Universidad de Huelva

Procesadores de lenguaje

Ejercicios del Tema 2

Ejercicio 2.1

Sean L = {a, aa, b} y M = {ab, b }. Describe LM y M^3 por enumercaión

Solución:

LM = { aab, ab, aaab, bab, bb }

M^3 = { ababab, ababb, abbab, abbb, babab, babb, bbab, bbb }

Ejercicio 2.2

Supongamos un lenguaje cuyos comentarios comiencen por << y terminen por >>. Escribe la expresión regular correspondiente a estos comentarios.

Solución:

La primera elección podría ser << .* >>, pero esto acepta cadenas como << a >> a >>.

Solución correcta: << ([^>] | > [^>])* >>

Solución correcta: << (>? [^>])* >>

Ejercicio 2.3

Escribir la expresión regular de un número entero que no acepte que el primer dígito sea cero salvo el número '0'.

Solución:

0 | [1-9][0-9]*

Ejercicio 2.4

Escribir una expresión regular para el conjunto de palabras reservaadas **integer**, **real** y **char**, para un lenguaje que acepte letras mayúsculas y minúsculas.

Solución:

((i|I)(n|N)(t|T)(e|E)(g|G)(e|E)(r|R) | (r|R)(e|E)(a|A)(l|L) | (c|C)(h|H)(a|A)(r|R))

([iI][nN][tT][eE][gG][eE][rR] | [rR][eE][aA][lL] | [cC][hH][aA][rR])

Ejercicio 2.5

¿Cuáles de las siguientes expresiones regulares para los comentarios de C son correctas? Da un contraejemplo para las erróneas.

a) / * .* */

b) / * [^*/]* */

c) / * ([^*] | * [^/])* * /

d) / * ([^*]* * + [^*/])* [^*]* *+ /

e) / * (** [^*/] | /)* *+ /

Solución:

a) incorrecta. Acepta algo como /* comentario */ sigue */

b) incorrecta: No acepta algo como /* comentario * sigue */

c) incorrecta: No acepta algo como /* comentario **/

d) correcta: ([^*]* * + [^*/])* representa cualquier cadena sin * que termine *[^*/].

e) correcta:

Ejercicio 2.6

Escribir una expresión regular para el comentario de una línea de C++.

Solución:

// [^\n]* \n

Ejercicio 2.7

Diseña expresiones regulares para los siguientes lenguajes:

a) Cualquier secuencia de caracteres encerrada entre llaves que no contenga ni el carácter | ni la llave cerrada.

b) Cualquier secuencia de caracteres encerrada entre llaves que no contenga la llave cerrada ni el carcater | salvo que vaya precedido de la barra invertida (\|).

c) Las direcciones IP en formato numérico (por ejemplo, 127.0.0.1).

Solución:

a) { [^|}]* }

b) { ([^|}] | \\ \|)* }

c) ([0-9] | [0-9][0-9] | [01][0-9][0-9] | 2[0-4][0-9] | 25[0-5]).
([0-9] | [0-9][0-9] | [01][0-9][0-9] | 2[0-4][0-9] | 25[0-5]).
([0-9] | [0-9][0-9] | [01][0-9][0-9] | 2[0-4][0-9] | 25[0-5]).
([0-9] | [0-9][0-9] | [01][0-9][0-9] | 2[0-4][0-9] | 25[0-5])

Ejercicio 2.8

¿Qué lenguajes representan las siguientes expresiones regulares?

a) 0 (0|1)* 0

b) (0|1)* 0 (0|1) (0|1)

c) 0* 1 0* 1 0* 1 0*

d) (00|11)* ((01|10)(00|11)* (01|10)(00|11)*)*

Solución:

a) binarios que empiecen y terminen en 0

b) binarios de al menos 3 dígitos cuyo tercer último dígito sea un 0.

c) binarios con tres dígitos 1.

d) cadenas de un número par de dígitos binarios. Si los transformamos en dígitos 0, 1, 2 y 3, representa las cadenas con un número por de 1s o 2s. Cadenas con un número poar de 0s y 1s.

Ejercicio 2.9

Escribe expresiones regulares para los siguientes lenguajes:

a) Todas las cadenas de letras que contengan las cinco vocales en orden (las vocales pueden repetirse.

b) Todas las cadenas de letras que estén en orden lexicográfico ascendente.

c) Comentarios que consisten en una cadena encerrada entre /* y */, sin ningún */ intermedio salvo que aparezca entre comillas.

d) Todas las cadenas de dígitos sin ningún dígito repetido

e) Todas las cadenas de dígitos con a lo sumo un dígito repetido

f) Todas las cadenas de 0 y 1 con un número par de 0s e impar de 1s.

g) Todas las cadenas de 0 y 1 que no contienen la subcadena 011.

Solución:

a) [a-df-hj-np-tv-z]* [b-hj-np-tv-z]* [b-df-np-tv-z]* [b-df-hj-tv-z]* [b-df-hj-np-z]*

b) [aA]*[bB]*[cC]* .... [zZ]*

c) / * ([^*] | “*/” | * [^/])* *$^{+}$/

d) Es posible aunque complicadísimo.

Solución errónea: [0-9]* --- puede aceptar duplicados

Solución inicial: (0|λ)(1|λ)(2|λ)(3|λ)(4|λ)(5|λ)(6|λ)(7|λ)(8|λ)(9|λ)

Ahora habría que hacer todas las permutaciones posibles: 3.628.800

Este tipo de problema se resuelve con [0-9]* y comprobación posterior.

e) Sería como el ejemplo anterior pero con 10 opciones para cada permutación.

Solución inicial: (0|λ)(1|λ)(2|λ)(3|λ)(4|λ)(5|λ)(6|λ)(7|λ)(8|λ)(9|λ)[0-9]

f) (00|11)* ((01|10)(00|11)* (01|10)(00|11)*)*

1

(00|11)* ((01|10)(00|11)* (01|10)(00|11)*)*

g) (1 (0^{+}10|λ))* (0^{+}1| 0^{+} |λ)

Ejercicio 2.10

Escribe los autómatas finitos deterministas para las siguientes expresiones:

a) (a|λ) b*

b) (a|λ) b* b

c) ((a|λ) b*)*

d) ((a|λ) b*)* b

Solución:

a) Estado1: (·a|λ)b* , (a|λ)·b* , (a|λ)b*·

Estado2: (a|λ)·b* , (a|λ)b*·

Transiciones: (1->2, "a"), (1->2, "b"), (2->2, "b")

Estados finales: (1,2)

b) Estado1: (·a|λ)b*b , (a|λ)·b*b , (a|λ)b*·b

Estado2: (a|λ)·b*b , (a|λ)b*·b

Estado3: (a|λ)·b*b , (a|λ)b*·b, (a|λ)b*b·

Transiciones: (1->2, "a"), (1->3, "b"), (2->3, "b")

Estados finales: (3)

c) Estado1: ((·a|λ)b*)*, ((a|λ)·b*)* , ((a|λ)b*)*·

Estado2: ((a|λ)·b*)*, ((a|λ)b*)*·, ((·a|λ)b*)* ¡¡Es el mismo que el 1 !!

Transiciones: (1->1, "a"), (1->1, "b")

Estados finales: (1)

d) Estado1: ((·a|λ)b*)*b, ((a|λ)·b*)*b , ((a|λ)b*)*·b

Estado2: ((a|λ)·b*)*b, ((·a|λ)b*)*b, ((a|λ)b*)*·b ¡¡ Es el mismo que el 1 !!

Estado2: ((a|λ)b*)*b·, ((a|λ)·b*)*b, ((·a|λ)b*)*b, ((a|λ)b*)*·b

Estado3: ((a|λ)·b*)*b, ((·a|λ)b*)*b, ((a|λ)b*)*·b ¡¡ Es el mismo que el 1 !!

Transiciones: (1->1, "a"), (1->2, "b"), (2->1, "a"), (2->2, "b")

Estados finales: (2)

Ejercicio 2.11

Escribe los autómatas finitos deterministas para las siguientes expresiones:

a) ab?c

b) ab?b

c) ab+c

d) ab+b

Solución:

a) Estado1: ·ab?c

Estado2: a·b?c, ab?·c

Estado3: ab?·c

Estado4: ab?c·

Transiciones: (1->2, “a”), (2->3, “b”), (2->4, “c”), (3->4, “c”)

Estados finales: (4)

b) Estado1: ·ab?b

Estado2: a·b?b, ab?·b

Estado3: ab?·b, ab?b·

Estado4: ab?b·

Transiciones: (1->2, “a”), (2->3, “b”), (3->4, “b”)

Estados finales: (3, 4)

c) Estado1: ·ab^{+}c

Estado2: a·b^{+}c

Estado3: ab^{+}·c, a·b^{+}c

Estado4: ab^{+}c·

Transiciones: (1->2, “a”), (2->3, “b”), (3->3, “b”), (3->4, “c”)

Estados finales: (4)

d) Estado1: ·ab^{+}b

Estado2: a·b^{+}b

Estado3: ab^{+}·b, a·b^{+}b

Estado4: a·b^{+}b, ab^{+}·b, ab^{+}b·

Transiciones: (1->2, “a”), (2->3, “b”), (3->4, “b”), (4->4, “b”)

Estados finales: (4)

Ejercicio 2.12 (1er parcial curso 04/05))

(a) Escribir una expresión regular que genere cadenas que comiencen y terminen por comillas (“), cuyo contenido admita cualquier carácter, incluido las comillas si van precedidas de la barra invertida (\”). Por ejemplo: **“Esta es una cadena \” que incluye comillas”**.

(b) Generar el autómata finito determinista a partir de la expresión anterior.

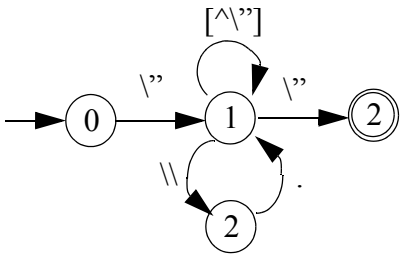
Solución:

(a) Vamos a considerar la barra invertida como carácter de escape para introducir los símbolo especiales. Con esta notación, la expresión para las comillas sería (\”) y la expresión para la bar-

ra sería (\\). El punto (.) se utiliza para designar a cualquier carácter. La espresión regular es la siguiente:

\" ([^\"] | \\ .)* \"

El autómata finito determinista de la expresión anterior es el siguiente:



Ejercicio 2.13 (1er parcial curso 05/06)

Los literales de tipo carácter en Java se pueden introducir de cuatro formas: caracteres imprimibles, caracteres con escape, caracteres en formato octal y caracteres unicode.

Los caracteres imprimibles son los que se representan por códigos ASCII mayores que 31 y menores que 256, a excepción de los siguientes: barra invertida (\ ), comilla simple (‘), comilla doble (“) y el código 127.

Los caracteres de escape se forman con la barra invertida seguida de otro símbolo. Las opciones son: salto de línea (\n), retorno de carro (\r), tabulador (\t), nulo (\f), barra invertida (\\ ), comilla simple (\’) y comilla doble (\”).

Los caracteres en formato octal se representan por la barra invertida seguida de uno, dos o tres dígitos octales. Por ejemplo, \0 \15 \163.

Los carateres en formato unicode se representan por medio de la barra invertida, seguida de una letra ‘u’ o ‘U’, seguida de cuatro dígitos en formato hexadecimal. Por ejemplo: \u005F \u007e.

Un literal de tipo carácter en Java se representa por una comilla simple, seguida de la representación del carácter en alguno de los cuatro formatos indicados anteriormente y terminado en comila simple. Por ejemplo: ‘A’, ‘\n’, ‘\163’, ‘\u007B’.

(a) Realizar una expresión regular que describa los literales de tipo carácter de Java.

(b) Realizar un autómata finito determinista para la expresión obtenida en el apartado anterior.

NOTA: para simplificar el problema, utilize ‘CHAR_IMP’ para denotar los caracteres imprimibles, sin necesidad de utilizar una expresión regular que los describa.

Solución:

(a) Realizar una expresión regular que describa los literales de tipo carácter de Java.

\’ (CHAR_IMP | \\ [nrtf\’”] | \\ [0-7] | \\ [0-7][0-7] || \\ [0-7][0-7][0-7] | \\ (u|U) [0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F]) \’

(b) Realizar un autómata finito determinista para la expresión obtenida en el apartado anterior.