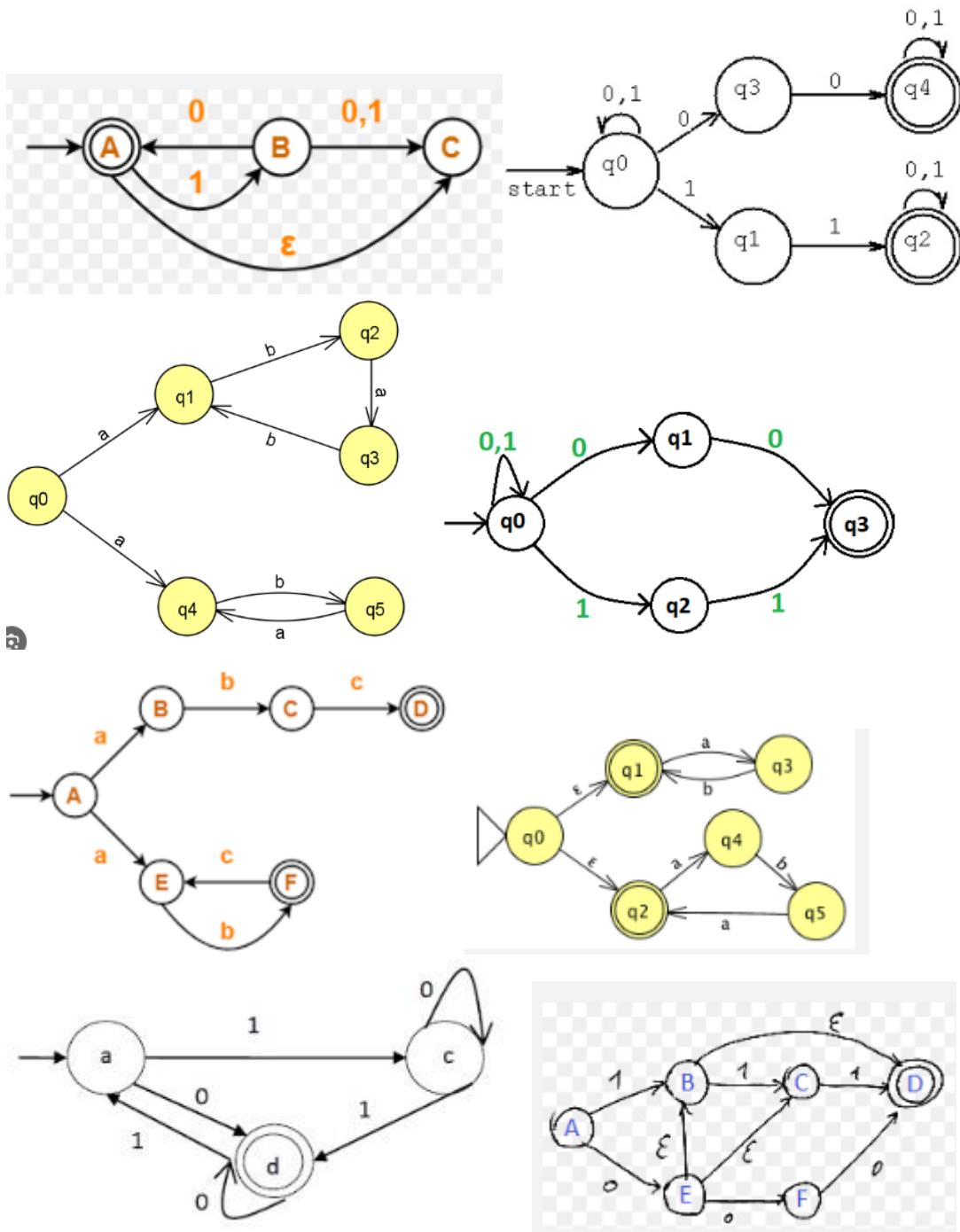


## Ejercicios Automatas:

Algunas posibles preguntas

- Definición Formal
- Tabla de transiciones
- Convertir a AFD o AFDmínimo
- Expresión regular
- Lenguaje que acepta

Casos prácticos



## Ejercicios Gramáticas

Algunas posibles preguntas:

- a) Árbol de derivaciones
- b) Definición formal de la gramática
- c) Tipo de gramática según Chomsky (y por qué)
- d) Si es gramática ambigua (y por qué)
- e) Lenguaje que genera
- f) Es posible expresión regular, escribirla

Casos prácticos:

<pre> sentence  -&gt; &lt;subject&gt; &lt;verb-phrase&gt; &lt;object&gt; subject   -&gt; This   Computers   I verb-phrase -&gt; &lt;adverb&gt; &lt;verb&gt;   &lt;verb&gt; adverb    -&gt; never verb      -&gt; is   run   am   tell object    -&gt; the &lt;noun&gt;   a &lt;noun&gt;   &lt;noun&gt; noun      -&gt; university   world   cheese   lies </pre>	$G_2 = (\{0, 1, 2\}, \{S, A\}, S, P)$ , where $P$ $ \begin{aligned} S &\rightarrow 0SA2 \\ S &\rightarrow \epsilon \\ 2A &\rightarrow A2 \\ 0A &\rightarrow 01 \\ 1A &\rightarrow 11 \end{aligned} $
$ \begin{aligned} S &\rightarrow AB \\ A &\rightarrow Ax   y \\ B &\rightarrow z \end{aligned} $	$G_4 = (\{0, 1\}, \{S, A, O, I, T\}, S, P)$ , where $P$ contains $ \begin{aligned} S &\rightarrow AT \\ A &\rightarrow 0AO \quad A \rightarrow 1AI \\ O0 &\rightarrow 0O \quad O1 \rightarrow 1O \\ I0 &\rightarrow 0I \quad I1 \rightarrow 1I \\ OT &\rightarrow 0T \quad IT \rightarrow 1T \\ A &\rightarrow \epsilon \quad T \rightarrow \epsilon \end{aligned} $
$ \begin{aligned} E &\rightarrow E t\_op T   T \\ t\_op &\rightarrow +   - \\ T &\rightarrow T f\_op F   F \\ f\_op &\rightarrow *   / \\ F &\rightarrow (E)   \text{int} \end{aligned} $	$ \begin{aligned} A &\rightarrow da   acB \\ B &\rightarrow aM   daN \\ M &\rightarrow bB   cBf \\ N &\rightarrow A   f \end{aligned} $
$ \begin{aligned} E &\rightarrow E op E   ( E )   \text{int} \\ op &\rightarrow +   -   *   / \end{aligned} $	$ \begin{aligned} S &\Rightarrow a \ T \\ T &\Rightarrow b \ T \\ T &\Rightarrow b \ V \\ V &\Rightarrow a \end{aligned} $

## Analizador léxico:

Algunas posibles preguntas:

- Construir distintos autómatas y uno que sea varios unidos.
- Aplicar a un código determinado y analizar dónde da error o no da error.
- Implementación de un autómata mediante IF anidados

Casos prácticos:

Autómata para comentarios definidos como %\* ..... \*%

Automata para nombres de variables según unas reglas, combinacion de minúsculas y números con guión bajo.

Autómatas para números naturales, Reales, complejos, decimales, numeros irracionales escritos de determinada forma.

Autómata para identificar el =

Automata para nombres de funciones: FUNCION( var1,var2, var3 , var4)

Autómata para operadores de comparaciones == <= >= < > !=

Autómata para operadores algebraicos: + - \* /

Autómata que junta todos los números en un único token.

¿Cómo te aseguras si implementas por if anidados que el token de la igualdad '=' y de la comparación '==' se identifican correctamente en cada caso?

## Limpieza de gramáticas:

Algunas posibles preguntas (Se puede pedir alguno de los pasos de limpieza):

- 1.1.a Reglas innecesarias
- 1.1.b Símbolos inaccesibles:
- 1.1.c Reglas superfluas
- 1.2. Eliminar símbolos no generativos
- 1.3. Eliminar reglas no generativas
- 1.4. Eliminar reglas de redenominación
- 2.a) Eliminar Factores izquierdos:
- 2.b) Eliminar recursividad a izquierdas
- 2.c) Eliminar ambigüedad

## Casos prácticos:

**Ejercicio 1:**  $G_1 = (\{0, 1, 2, 3\}, \{S, A, B, C, D, E\}, S, P_1)$   
 $P_1 = \{(S := 0A / 1B), (A := A / 1B / 0), (B := 0C / 0E / 1), (C := 1), (E := 0E), (D := 0A / 1B / 0)\}$

**Ejercicio 2:**  $G_2 = (\{0, 1, 2\}, \{S, A, B, C\}, S, P_2)$   
 $P_2 = \{(S := 0A / 1), (A := A / 1B0 / 1), (C := 0 / 1B / 1), (B := 1A / A0 / 1B)\}$

**Ejercicio 3:**  $G_3 = (\{a, b, c\}, \{S, A, B, C, D, E\}, S, P_3)$   
 $P_3 = \{(S := aBb / \lambda), (A := bB / Ca / A), (B := bA / b / a / bE), (C := a / bB / aD), (D := a), (E := aE / E)\}$

**Ejercicio 4:**  $G_4 = (\{0, 1, 2\}, \{Q, R, S, T\}, Q, P_4)$   
 $P_4 = \{(Q := 1R0 / \lambda), (R := 0S1 / 0T / 1), (T := 0R / RT1), (S := 0)\}$

## Ejercicio 16

$G_1 = (\{0, 1\}, \{S, A, B\}, S, P_1)$   
 $P_1 = \{S := A B1 | \lambda, A := BA | \lambda, B := 0A | \lambda\}$

## Ejercicio 17

$G_2 = (\{0, 1, 2, 3\}, \{S, A, B, C, D\}, S, P_2)$   
 $P_2 = \{S := C0 | \lambda | D10, A := 1C3, B := B, C := 1 | \lambda | 0, D := 1D\}$

## Ejercicio 18

$G_3 = (\{a, b, c, d\}, \{A, B, C, D\}, A, P_3)$   
 $P_3 = \{A := bBa, B := bDa | aC | b | \lambda, C := BB | A, D := \lambda | a | b\}$

## Ejercicio 19

$G_1 = (\{a, b\}, \{P, Q, R\}, P, P_1)$   
 $P_1 = \{P := abP | aQ, Q := a | bR, R := Ra | b\}$

## Ejercicio 20

$G_2 = (\{a, b, c\}, \{S, A, B, C, D\}, S, P_2)$   
 $P_2 = \{S := AB | c, A := aC, B := aD, C := Ca | Cab | b, D := b\}$

## Ejercicio 21

$G_3 = (\{a, b, c\}, \{S, A, B\}, S, P_3)$   
 $P_3 = \{S := aAb, A := aB | a | Ac, B := c\}$

## Ejercicio 22

$G_4 = (\{a, b\}, \{M, N, P\}, M, P_4)$   
 $P_4 = \{M := Ma | aP | b, N := aP | a, P := b | aN | Pb\}$

## Ejercicio 23

$G_5 = (\{a, b\}, \{M, P\}, M, P_5)$   
 $P_5 = \{M := Pa | b, P := Mb | b\}$

## Analizador Sintáctico LL(1)

Algunas posibles preguntas (coinciden con los pasos):

- 1) Hallar FIRST y FOLLOW para construir la tabla parser
- 2) Construir tabla parser a partir de FIRST y FOLLOW
- 3) Obtener árbol de derivación a partir de una cadena de terminales y tabla parser (poneos ejemplos de una producción que sepáis que es válida y otra que no).

Casos prácticos:

$$\begin{aligned}S &\rightarrow ( A ) \\A &\rightarrow CB \\B &\rightarrow ; \ A \mid \epsilon \\C &\rightarrow x \mid S\end{aligned}$$

$$\begin{aligned}S &\rightarrow A \ B \ c \\A &\rightarrow a \mid \epsilon \\B &\rightarrow b \mid \epsilon\end{aligned}$$

$$\begin{aligned}S &\rightarrow a \ S \ e \mid B \\B &\rightarrow b \ B \ e \mid C \\C &\rightarrow c \ C \ e \mid d\end{aligned}$$

## Analizador Sintáctico LR(1)

Algunas posibles preguntas (coinciden con los pasos):

- 1) Hallar Autómata para construir la tabla parser
- 2) Construir tabla parser a partir de autómata y reglas
- 3) Obtener árbol de derivación a partir de una cadena de terminales y tabla parser

Casos prácticos:

$$\begin{array}{ll} S \rightarrow uBDz & \\ B \rightarrow Bv & \\ B \rightarrow w & S \rightarrow ( L ) \\ D \rightarrow EF & \\ E \rightarrow y & S \rightarrow a \\ E \rightarrow \epsilon & L \rightarrow L , S \\ F \rightarrow x & \\ F \rightarrow \epsilon & L \rightarrow S \end{array}$$

$$\begin{array}{l} S \rightarrow P M \\ P \rightarrow E \\ | \quad q \\ M \rightarrow n E \\ E \rightarrow , o \\ | \quad E \\ o \rightarrow u \\ | \quad u, o \end{array}$$

## Otros ejercicios:

- Autómata a pila para paréntesis, corchetes y llaves balanceados (o variante sin alguno de ellos).
- Interpretar lo que hace una Maquina de Turing dada y su funcionalidad.



