

Recopilación de ejercicios sobre expresiones regulares en exámenes de Compiladores e intérpretes

IG29: Compiladores e intérpretes

Segunda sesión de teoría

Bloque 1: Modelado

Ejercicio 1

Valor: 1,00 puntos

Modela mediante una expresión regular, sin utilizar nombres ni la expresión regular punto, cada uno de los siguientes lenguajes:

- A. El de las secuencias de uno o más dígitos que no contienen dos dígitos pares consecutivos.
- B. El de los comentarios que empiezan por un asterisco y un uno y terminan con el fin de esa línea o bien empiezan por un asterisco y un dos y terminan con el fin de la línea siguiente.

Ejercicio 2

Valor: 1,00 puntos

Modela mediante una única expresión regular, sin utilizar nombres, el lenguaje formado por todas las cadenas de caracteres que cumplen simultáneamente las tres condiciones siguientes:

- Sus tres primeros caracteres son, por este orden, una letra i, una ene y otra i.
- Sus tres últimos caracteres son, por este orden, una letra efe, una i y una ene.
- Entre las tres letras iniciales y las tres letras finales de la cadena aparece una secuencia de uno o más caracteres, ninguno de los cuales es ni una letra ni un salto de línea.

Observa que cada una de las seis letras de la cadena debe poder ser, indistintamente, mayúscula o minúscula, ya que no se ha impuesto ninguna restricción al respecto.

Ejercicio 3

Valor: 0,50 puntos

Sea L_1 el lenguaje de todas las cadenas que pueden formarse utilizando únicamente cero o más dígitos binarios y asteriscos con la restricción de que no se permiten más de dos asteriscos consecutivos. Así, por ejemplo, las siguientes cadenas pertenecerían a L_1 : 101, **, *1*1*1, *00*1*, λ . Pero no estas otras: +001, 1****1, 000***, 911...

Modela el lenguaje L_1 mediante una expresión regular, sin utilizar nombres.

Ejercicio 4

Valor: 0,50 puntos

Sea L_2 el lenguaje de todas las cadenas que pueden formarse utilizando únicamente dígitos decimales y asteriscos con la restricción de que en cada cadena debe haber un único grupo de varios asteriscos consecutivos. Así, por ejemplo, las siguientes cadenas pertenecerían a L_2 : **, *10*1**, 19***99. Pero no estas otras: **1**, x****x, 012345, 1*1, λ ...

Modela el lenguaje L_2 mediante una expresión regular, sin utilizar nombres.

Ejercicio 5

Valor: 0,50 puntos

Sea L_3 el lenguaje de todas las cadenas formadas por cero o más letras minúsculas y que no tienen tres bes seguidas en su interior. Así, por ejemplo, las siguientes cadenas pertenecerían a L_3 : xyz, bb, bebebe, baobab, λ . Pero no estas otras: Gato, abbbba, xxxbbb, sa1u2...

Modela el lenguaje L_3 mediante una expresión regular, sin utilizar nombres.

Ejercicio 6

Valor: 1,50 puntos

Modela mediante una expresión regular, sin utilizar nombres, cada uno de los dos lenguajes siguientes:

- A. El lenguaje de todas las secuencias no vacías que se pueden formar con signos más y letras mayúsculas respetando las dos condiciones siguientes: no habrá dos signos más consecutivos ni tampoco dos letras consecutivas. Por ejemplo: +U+J+I+.
- B. El de los identificadores de cierto lenguaje de programación donde éstos:
- Están formados por dígitos decimales y, opcionalmente, también por letras minúsculas.
 - Su primer carácter es un cero o un uno.
 - Terminan con el primer carácter que, no siendo el primero del identificador, es igual a éste.

Por ejemplo: 1estce007uji1.

Ejercicio 7

Valor: 2,50 puntos

Modela mediante una expresión regular, sin utilizar nombres, cada uno de los dos lenguajes siguientes:

- A. El conjunto de los ficheros no vacíos compuestos por líneas con la siguiente estructura: cada línea debe acabar en un carácter salto de línea y constar de un cierto número (posiblemente nulo) de posiciones, donde cada posición consistirá en un carácter tabulador que podrá ir, o no, seguido de un identificador. Cada identificador estará formado, a su vez, por letras mayúsculas o minúsculas, pero en un mismo identificador no podrán combinarse ambas cajas.
- B. El lenguaje de todas las cadenas que se pueden formar con los dígitos 0, 1 y 2 respetando las dos condiciones siguientes: la cadena no será vacía ni habrá ninguna subsecuencia 01 en ella. Por ejemplo, 02210 pertenecerá a este lenguaje, pero no 2012.

Ejercicio 8

Valor: 2,00 puntos

Modela mediante una expresión regular, sin utilizar nombres, cada uno de los dos lenguajes siguientes:

- A. El conjunto de los ficheros compuestos por cero o más líneas con la siguiente estructura: cada línea debe consistir en una suma de uno o más operandos y acabar en una secuencia de dos caracteres: punto y coma y salto de línea. El primer sumando de cada línea será un identificador formado por letras mayúsculas; cada uno de los restantes sumandos será un dígito octal. Por ejemplo:

A+2+3+0+7;
UNO;
Z+7;

- B. El lenguaje de todas las cadenas no vacías que se pueden formar con las letras minúsculas a, b y c sin que haya dos consonantes seguidas. Por ejemplo, baaaca pertenecerá a este lenguaje, pero no λ , abc o bb.

Ejercicio 9

Valor: 0,50 puntos

Sea L_4 el lenguaje formado únicamente por la cadena vacía y tres cadenas de longitud 1: la letra be minúscula, el guion y la letra jota minúscula. Escribe, sin utilizar el carácter \backslash ni el carácter |, una expresión regular que modele el lenguaje L_4 .

Ejercicio 10

Valor: 1,00 puntos

Modela mediante una expresión regular, sin utilizar nombres, cada uno de los dos lenguajes siguientes:

A. El conjunto L_5 de todas las cadenas que cumplen las siguientes condiciones:

- Cada cadena contiene en su interior las dos subcadenas siguientes: blanco y negro.
- Las cadenas están formadas íntegramente por letras minúsculas.

Así, la cadena monegroshablancosa pertenece a L_5 , pero no muyblanco o BlancoYNegro.

B. El conjunto L_6 de todas las cadenas que cumplen las siguientes condiciones:

- Cada cadena consiste en una única línea, correctamente terminada en el correspondiente carácter salto de línea.
- Aparte del carácter de salto, la línea consta de, al menos, otros dos caracteres.
- Ningún carácter de la línea es un dígito decimal.

Ejercicio 11

Valor: 1,50 puntos

Modela mediante una expresión regular, sin utilizar nombres, cada uno de los dos lenguajes siguientes:

A. El conjunto de literales numéricos de tres dígitos formado por todos los posibles excepto 007. Así, por ejemplo, 033, 921 y 777 pertenecen a este lenguaje, pero no efe, 33 ni, obviamente, 007.

B. El conjunto de todos los posibles comentarios de un hipotético lenguaje de programación en el que éstos comienzan con una secuencia de dos barras verticales `||` y acaban con una nueva barra `|` o con una secuencia `>>`, lo que suceda primero. Así, por ejemplo, `||ho1a>|`, `||x>y>>`, `||barras|` y `||>>` serían cuatro posibles comentarios correctos, pero no `ho1a`, `||x/3>` o `||2+2|4>>`.

Al responder, no utilices expresiones regulares de más de cuarenta caracteres de longitud ni caracteres de escape innecesarios.

Ejercicio 12

Valor: 0,75 puntos

Se desea poder representar cadenas no vacías en las que no aparezcan saltos de línea ni tabuladores. Para ello se utilizarán literales con las siguientes características:

- En el literal, el valor de la cadena vendrá delimitado mediante paréntesis.
- Si el valor de la cadena incluye algún paréntesis, éste, sea abierto o cerrado, deberá aparecer duplicado en el literal.

Así, por ejemplo, la cadena de dieciocho caracteres Algorítmica (II24) quedaría representada mediante el literal siguiente: `(Algorítmica ((II24)))`; por su parte, `:)+<` se representaría como `(:)))+<`.

Modela este lenguaje de literales de cadena mediante una expresión regular, sin utilizar nombres ni caracteres de escape innecesarios.

Ejercicio 13

Valor: 2,00 puntos

Se desea poder representar horas del día con una precisión de, a lo sumo, un minuto; visto de otro modo, se trata de disponer de una notación cómoda para representar cualquiera de los 1.440 minutos del día, considerando que el primer minuto es el número cero y, por lo tanto, el último minuto del día será el 1.439. El usuario de la notación debe poder expresar una misma hora de diversas formas, según sus preferencias, pero respetando algunas restricciones:

- En principio, la notación básica será de cinco caracteres: dos dígitos (la hora), un punto y dos dígitos (el minuto dentro de la hora). Lógicamente, sólo serán literales de hora válidos de 00.00 a 23.59 y no, por ejemplo, 00.75, 24.00 ni 25.10.
 - Si el primer dígito en un literal de hora es un cero, podrá ser omitido, como en 9.30; sin embargo, tras esa omisión no será posible otra: por ejemplo, 00.15 podrá expresarse como 0.15, pero no como .15.
 - Para separar horas de minutos, será posible utilizar el carácter dos puntos en lugar del punto, como en 16:20.
 - A partir del mediodía, se permite reiniciar la cuenta de las horas si al final del literal se añade una te mayúscula o minúscula. Así, por ejemplo, 12.00 podrá expresarse como 00.00T y 16:20, como 4:20t.
 - Análogamente, a la expresión de una hora previa al mediodía se le podrá añadir al final una eme, también mayúscula o minúscula, y así, 11.15M será equivalente a 11.15; pero, por ejemplo, 12.00m o 16:20M no serían literales válidos.
 - Si los dos dígitos correspondientes a minutos son 00, podrán omitirse, siempre y cuando también se omita el correspondiente separador. Así, la seis de la tarde podrán expresarse, por ejemplo, como 18 o 6t.
 - Cuando se omitan los minutos en una hora exacta, en su lugar podrá hacerse constar una hache minúscula, salvo que otra letra (eme o te) forme también parte del literal. Así, aunque las seis de la tarde puedan expresarse como 18h, ninguna de las siguientes cadenas sería un literal correcto: 18H, 18.00h, 6th o 6Ht.
- A. Modela este lenguaje de literales de hora mediante una expresión regular que no tenga más de ochenta caracteres de longitud, sin utilizar nombres ni caracteres de escape innecesarios. Considera la posibilidad de expresar el lenguaje como una disyunción entre dos alternativas: por una parte, literales sin eme ni te y, como segundo término de la disyunción, literales con eme o te.
- B. Escribe una función `lex2min` que reciba como parámetro un componente léxico `comlex` con un atributo `lexema` y calcule en un nuevo atributo, `comlex.mindia`, el minuto del día representado mediante el correspondiente literal de hora. Así, por ejemplo, si el valor de `comlex.lexema` fuera 10.25, `comlex.mindia` debería crearse con el valor 625; o, si el literal fuera 8T, debería calcularse que el correspondiente minuto es el 1.200. Puedes utilizar, según te resulte más cómodo, Python o pseudocódigo. En este segundo caso, puedes suponer dada una única función auxiliar capaz de transformar cadenas en valores numéricos: una llamada `dig2int` que, dado un único dígito, devuelva su valor entero asociado.

Ejercicio 14

Valor: 2,25 puntos

Modela mediante una expresión regular que no tenga más de treinta caracteres de longitud, sin utilizar nombres ni caracteres de escape innecesarios, cada uno de los siguientes lenguajes:

- A. El lenguaje de los literales enteros sin signo que sirven para representar en base diez valores entre quince y cien, ambos incluidos; además, cabe señalar que este lenguaje no prohíbe ceros innecesarios a la izquierda. Así, pertenecerán al lenguaje, por ejemplo, 15, 0082 y 100, pero no XVI, 0101 o cuarenta.
- B. El lenguaje de las cadenas no vacías que están formadas íntegramente por dígitos, o bien íntegramente por letras, o bien no contienen ningún carácter alfanumérico. Así, pertenecerán a este lenguaje, por ejemplo, 2009, XEmacs y :)<, pero no F54, e-mail o λ.
- C. El lenguaje de las siglas delimitadas por secuencias no vacías de asteriscos, donde se entiende que son siglas las secuencias formadas por dos o más letras mayúsculas con un punto inmediatamente detrás de cada una de tales letras. Así, pertenecerán a este lenguaje, por ejemplo, *S.A.*, ***M.A.S.H.*** y **X.Y.*, pero no *A.*, P.I.B. o *ETC...*

Bloque 2: Análisis

En las tablas que siguen, cada fila de la tabla corresponde a una expresión regular; cada columna, a una cadena de caracteres. Debes completar esas tablas escribiendo SÍ o NO en cada una de sus celdas, según si la cadena que encabeza la columna pertenece o no al lenguaje representado por la expresión regular que encabeza la fila.

Ejercicio 15

Valor: 1,00 puntos

| | λ | bca | xxx | x | (xxx) | (z |
|---|-----------|-----|-----|---|-------|----|
| $[\text{^}abc]?$ | | | | | | |
| $([\text{^}a][\text{^}b][\text{^}c])^*$ | | | | | | |
| $[\text{^}(abc)]^+$ | | | | | | |
| $\backslash([\text{^}abc]\backslash)^*$ | | | | | | |

Ejercicio 16

Valor: 1,00 puntos

| | Pa | Pan | PanN | PpAa | Pan* | Xan* | A | \.\.\ | .\.* | .\an |
|--------------------------------|----|-----|------|------|------|-------|---|-------|------|------|
| $(Pp)(Aa)(Nn)^*$ | | | | | | | | | | |
| $[P p][A a][N n]$ | | | | | | | | | | |
| $\.\.\.*$ | | | | | | | | | | |
| $[\text{^}Aa](an)\backslash^*$ | | | | | | | | | | |

Ejercicio 17

Valor: 1,00 puntos

| | () | (01) | (^ab) | 01 | \(01\) | . | (01)* | (. abab) | *) | 01010 |
|--|----|------|-------|----|--------|---|-------|----------|----|-------|
| $\backslash((\text{^}ab)^*\backslash)$ | | | | | | | | | | |
| $(01)^*$ | | | | | | | | | | |
| $[\text{^}.\.]$ | | | | | | | | | | |
| $[(ab)^*\backslash]$ | | | | | | | | | | |

Ejercicio 18

Valor: 1,25 puntos

| | x | (x) | z y | (xyy) | x | ^xxx | x | ^a | x\ | (^.) | (xy |
|---|---|-----|-----|-------|---|------|---|----|----|------|-----|
| $(\text{^}[x.\])^+$ | | | | | | | | | | | |
| $x\ [y z]^+$ | | | | | | | | | | | |
| $\backslash(xy^*\backslash)^+$ | | | | | | | | | | | |
| $\backslash ^[\text{^}]^+\backslash ^*$ | | | | | | | | | | | |

Ejercicio 19

Valor: 1,00 puntos

| | (ab) | b7 | b52s | [\text{^}ab] | (x | b2s | (pe | bas? | ((| xxx | λ |
|----------------------------------|------|----|------|--------------|----|-----|-----|------|----|-----|-----------|
| $\backslash([\text{^}ab])$ | | | | | | | | | | | |
| $(([\text{^}a][\text{^}b])^*)^+$ | | | | | | | | | | | |
| $b[3-72]s$ | | | | | | | | | | | |
| $b[\text{^}0-59]s?$ | | | | | | | | | | | |