

Teoría de Autómatas y Lenguajes Formales


Prueba de Evaluación Final

Autores:

Araceli Sanchis de Miguel
Agapito Ledezma Espino
Jose A. Iglesias Martínez
Beatriz García Jiménez
Juan Manuel Alonso Weber



UNIVERSIDAD CARLOS III DE MADRID TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES. GRADO EN INGENIERÍA INFORMÁTICA.
--

	<i>APELLIDOS</i> <i>(En mayúsculas)</i>			
	<i>NOMBRE</i> <i>(En mayúsculas)</i>			
	NIA		DNI	

INSTRUCCIONES PARA LA REALIZACIÓN DEL EXAMEN

- Para que una pregunta de test puntúe, todas las respuestas correctas deben estar marcadas y ninguna de las incorrectas lo ha de estar. En ningún caso las preguntas de test restan puntos.
- **Tiempo de examen (TEST + PROBLEMAS): 3 horas y media.**
- **A los 40 minutos del inicio se recogerán las hojas de TEST.**

TEST (Responder en la hoja del enunciado) 3 puntos.

1. Marque las afirmaciones verdaderas:
 - a. Si $\Sigma = \{a, b, c\}$ entonces $\Sigma^0 = \lambda$.
 - b. $\Sigma = \{a, b, \lambda\}$ es un alfabeto.
 - c. Si $\Sigma = \{a, b, c\}$, entonces $\{a, b, c\}$ puede ser un lenguaje sobre Σ .
 - d. Si $\Sigma = \{a, b, c\}$, entonces \emptyset no es un lenguaje sobre Σ .

2. Marque las afirmaciones verdaderas:
 - a. $ABC ::= AB$ es una regla de una gramática de tipo 1, sensible al contexto.
 - b. Toda gramática regular también es en una gramática independiente del contexto.
 - c. Toda gramática de tipo 0 puede transformarse en una gramática sensible al contexto equivalente.
 - d. Toda gramática de tipo 0 sin estructura de frase puede transformarse en una gramática equivalente de tipo 0 con estructura de frase.

3. Marque las afirmaciones verdaderas:
 - a. Si la cardinalidad del conjunto cociente Q/E_0 es 3, es imposible que la cardinalidad de Q/E_1 sea 2.
 - b. Si $Q/E_0 = Q/E_1$ entonces el autómata finito es mínimo.
 - c. En un autómata de 6 estados, Q/E_3 siempre es Q/E .
 - d. Para determinar si dos autómatas finitos son equivalentes bastará saber si sus autómatas mínimos son isomorfos.

4. Marque las afirmaciones verdaderas:
 - a. Podemos utilizar expresiones regulares para describir cualquier lenguaje independiente del contexto.
 - b. Para que dos AFD sean equivalentes, su número de estados debe ser el mismo.

- c. Todo AFND puede transformarse en un AFD equivalente.
- d. Dado un AFD siempre es posible encontrar la expresión regular correspondiente al lenguaje que acepta.

5. Marque las afirmaciones verdaderas:

- a. Si α es una expresión regular, entonces $\alpha\alpha^*=\alpha^*$
- b. Si α es una expresión regular, entonces $\alpha\alpha^*=\alpha^*\alpha$
- c. $D_b(a^*(a+b)^*) = (a+b)^*$
- d. $\delta(a^*bb) = \lambda$

6. Marque las afirmaciones verdaderas:

- a. Existe algún Autómata a Pila (AP) capaz de reconocer el lenguaje vacío ($L = \emptyset$).
- b. Un AP puede carecer de estados finales.
- c. A partir de un AP que reconoce un lenguaje por estados finales puede construirse otro AP que reconoce un lenguaje por vaciado de pila.
- d. Un AP puede tener un solo estado.

7. Marque las afirmaciones verdaderas:

- a. El lenguaje $L = \{a^{n+2}b^n\}$ puede ser reconocido por un AP.
- b. Un lenguaje que tiene λ como una de sus palabras puede ser aceptado por algún AP.
- c. Si en un AP $\Gamma = \{A\}$, $Q = \{p, q\}$, siendo p estado inicial, entonces la correspondiente gramática tendrá entre sus reglas $S ::= (p, A, p) \mid (p, A, q)$
- d. $f(p, x, R) = (p, R)$ corresponde al movimiento $(p, xB, RP) \mid (p, B, RP)$.

8. Marque las afirmaciones verdaderas:

- $L((0+1)^*) = (L(0) \cup L(1))^*$.
- Si la ecuación característica correspondiente a un AF es $X_1 = 1 + X_1 + 0 + X_2 + 0 + 1 + X_0$, entonces el autómata es no determinista.
- Un autómata que acepta el lenguaje expresado por λ puede tener dos estados p (inicial) y q (final) con $f(p, \lambda) = q$.
- Siendo α y β expresiones regulares: $(\alpha\beta)^* = \lambda + (\alpha\beta)^*\beta$.

9. Marque las afirmaciones verdaderas:

- En una máquina de Turing transductora, si la entrada no está bien formada, debe acabar en estado no-final.
- Toda Máquina de Turing tiene un AFD equivalente.
- Una máquina de Turing no puede modificar el contenido de la cinta.
- Una máquina de Turing puede desplazarse varias celdas a la vez después de leer un símbolo.

10. Marque las afirmaciones verdaderas:

- En una máquina de Turing: $f(p,a) \rightarrow (p,a,i)$ indica que mientras lee "a", se sigue en el estado p , se re-escibe "a", y se mueve la cabeza de lectura a la izquierda.
- En una máquina de Turing: $f(p,a) \rightarrow (p,a,i)$ indica que cuando se encuentra "a", se sigue en el estado p , se re-escibe "i", y se mueve la cabeza de lectura a la izquierda.
- En una máquina de Turing $f(r,a) \rightarrow (r,c,d)$ indica que se cambian las "aes" encontradas por "ces" y se cambia de estado.
- En una máquina de Turing $f(q,1) \rightarrow (q,1,d)$ indica que siempre que se encuentra el símbolo "1" se re-escibe.

UNIVERSIDAD CARLOS III DE MADRID TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES. GRADO EN INGENIERÍA INFORMÁTICA.
--

	APELLIDOS <i>(En mayúsculas)</i>			
	NOMBRE <i>(En mayúsculas)</i>			
	NIA		DNI	

Problema 1 (2'5 puntos)

Las especificaciones de un lenguaje de programación son las siguientes:

- Todo programa empieza con la palabra reservada “*program*” y finaliza con la palabra reservada “*end*”.
- El lenguaje incluye cinco tipos de sentencias:
 - Lectura de una variable: *read identificador;*
 - Escritura de una variable: *write identificador;*
 - Sentencias de asignación: *identificador := expresión_aritmética;*
identificador := número_real;
 - Sentencias condicionales: *if condicional then sentencias else sentencia(s) endIf*
 - condicional: *identificador = expresión_aritmética*
identificador = número_real
 - Sentencias de iteración: *while condición do sentencia(s) endWhile*
- En estas dos últimas sentencias (condicional y de iteración), *condicional* es una única sentencia de condición.
- Las sentencias de lectura, escritura y deben acabar en “;”.
- Respecto a los tipos de datos:
 - Los identificadores consisten en una letra seguida o no de cualquier combinación de letras o dígitos.
 - Los números reales utilizan el “.” para separar la parte entera de la decimal.
 - Las expresiones aritméticas deben contener al menos uno de los operadores: “+”, “-”, “*” y “/”. Está permitido el uso de paréntesis siempre que el número de abiertos y cerrados esté emparejado.

Se pide:

1. Determinar una gramática para generar las **palabras incluidas en el lenguaje definido**, es decir, cualquier programa válido en este lenguaje de programación.
2. Sólo para reconocer **expresiones aritméticas** válidas, diseñar un autómata de pila por vaciado. Para el diseño de este AP, simplificar identificador y números reales como símbolos terminales de la gramática.

Problema 2 (2'25 puntos)

Dada la siguiente gramática $G_1 = (\{1, 2\}, \{A, M, N\}, A, P)$ siendo el conjunto de producciones P las siguientes:

$$\begin{aligned} P &= \{A \rightarrow MN \mid M \\ &\quad N \rightarrow 1 \\ &\quad M \rightarrow N \mid 1M21 \mid 121 \\ &\quad N \rightarrow 2M1 \mid \lambda \\ &\quad M \rightarrow 2 \} \end{aligned}$$

Se pide:

1. Contextualizar la gramática dentro de la Jerarquía de Chomsky.
2. Identificar aquellas producciones de G_1 que son válidas en una gramática de tipo 3.
3. Quitar las reglas identificadas en el apartado 2 (sin sustituirlas por ninguna otra), y generar así una nueva gramática (G_2) que no será equivalente.
4. Limpiar y bien formar la gramática resultante (G_2) y obtener a partir de ella las gramáticas equivalentes en FNC (G_3) y FNG (G_4).
5. Comprobar que las palabras: λ , 121 y 12121211 pueden ser generadas por las gramáticas G_2 , G_3 y G_4 .

Problema 3 (2'25 puntos)

Diseñar una Máquina de Turing que calcule el cociente de la división entre números naturales en codificación unaria. Considerar que el dividendo **siempre** es mayor que el divisor, y que **la división es siempre exacta** (sin resto).

Utilizar la codificación unaria en la que: 1 se representa como 1, 2 se representa como 11, 3 se representa como 111, etc. En la cinta inicialmente habrá $dividendo \div divisor$, y deberá finalizar con $dividendo \div divisor = cociente$. Es decir, se debe preservar el contenido original de la cinta.

Por ejemplo, si al inicio hubiera: $1111 \div 11$
le correspondería el fin: $1111 \div 11 = 11$
(en codificación decimal: $4 \div 2 = 2$);

Y si hubiera al inicio: $11111111 \div 111$
le correspondería el fin: $11111111 \div 111 = 111$
(en codificación decimal: $9 \div 3 = 3$);

Se pide:

- a) Definición formal de la séptupla de la Máquina de Turing. Incluir el diagrama de transiciones (no la lista, ni la tabla de la función de transición).
- b) Descripción detallada del algoritmo implementado en la Máquina de Turing.
- c) Explicación del significado de:
 - cada símbolo del alfabeto de la cinta no definido en este enunciado,
 - cada uno de los estados y transiciones, o grupo de estados y transiciones.