Modelos de Computación

LibreIM

Doble Grado de Informática y Matemáticas

Universidad de Granada

libreim.github.io/apuntesDGIIM

Modelos de Computación

LibreIM

Doble Grado de Informática y Matemáticas

Universidad de Granada

libreim.github.io/apuntesDGIIM

Índice

Parte I.
Teoría

1. Introcucción a la Computación

1.1. Breve introducción histórica a la computación

La computción trata de responder a 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. . . ?

1.1.1. El problema de la parada

£Existe un programa que lea un programa y unos datos y nos diga si ese programa termina realiza ciclos indefinidamente? Concluimos que no existe, ya que si existiera (programa `Stops(P, x)`) podríamos construir el algoritmo `Turing(P )` con entrada `P`.

```
If Stops(P,P) GOTO 1
```

£Cuál sería el resultado de `Turing(Turing)`? El programa comprueba si `Turing` termina con `Turing` como dato. Por lo tanto existe un programa que termina y que puede no terminar.

1.2. Alfabetos

> **Definición 1.1 (Alfabeto).** Un alfabeto es un conjunto finito A. Sus elementos se llamarán *símbolos* o *letras*. Notaremos los alfabetos con letras mayúsculas $(A, B, C, ...)$ y a los símbolos con letras minúsculas o números $(a, b, c, 1, 2, ...)$.

Ejemplo 1.1. $A = \{0, 1\}$, $B =$ <!– compeltar el ejemplo –>

> **Definición 1.2 (Palabra).** Una palabra sobre el alfabeto A es una sucesión finita de elementos de A

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

> **Definición 1.3 (Longitud de palabra).** Si $u \in A^*$ entonces la longitud de la palabra u es el número de símbolos de A que contiene. Lo notamos como $|u|$.

La palabra vacía es la palabra de longitud cero. La notaremos ε.

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).

Definición 1.4 (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 $v \neq u$.

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

Ejemplo 1.2. Si $u = 010$, entonces $u^3 = 010010010$.

Proposición 1.1. El conjunto de lenguajes sobre A^* (si A no es vacío) nunca es numerable.

Definición 1.5. Llamamos *lenguaje generado* por una gramática $G = (V, T, P, S)$ al conjunto de cadenas formadas por símbolos terminales que son derivables a partir del símbolo de partida.

1.2.1. Jerarquía de Chomsky

- Tipo 0: Cualquier gramática, sin restricciones. Son los lenguajes recursivamente enumerables.
- Tipo 1: Si todas las producciones tienen la formaa $\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$, donde
- Tipo 2: Si cualquier producción tiene la forma $A \to \alpha$, donde $A \in V, \alpha \in (V \cup T)^*$. Los lenguajes que se generan son lenguajes independientes del contexto.
- Tipo 3: Son independientes del contecxto y además toda producción es de la forma: $A \to uB$ ó $A \to u$, donde $u \in T^*$ y $A, B \in V$. Se generan conjuntos regulares.

1.2.2. Clases de lenguajes

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 languajes de tipo i se denota por L_i.

Proposición 1.2. $L_3 \subseteq L_2 \subseteq L_1 \subseteq L_0$

2. Autómatas finitos y expresiones regulares

2.1. Autómata finito determinista

Veamos un ejemplo. Supongamos que queremos reconocer palabras que son direcciones de correo electrónico del tipo `nombre@dominio.exten`, donde `nombre` es una palabra formada por dígitos y caracteres alfabéticos, y `ddominio` y `extensión` son palabras formadas por símbolos alfabéticos. £Cómo podemos especificar un algoritmo que identifique las palabras que corresponden a este patrón?

Definición 2.1. Un *autómata finito* es una quintupla $M = (Q, A, \delta, q_0, F)$ donde:

- Q es un conjunto finito llamado *conjunto de estados*
- A es un alfabeto llamado *alfabeto de entrada*.
- δ es una aplicación llamada *función de transición* $\delta : Q \prod A \rightarrow Q$.
- q_0 es un elemento de Q, llamado *estado inciial*.
- F es un subconjunto de Q, llamado conjunto de *estados finales*.

Definición 2.2 (Diagrama de transición). Es un grafo en el que:

- Hay un nodo por cada estado
- Por cada transición $\delta(q, a) = p$ hay un arco de q a p con la etiqueta a.
- El estado inicial está indicado con un ángulo entrante. Los estados finales están indicados con una doble circunferencia.

Parte II.
Ejercicios

3. Introducción a la Computación

Ejercicio 3.1. Demostrar que la gramática

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

Ejercicio 3.2. Encontrar el lenguaje generado por la gramática $G = (\{A, B, C\}, \{a, b\}, P, S)$ donde P contiene las siguientes producciones

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

Solución. Se genera el lenguaje vacío, ya que se entra en un ciclo. Para quitar S hay que añadir una A, pero para quitar una A es necesario una S.

Ejercicio 3.3. Encontrar una

4. Tema 1 - Relación

Ejercicio 4.1. Describir el lenguaje generado por la siguiente gramática,

$$\begin{aligned} S &\to XYX \\ X &\to aX|bX|\epsilon \\ Y &\to bbb \end{aligned}$$

Solución. $L(G) = \{ub^3w : u, w \in \{a, b\}^*\} \equiv$ Palabras que contienen al menos 3 b seguidas.

Ejercicio 4.2. Describir el lenguaje generado por la siguiente gramática,

$$\begin{aligned} S &\to aX \\ X &\to aX|bX|\epsilon \end{aligned}$$

Solución. $L(G) = \{au : u \in \{a, b\}^*\} \equiv$ Palabras que empiezan por a.

Ejercicio 4.3. Describir el lenguaje generado por la siguiente gramática,

$$S \to XaXaX$$
$$X \to aX|bX|\epsilon$$

Solución. $L(G) = \{uavaw : u, v, w \in \{a, b\}^*\}$ ≡ Palabras que al menos tienen 2 a.

Este lenguaje se puede generar por una gramática de tipo 3, luego es de tipo 3, con las siguientes reglas:

$$S \to bS|aX$$
$$X \to bX|aY$$
$$Y \to bY|aY|\epsilon$$

Ejercicio 4.4. Describir el lenguaje generado por la siguiente gramática,

$$S \to SS|XaXaX|\epsilon$$
$$X \to bX|\epsilon$$

Solución. $L(G) = \{(b^i ab^j ab^k)^m : i, j, k \in \mathbb{N}, m\ espar\}$ ≡ Palabras con el patron siguiente repetido un número n par, 2 a con posibilidad de b a los lados.

Este lenguaje es de tipo 3, ya que se puede generar por la siguiente gramática de tipo 3:

$$S \to \epsilon|X$$
$$X \to bX|aY$$
$$Y \to bY|aZ|aS$$
$$Z \to aY|bZ|\epsilon$$

Ejercicio 4.5. Encontrar la gramática libre de contexto que genera el lenguaje sobre el alfabeto $\{a, b\}$ de las palabras que tienen más a que b (al menos una más).

Solución. (Por confirmar)

$$S \to aX|bSS$$
$$X \to bS|S|\epsilon$$

Ejercicio 4.6. Encontrar gramáticas de tipo 2 para los siguientes lenguajes sobre el alfabeto $\{a, b\}$. En cada caso determinar si los lenguajes generados son de tipo 3, estudiando si existe una gramática de tipo 3 que los genera.

1. Palabras en la que el número de b no es tres.
2. Palabras que tienen 2 o 3 b.
3. Palabras que no contienen la subcadena ab.

4. Palabras que no contienen la subcadena baa.

Solución. Todos son de tipo 3.

1. Reglas:

$$\begin{gathered} S \to aS|bS_1|\epsilon \\ S_1 \to aS_1|bS_2|\epsilon \\ S_2 \to aS_2|bS_3|\epsilon \\ S_3 \to aS_3|bS_4 \\ S_4 \to aS_4|bS_4|\epsilon \end{gathered}$$

2. Reglas:

$$\begin{gathered} S \to aS|bS_1 \\ S_1 \to aS_1|bS_2 \\ S_2 \to aS_2|bS_3|\epsilon \\ S_3 \to aS_3|\epsilon \end{gathered}$$

3. Reglas:

$$\begin{gathered} S \to aA|bB|\epsilon \\ A \to aA|\epsilon \end{gathered}$$

4. Reglas:

$$\begin{gathered} S \to aS|bB|\epsilon \\ B \to bB|aA|\epsilon \\ A \to bB|\epsilon \end{gathered}$$

Ejercicio 4.7. Encontrar una gramática libre del contexto que genere el lenguaje

$$L = \{1u1 | u \in \{0,1\}^*\}$$

Solución. Gramática tipo 2:

$$\begin{gathered} S \to 1X1 \\ X \to 0X|1X|\epsilon \end{gathered}$$

Si es gramática tipo 3:

$$\begin{gathered} S \to 1A \\ A \to 1A|0A|1 \end{gathered}$$

Ejercicio 4.8. Encontrar si es posible una gramática lineal por la derecha o una gramática libre del contexto que genere el lenguaje L supuesto que $L \subset \{a, b, c\}^*$ y verifica:

- $u \in L \iff u$ no contiene dos símbolos b consecutivos.
- $u \in L \iff u$ contiene dos símbolos b consecutivos.
- $u \in L \iff u$ contiene un número impar de símbolos c.
- $u \in L \iff u$ no contiene el mismo número de símbolos b que de símbolos c

(Mismo ejercicio que el 19)

Solución. Todas son de tipo 3 menos el 4.

1. Reglas:

$$S \to aS|bX|cS|\epsilon$$
$$X \to aS|cS|\epsilon$$

2. Reglas (comprobar):

$$S \to aS|bS|cS|bbX$$
$$X \to aX|bX|cX|\epsilon$$

3. Reglas:

$$S \to aS|bS|cX$$
$$X \to aX|bX|cS|\epsilon$$

4. Reglas:

$$S \to S_1|S_2$$
$$S_1 \to B|BS_1$$
$$S_2 \to C|CS_2$$
$$B \to bX|cBB|aB$$
$$C \to cX|bCC|aC$$
$$X \to bC|aX|cB|\epsilon$$

Ejercicio 4.9. Encontrar las gramáticas:

1. Dado el alfabeto $A = \{a, b\}$ determinar si es posible encontrar una gramática libre de contexto que genere las palabras de longitud impar, y mayor o igual que 3, tales que la primera letra coincida con la letra central de la palabra.
2. Dado el alfabeto $A = \{a, b\}$ determinar si es posible encontrar una gramática libre de contexto que genere las palabras de longitud par, y mayor o igual que 2, tales que las dos letras centrales coincidan.

Solución. Gramáticas de tipo 2.

1. Reglas:

$$S \to aAX|bBX$$
$$A \to a|XAX$$
$$B \to b|XBX$$
$$X \to a|b$$

2. Reglas:

$$S \to aa|bb|XSX$$
$$X \to a|b$$

Ejercicio 4.10. Determinar si el lenguaje generado por la gramática

$$S \to SS$$
$$S \to XXX$$
$$X \to aX|Xa|b$$

es regular. Justificar la respuesta.

Solución. Obtenemos el lenguaje que es: $L(G) = \{u \in \{a,b\}^* : N_b(u) = 3k, k \in \mathbb{N}\} \equiv$ Palabras que contienen b un número múltiplo de 3. Con memoria finita podemos reconocer si hay un número múltiplo de b (cada 3), luego debemos encontrar una gramática de tipo 3 (regular):

$$S \to aS|bS_1$$
$$S_1 \to aS_1|bS_2$$
$$S_2 \to aS_2|bS_3$$
$$S_3 \to aS_3|bS_1|\epsilon$$

Ejercicio 4.11. Dado un lenguaje L sobre un alfabeto A, caracterizar cuando $L^* = L$

Solución. L tiene que cumplir: $\epsilon \in L$ y que L sea submonoide. L es submonoide si: $u, v \in L \to uv \in L$

Ejercicio 4.12. Dado un lenguaje L sobre un alfabeto A, determinar si L^* es siempre, nunca o a veces numerable.

Solución. L* es siempre numerable porque A* es numerable (visto en clase) y L* está contenido en A*.

Ejercicio 4.13. Dados dos homomorfismos $f : A^* \to B^*, g : A^* \to B^*$, se dicen que son iguales si $f(x) = g(x), \forall x \in A^*$. £Existe un procedimiento algorítmico para comprobar si dos homomorfismos son iguales?

Solución. Sea $x_0 \in A^* \Longrightarrow x_0 = a_0 ... a_{n-1}, \quad , n = |A|$. Apliquemos f y g a x_0.

$$f(x_0) = f(a_0...a_{n-1}) = f(a_0)...f(a_{n-1}) = g(a_0)...g(a_{n-1}) = g(a_0...a_{n-1}) = g(x_0)$$

Luego para que esa igualdad se cumpla, f y g tienen que asignarle a cada elemento de A (que es **finito**), el mismo elemento de B. Por lo que el procedimiento algorítmico si existe, y sería comprobar que:

$$f(a_i) = g(a_i), \quad \forall i \in \{0, ..., n-1\} \text{ con } a_i \in A$$

Ejercicio 4.14. Sea $L \subseteq A^*$ un lenguaje arbitrario. Sea $C_0 = L$ y definamos los lenguajes S_i y C_i, para todo $i \geq 1$, por $S_i = C_{i-1}^+$ y $C_i = \overline{S_i}$

1. £Es S_i siempre, nunca o a veces igual a C_2? Justifica la respuesta.
2. Demostrar que $S_2 = C_3$, cualquiera que sea L. (Pista: Demuestra que C_3 es cerrado para la concateación).

Solución.

Ejercicio 4.15. Demuestra que para todo alfabeto A, el conjunto de los lenguajes finitos sobre dicho alfabeto es numerable.

Solución.

Ejercicio 4.16. Dada la gramática $G = (\{S, A\}, \{a, b\}, P, S)$ donde $P = \{S \to abAS, abA \to baab, S \to a, A \to b\}$. Determinar el lenguaje que genera.

Solución. Debido a $S \longrightarrow abAS$ y $S \longrightarrow a$, vemos que todas las palabras de $L(G)$ van a terminar en a. Después de aplicar n veces la primera regla, tenemos dos opciones: sustituir abA por $baab$, provocando así que después de cada 2 a, siempre vaya una b (ya que lo único que puede seguir en la palabra es abA), o dejar ab y sustituir A por b, luego siempre que nos encontremos a sola, le seguirán 2 b.

Por lo tanto, el lenguaje generado es:

L(G) = {palabras en las que la cadena a es seguida por bb, y la cadena aa es seguida por b, y terminan en a.}

Ejercicio 4.17. Sea la gramática $G = (V, T, P, S)$ donde:

- $V = \{< numero >, < digito >\}$
- $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $S =< numero >$
- Las reglas de producción P son:
  - $< numero > \to < numero >< digito >$
  - $< numero > \to < digito >$
  - $< digito > \to 0|1|2|3|4|5|6|7|8|9$

Determinar el lenguaje que genera.

Solución. El símbolo inicial es $S =< numero >$, o sea que siempre empezaremos por un número, que podremos sustituir por cualquier dígito del 0 al 9, generando así los números del 0 al 9. También podríamos aplicar la primera regla y volver a hacer lo anterior, generando los números del 10 al 19, y así hasta que queramos.

Por lo tanto, el lenguaje generado es: L(G) = $\{\{0,1,2,3,4,5,6,7,8,9\}^*\} = \mathbb{N}$

Ejercicio 4.18. Sea la gramática $G = (\{A, S\}, \{a, b\}, S, P)$ donde las reglas de producción son:

$$\begin{gathered} S \to aS \\ S \to aA \\ A \to bA \\ A \to b \end{gathered}$$

Determinar el lenguaje generado por la gramática.

Solución. Aplicando cualquier regla de producción con S, la palabra va a empezara por a. Después de eso solo tenemos dos opciones, o seguir añadiendo a o parar cuando apliquemos la regla $S \longrightarrow aA$ y hacer lo mismo con las b.

Por lo tanto, el lenguaje generado es: $L(G) = \{a^i b^j, \quad i, j > 0\}$