



ANÁLISIS LÉXICO

OLC1-A
Clase 2

Temas

01

Análisis Léxico

02

Componentes Léxicos

03

Expresiones
Regulares

04

Autómatas Finitos No
Deterministas

05

Autómatas Finitos
Deterministas

Análisis léxico

```

type="text/css">
body,div,h1,h2,h3 {
padding:0;
font-weight:100;
body {
background: #f0f0f0;
color: #555;

```

Análisis Léxico

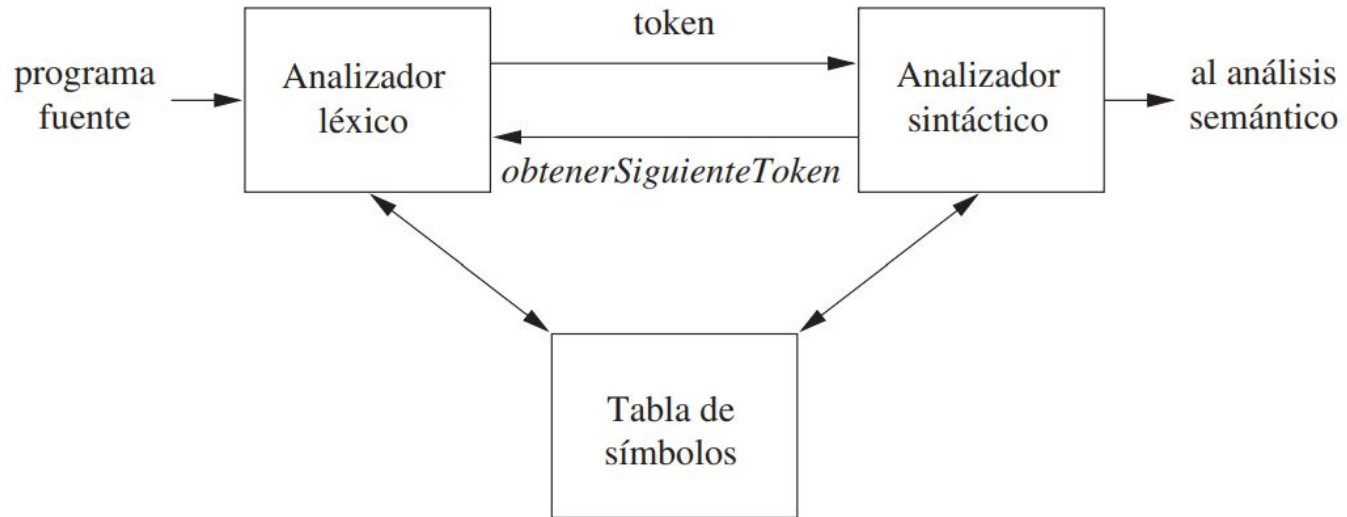
- Primera Fase del compilador
- Desglosa el código fuente
- Ignora espacios en blanco y comentarios
- Genera errores léxicos



Analizador Léxico

El analizador léxico o scanner lee el flujo de caracteres que componen el programa fuente y los agrupa en secuencias significativas llamadas lexemas.







02

Componentes Léxicos

Token

Un token es un par que consiste en un nombre de token y un valor de atributo opcional.

Lexema

Es una secuencia de caracteres en el programa fuente, que coinciden con el patrón para un token y que el analizador léxico identifica como una instancia de ese token.

Patrón

Una descripción de la forma que pueden tomar los lexemas de un token.

Ejemplo

posicion = inicial + velocidad * 60



Analizador léxico



$\langle \mathbf{id}, 1 \rangle \langle = \rangle \langle \mathbf{id}, 2 \rangle \langle + \rangle \langle \mathbf{id}, 3 \rangle \langle * \rangle \langle 60 \rangle$

.

Ejemplo

`posicion = inicial + velocidad * 60`

Token	Lexema	Patrón
id	posicion	Una o más letras
signo_igual	=	Signo igual
id	inicial	Una o más letras
signo_mas	+	Signo mas
id	velocidad	Una o más letras
signo_multiplicacion	*	Signo multiplicación
numero	60	Uno o mas digitos



03

Expresiones Regulares

Las expresiones regulares son patrones que se utilizan para hacer coincidir combinaciones de caracteres en cadenas.

Son cadenas de caracteres basadas en reglas sintácticas que permiten describir **secuencias de caracteres**.

De gran importancia para la tecnología de la información y, especialmente, para el desarrollo de software.

(. *) Exp[resio]nes
regul[are]s

Operaciones

Operación	Símbolo	Descripción	Ejemplo
Cerradura	*	Es el conjunto de cadenas que se obtienen al concatenar cero o más veces.	a^*
Concatenación		Se concatenan todas las cadenas que se forman al tomar una cadena del primer lenguaje y una cadena del segundo lenguaje.	ab
Unión		Es una operación que resulta en otro lenguaje, cuyos elementos son los mismos de los lenguajes iniciales.	$a \mid b$

Extensiones

Operador	Símbolo	Descripción	Ejemplo
Una o más instancias	+	Representa la cerradura positivo de una expresión regular y su lenguaje.	a+
Cero o una instancia	?	El operador unario postfijo ? significa “cero o una ocurrencia”. Es decir, r? es equivalente a r	a?
Clases de caracteres	[]	Forman una secuencia lógica, por ejemplo, letras mayúsculas, minúsculas o dígitos consecutivos.	[a-z] [abc]

04

Automátas



Autómatas Finitos

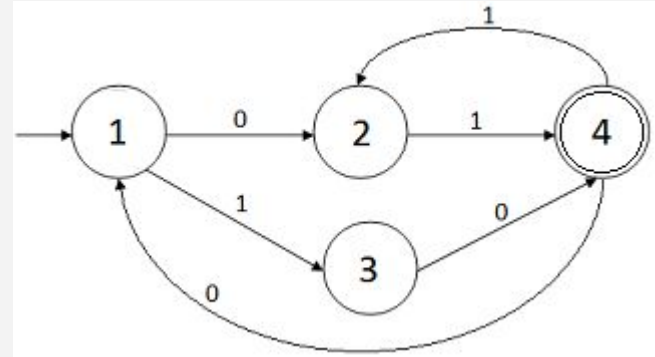
- Los autómatas finitos son reconocedores; sólo dicen “sí” o “no” en relación con cada posible cadena de entrada.

Se pueden representar mediante:

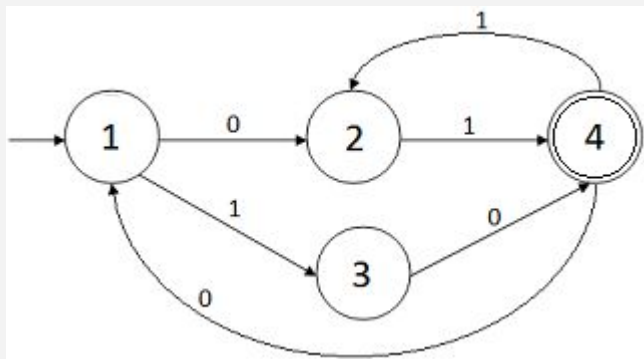
1. Diagramas de transición
2. Tablas de transición

Elementos:

1. Conjunto de estados
2. Alfabeto
3. Estado inicial
4. Estados finales
5. Función de transición



Autómatas Finitos



Alfabeto estados	0	1
1	2	3
2	-	4
3	4	-
4	1	2

Autómatas Finitos No Deterministas

No tienen restricciones en cuanto a las etiquetas de sus líneas. Un símbolo puede etiquetar a varias líneas que surgen del mismo estado, y, la cadena vacía, es una posible etiqueta.

Cada combinación de estados produce varios estados.

Son posibles transiciones con λ .

Autómatas Finitos Deterministas

Tienen, para cada estado, y para cada símbolo de su alfabeto de entrada, exactamente una línea con ese símbolo que sale de ese estado.

Cada combinación de estados produce un solo estado.

No son posibles transiciones con λ

Autómata Finito No Determinista AFND

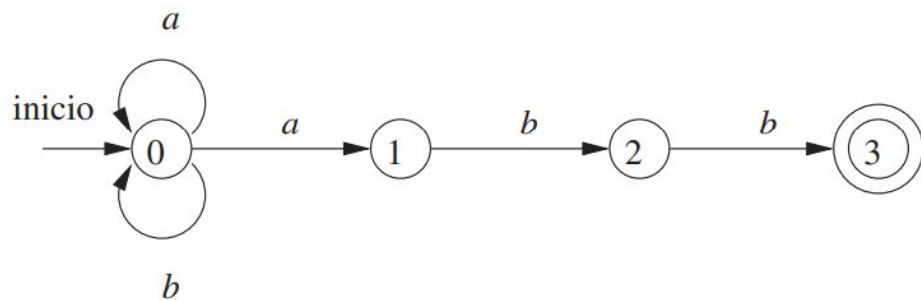
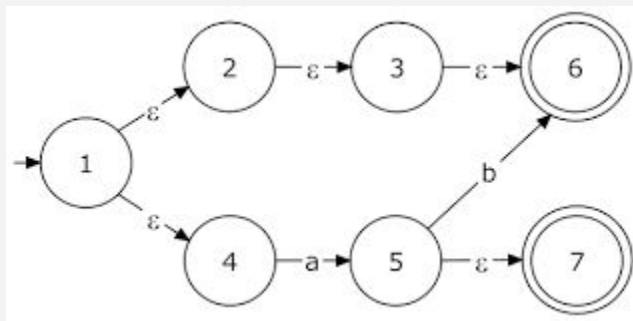


Figura 3.24: Un autómata finito no determinista



Autómata Finito Determinista AFD

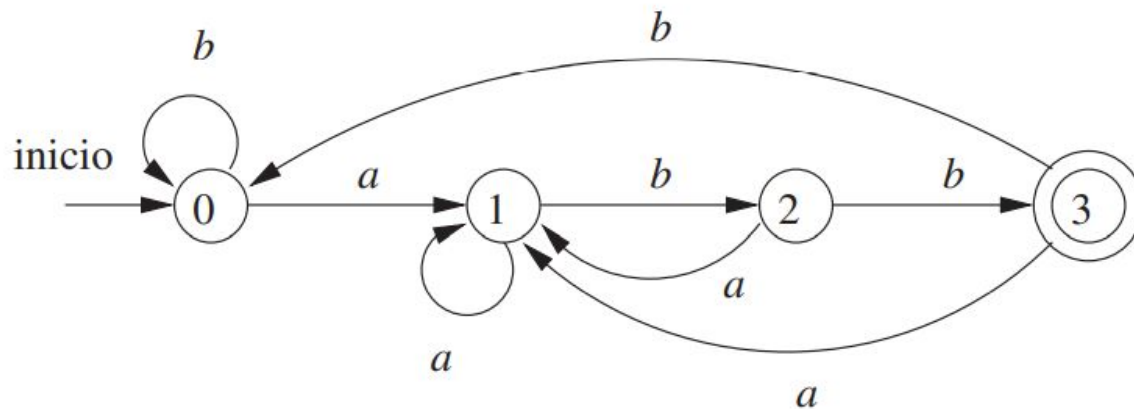


Figura 3.28: AFD que acepta a $(a|b)^*abb$