,A}(S) = N_{b,B}(S) = 0$
- Al aplicar cualquier regla $\alpha_1 \Rightarrow \alpha_2$, si $N_{a,A}(u\alpha_1) = N_{b,B}(\alpha_1)$, entonces $N_{a,A}(\alpha_2) = N_{b,B}(\alpha_2)$
- Luego al final $N_{a,A}(u) = N_{b,B}(u)$, y como u no contiene variables, $N_a(u) = N_b(u)$, como se quería demostrar.

Algoritmo de Generación

Generación por la izquierda, un símbolo cada vez.

- Para generar una a
  - Si a último símbolo de la palabra, aplicar $A \to a$
  - Si no es el último símbolo
    - Si la primera variable es S aplicar $S \to aB$
    - Si la primera variable es B aplicar $B \to aBB$
    - Si la primera variable es A
      - Si hay más variables aplicar $A \to a$
      - Si no hay más, aplicar $A \to aS$
- Para generar una b
  - Si b último símbolo de la palabra, aplicar $B \to b$
  - Si no es el último símbolo
    - Si la primera variable es S aplicar $S \to bA$
    - Si la primera variable es A aplicar $A \to bAA$
    - Si la primera variable es B
      - Si hay más variables aplicar $B \to b$
      - Si no hay más, aplicar $B \to bS$

Condiciones de Garantía

- Las palabras generadas tienen primero símbolos terminales y después variables.
- Se genera un símbolo de la palabra en cada paso de derivación
- Las variables que aparecen en la palabra pueden ser:
  - Una cadena de A (si hemos generado más b que a)
  - Una cadena de B (si hemos generado más a que b)
  - Una S si hemos generado las mismas a que b
- Antes de generar el último símbolo tendremos como variables:
  - Una A si tenemos que generar a
  - Una B si tenemos que generar b
- Entonces aplicamos la primera opción para generar los símbolos y la palabra queda generada.

Ejemplo

Sea $G = (\{S, X, Y\}, \{a, b, c\}, P, S)$ donde P tiene las reglas,

$$
\begin{array}{llll}
S \rightarrow abc & S \rightarrow aXbc & Xb \rightarrow bX & Xc \rightarrow Ybcc \\
bY \rightarrow Yb & aY \rightarrow aaX & aY \rightarrow aa &
\end{array}
$$

Lenguaje Generado

$$S \rightarrow abc \qquad S \rightarrow aXbc \qquad Xb \rightarrow bX \qquad Xc \rightarrow Ybcc$$
$$bY \rightarrow Yb \qquad aY \rightarrow aaX \qquad aY \rightarrow aa$$

Esta gramática genera el lenguaje: $\{a^n b^n c^n \mid n = 1, 2, \ldots\}$.

Inicialmente tenemos dos posibilidades:

$$S \Longrightarrow abc, \qquad S \Longrightarrow aXbc$$

Con la primera generamos la palabra abc.
Si seguimos la segunda opción, sólo se puede seguir cómo:

$$aXbc \Longrightarrow abXc \Longrightarrow abYbcc \Longrightarrow aYbbcc$$

Lenguaje Generado: $\{a^n b^n c^n \mid n = 1, 2, \ldots\}$

$S \rightarrow abc \qquad S \rightarrow aXbc \qquad Xb \rightarrow bX \qquad Xc \rightarrow Ybcc$

$bY \rightarrow Yb \qquad aY \rightarrow aaX \qquad aY \rightarrow aa$

Ya tenemos abc.

Habíamos generado también: $aYbbcc$. En este momento podemos aplicar dos reglas:

- $aY \rightarrow aa$, en cuyo caso producimos $aabbcc = a^2b^2c^2 \in L(G)$
- $aY \rightarrow aaX$, en cuyo caso producimos $aaXbbcc$

A partir de $aaXbbcc$ repetimos el proceso: sólo se puede mover la X a la derecha hasta la frontera $b-c$. Entonces se añade una b y una c y se cambia X por Y. Después, se mueve la Y a la izquierda hasta que encuentra la frontera $a-b$. Entonces, tiene dos opciones: añadir sólo a, obteniendo la siguiente palabra, o aX con lo que se vuelven a generar las otras palabras.

Jerarquía de Chomsky

- **Tipo 0** Cualquier gramática. Sin restricciones. **Lenguajes recursivamente enumerables**.
- **Tipo 1** Si todas las producciones tienen la forma

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$$

donde $\alpha_1, \alpha_2, \beta \in (V \cup T)^*, A \in V,$ y $\beta \neq \varepsilon$, excepto posiblemente la regla $S \rightarrow \varepsilon$, en cuyo caso S no aparece a la derecha de las reglas. **Lenguajes dependientes del contexto**.

- **Tipo 2** Si cualquier producción tiene la forma

$$A \to \alpha$$

donde $A \in V, \alpha \in (V \cup T)^*$.
Lenguajes Independientes del Contexto

- **Tipo 3** Si toda regla tiene la forma

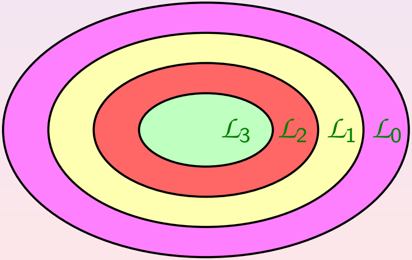
$$A \to uB \text{ ó } A \to u$$

donde $u \in T^*$ y $A, B \in V$.
Conjuntos Regulares

Un lenguaje se dice que es de tipo i ($i = 0, 1, 2, 3$) si y solo si es generado por una gramática de tipo i.
La clase o familia de lenguajes de tipo i se denota por $\mathcal{L}_i$.

Propiedad

$$\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0$$



Ejercicios

Demostrar que la gramática

$$G = (\{S\}, \{a, b\}, \{S \rightarrow \varepsilon, S \rightarrow \ \mathrm{aSb}\}, S)$$

genera el lenguaje $L = \{a^i b^i \mid i = 0, 1, 2, \ldots\}$

Inicialmente tenemos dos opciones:

$$S \Rightarrow \varepsilon, \qquad S \Rightarrow aSb$$

Con eso generamos la palabra vacía, o continuamos generando.
Otra vez hay dos opciones:
$S \Rightarrow aSb \Rightarrow ab, \qquad S \Rightarrow aSb \Rightarrow aaSbb$

Demostrar que la gramática

$$G = (\{S\}, \{a,b\}, \{S \to \varepsilon, S \to \text{aSb}\}, S)$$

genera el lenguaje $L = \{a^i b^i \mid i = 0,1,2,\ldots\}$

Si seguimos este procedimiento, nos encontramos que podemos ir generando todas las palabras de la forma $a^i b^i$, y siempre nos queda la palabra $a^i S b^i$ para seguir generando las palabras de mayor longitud.

Por otra parte, estas son las únicas palabras que se pueden generar.

Ejercicios

Encontrar el lenguaje generado por la gramática $G = (\{A, B, C\}, \{a, b\}, P, S)$ donde P contiene las siguientes producciones

$$
\begin{array}{lll}
S \rightarrow aAB & bB \rightarrow a & Ab \rightarrow SBb \\
Aa \rightarrow SaB & B \rightarrow SA & B \rightarrow ab
\end{array}
$$

El resultado es el Lenguaje vacío: nunca se puede llegar a generar una palabra con símbolos terminales. Siempre que se sustituye S aparece A, y siempre que se sustituye A aparece S.

Ejercicios

Encontrar una gramática libre del contexto para generar cada uno de los siguientes lenguajes

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

donde $\mathbb{N}$ es el conjunto de los números naturales incluyendo el 0.

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

$$S \to aSb$$

$$S \to \varepsilon$$

$$S \to Sb$$

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

Este lenguaje es generado por la siguiente gramática:

$$S \to aSb$$

$$S \to B, \quad B \to bBa, \quad B \to \varepsilon$$

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

Podemos generar $\{a^i b^i \mid i \in \mathbb{N}\}$ con:

$$S_1 \to aS_1 b, \quad S_1 \to \varepsilon$$

El lenguaje L se puede generar añadiendo:

$$S \to S_1 S_1$$

siendo S el símbolo inicial.

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

Podemos generar $\{a^i b^i \mid i \in \mathbb{N}\}$ con:

$$S_1 \to aS_1 b, \quad S_1 \to \varepsilon$$

y $\{b^i a^i \mid i \in \mathbb{N}\}$ con

$$S_2 \to bS_2 a, \quad S_2 \to \varepsilon$$

El lenguaje L se puede generar añadiendo:

$$S \to S_1, \quad S \to S_2$$

siendo S el símbolo inicial.

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

Este lenguaje se genera con la gramática:

$$S \to aSa, \quad S \to bSb, \quad S \to \varepsilon$$

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

Este lenguaje se genera con la gramática:

$$S \to aSc, \quad S \to B,$$

$$B \to bBc, \quad B \to \varepsilon$$

Ejercicios

Determinar si la gramática $G = (\{S, A, B\}, \{a, b, c, d\}, P, S)$ donde P es el conjunto de reglas de producción:

$$\begin{array}{lll} S \to AB & A \to Ab & A \to a \\ B \to cB & B \to d & \end{array}$$

genera un lenguaje de tipo 3.
Esta gramática genera el lenguaje: $\{ab^i c^j d \ : \ i,j \in \mathbb{N}\}$

Y este lenguaje se puede generar mediante la gramática:

$S \to aB, \quad B \to bB, \quad B \to C, \quad C \to cC, \quad C \to d$
Como esta gramática es de tipo 3, el lenguaje lo es.