TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES
2^o curso. **GESTIÓN**

Problemas varios. 30 de marzo de 1999

1. Obtener una máquina de Moore mínima equivalente a la máquina de Mealy dada por la siguiente tabla:

	a	b	c
1	2/1	4/2	1/0
2	2/2	3/2	2/1
3	4/2	3/2	2/2
4	4/2	4/2	2/2

2. (a) Dados tres lenguajes A, B y C sobre un alfabeto cualquiera, hay una relación de contención entre los lenguajes $A(B-C)$ y $AB-AC$. ¿Cuál?.

   (b) Obtener un ejemplo de lenguajes A, B y C tales que $A(B-C) \neq AB-AC$.

3. Justificar por qué

   (a) La unión de lenguajes independientes de contexto lo es.

   (b) La concatenación de lenguajes independientes de contexto lo es.

   (c) El cierre de un lenguaje independiente de contexto lo es.

4. Sea L un lenguaje sobre un alfabeto A. Definimos

$$INIT(L) = \{x \in A^*/\exists y \in A^* : xy \in L\}$$

   o sea el conjunto de cadenas que son prefijo de alguna cadena de L.

   (a) Razonar por qué $\epsilon \in L$ y $L \subseteq INIT(L)$, si $L \neq \emptyset$.

   (b) Calcular $INIT(L_1)$ para $L_1 = ab^*a$.

   (c) Obtener un reconocedor finito determinista para L_1.

   (d) Obtener un reconocedor finito determinista para $INIT(L_1)$.

   (e) Justificar que, si L es regular, $INIT(L)$ también lo es, mostrando cómo se construiría un reconocedor finito para este último a partir de un reconocedor finito determinista para el primero.

(f) ¿Cómo podría construirse a partir de una gramática lineal por la derecha ($A \rightarrow aB \mid b$) para un lenguaje regular L, una gramática para $INIT(L)$?.

5. Considérese el lenguaje $L = \{a^n b^m \ / \ 0 \leq n < m\}$.

   (a) Demostrar que L no es regular.

   (b) Obtener una gramática independiente de contexto que genere L.

   (c) Deducir, partiendo de la gramática anterior, un autómata a pila que reconozca L.

   (d) Construir razonadamente un autómata a pila determinista que reconozca L.

6. Se considera la gramática G:

$$\begin{array}{lcl} S & \rightarrow & AA \\ A & \rightarrow & AAA \mid a \mid bA \mid Ab \end{array}$$

   (a) Hallar la cadena más corta de $L(G)$ que demuestre que G es ambigua.

   (b) Demostrar que, $\forall k \geq 0$, $A \Rightarrow^* A^{2k+1}$ en G

   (c) Demostrar que , $\forall m, n \geq 0$, $A \Rightarrow^* b^m a b^n$ en G

   (d) Demostrar que si $x \in (a|b)^*$ y $|x|_a$ es un número par estrictamente positivo, entonces $x \in L(G)$

   (e) Demostrar que, si $A \Rightarrow^* x$ y $x \in (a|b)^*$, entonces $|x|_a$ es impar. ¿Quién es $L(G)$?

   (f) Obtener razonadamente una gramática regular, una expresión regular y un autómata finito determinista en forma mínima para $L(G)$.

7.

   (a) Sea M un reconocedor finito determinista con k estados. Demostrar que si M acepta una cadena de longitud k o mayor, entonces $L(M)$ es infinito.

   (b) Si L es un lenguaje finito (luego regular) del que se sabe que $a^{2045} \in L$, ¿qué se puede deducir sobre el número de estados de un reconocedor finito determinista para L?.

8. Las dos gramáticas siguientes son $LALR(1)$:

$$G1: \quad S \to Saa \mid a$$

$$G2: \quad S \to aaS \mid a$$

   (a) Simular el análisis por desplazamiento-reducción para la cadena a^5 en cada uno de los casos.

   (b) ¿Cuál de las dos es preferible para ser utilizada por YACC y por qué?

9. Considérese la gramática G siguiente:

$$\begin{array}{lcl} S & \to & (A) \\ A & \to & A; E \mid E \\ E & \to & x \mid S \end{array}$$

   Cada terminal x representa un número real. La gramática genera listas de elementos que son números o nuevamente listas.

   (a) Obtener, a partir de G, una gramática $LL(1)$, y su tabla de análisis sintáctico predictivo.

   (b) Escribir el pseudocódigo de tres de los procedimientos del analizador predictivo basado en la gramática anterior.

   (c) Diseñar un esquema de traducción sobre la gramática de partida que calcule la media aritmética de cada lista generada por cada símbolo S, obteniendo finalmente la media de las listas. Por ejemplo, para la entrada "(2;(1;3);5)" la traducción debe obtener el valor 3, que es (2+(1+3)/2+5)/3.

10. (a) Expresión regular para el lenguaje formado por las cadenas de ceros y unos tales que el tercer símbolo por la derecha sea un 0. Reconocedor finito no determinista que acepte dicho lenguaje.

   (b) Análogamente, pero por la izquierda.

   (c) Construir reconocedores finitos deterministas mínimos para los dos lenguajes anteriores.

11. (a) Obtener el lenguaje (L_1) que genera la gramática dada por las reglas

$$S \to AB \quad A \to cAc \mid a \quad B \to cBc \mid b$$

   (b) Demostrar que L_1 no es regular.

(c) Construir una gramática independiente del contexto que genere el lenguaje

$$L_2 = \{u_1 a v_1 u_2 b v_2 / u_1, u_2, v_1, v_2 \in (a|b)^* \wedge |u_1| = |u_2| \wedge |v_1| = |v_2|\}$$

(INDICACION : $|v_1 u_2| = |u_1| + |v_2|$ y $p + q = q + p$).

(d) Construir un autómata a pila que reconozca el lenguaje L_2

(e) Consideremos ahora el lenguaje $L_3 = \{ww/w \in (a|b)^*\}$ (que no es independiente del contexto). Demostrar que su complementario sí es independiente del contexto, construyendo razonadamente una gramática para él. (INDICACION: si $x \notin L_3$, entonces, o bien su longitud es impar, o puede ponerse en la forma $u_1 e_1 v_1 u_2 e_2 v_2$ donde $e_1, e_2 \in \{a, b\}$, u_1, v_1, u_2 y v_2 son cadenas de $(a|b)^*$ verificando $|u_1| = |u_2|$ y $|v_1| = |v_2|$, es decir, si se "parte" x por la mitad, en la primera de ellas hay al menos un caráter que es distinto del que ocupa la misma posición en la segunda mitad).

(f) ¿Puede ser determinista un autómata a pila que reconozca por estado final el lenguaje L_3?.

(g) Explicar cómo podrían diseñarse máquinas de Turing para reconocer L_3 y su complementario.

12. Se considera la gramática G siguiente:

$$\begin{array}{lcl} S & \rightarrow & (A) \\ A & \rightarrow & xB \mid SB \\ B & \rightarrow & ;A \mid \lambda \end{array}$$

(a) Construir la tabla de análisis sintáctico predictivo para G, mostrando los conjuntos *primeros* y *siguientes* para cada símbolo auxiliar.

(b) Supongamos que cada terminal x representa un número real. Diseñar un esquema de traducción que calcule la media aritmética de cada lista generada por cada símbolo S, obteniendo finalmente la media de las listas. Por ejemplo, para la entrada "(2;(1;3);5)", la traducción debe obtener el valor 3, que es $(2 + (1 + 3)/2 + 5)/3$.

(c) Escribir código YACC para dicho esquema de traducción.

(d) Escribir código LEX para el analizador léxico que requeriría el traductor.

13. (a) Sea L un lenguaje regular sobre el alfabeto $\{a, b\}$. Demostrar que si L contiene cadenas de la forma $a^n b^n$ para valores de n arbitrariamente grandes, entonces L debe contener alguna cadena de la forma $a^n b^m$ para ciertos n y m con $n \neq m$.

    (b) Demostrar qye el complementario de un lenguaje recursivamente numerable o es recursivo o no es recursivamente numerable.

14. Se considera la gramática G siguiente:

$$\begin{array}{lcl} S & \to & aSB \mid aB \\ B & \to & b \mid bb \end{array}$$

    (a) ¿Qué lenguaje genera?

    (b) Diseñar un autómata pila que lo reconozca.

    (c) Para dicho autómata, mostrar un comportamiento sobre las cadenas de entrada que se muestran a continuación, que permita deducir si dichas cadenas son o no reconocidas: ab, abb, $abbb$, $aabbb$.

    (d) Para alguna de dichas cadenas ¿hay más de un comportamiento de reconocimiento?. ¿Por qué?

    (e) Modificar la gramática para obtener una equivalente con la que sea razonable intentar un Análisis Sintáctico Predictivo, y calcular entonces la T.A.S.P. ¿Qué ocurre y por qué?.

    (f) Escribir un analizador sintáctico predictivo basado en la gramática G_1 siguiente:

$$\begin{array}{lcl} S & \to & aSB \mid aB \\ B & \to & b \mid bc \end{array}$$

    (g) Considerar de nuevo la gramática G ¿Provocará algún conflicto en YACC? Si es así, explicar cuál, por qué, cómo será resuelto y si tal resolución reduce el lenguaje que se reconocerá o no. En caso contrario, razonarlo. (INDICACIÓN: considerar la cadena $aabbb$).

    (h) Describir y construir una máquina de Turing con una sola cinta que reconozca el lenguaje generado por G.

15. (a) Demostrar que la función $f(n) = 2^n$ es recursiva primitiva.

    (b) Demostrar que la función $l(n) = \log_2 n$ es $\mu-$recursiva.

16. (a) Construir una máquina secuencial de Moore que obtenga como salida la suma módulo cinco de las cifras decimales que se le van presentando en la entrada. El alfabeto de entrada será $\{0, 1 \ldots 9\}$

y el de salida $\{0, 1, 2, 3, 4\}$. Por ejemplo, para la cadena de entrada 642531, la de salida será 102201.

(b) Basándose en la máquina anterior, mostrar cómo podría hallarse una expresión regular para el conjunto de cadenas de dígitos que verifiquen simultáneamente las tres condiciones siguientes:

  i. Están formadas exclusivamente por unos y doses
  ii. La suma de sus dígitos es un múltiplo de 5
  iii. Ninguno de sus prefijos propios verifica la propiedad *ii*

  Indicar un conjunto infinito de cadenas que verifique estas tres propiedades.

17. Considérese la siguiente definición recursiva de un lenguaje L_1 sobre el alfabeto $\{a, b, c\}$:

- $b \in L_1$ y $\epsilon \in L_1$
- si $x \in L_1$, entonces $axb \in L_1$ y $bxa \in L_1$
- si $x, y \in L_1$, entonces $xy \in L_1$
- No hay otras cadenas en L_1

Se pide:

(a) Construir una gramática independiente de contexto G que genere L_1. Justificar que $L_1 = \{x \in (a|b)^*/|x|_b \geq |x|_a\}$ ¿Es G ambigua?

(b) Obtener, a partir de la gramática anterior, un autómata a pila que reconozca $L_1 - \{\epsilon\}$

(c) ¿Quién es L_1^2? ¿Quién es L_1^*?

(d) Construir un autómata a pila determinista para $L_1\{c\}$.

(e) Obtener justificadamente una gramática que genere el lenguaje $L = \{x \in (a|b)^*/|x|_b \neq |x|_a\}$

(f) Razonar por qué L no es regular.

(g) ¿Quién es L^2?. ¿Quién es L^*?

18. (a) Construir una gramática que genere los números romanos desde 1 hasta 3999. Los números romanos serán cadenas formadas por los símbolos I, V, X, L, C, D, M, que corresponden a los valores respectivos 1, 5, 10, 50, 100, 500, 1000. Los símbolos se escriben en general en orden decreciente de valor y sus valores se suman, excepto en los casos que se obtienen de las siguientes reglas:

i. I, X, C y M pueden repetirse hasta tres veces consecutivas.

ii. V, L y D sólo pueden aparecer una vez en la cadena.

iii. I, X,y C restan cuando aparecen delante de otro de mayor valor, en cuyo caso lo harán una sola vez. El símbolo I sólo resta a V y a X, X sólo a L y a C y C sólo a D y a M.

(b) Construir, utilizando Lex y/o Yacc, un programa que lea números romanos (de 1 a 3999), uno por línea, y muestre para cada uno su valor en nuestro sistema de numeración.

19. (a) Se sabe que, dado un lenguaje L sobre un alfabeto cualquiera, en general $L \subseteq L^+ \subseteq L^*$. Dar un ejemplo, si es posible, de lenguajes sobre el alfabeto $\{a, b\}$ que verifiquen respectivamente:

i. $L \neq L^+ \neq L^*$

ii. $L = L^+ \neq L^*$

iii. $L \neq L^+ = L^*$

iv. $L = L^+ = L^*$

20. (a) Probar que la función $f(x) = x^2$ es recursiva primitiva (partiendo de la definición de estas funciones). Probar que la función parcial $r(x) = |\sqrt{(x)}|$ es μ-recursiva.

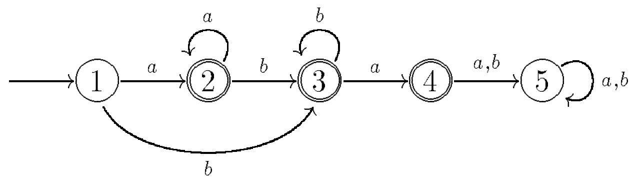
21. Obtener una expresión regular que represente los lenguajes definidos en los siguientes apartados

(a) $(a \mid b^*) \cap (a \mid b)^*$

(b) $(a \mid b)b(a \mid b)^* \cap b(a \mid b)^*$

(c) $aa^* \cap a^*b^+$

(d) El lenguaje L_1 reconocido por el siguiente autómata finito:



(e) $L_1 - a^*bb^*$

(f) $L_1 - aa^*b^*$

22. (a) Demostrar que el lenguaje $L_2 = \{a^n b^m a^n / n, m \geq 0\}$ no es regular.

(b) Demostrar que L_2 es independiente de contexto y obtener una gramática sin reglas simples, no recursiva a la izquierda y que no admita factorización a la izquierda que lo genere.

(c) ¿Es la gramática del apartado anterior $LL(1)$?.

(d) Obtener el pseudocódigo de los procedimientos de un analizador sintáctico predictivo para el lenguaje

$$L_3 = \{a^n b^m a^n / n \geq 0, m > 0\}$$

23. Justificar los siguientes enunciados:

(a) Si L y L' son recursivos, $L \cap L'$ también.

(b) Si L y L' son recursivos, LL' también.

(c) Si L y L' son recursivamente numerables, LL' también.

(d) La función $I(x) = \begin{cases} 0 & \text{si } x \text{ es par} \\ 1 & \text{si } x \text{ es impar} \end{cases}$ es recursiva primitiva.

(e) La función $H(x) = \begin{cases} x/2 & \text{si } x \text{ es par} \\ (x-1)/2 & \text{si } x \text{ es impar} \end{cases}$ es recursiva primitiva

24. (a) Probar que las funciones

$$c(x) = x^2$$

y

$$sc(x) = \begin{cases} 1 & \text{si } x = 0 \\ 0 & \text{si } x < 0 \end{cases}$$

son recursivas primitivas.

(b) Obtener el valor de $rc(x) = \mu_y(x \dot{-} c(y) = 0)$ para $x = 0, 1 \ldots 10$ Describir las funciones rc, $c \circ rc$ y $sc \circ c \circ rc$.

(c) Justificar si el lenguaje $\{1^{i^2} / i \geq 0\}$ es o no

i. independiente de contexto

ii. recursivo

iii. recursivamente numerable

25. Las dos gramáticas siguientes son $LALR(1)$:

$$G1: \quad S \rightarrow Saa \mid a$$

$$G2: \quad S \rightarrow aaS \mid a$$

(a) Simular el análisis por desplazamiento-reducción para la cadena a^5 en cada uno de los casos.

(b) ¿Cuál de las dos es preferible para ser utilizada por YACC y por qué?

(c) ¿Es alguna de las dos $LL(1)$?. Si lo son, calcular su TASP. Si no, realizar si es posible las transformaciones necesarias para obtener otras equivalentes que lo sean, calculando su TASP.

(d) Simular los análisis predictivos basados en las gramáticas del apartado anterior para a^5 y a^2

26. (a) Probar que la intersección de dos lenguajes regulares es regular.

    (b) Demostrar que si un lenguaje L es regular, el conjunto de cadenas de L cuya longitud es par también lo es. ¿Es regular el conjunto de cadenas de L de longitud impar?.

    (c) Expresión regular para el subconjunto de cadenas de $c^*(a|bc^*)$ cuya longitud sea par.

    (d) Construir un reconocedor finito determinista en forma mínima para el lenguaje $c^*(a|bc^*)^*$. (NOTA: no es el mismo lenguaje del apartado anterior).

27. Considérese la gramática G dada por las reglas:

$$\begin{array}{lcl} S & \rightarrow & Rc \\ R & \rightarrow & aRbR \mid \lambda \end{array}$$

    (a) Demostrar que que todas las cadenas que genera G tienen la forma xc, donde $x \in (a|b)^*$ y que cumple además:

    (p) $|x|_a = |x|_b$ (el mismo número de aes que de bes)
    (q) todo prefijo y de x verifica $|y|_a \geq |y|_b$

    Observar que las cadenas x son las cadenas de paréntesis equilibrados, si redenominamos a como (y b como).

    El recíproco del enunciado anterior es también cierto: toda cadena de la forma xc donde $x \in (a|b)^*$ y x verifica (p) y (q) es generable por G. (De este hecho no se pide demostración, aunque debe usarse en el apartado c).

    (b) Deducir de G un autómata a pila que reconozca el lenguaje $L(G)$.

    (c) El autómata a pila del apartado anterior es no determinista. Basándose en las propiedades de $L(G)$, construir razonadamente un autómata a pila determinista reconocedor de $L(G)$.

    (d) Deducir del apartado anterior otra gramática que genere $L(G)$.

(e) Las cadenas de $L(G)$ están compuestas por la concatenación de subcadenas equilibradas de aes y bes (grupos), y a continuación la c final. Por ejemplo $aabbabc$ tiene dos grupos, $aabbababc$ tres, mientras que $aababbc$ sólo tiene de un grupo. Construir, sobre la gramática G de partida, un esquema de traducción que obtenga el número de grupos que tiene una determinada cadena.

(f) Escribir el pseudocódigo de un traductor predictivo que implante dicho esquema de traducción.

28. Explicar cómo podría construirse una Máquina de Turing que genere el lenguaje $\{a^{2^i}/i \in N, i \geq 0\}$. Las cadenas del lenguaje deben aparecer en la cinta de salida separadas por el símbolo #.

29. La siguiente gramática (G) genera ciertas expresiones polinómicas sobre la indeterminada $\mathbf{x}$ y la constante $\mathbf{c}$.

$$\begin{array}{lcl} P & \rightarrow & P+T \mid P-T \mid T \\ T & \rightarrow & \mathbf{c}*Q \mid Q \mid \mathbf{c} \\ Q & \rightarrow & Q*F \mid F \\ F & \rightarrow & \mathbf{x} \mid (P) \end{array}$$

(a) Obtener una gramática equivalente $LL(1)$ y su TASP.

(b) Explicar el significado de que la casilla $TASP(F,*)$ esté vacía, buscando un ejemplo en el que se acceda a ella, y mostrando la acción de un analizador predictivo trabajando en modo de pánico.

(c) Añadir a la $TASP$ elementos de sincronización. ¿Qué cambia en el reconocimiento de las cadenas "$x*x$" y "$x**x$" respecto a no considerar estos elementos?

(d) Obtener un esquema de traducción para G que calcule el grado del polinomio reconocido.

(e) Obtener el correspondiente esquema de traducción para la gramática $LL(1)$ obtenida anteriormente.

(f) Pseudocódigo de los procedimientos asociados a P y P' del traductor predictivo dirigido por la sintaxis que se obtendría de los apartados anteriores.

30. Dado el autómata siguiente, obtener un R.F.D. mínimo equivalente, el lenguaje que reconoce, y gramáticas regulares por la izquierda y por la

derecha para tal lenguaje.

	0	1
$->1$	$2,4$	$3,4$
2		$3,4$
3	$2,4$	
(4)		

31. (a) Considérense las expresiones siguientes, que tienen el mismo valor (aritméticamente):
    i. $ax^3 + bx^2 + cx + d$
    ii. $x^3a + x^2b + xc + d$
    iii. $((ax + b)x + c)x + d$

    Mostrar el árbol sintáctico y la notación polaca inversa para cada una de ellas. ¿Hay alguna ventaja en elegir una u otra forma en cuanto a número de operaciones o espacio de almacenamiento para el código intermedio?. Mostrar igualmente el grafo dirigido acíclico para ambas, mediante números de valor. ¿Qué diferencias hay ahora?.

    (b) Explicar brevemente qué problemas pueden encontrarse al intentar un reconocimiento descendente recursivo sobre una gramática, y cómo pueden, si es que pueden, solucionarse.

32. Se considera la gramática siguiente:

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

    (a) Describir todas las cadenas de $L(G)$ en cuyo árbol de derivación no aparezca S en más nodo que el raiz, y todas las cadenas de $L(G)$ en las cuales aparezcan exactamente 3 símbolos x

    (b) Construir una T.A.S.P. para G. Realizar la simulación del análisis sintáctico predictivo basado en la tabla anterior para la cadena $(x; (x; x); x)$

    (c) Añadir a la tabla entradas de sincronización donde se considere conveniente. Dar ejemplos de cadenas que provoquen accesos a cada una de las entradas vacías o con *sinc* de la tabla, indicando aquéllas que sean inaccesibles si las hay. ¿Sería conveniente añadir o modificar alguna entrada *sinc*?.

(d) Supongamos que el símbolo terminal x representa un número real. Diseñar un esquema de traducción que calcule la media aritmética de cada lista generada por cada símbolo S. Por ejemplo, para la entrada $(2;(1;3);5)$, la traducción debe obtener el valor 3 $((2+(1+3)/2+5)/3))$.

(e) Escribir el pseudocódigo de los procedimientos asociados a S y a B del traductor predictivo.

(f) Escribir una especificación Lex para el analizador léxico de este traductor.

33. Considérense las expresiones siguientes, que tienen el mismo valor (aritméticamente):

 (a) $a + a * (a + a * (a + a * a))$

 (b) $a + a * a + a * a * a + a * a * a * a$

 Mostrar el árbol sintáctico y la notación polaca inversa para cada una de ellas. ¿Hay alguna ventaja en elegir una u otra forma?. Mostrar igualmente el grafo dirigido acíclico para ambas, mediante números de valor. ¿Qué diferencias hay ahora?.