informatika fakultatea facultad de informática

Lenguajes, Computación y Sistemas Inteligentes

Curso 2020-21 – Grupo 01

Evaluación de Conjunto - 11 de enero de 2021

Apellidos, Nombre:

El examen consta de 14 preguntas con distinta valoración. Las cuatro primeras **0,25** puntos, las siguientes cinco **0,5** puntos, las tres siguientes **1,5** puntos y las dos últimas **1** punto. La nota final que puede obtenerse es **10**. La nota necesaria para aprobar es 5.

Duración: tres horas

1	2	3	4	5	6	7	8	9	10	11	12	13	14
0,25	0,25	0,25	0,25	0,5	0,5	0,5	0,5	0,5	1,5	1,5	1,5	1	1

TOTAL

1. (0,25 puntos) Hemos estudiado fundamentalmente tres clases de lenguajes (regulares, independientes de contexto y recursivamente enumerables) y nos preguntamos qué características tienen los autómatas que reconocen cada una de las clases.Escoge para cada tipo de lenguajes la opción (1,2,3):

Lenguajes **Regulares** A__1 B__2 C__2

Lenguajes **Independientes de Contexto** A__2 B__3 C__2

Lenguajes **Recursivamente Enumerables** A__3 B__2 C__3

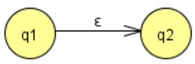
A1 Solo tienen estados	**B1** Tienen que ser deterministas	**C1** Pueden procesar cada palabra alternando a voluntad lecturas hacia la derecha o hacia la izquierda
A2 Además de los estados tienen una memoria auxiliar a la que solo se accede por uno de los extremos	**B2** No importa si son deterministas o no	**C2** Cada palabra debe leerse de una sola pasada de izquierda a derecha
A3 Además de los estados tienen una memoria auxiliar en la que se puede acceder en cualquier posición mediante movimientos secuenciales en ambas direcciones	**B3** Tienen que ser no deterministas	**C3** Tanto la palabra procesada como las anotaciones hechas pueden leerse alternando a voluntad lecturas hacia la derecha o hacia la izquierda

2. (0,25 puntos) Iniciamos la construcción de un autómata finito en **JFLAP**. Introducimos dos estados q1 y q2 y una transición vacía entre ellos

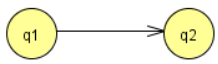
Señala con un círculo **cuál o cuáles** de los siguientes diagramas nos presentaría la ventana de aplicación.

a)

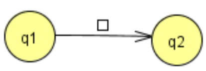
b)



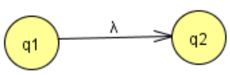
c)



d)



e)



3. (0,25 puntos) Completa los términos que faltan para que los siguientes tipos de autómata se correspondan con su función de transición.

a) Autómata finito con transiciones vacías: δ: Q × $\boxed{(\Sigma \cup \{\varepsilon\})}$ — → $\wp$(Q)

b) Autómata con Pila: δ: Q × (Σ ∪ {ε}) × $\boxed{\Gamma}$ — → $\wp$(Q × Γ*)

---

4. (0,25 puntos) Indica si las siguientes afirmaciones son verdaderas o falsas. Rodea con un círculo **V** si es verdadera y **F** si es Falsa. Las respuestas incorrectas penalizan.

(V / F) La tesis de Church-Turing se demuestra aplicando la técnica de diagonalización.

(V / F) Para poder usar una máquina M como bloque dentro de otra es necesario que M pare en todos los casos.

(V / F) El problema de parada no es computable/decidible porque no existe ningún algoritmo capaz de decidir cuando los programas van a parar o no.

(V / F) El test de Turing es una prueba desafío-respuesta utilizada para determinar cuándo un usuario remoto es una persona o un programa.

(V / F) No se ha podido demostrar si P ⊆ NP.

---

5. (0,5 puntos) Escribe para cada una de las siguientes expresiones regulares con notación **JavaScript** su notación equivalente como expresión regular formal utilizando por tanto únicamente las operaciones permitidas en la definición.

[**a**-**cf**]	(a U b U c U f)
(**ab**|**ba**){2,}	(ab U ba)(ab U ba)$^{+}$
ac?**b***	a(c U ε)b*
[**a|:z**]	(a U | U : U z)
[**ac**]{2}	(aUc)(aUc)

6. (0,5 puntos) Considera la siguiente definición inductiva de un lenguaje L sobre el alfabeto {0,1}:

- **00**$\in$L
- $\forall x \in L \Rightarrow$ **1**x, x**1**$\in$ L

y describe formalmente el lenguaje L*.

$L=\{1^n001^m: n,m\geq 0\}$

$L^*=(1^*001^*)^* = \{1, 00\}^* - \{1\}^+$

7. (0,5 puntos) Averigua qué lenguaje es **X** sabiendo que satisface el siguiente sistema de ecuaciones sobre expresiones regulares:

$X = \mathbf{a}^+X \cup \mathbf{b}Y \cup \mathbf{c}Z$
$Y = \mathbf{a}Y \cup \mathbf{ab}X \cup \varepsilon$
$Z = \mathbf{a}Z \cup \mathbf{bb}Z$

Eliminamos Z:

$Z = aZ \cup bbZ \Rightarrow Z = (a \cup bb)Z \Rightarrow Z = (a \cup bb)^*\varnothing = \varnothing$

Obtenemos:

$X = a^+X \cup bY$

$Y = aY \cup abX \cup \varepsilon$

Eliminamos Y:

$Y = aY \cup abX \cup \varepsilon \Rightarrow Y = a^* (abX \cup \varepsilon)= a^+bX \cup a^*$

Obtenemos:

$X = a^+X \cup b(a^+bX \cup a^*) = a^+X \cup ba^+bX \cup ba^* = (a^+ \cup ba^+b) X \cup ba^*$

Por tanto $X=(a^+ \cup ba^+b)^*ba^*$

8. (0,5 puntos) El programa Prolog que te presentamos corresponde al trabajado en el laboratorio 2. ¿Qué regla o reglas podemos añadir para definir un menú "plato del día" consistente en un único plato que puede ser entrada o principal y postre?

```
% menu
entrada(paella).
entrada(gazpacho).

carne(filete_de_cerdo).
carne(pollo_asado).

pescado(trucha).
pescado(bacalao).

postre(flan).
postre(naranja).

% plato_principal(P)  P es un plato principal si es carne o pescado
plato_principal(P):-
     carne(P).
plato_principal(P):-
     pescado(P).

% comida(Entrada, Principal, Postre)

comida(Entrada, Principal, Postre):-
     entrada(Entrada),
     plato_principal(Principal),
     postre(Postre).



% añade las nuevas reglas
```

```
plato_del_dia(Entrada, Postre):-
     entrada(Entrada),
     postre(Postre).

plato_del_dia(Principal, Postre):-
     plato_principal(Principal),
     postre(Postre).
```

9. **(0,5 puntos)** Hay una propiedad indeseable que tienen algunas Gramáticas Independientes del Contexto que tratamos de evitar en la medida de lo posible. La siguiente gramática la tiene:

$S \rightarrow S\mathbf{a}S \mid S\mathbf{b}S \mid A$

$A \rightarrow \mathbf{c}A \mid \mathbf{d}A \mid \mathbf{c} \mid \mathbf{d}$

¿Cuál es esa propiedad? ¿Por qué esta gramática tiene esta propiedad? Demuéstralo.

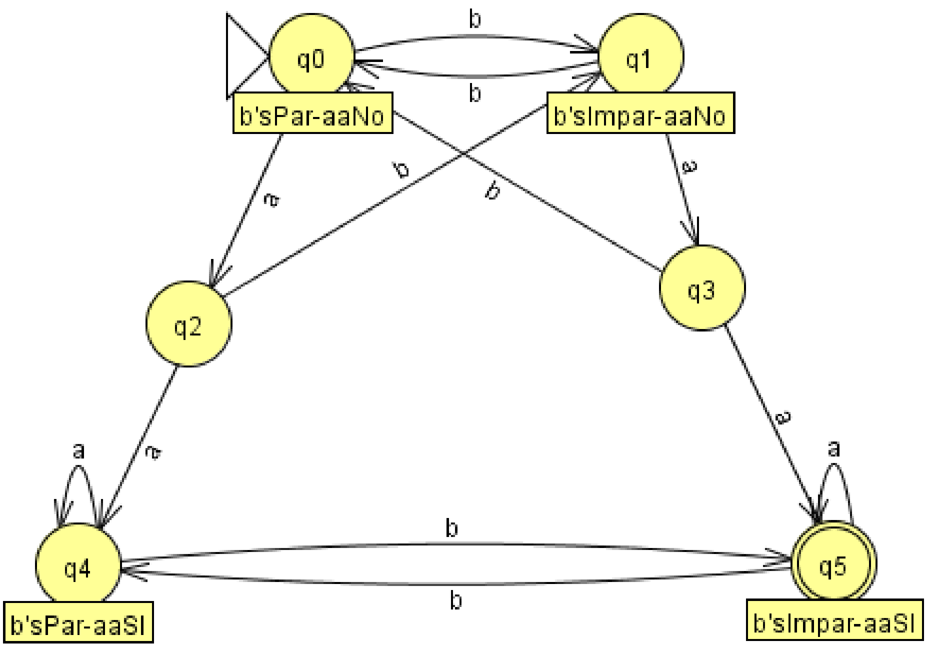
La propiedad es la ambigüedad. Esta gramática es ambigua porque existen palabras con más de un árbol de derivación, o equivalentemente, con más de una derivación a la izquierda. Por ejemplo, para la palabra **cadbc** tenemos las siguientes derivaciones:

$\mathbf{S \Rightarrow SaS \Rightarrow AaS \Rightarrow caS \Rightarrow caSbS \Rightarrow caAbS \Rightarrow cadbS \Rightarrow cadbA \Rightarrow cadbc}$

$\mathbf{S \Rightarrow SbS \Rightarrow SaSbS \Rightarrow AaSbS \Rightarrow caSbS \Rightarrow caAbS \Rightarrow cadbS \Rightarrow cadbA \Rightarrow cadbc}$

10. (1,5 puntos) Diseña un autómata finito M (de cualquier tipo) que reconozca el lenguaje L. **Es imprescindible justificar su diseño.**

$$L = \{ x \in \{ \mathbf{a},\mathbf{b} \}^* : |x|_b \% 2 \neq 0 \wedge |x|_{aa} \geq 1 \}$$



11. (1,5 punto) Considera los autómatas finitos M1=(Q, Σ, δ_1, q_0, $\{q_1,q_2,q_4\}$) y M2=(Q, Σ, δ_2, q_0, $\{q_1,q_3,q_4\}$) siendo Σ={**0**,**1**} siendo Q=$\{q_0,q_1,q_2,q_3,q_4\}$ y las siguientes funciones de transición:

δ_1	**0**	**1**
q_0	$\{q_0, q_1\}$	$\{q_2\}$
q_1	-	$\{q_1, q_2, q_3\}$
q_2	$\{q_4\}$	-
q_3	$\{q_4\}$	-
q_4	-	$\{q_2\}$

δ_2	**0**	**1**	**ε**
q_0	$\{q_0, q_1\}$	$\{q_3\}$	$\{q_2\}$
q_1	-	$\{q_1, q_4\}$	-
q_2	$\{q_4\}$	-	-
q_3	-	-	$\{q_2\}$
q_4	-	$\{q_3\}$	-

a) Aplica el algoritmo que permite obtener el AFD equivalente a M1.

b) Aplica el algoritmo que permite obtener el AFD equivalente a M2.

c) Comprueba que se obtiene esencialmente el mismo AFD. Si no es así habrás cometido algún error. Por si te ayuda a localizar el error, puedes tener en cuenta que las palabras **000**, **0011**, **0110**, **1010** sí están en el lenguaje y las palabras **0100**, **1011**, **1101**, **1000** no.

Nota: para que se den como válidos los AFD's obtenidos es imprescindible que se pueda observar cómo has aplicado el algoritmo correspondiente.

$q_0=p_0$

$\delta(q_0, 0) = \{q_0, q_1\}=p_1$

$\delta(q_0, 1) = \{q_2\}=p2$

$\delta(p_1, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\}=p_1$

$\delta(p_1, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_2\} \cup \{q_1,q_{2,q3}\}=\{q_1,q_{2,q3}\}=p_3$

$\delta(p_2, 0) = \delta(q_2, 0) = \{q_4\}=p_4$

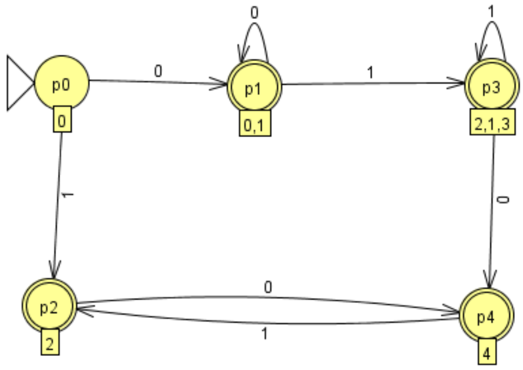
$\delta(p_2, 1) = \delta(q_2, 1) = \varnothing$

$\delta(p_3, 0) = \delta(q_1, 0) \cup \delta(q_2, 0) \cup \delta(q_3, 0)= \varnothing \cup \{q_4\} \cup \{q_4\}= \{q_4\}= p_4$

$\delta(p_3, 1) = \delta(q_1, 1) \cup \delta(q_2, 1) \cup \delta(q_3, 1)= \{q_1,q_{2,q3}\} \cup \varnothing \cup \varnothing= \{q_1,q_{2,q3}\}= p_3$

$\delta(p_4, 0) = \delta(q_4, 0) = \varnothing$

$\delta(p_4, 1) = \delta(q_4, 1) = \{q_2\}= p_2$



$p_0=C(q_0)= \{q_0, q_2\}$

$\delta(p_0, 0) = C(\delta(q_0, 0) \cup \delta(q_2, 0)) = C(\{q_0, q_1\} \cup \{q_4\})=\{q_0, q_1, q_2, q_4\}=p_1$

$\delta(p_0, 1) = C(\delta(q_0, 1) \cup \delta(q_2, 1)) = C(\{q_3\} \cup \varnothing)=\{q_3, q_2\}=p_2$

$\delta(p_1,0)=C(\delta(q_0,0)\cup\delta(q_1,0)\cup\delta(q_2,0)\cup\delta(q_4,0))=C(\{q_0,q_1\}\cup\varnothing\cup\{q_4\}\cup\varnothing)=\{q_0,q_1,q_2,q_4\}=p_1$

$\delta(p_1,1)=C(\delta(q_0,1)\cup\delta(q_1,1)\cup\delta(q_2,1)\cup\delta(q_4,1))=C(\{q_3\}\cup\{q_1,q_4\}\cup\varnothing\cup\{q_3\})=\{q_1,q_2,q_3,q_4\}=p_3$

$\delta(p_2,0)=C(\delta(q_2,0)\cup\delta(q_3,0))=C(\{q_4\}\cup\varnothing)=\{q_4\}=p_4$

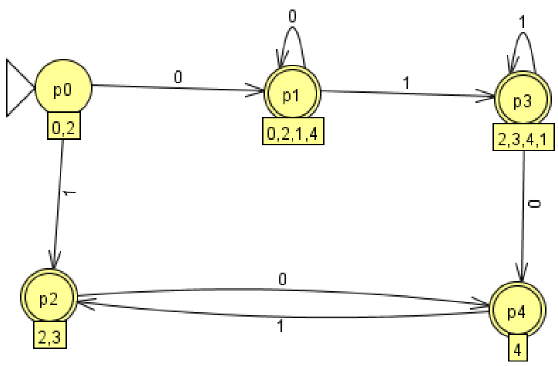
$\delta(p_2,1)=C(\delta(q_2,1)\cup\delta(q_3,1))=C(\varnothing\cup\varnothing)=\varnothing$

$\delta(p_3,0)=C(\delta(q_1,0)\cup\delta(q_2,0)\cup\delta(q_3,0)\cup\delta(q_4,0))=C(\varnothing\cup\{q_4\}\cup\varnothing\cup\varnothing)=\{q_4\}=p_4$

$\delta(p_3,1)=C(\delta(q_1,1)\cup\delta(q_2,1)\cup\delta(q_3,1)\cup\delta(q_4,1))=C(\{q_1,q_4\}\cup\varnothing\cup\varnothing\cup\{q_3\})=\{q_1,q_2,q_3,q_4\}=p_3$

$\delta(p_4, 0) = C(\delta(q_4, 0)) = \varnothing$

$\delta(p_4, 1) = C(\delta(q_4, 1)) = C(\{q_3\})= \{q_3, q_2\}=p_2$



12. (1,5 puntos) Diseña una **gramática independiente del contexto** que genere el lenguaje **L**. Es imprescindible justificar su diseño.

$$L = \{ a^i b^n c^k : (3i = n \lor 2i = k) \land i,n,k \geq 0 \}$$

$S \rightarrow AC \mid D$

$A \rightarrow aAbbb \mid \varepsilon$

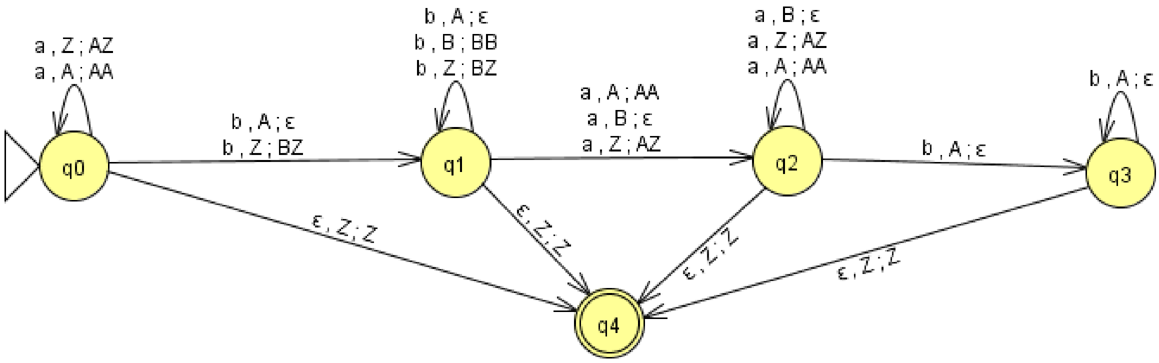
$C \rightarrow cC \mid \varepsilon$

$D \rightarrow aDcc \mid B$

$B \rightarrow bB \mid \varepsilon$

La gramática ha de generar palabras de dos tipos aib3ick que generamos a partir de la primera regla. Con la A generamos las a's y b's con la relación pedida y el orden pedido y con la C cualquier número de c's. A partir del no terminal D generamos las palabras de la forma aibnc2i comenzando a generar por lso extremos las a's y c's con la relación pedida y después B genera en medio cualquier número de b's.

13. (1 punto) Describe el lenguaje reconocido por el siguiente autómata con pila. Elige una palabra representativa del lenguaje y describe el cómputo que la acepta.



L(M) = $\{a^i b^k a^m b^n : i,m,k,n \geq 0 \wedge i+m=k+n\}$

Ejemplo de cómputo para la palabra **aaabbabb**

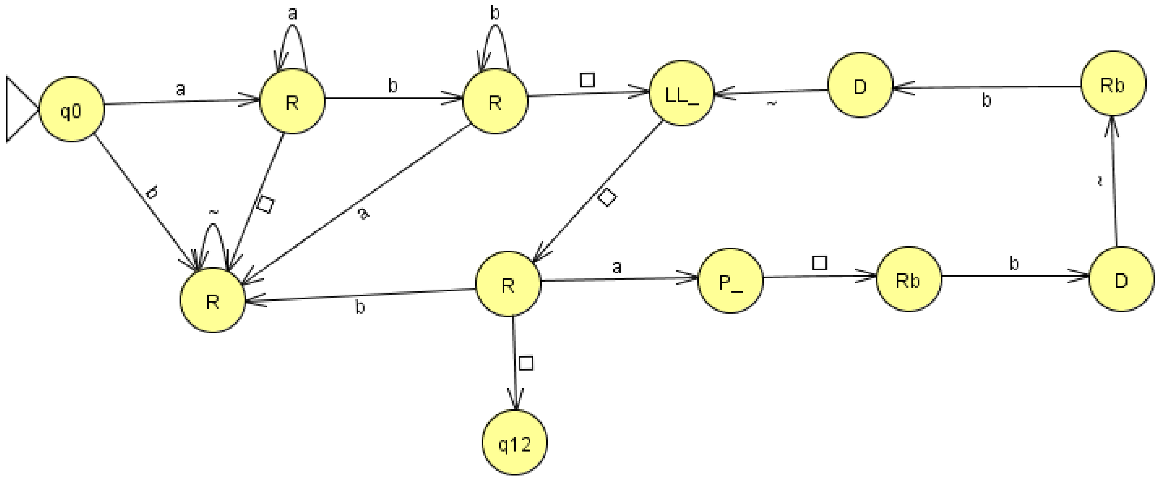
$(q_0, aaabbabb, Z) \vdash (q_0, aabbabb, AZ) \vdash (q_0, abbabb, AAZ) \vdash (q_0, bbabb, AAAZ) \vdash (q_1, babb, AAZ) \vdash (q_1, abb, AZ) \vdash (q_2, bb, AAZ) \vdash (q_3, b, AZ) \vdash (q_3, \varepsilon, Z) \vdash (q_4, \varepsilon, Z)$

Ejemplo de cómputo para la palabra **aabbbaab**

$(q_0, aabbbaab, Z) \vdash (q_0, abbbaab, AZ) \vdash (q_0, bbbaab, AAZ) \vdash (q_1, bbaab, AZ) \vdash (q_1, baab, Z) \vdash (q_1, aab, BZ) \vdash (q_2, ab, Z) \vdash (q_2, b, AZ) \vdash (q_3, \varepsilon, Z) \vdash (q_4, \varepsilon, Z)$

14. (1 punto) La siguiente Máquina de Turing M con alfabeto de entrada {a,b}, calcula una función de la forma

$$\varphi_M(x) = \begin{cases} \varepsilon & si\ x \in L \\ \perp & c.c. \end{cases}$$



Indica cuál es el lenguaje L.

$L = \{a^n b^{2n} : n \geq 0\}$