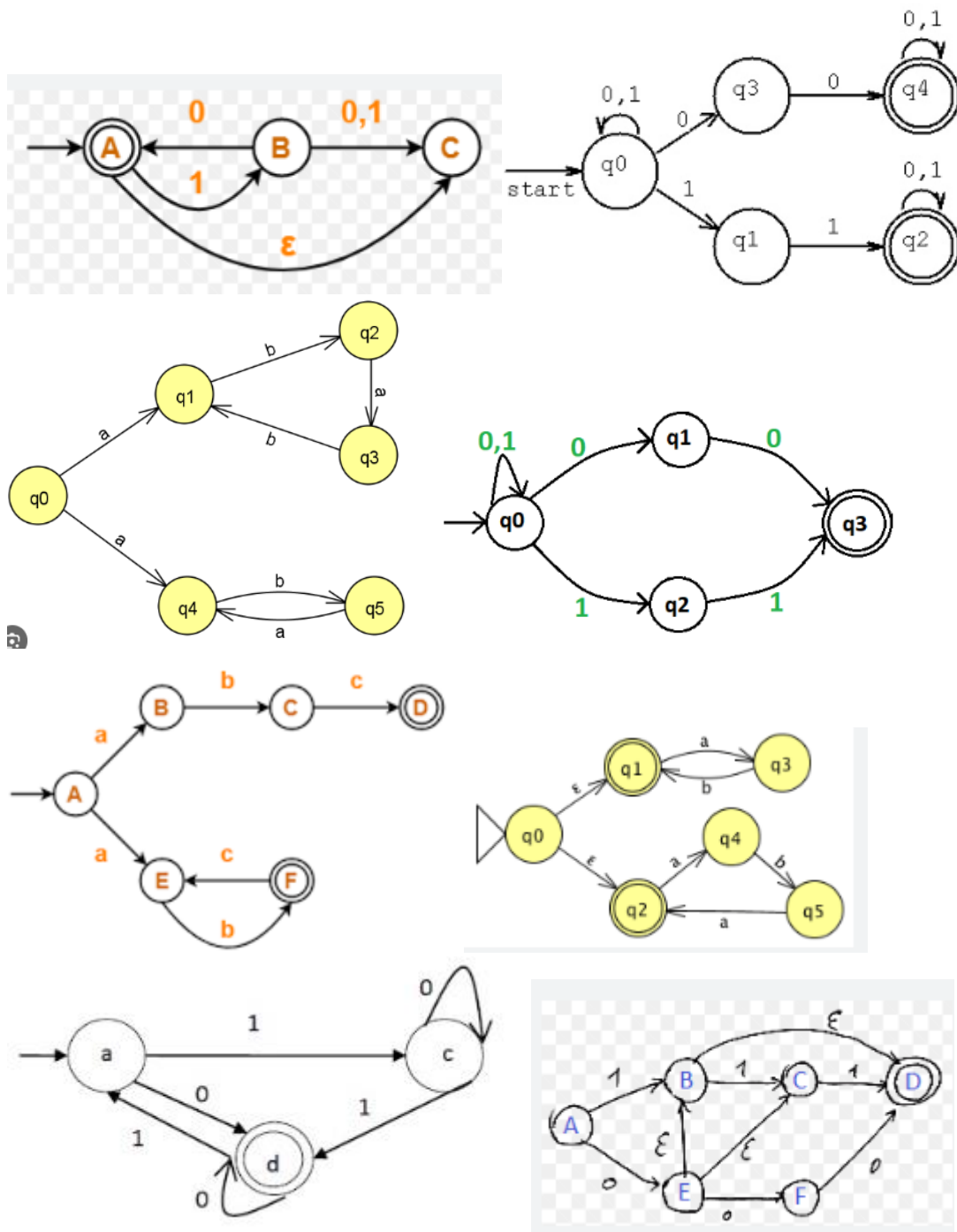


Ejercicios Automatas:

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:

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

Casos prácticos:

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

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 junte 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 red denominación
- 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: $G_2 = (\{0, 1, 2\}, \{S, A, B, C\}, S, P_2)$
 $P_2 =$
 $\{(S := 0A / 1), (A := A / 1B / 1), (C := 0 / 1B / 1),$
 $(B := 1A / A / 1B)\}$

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: $G_4 = (\{0, 1, 2\}, \{Q, R, S, T\}, Q, P_4)$
 $P_4 =$
 $\{(Q := 1R / \lambda), (R := 0S / 0T / 1), (T := 0R / RT / 1), (S := 0)\}$

Ejercicio 16

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

Ejercicio 17

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

Ejercicio 18

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

Ejercicio 19

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

Ejercicio 20

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

Ejercicio 21

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

Ejercicio 22

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

Ejercicio 23

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

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:

$$S \rightarrow (A)$$

$$A \rightarrow CB$$

$$B \rightarrow ; A \mid \epsilon$$

$$C \rightarrow x \mid S$$

$$S \rightarrow A B c$$

$$A \rightarrow a \mid \epsilon$$

$$B \rightarrow b \mid \epsilon$$

$$S \rightarrow a S e \mid B$$

$$B \rightarrow b B e \mid C$$

$$C \rightarrow c C e \mid d$$

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:

$S \rightarrow \mathbf{uBDz}$

$B \rightarrow B\mathbf{v}$

$B \rightarrow \mathbf{w}$

$D \rightarrow EF$

$E \rightarrow \mathbf{y}$

$E \rightarrow \varepsilon$

$F \rightarrow \mathbf{x}$

$F \rightarrow \varepsilon$

$S \rightarrow (\ L \)$

$S \rightarrow \mathbf{a}$

$L \rightarrow L \ , \ S$

$L \rightarrow S$

$S \rightarrow P \ M$

$P \rightarrow E$

$\quad | \quad \mathbf{q}$

$M \rightarrow \mathbf{n} \ E$

$E \rightarrow , \ O$

$\quad | \quad E$

$O \rightarrow \mathbf{u}$

$\quad | \quad \mathbf{u}, O$

Otros ejercicios:

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

