Escuela Técnica Superior de Ingenieros Informáticos (UPM)

LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

FINAL - 1ª EVALUACIÓN (22 de enero de 2020)

Apellidos: SOLUCIÓN Nombre:

Ejercicio 1:

Dada la gramática G:

G = { Σ_T = { a , b }, Σ_N = { S , A }, S, $\mathcal{P}$} con las siguientes producciones:

$$\mathcal{P} \equiv \left| \begin{array}{l} S ::= AAA \\ A ::= AAA \mid bA \mid Ab \mid a \mid b \end{array} \right.$$

a) Definir gramática ambigua.

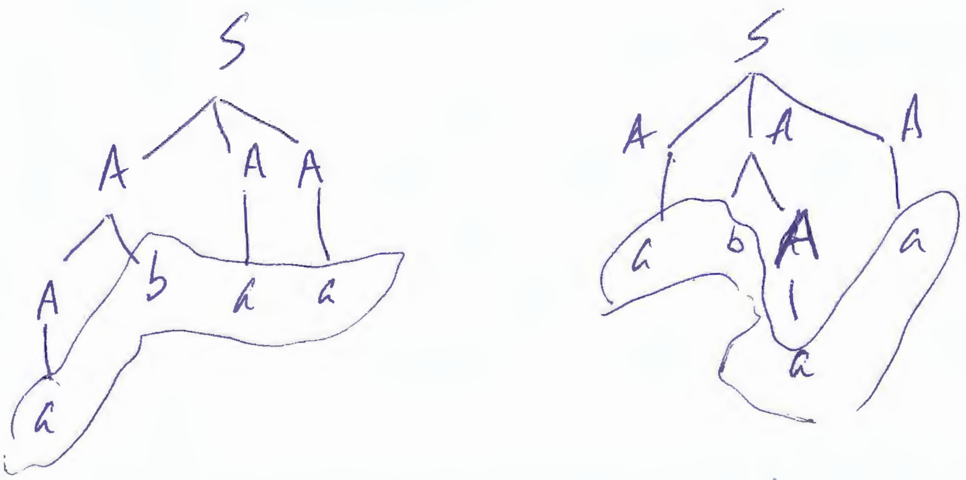
b) Probar que G es gramática ambigua.

25 minutos

---

b/ Una gramática es ambigua si tiene al menos una palabra ambigua.

La palabra x = abaa es ambigua, porque tiene 2 árboles de derivación distintos y 2 derivaciones por la izquierda diferentes



$S \to AAA \to Ab\,AA \to ab\,AA \to abaA \to abaa$

$S \to AAA \to a\,AA \to a\,bAA \to abaA \to abaa$

Escuela Técnica Superior de Ingenieros Informáticos (UPM)

LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

FINAL - 1ª EVALUACIÓN (22 de enero de 2020)

Apellidos: **SOLUCIÓN** Nombre:

Ejercicio 2:

Dada la expresión regular R_0 = 0 (1* + 0*0) 1, calcular el Autómata Finito que la reconoce, por medio de derivadas.

30 minutos

$\mathbf{D_0(R_0)} = D_0(0(1^* + 0^*0)1) = \mathbf{(1^* + 0^*0)1 = R_1}$

$D_1(R_0) = \emptyset$

$\mathbf{D_0(R_1)} = D_0((1^* + 0^*0)1) = D_0(1^* + 0^*0) \cdot 1 + D_0(1) = [D_0(1^*)+ D_0(0^*0)]1 + \emptyset =$
$= [\emptyset + (D_0(0^*)0 + D_0(0))]1 = (0^*0 + \lambda)1 = \mathbf{0^*01 + 1 = R_2}$

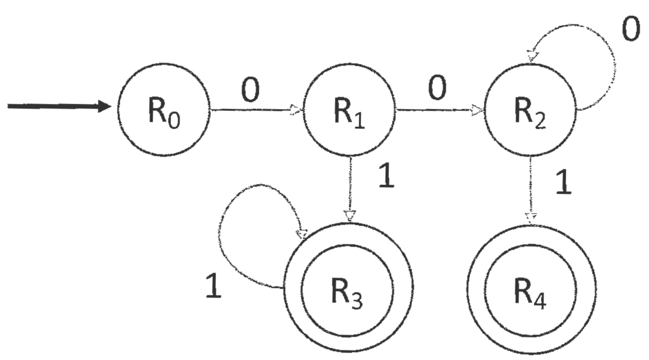
$\mathbf{D_1(R_1)} = D_1((1^* + 0^*0)1) = D_1(1^* + 0^*0) \cdot 1 + D_1(1) = [D_1(1^*) + D_1(0^*0)]1 + \lambda =$
$= (1^* + \emptyset)1 + \lambda = \mathbf{1^*1 + \lambda = R_3, (\lambda \in R_3)}$

$\mathbf{D_0(R_2)} = D_0(0^*01 + 1) = D_0(0^*01) + D_0(1) = (D_0(0^*)01 + D_0(01)) + \emptyset = 0^*01 + 1 = \mathbf{R_2}$

$\mathbf{D_1(R_2)} = D_1(0^*01 + 1) = D_1(0^*01)+D_1(1) = (D_1(0^*)01+D_1(01)) + \lambda = \emptyset+\emptyset+\lambda = \lambda \mathbf{= R_4}$

$\mathbf{D_0(R_3)} = D_0(1^*1 + \lambda) = \emptyset$

$\mathbf{D_1(R_3)} = D_1(1^*1 + \lambda) = D_1(1^*1)+D_1(\lambda) = (D_1(1^*)1+D_1(1))+\emptyset = \mathbf{1^*1 + \lambda = R_3, (\lambda \in R_3)}$



Escuela Técnica Superior de Ingenieros Informáticos (UPM)

LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

FINAL - 2ª EVALUACIÓN (22 de enero de 2020)

Apellidos: **SOLUCIÓN** Nombre:

Ejercicio 1:

Sea la gramática G = { Σ_T, Σ_N, S, $\mathbb{P}$ } donde Σ_T = { 1 , 2 }, Σ_N = { S , D }, S = axioma y cuyas producciones $\mathbb{P}$ son:

S : : = 1D
D : : = 1D2 | 22

a) Obtener a partir de la gramática G (utilizando el método 2) un AP por vaciado de pila que genere el mismo lenguaje. (6 puntos)

b) Comprobar el reconocimiento en el AP de las palabras 112 y 11222. (2 puntos)
y su generación en la gramática G. (1 punto)

c) ¿Qué lenguaje reconoce el AP y genera la gramática G? (1 punto)

25 minutos

---

a) Construimos AP utilizando el método 2:

G = { { 1 , 2 } , { S , D } , S , $\mathbb{P}$ } Producciones $\mathbb{P}$: S : : = 1D ; D : : = 1D2 | 22

AP = { { 1 , 2 } , { 1 , 2 , S } , { q } , q , S , *f* , ∅ }

Algoritmo (para obtener los movimientos)

1. X ∈ { $\Sigma_N \cup \Sigma_T$}, A ∈ Σ_N
   ∀ A : : = X , producción de la gramática, en AP se hace: (q X) ∈ *f* (q λ A)

2. ∀ a ∈ Σ_T, entonces, (q λ) ∈ *f* (q a a)

Movimientos del AP

1) *f* (q λ S) = (q 1D)
2) *f* (q λ D) = (q 1D2) , (q 22)
3) *f* (q 1 1) = (q λ)
4) *f* (q 2 2) = (q λ)

b) Comprobamos la aceptación en AP y la generación en G de las palabras 112 y 11222:

AP (112): (q , 112 , S) ⊢ (q , 112 , 1D) ⊢ (q , 12 , D) ⊢ (q , 12 , 1D2) ⊢ (q , 2 , D2) ⊢ (NO)
(hay más caminos pero que no llevan a la aceptación)

AP (11222): (q , 11222 , S) ⊢ (q , 11222 , 1D) ⊢ (q , 1222, D) ⊢ (q , 1222, 1D2) ⊢ (q , 222 , D2) ⊢ (q , 222 , 222) …) ⊢ (q , 22 , 22) …) ⊢ (q , 2 , 2) ⊢ (q , λ , λ) (SI)

G (112): S → 1D → 11D2 → 1122 (NO) ; S → 1D → 122 (NO).
G (11222): S → 1D → 11D2 → 11222 (SI)

c) El lenguaje que genera G y que acepta el AP es: L = { $1^n 2^{n+1}$ / n ≥ 1 }

Escuela Técnica Superior de Ingenieros Informáticos (UPM)

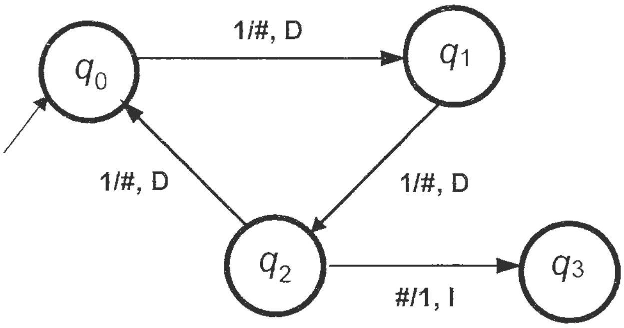
LENGUAJES FORMALES, AUTÓMATAS Y COMPUTABILIDAD

FINAL - 2ª EVALUACIÓN (22 de enero de 2020)

Apellidos: **SOLUCIÓN** Nombre:

Ejercicio 2:

Sea la Máquina de Turing M definida según el siguiente grafo:



Y cuya configuración inicial es la siguiente:

w
↑
q_0

Donde $w \in 1^*$ es un número entero codificado en unario. M inicialmente está en el estado q_0 leyendo el primer 1 de w.

a) ¿Qué función aritmética sobre cada w calcula M? ¿Cuál es la configuración final de M tras recibir las entradas de los apartados a.1), a.2) y a.3)? (2 puntos)

a.1) # 1 # (↑ q_0) a.2) # 1 1 # (↑ q_0) a.3) # 1 1 1 # (↑ q_0)

b) Escribir (y describir brevemente) el contenido inicial de la cinta de la Máquina de Turing Universal (MTU) programada para simular a la máquina M con la entrada del apartado a.2). Utilizar la siguiente codificación binaria:

$q_0 \equiv 00$; $q_1 \equiv 01$; $q_2 \equiv 10$; $q_3 \equiv 11$; Izquierda I $\equiv$ 1; Derecha D $\equiv$ 0 (2 puntos)

c) Durante la simulación del primer movimiento de M con la entrada del apartado a.2): ¿En qué estado termina el módulo transcriptor de la MTU? ¿A qué estado accede el módulo simulador tras recolocar el *? ¿Por qué? Explicar brevemente. (3 puntos)

d) Escribir (y describir brevemente) el contenido de la cinta de la MTU después de simular el primer movimiento que realiza la máquina M con la entrada del apartado a.2). (1 punto)

e) Escribir (y describir brevemente) el contenido final de la cinta de la MTU cuando termine de simular a la máquina *M* con la entrada del apartado a.2). ¿En qué estado se para la MTU? ¿Por qué? (2 puntos)

NOTA: Todos los apartados se responderán en la carilla de atrás.
Durante el examen se da fotocopia con el grafo de los tres módulos de la MTU.

30 minutos

Continuación ejercicio 2.

Apartado a) Configuraciones finales con a.1), a.2) y a.3) y función aritmética que calcula M

a.1) $q_0 1 \vdash \# q_1 \#$

a.2) $q_0 11 \vdash \# q_1 1 \vdash \#\# q_2 \# \vdash \# q_3 \# 1$

a.3) $q_0 11 \vdash \cdots \vdash q_0 \#\# \ldots$

$$f = \begin{cases} 1 & \text{si } (w+1) \bmod 3 = 0 \\ 0 & \text{en otro caso} \end{cases}$$

Apartado b) Cinta inicial de la MTU programada con la entrada del aptdo. a.2)

celda en blanco q_0 q_1 1 # D

...	#	*	1	0	ǂ	0	0	1	ǂ	0	0	1	0	1	0	0	ǂ	0	1	1	1	0	0	0	ǂ	1	0	1	0	
0	0	0	ǂ	1	0	0	1	1	1	1	ǂ	#	·	·	·															...

El * se sitúa sobre la celda que lee inicialmente M y que contiene un 1 que se guarda en la última celda del REG de inicio. Hay 4 registros (uno por cada movimiento diferente de M)

Apartado c)

Estado en el que termina el módulo transcriptor: q_{12} ¿Por qué? Para memorizar el valor 0 que indica desplazamiento a la dcha. Ese valor no se puede transcribir en las celdas del REG. de inicio por lo que se memoriza. Si es 0 (q_{12}). Si es 1 (q_{13})

Estado al que accede la MTU tras recolocar el *: q_{22} ¿Por qué? Porque en el estado q_{20} el módulo simulador se encuentra un 1 que memoriza transitando a q_{22} para almacenarlo posteriormente en la última celda del REG de inicio.

Apartado d) Cinta de la MTU tras simular el primer movimiento (escribid sólo la parte de la cinta que cambia)

...	#	0	*	0	ǂ	0	1	1	ǂ	·	·	·	

Se borra el 1er uno.
El * se desplaza a la derecha ⇒ se lee un 1
El control pasa de q_0 (00) a q_1 (01)

Apartado e) Cinta de la MTU cuando para:

M se para en q_3 (11) leyendo un #

...	#	0	*	1	ǂ	B	1	0	ǂ	A	A	B	A	B	A	A	ǂ	A	BB	B	A	A	A	ǂ	B	A	B
A	A	A	A	ǂ	B	A	A	B	B	BB	ǂ	#															...

q_5

¿En qué estado se para la MTU? q_5 ¿Por qué? Porque el módulo localizador busca ǂ110ǂ al comienzo de los diferentes registros. Ninguno de ellos comienza por esa secuencia por lo que son marcados con A's y B's. Al buscar un siguiente registro aparece la primera celda en blanco por la derecha en q_5