

# **LENGUAJES, GRAMÁTICAS Y AUTÓMATAS**

**UN ENFOQUE PRÁCTICO**

**Pedro Isasi Viñuela**  
**Paloma Martínez Fernández**  
**Daniel Borrajo Millán**  
Universidad Carlos III de Madrid



**ADDISON-WESLEY**

Harlow, Inglaterra • Reading, Massachusetts • Menlo Park, California • Nueva York  
Don Mills, Ontario • Amsterdam • Bonn • Sydney • Singapur  
Tokio • Madrid • San Juan • Milán • México • Seúl • Taipei

# Índice General

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Lenguajes, Gramáticas y Autómatas . . . . .	1
1.2	Estructura del libro . . . . .	5
1.3	Notaciones . . . . .	6
<b>2</b>	<b>Lenguajes y Gramáticas Formales</b>	<b>7</b>
2.1	Lenguajes . . . . .	7
2.1.1	Definiciones básicas . . . . .	7
2.1.2	Operaciones con palabras . . . . .	8
2.1.3	Operaciones con lenguajes . . . . .	9
2.1.4	Otras definiciones . . . . .	11
2.2	Gramáticas formales . . . . .	13
2.2.1	Definiciones . . . . .	13
2.2.2	Tipos de Gramáticas . . . . .	16
2.2.3	Árboles de derivación . . . . .	19
2.2.4	Ambigüedad . . . . .	20
2.2.5	Recursividad . . . . .	21
2.2.6	Factorización a izquierdas . . . . .	25
	<b>Ejercicios</b> . . . . .	<b>27</b>
<b>3</b>	<b>Gramáticas Regulares y Autómatas Finitos</b>	<b>43</b>
3.1	Gramáticas regulares . . . . .	43
3.2	Máquinas Secuenciales . . . . .	46
3.2.1	Definición . . . . .	46
3.2.2	Representación . . . . .	49
3.2.3	Extensión a palabras de la entrada y salida . . . . .	51
3.2.4	Equivalencia de Máquinas Secuenciales . . . . .	55

3.2.5	Equivalencia de Máquina de Mealy y Máquina de Moore . . . . .	61
3.3	Autómatas Finitos Deterministas (AFD) . . . . .	63
3.3.1	Definición . . . . .	63
3.3.2	Representación de un AFD . . . . .	65
3.3.3	Conceptos relativos a AFDs . . . . .	66
3.3.4	Equivalencia de AFD . . . . .	68
3.4	Autómatas Finitos No Deterministas (AFND) . . . . .	75
3.4.1	Definición . . . . .	75
3.4.2	Representación . . . . .	76
3.4.3	Conceptos asociados a AFNDs . . . . .	77
3.4.4	Autómata Finito asociado a una $G_3$ . . . . .	81
3.5	Expresiones regulares (ER) . . . . .	83
3.5.1	Definiciones . . . . .	83
3.5.2	Teoremas de Kleene . . . . .	85
3.6	Autómatas de Células de McCulloch-Pitts . . . . .	97
3.6.1	Definición . . . . .	97
3.6.2	Representación . . . . .	98
3.6.3	Construcción de un AF equivalente . . . . .	101
3.6.4	Construcción de un Autómata de Células equivalente a un AF . . . . .	106
3.7	Autómatas probabilísticos . . . . .	107
3.7.1	Definición . . . . .	108
3.7.2	Matrices de probabilidad de transición . . . . .	108
3.7.3	Vectores de estados . . . . .	109
3.7.4	Lenguaje aceptado por un AFP . . . . .	111
3.7.5	AF como AFP . . . . .	113
	<b>Ejercicios</b> . . . . .	<b>115</b>
<b>4</b>	<b>Gramáticas Independientes del Contexto y Autómatas a Pila</b>	<b>237</b>
4.1	Gramáticas Independientes del Contexto . . . . .	237
4.1.1	Definiciones . . . . .	237
4.1.2	Forma Normal de Chomsky (FNC) . . . . .	242
4.1.3	Forma Normal de Greibach (FNG) . . . . .	246
4.2	Autómatas a Pila (AP) . . . . .	248
4.2.1	Definición . . . . .	248
4.2.2	Movimientos . . . . .	251
4.2.3	Descripción instantánea . . . . .	254

4.2.4	Autómatas a Pila Deterministas . . . . .	255
4.2.5	Lenguaje aceptado por un AP . . . . .	256
4.2.6	Autómatas a Pila y Gramáticas de tipo 2 . . . . .	257
	<b>Ejercicios . . . . .</b>	<b>263</b>
<b>5</b>	<b>Gramáticas y autómatas generales . . . . .</b>	<b>321</b>
5.1	Máquinas de Turing . . . . .	321
5.1.1	Definición . . . . .	321
5.1.2	Movimiento . . . . .	323
5.1.3	Lenguaje reconocido por una Máquina de Turing . . . . .	326
5.1.4	Variantes de las Máquinas de Turing . . . . .	326
5.1.5	Máquina de Turing Universal (MTU) . . . . .	327
5.1.6	Máquinas de Turing y computación . . . . .	329
5.2	Autómatas Linealmente Acotados . . . . .	330
	<b>Ejercicios . . . . .</b>	<b>331</b>
<b>6</b>	<b>Aplicaciones . . . . .</b>	<b>343</b>
6.1	Construcción de compiladores . . . . .	343
6.1.1	Analizador Léxico . . . . .	346
6.1.2	Analizador Sintáctico . . . . .	351
6.2	Análisis del lenguaje natural . . . . .	357
6.3	Aplicaciones de Control . . . . .	368
6.4	Más aplicaciones . . . . .	372

# Prefacio

Los fundamentos teóricos de la informática, que han dado en llamarse Informática Teórica, provienen de diferentes ramas del conocimiento. Este aspecto, unido a la fuerte componente teórica que esta disciplina requiere, como no podría ser de otra forma, hace que a menudo los textos de Informática Teórica sean difíciles de leer. Sin embargo, detrás de toda esta componente teórica se esconde una disciplina amena, muy instructiva y con grandes posibilidades de aplicación. Todo esto queda a menudo oculto entre definiciones, teoremas y demostraciones. Este libro pretende ser una manera de superar este escollo. No pretende ser una sustitución de otros textos con mayor contenido teórico, sino más bien un complemento a los mismos. Todos los temas han sido tratados con el rigor exigible, aún cuando no aparezca una fuerte carga de teoremas y demostraciones. De esta forma, puede constituir un libro ideal para iniciarse en los temas tratados, o una guía rápida de consulta de los diferentes contenidos. En cualquier caso, se trata de un texto autocontenido y que no exige de conocimientos previos por parte del lector para su comprensión.

Por otra parte, tiene un alto contenido práctico, entendido como un gran número de ejercicios, en los que reflejar y complementar los contenidos teóricos. Creemos, desde nuestra perspectiva de la docencia de la asignatura que abarca estos temas en la universidad, que la mejor forma de comprender y asentar los conocimientos es con la práctica. Afortunadamente, las materias de la Informática Teórica son muy versátiles a la hora del planteamiento de ejercicios, y estos sirven de apoyo fundamental e imprescindible al texto. Es por esto que hemos querido que en el título apareciera explícitamente “un enfoque práctico”, para resaltar el hecho de que se trata de un libro con dos partes diferenciadas; por un lado se trata de un libro de teoría autocontenido, y, por otro, un libro de ejercicios completo. Esto queda remarcado por el hecho de que los ejercicios aparecen todos juntos al final de cada capítulo, como si de otro capítulo independiente se

tratase. Por último, hemos querido añadir un capítulo de aplicaciones, para mostrar algunos de los numerosos campos en los que se pueden utilizar estos conceptos.

Pedro Isasi, Paloma Martínez, y Daniel Borrajo

Leganes, Septiembre de 1997

# Capítulo 1

## INTRODUCCIÓN

Puesto que este libro abarca la Teoría de Gramáticas, Lenguajes y Máquinas Abstractas (Autómatas), en esta introducción se comenzará aclarando el significado de cada uno de los conceptos y la relación existente entre ellos. A continuación, se describirá la estructura del libro y algunas aclaraciones sobre la notación empleada a lo largo del texto.

### 1.1 Lenguajes, Gramáticas y Autómatas

Toda comunicación involucra la utilización de un lenguaje. Así, por ejemplo, las personas se comunican con el resto en los diferentes idiomas (lenguajes naturales) o con las máquinas (lenguajes artificiales) a través de conjuntos de símbolos. Se define lenguaje como un conjunto de palabras, también llamadas cadenas o sentencias, que están formadas por símbolos de un alfabeto. Así, por ejemplo, el idioma español está formado por un conjunto de palabras compuestas por letras (símbolos) del alfabeto español. Una gramática da cuenta de la estructura de un lenguaje, es decir, de las sentencias que lo forman, proporcionando las formas válidas en que se pueden combinar los símbolos del alfabeto. En el caso del español, las oraciones deben ajustarse a una gramática. Una consideración importante es la distinción entre lenguajes formales, que son los que se tratarán en este libro, y lenguajes naturales (inglés, español, etc.). Se puede decir que la diferencia estriba en que los lenguajes formales (como pueden ser los lenguajes de

programación) obedecen a reglas preestablecidas y por tanto, se ajustan a ellas, no evolucionan y han sido creados para un fin específico. Sin embargo, los lenguajes naturales (utilizados por el hombre) existen “per se” y las reglas gramaticales que rigen su estructura han sido desarrolladas con posterioridad para explicar esta última.

Una máquina abstracta o autómeta es un dispositivo teórico capaz de recibir y transmitir información. Para realizar esta labor manipula cadenas de símbolos que se le presentan a la entrada produciendo otras tiras o cadenas de símbolos a su salida mediante un conjunto de estados que contienen la información necesaria para poder deducir, dado un símbolo de entrada, cuál será el símbolo de salida en cada momento. A pesar de la conexión que existe entre estos conceptos (figura 1.1), los trabajos iniciales sobre gramáticas y máquinas abstractas tienen orígenes distintos.

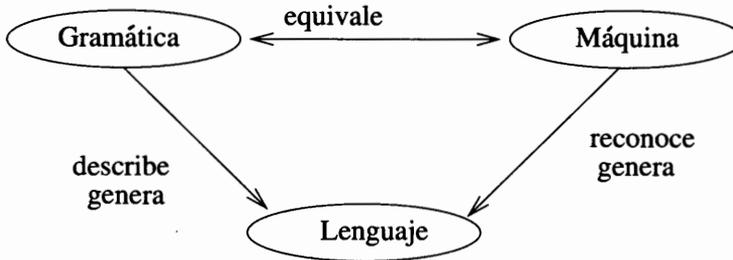


Figura 1.1: Conexiones entre los conceptos de lenguaje, gramática y autómeta.

El concepto de gramática procede de los estudios de Chomsky en su búsqueda de una descripción formalizada de las oraciones de un lenguaje natural. Chomsky clasificó las gramáticas en cuatro grandes grupos ( $G_0, G_1, G_2, G_3$ ) cada uno de los cuales incluye a los siguientes ( $G_3 \subseteq G_2 \subseteq G_1 \subseteq G_0$ ). Las gramáticas Tipo 0 se denominan gramáticas sin restricciones o gramáticas de estructura de frases; las gramáticas Tipo 1 se denominan sensibles al contexto; las gramáticas Tipo 2 se conocen como gramáticas independientes del contexto y, por último, las gramáticas Tipo 3 se denominan gramáticas regulares. Cada tipo añade restricciones al tipo inmediatamente superior y la jerarquía va desde la más general a la más restrictiva. Cada una de estas gramáticas es capaz de generar un tipo de lenguaje. Un lenguaje  $L$  se llama del tipo  $i$  ( $i=0, 1, 2, 3$ ) si existe una gramática  $G$  del tipo  $i$  capaz de generar o describir ese lenguaje.

Estos estudios previos sobre teorías de gramáticas formales y lenguajes

crearon las bases de la lingüística matemática, la cual tendría aplicación no solo al estudio del lenguaje natural sino también a los lenguajes formales. Así, los lingüistas distinguen entre gramática particular (propiedades de lenguajes concretos o artificiales) y gramática universal (propiedades generales que pueden aplicarse a cualquier lenguaje humano).

La teoría de los autómatas proviene del campo de la ingeniería eléctrica. Shannon publicó varios trabajos donde demostraba la aplicación de la lógica matemática a los circuitos combinatorios y secuenciales. Posteriormente, sus ideas se desarrollaron para dar lugar a la Teoría de Autómatas. Moore publicó el primer estudio riguroso sobre autómatas y, a finales de los años 50, se comenzó a ver la utilidad de los autómatas en relación con los lenguajes.

La teoría de lenguajes y gramáticas formales tiene una relación directa con la teoría de máquinas abstractas, siendo posible establecer entre ambas un isomorfismo. Dado que las gramáticas proporcionan las reglas utilizadas en la generación de las cadenas de un lenguaje, se puede establecer una conexión entre la clase de lenguajes generados por ciertos tipos de gramáticas y la clase de lenguajes reconocibles por ciertas máquinas. Así, se pueden identificar los lenguajes del tipo 0 con la clase de lenguajes reconocidos por una Máquina de Turing; los lenguajes del tipo 1 con los Autómatas Linealmente Acotados; los lenguajes de tipo 2 con los Autómatas a Pila y, por último, los lenguajes de tipo 3 con los Autómatas Finitos, los Autómatas Probabilísticos y los Autómatas de Células de McCulloch-Pitts. Al igual que ocurría con las gramáticas, cada tipo de máquina abstracta añade restricciones al tipo de máquina del nivel superior. Todas poseen una cinta de donde leen los símbolos de entrada, un conjunto de estados que representan diferentes fases del análisis de las palabras de entrada, un lugar donde generar la salida, y, en algunos casos, cuentan con dispositivos auxiliares de memoria. Las diferencias entre ellas estriban en la capacidad para escribir en la cinta de entrada, en los distintos tipos de movimientos que pueden realizar sobre la cinta, si tienen o no memoria auxiliar, etc.

Establecidas las reglas de una gramática, una cadena de símbolos pertenecerá al correspondiente lenguaje si tal cadena se ha formado obedeciendo esas reglas. A partir de una gramática se puede construir una máquina reconocedora o aceptadora del lenguaje generado por esa gramática, de tal forma que cuando reciba a su entrada una determinada cadena de símbolos indicará si dicha cadena pertenece o no al lenguaje. Una máquina reconoce un lenguaje  $L$  si es capaz de reconocer todas las sentencias pertenecientes a  $L$  y de no reconocer ninguna sentencia que no pertenezca a  $L$ . La figura 1.1