Fundamentos de los Lenguajes Informáticos

Grado en Ingeniería Informática

Hoja de ejercicios 8

Ejercicios adicionales

Ejercicio 1 Demuestra que la función de transición extendida cumple $\hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y)$ para todo estado q y cadenas x e y.

Ejercicio 2 Dado $\Sigma = \{a, b\}$, encuentra un lenguaje L tal que $|L^2| = |L| + 1$ y $|L| \geq 2$. Encuentra otros cinco lenguajes que cumplan dicha propiedad.

Ejercicio 3 Diseña un algoritmo que parta de un AFD A y un número n y calcule el número de cadenas de longitud n aceptadas por A.

Ejercicio 4 Demuestra que el siguiente lenguaje es regular, construyendo un AFD apropiado y demostrando en detalle cuál es el lenguaje aceptado por el AFD:

$$\{w \in \{0,1\}^* \mid w \text{ representa un múltiplo de 3 en binario}\}$$

(puede haber ceros al principio de una cadena y ϵ representa el número 0).

Ejercicio 5 Existen algunas simplificaciones para las construcciones usadas para convertir una expresión regular en un ϵ-AFN. He aquí tres de ellas:

1. Para el operador de unión, en lugar de crear nuevos estados inicial y de aceptación, se combinan los dos estados iniciales en un estado con todas las transiciones de ambos estados. Del mismo modo, se combinan los dos estados de aceptación, teniendo todas las transiciones que ir al estado combinado.
2. Para el operador de concatenación, se combina el estado de aceptación del primer autómata con el estado inicial del segundo.
3. Para el operador de clausura, simplemente se añaden transiciones ϵ del estado de aceptación al estado inicial y viceversa.

Cada una de estas simplificaciones, por sí misma, genera una construcción correcta, es decir, el ϵ-AFN resultante para cualquier expresión regular acepta el lenguaje de la expresión. ¿Qué subconjuntos de los cambios 1, 2 y 3 pueden aplicarse a la vez, dando lugar a un autómata correcto para toda expresión regular?

Ejercicio 6 Demuestra que los siguientes lenguajes no son regulares:

1. $\{0^{n^3} \mid n \in \mathbb{N}\}$.
2. $\{0^p \mid p \text{ es primo}\}$.
3. $\{0^{n!} \mid n \in \mathbb{N}\}$.

Ejercicio 7 Estudia si el siguiente lenguaje es regular: $\{www \mid w \in \Sigma^*\}$.

Ejercicio 8 Si L es un lenguaje y a es un símbolo, L/a, el cociente de L entre a, es el conjunto de cadenas w tales que wa pertenece a L. Por ejemplo, si $L = \{a, aab, baa\}$ entonces $L/a = \{\epsilon, ba\}$. Demuestra que si L es regular, también lo es L/a.

Ejercicio 9 Si $w = a_1 \dots a_n$ y $x = b_1 \dots b_n$ son cadenas de la misma longitud, se define $alt(w, x)$ como la cadena $a_1 b_1 a_2 b_2 \dots a_n b_n$ en la que los símbolos de w y de x se alternan, empezando por w. Si L y M son lenguajes, se define $alt(L, M)$ como el conjunto de cadenas de la forma $alt(w, x)$ donde w es una cadena de L y x es una cadena en M de la misma longitud. Demuestra que si L y M son regulares entonces $alt(L, M)$ también lo es.

Ejercicio 10 Si L es un lenguaje, se define *mitad*(L) como el conjunto de las primeras mitades de las cadenas de L, es decir, $\{w \mid \text{existe } x \text{ con } |x| = |w| \text{ y } wx \in L\}$. Demuestra que si L es regular entonces *mitad*(L) también lo es.

Ejercicio 11 Si L es un lenguaje, se define *tercio*(L) como el conjunto de las cadenas que son tercios centrales en las cadenas de L, es decir, $\{w \mid \text{existen } x, y \text{ con } |x| = |w| = |y| \text{ y } xwy \in L\}$. Demuestra que si L es regular entonces *tercio*(L) también lo es.

Ejercicio 12 Dado $L \subseteq \{0,1\}^*$, sea $L' = \{xy \mid x1y \in L\}$, esto es, L' está formado por las cadenas obtenidas a partir de la eliminación de un 1 en una cadena de L. Demuestra que L' es regular si L lo es.

Ejercicio 13 Ejercicio 7 en las pátinas 317–318 del libro *Automata and Computability* de Dexter C. Kozen (Undergraduate Texts in Computer Science, Springer, 1997).

Ejercicio 14 Diseña una gramática incontextual para el lenguaje consistente en el conjunto de todas las cadenas sobre el alfabeto $\{a, b\}$ que *no* son de la forma ww.

Ejercicio 15 Diseña una gramática incontextual para el lenguaje de las palabras con un número par de $a$es sobre el alfabeto $\Sigma = \{a, b\}$. Demuestra rigurosamente que la gramática incontextual efectivamente genera dicho lenguaje.

Ejercicio 16 Sea el conjunto $a^*b^*c^* \setminus \{a^n b^n c^n \mid n \geq 0\}$ de las cadenas de $a$es seguidas de $b$es seguidas de $c$es tales que el número de $a$es, $b$es y $c$es no es el mismo. Define una gramática incontextual para este conjunto y demuestra informalmente que la gramática es correcta.

Ejercicio 17 Sea G una gramática incontextual sin ninguna producción con ϵ en el lado derecho. Si $w \in L(G)$, la longitud de w es n y w tiene una derivación en m pasos, demuestra que w tiene un árbol de derivación con $n + m$ nodos.

Ejercicio 18 Dada una cadena $w = w_1 \ldots w_n$, se define $w^R = w_n \ldots w_1$. Dado un lenguaje L, se define $L^R = \{w^R \mid w \in L\}$.

1. Demuestra que si L es un lenguaje regular, también lo es L^R.
2. Sea el alfabeto
$$\Sigma = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}.$$
Σ contiene todas las columnas de ceros y unos. Una cadena de símbolos sobre Σ da lugar a tres filas de ceros y unos. Supongamos que cada fila representa un número binario y sea
$$L = \{w \in \Sigma^* \mid \text{la fila inferior de } w \text{ es la suma de las dos superiores}\}.$$
Demuestra que L es regular. *Indicación*: utiliza el apartado anterior.

Ejercicio 19 Sea L un lenguaje. Se define *descolgar*(L) como el lenguaje formado por todas las cadenas que se pueden obtener eliminando un símbolo de una cadena de L. Así, *descolgar*$(L) = \{wv \mid wxv \in L, \text{ donde } w, v \in \Sigma^*, x \in \Sigma\}$. Demuestra que la clase de los lenguajes regulares es cerrado bajo la operación *descolgar*.

Ejercicio 20 Verdadero o falso:

1. Si un lenguaje L es aceptado por un AFN con n estados, debe ser aceptado por algún AFD con 2^n estados.
2. Si un lenguaje L es aceptado por un AFD con 2^n estados, debe ser aceptado por algún AFN con n estados.

Ejercicio 21 ¿Es posible determinar, para todo lenguaje incontextual sin ϵ, una gramática tal que todas sus producciones sean de la forma $A \to BCD$ o de la forma $A \to a$? Demuéstralo o encuentra un contraejemplo.

Ejercicio 22 Dado un lenguaje L, se define $\mathit{inicio}(L) = \{w \mid wx \in L \text{ para algún } x\}$. Demuestra que $\mathit{inicio}(L)$ es incontextual si L lo es.

Ejercicio 23 Dadas dos cadenas x e y, $\mathit{barajar}(x, y)$ es el conjunto de cadenas que se pueden obtener intercalando las posiciones de x y de y de cualquier manera. Por ejemplo, $\mathit{barajar}(01, 110) = \mathit{barajar}(110, 01) = \{01110, 01101, 11001, 10110, 11010, 10101\}$. Si L_1 y L_2 son lenguajes, se define $\mathit{barajar}(L_1, L_2) = \{\mathit{barajar}(x, y) \mid x \in L_1, y \in L_2\}$. Demuestra que si L_1 es incontextual y L_2 es regular, entonces $\mathit{barajar}(L_1, L_2)$ es incontextual.

Ejercicio 24 Una máquina de Turing se para si entra en un estado q señalando un símbolo X de la cinta tal que $\delta(q, X)$ no está definido. Dada una máquina de Turing M, se define $L'(M)$ como el lenguaje de todas las cadenas sobre las que M para. Demuestra que el conjunto de los lenguajes definidos por máquinas de Turing atendiendo a los estados finales coincide con el conjunto de los lenguajes aceptados por máquinas de Turing atendiendo a su parada; es decir:

$$\{L(M) \mid M \text{ máquina de Turing}\} = \{L'(M) \mid M \text{ máquina de Turing}\}.$$

Ejercicio 25 Describe informalmente una máquina de Turing que, dada una lista de palabras sobre $\{0, 1\}$ separadas por $*$, acepte si todas las cadenas son distintas. Se trata, por tanto, de reconocer el lenguaje

$$E = \{*x_1*x_2*\ldots*x_n \mid x_i \in \{0, 1\}^* \text{ y } x_i \neq x_j \text{ para cada } i \neq j\}.$$

Ejercicio 26 Una máquina con k pilas es un autómata con pila determinista, pero con k pilas. Un movimiento de esta máquina depende del estado actual, del símbolo de entrada leído y de la cima de cada una de las pilas, y da lugar a un nuevo estado y a reemplazar el símbolo en la cima de cada pila por una cadena de símbolos, posiblemente distinta para cada una. Por tanto, una regla de transición típica tiene la forma

$$\delta(q, a, X_1, \ldots, X_k) = (p, \gamma_1, \ldots, \gamma_k).$$

Se supone, además, que existe un símbolo especial \$ que solo aparece al final de la entrada de la máquina con k pilas y que no forma parte de la misma. Demuestra que si L es un lenguaje aceptado por una máquina de Turing entonces L también es aceptado por una máquina con 2 pilas.

Ejercicio 27 Demuestra que los siguientes lenguajes son recursivos:

1. $L_1 = \{(R, S) \mid R \text{ y } S \text{ son expresiones regulares y } L(R) \subseteq L(S)\}$.
2. $L_2 = \{G \mid G \text{ es una GI sobre } \{0, 1\} \text{ y } 1^* \cap L(G) \neq \emptyset\}$.

Ejercicio 28 Demuestra que el lenguaje $L_3 = \{M \mid L(M) \neq \emptyset\}$ es r.e.

Ejercicio 29 Demuestra que, dada una máquina de Turing M y un estado q, el problema de determinar si M alcanza el estado q es indecidible.

Ejercicio 30 Demuestra que ni el lenguaje $L_4 = \{M \mid L(M) \text{ es regular}\}$ ni su complementario son r.e.