Grupo Miércoles 10:00-12:00 semanas B

—Práctica 1—

Autor: Athanasios Usero

NIP: 839543

NOTA:

Se va a emplear, a la hora de trabajar con expresiones regulares, la notación que se utiliza en *Flex,* con el fin de hacer más fácil la posterior lectura de los ficheros .l .

Ejercicio 5:

1. Resumen

En este problema se plantea el formateo de ficheros de texto para cuatro condiciones, todas ellas determinadas por el número y posición en cada línea. Es por ello que se ha de tener extremada precaución en el orden de declaración de reglas sintácticas, para evitar problemas de prioridades.

Los lenguajes regulares desarrollados son, en sumo, la concatenación o cerradura de Kleene de un lenguaje elemental, el del espacio en blanco:

$$L_{e\_blanco} = [\ ]$$

Primero, para poder reconocer líneas de texto vacías, se ha tenido en cuenta que una línea de texto está vacía no cuando no contiene ningún elemento, sino cuando solo contiene (cero o más veces) el espacio en blanco. Se ha creado el siguiente patrón para reconocerlo:

$$r_1 = \text{^}[\ ]^*[\backslash n] \qquad (1)$$

Como se puede apreciar, se ha indicado que la línea tiene que comenzar (^) con 0 o más espacios en blanco (mediante la cerradura de Kleene *), y que tiene que finalizar con un salto de línea (conjunto [\n]), que indica, a su vez, que esta línea finaliza. Así, se evitan confusiones con líneas que acaban o empiezan con espacios en blanco pero contienen otros símbolos y se consume también el salto de línea para efectivamente eliminar toda la línea. A este patrón no se le ha asociado ninguna acción.

Para sustituir los espacios en blanco al inicio de línea por 4 espacios en blanco se ha creado el siguiente patrón asociado a la acción correspondiente:

$$r_2 = \text{^}[\ ]^+ \qquad (2)$$

El patrón identifica las líneas que empiezan con la cerradura positiva (+) del espacio en blanco, puesto que se especifica que la sustitución se realiza en caso de haber espacios en blanco (es decir, 1 o más).

De manera análoga, se ha creado un patrón para reconocer líneas que finalicen con espacios en blanco, pero a este patrón no se le asocia ninguna acción para que simplemente consuma y elimine esos espacios en blanco:

$$r_3 = [\ ]^+\$ \qquad (3)$$

Por último, se desarrolla el siguiente patrón para reconocer dos o más espacios en blanco intermedios, y se le asocia la acción de escribir un espacio en blanco:

$$r_4 = [\ ][\ ]^+ \qquad (4)$$

Aquí lo que se indica es la concatenación de un espacio en blanco y uno o más espacios en blancos (cerradura positiva), es decir, 2 o más espacios en blanco.

Nótese la importancia de orden mencionada. Se han declarado antes (2) y (3), puesto que dos o más espacios en blanco al inicio o al final también se reconocen en (4). De este modo, en caso de poder aplicar más de una regla, se aplica (2) o (3), aplicando así (4) solo en casos de espacios en blanco intermedios (como se pretendía)

2. Pruebas

A modo de prueba se ha elaborado el siguiente fichero de texto a analizar, donde la primera, tercera, quinta y sexta línea son vacías, pero la quinta contiene dos espacios en blanco y la sexta más de dos. Todas estas líneas deberían eliminarse. Además, en las líneas dos, cinco y siete hay palabras separadas por 2 o más espacios, que deberían ser sustituidos por 1. Por último, la segunda línea empieza y termina con 2 espacios (comprobamos que no hay confusión con la regla (4)), y la última empieza y termina con un espacio. Tras el formateo deberían empezar con 4 espacios en blanco y terminar sin ninguno. Asimismo, el inicio y final de la quinta línea debería quedar intacto, al no empezar ni terminar con ningún espacio en blanco.

ENTRADA:

¶

··Esta·linea·empieza·y·termina···con·dos·espacios·en·blanco··¶

¶

··¶

Para·····ser·formateado·que·termina·sin·ningún·espacio¶

.....¶

linea que·····empieza·y·termina·con·un·espacio·¶

SALIDA:

····Esta·linea·empieza·y·termina·con·dos·espacios·en·blanco¶

Para·ser·formateado·que·termina·sin·ningún·espacio¶

····linea·que·empieza·y·termina·con·un·espacio¶

Ejercicio 6:

1. Resumen

En este ejercicio se solicita el análisis de un texto e informe de ciertas características. Las expresiones pueden intersecar en ciertos casos, lo que se resuelve con el orden de las declaraciones y con variables contadoras. Asimismo, en la sección de declaraciones se definen dos nombres para representar el conjunto de dígitos numéricos y las consonantes (mayúsculas o minúsculas):

$$DIGITO\ [0-9]$$
$$CONSONANTE\ [b-zB-Z]\{-\}[aeiouAEIOU]$$

Primeramente, se han creado los patrones para reconocer líneas en blanco y líneas (en general):

$$r_1 = \text{ ^}[\ ]^*[\backslash n] \qquad (5)$$
$$r_2 = [\backslash n] \qquad (6)$$

A ambos patrones se les asocia la acción de incrementar en 1 el valor del contador nl (número de líneas) y asignarle a una variable auxiliar el valor de ese número de líneas (tendrá importancia después), pero a (5) también se le asocia la acción de aumentar en uno el contador nb (nº líneas en blanco) y sumar al contador de caracteres nc el número de símbolos consumidos menos 1 (el salto de línea no cuenta como tal). Ahora bien, se puede observar que (6) contiene todas las posibles líneas, también las líneas vacías. De ahí que se declare primero (5) y se le otorgue prioridad.

Se ha creado el siguiente patrón para reconocer un número, considerado este como una sucesión de dígitos (separada o no):

$$r_3 = \{DIGITO\}^+ \qquad (7)$$

En (7), la expresión regular define la cerradura positiva de dígitos, con independencia del contexto que les rodee. A este patrón se le asocian las acciones de aumentar en uno el contador de números nt y el contador de caracteres nc en el valor de caracteres consumidos.

Para reconocer el número de líneas con algún signo de puntuación, se emplea el siguiente patrón:

$$r_4 = [.,:;] \quad (8)$$

Dicha expresión regular se define como un elemento del conjunto de signos de puntuación, pero no parece ajustarse al requisito. Para ello, son importantes las acciones asociadas. Siempre que se reconozca y consuma un signo de puntuación se aumentará el contador de caracteres en uno, pero solo se aumentará el valor del contador de líneas con signo de puntuación ns si es la primera vez que aparece uno en dicha línea. Para ello, en una variable aux se ha actualizado al final de cada línea el valor del número de líneas recorridas, y cuando se lee un signo, se aumenta el valor de aux y también el de ns. De este modo, ningún otro signo de puntuación contará más que como carácter hasta que finalice la línea y se actualice aux.

Finalmente, como patrón para reconocer líneas que empiezan (más bien, su primer símbolo que no es un espacio empieza) por consonante (con la acción asociada de actualizar los contadores de líneas con consonante y número de caracteres) se propone:

$$r_5 = \text{^}[\ ]^*\{CONSONANTE\} \quad (9)$$

Tampoco hay que olvidar el patrón "residual"

$$r_6 = . \quad (10)$$

Que sirve para consumir todo carácter no contenido en otros patrones y actualizar el número de caracteres en el texto. Se define último para evitar confusiones con cualquier otro carácter que se forme parte de otras reglas.

2. **Pruebas**

El fichero de entrada que se analizará consistirá en 8 líneas, y la primera, tercera y séptima serán vacías (con espacios o no). Además, se han escrito 4 números en la sexta y 1 en la segunda; mientras que hay signos de puntuación en la segunda, cuarta y sexta línea. Además, solo hay una línea que empiece por consonante, pues en la cuarta y sexta línea es consonante la primera letra, no el primer símbolo. El resto de símbolos no reconocidos por un patrón deberán consumirse para contabilizar caracteres.

ENTRADA:

¶

Comienzo, con consonante y num23 ¶

¶

No consonante, linea con signo; ¶

inicio con vocal y no signo¶

hay 4, numeros: 2 3 numeros23.¶

¶

es una linea sin digitos consonante ni signo solo caracteres¶

SALIDA:

TL: 8¶

TB: 3¶

TC: 190¶

TN: 5¶

TS: 3¶

TCons: 3¶

Ejercicio 7:

1. Resumen

Para este ejercicio, trabajaremos cada orden con su expresión regular con notación de Flex, pero teniendo en cuenta que la orden egrep ya implica que con que una parte de una línea sea reconocida por el patrón de la expresión regular se escribirá toda la línea. Si hace falta, se puntualizarán otros cambios para su correcto funcionamiento en Hendrix.

Para reconocer líneas que empiecen y finalicen con un dígito se propone la siguiente expresión regular:

$$r_1 = \text{\^{}}[\ ]^*[0-9].^*\,[0-9][\ ]^*\$ \qquad (11)$$

Este patrón reconoce una línea que empieza con un dígito, al que se le concatena la cerradura de Kleene del lenguaje de caracteres, y que debe terminar con otro dígito (desestimando espacios en blanco en extremos). La cerradura de Kleene es importante, pues sin ella solo se reconocerían líneas con dos dígitos.

Para la segunda orden el lenguaje regular propuesto se puede contemplar como la concatenación de un identificador, uno o más espacios en blanco (al inicio y final cero o más), y una IP. Consideremos, pues, que $r_2 = \text{\^{}}\, r_{22} * r_{21} * [\ ] * r_{22} * r_{23} * r_{22}\, \$$, con

$$r_{2_1} = [a-zA-Z][a-zA-Z0-9]^* \qquad (12)$$

$$r_{2_2} = [\ ]^* \qquad (13)$$

$$r_{2_3} = ([0-9]\{1,3\}[\ .])\{3\}[0-9]\{1,3\} \qquad (14)$$

La expresión regular (12) permite reconocer una palabra que empieza por letra y a la que le siguen una secuencia indefinida de letras y dígitos. La expresión regular (13) reconoce palabras donde aparecen 3 veces una sucesión de 1 hasta 3 dígitos seguidos de un punto y termina con otra sucesión de 1 hasta 3 dígitos. Ahora bien, Hendrix no reconoce las llaves para indicar repetición, por lo que en el caso de repetir un bloque tres veces se hace manualmente, y cada número se representa con la concatenación de un dígito con otros dos dígitos que pueden aparecer o no (mediante el comando *?*).

Para la última orden, se tiene en cuenta que un número es impar si su último dígito es impar, de modo que el patrón que reconoce la expresión regular

$$r_3 = [0-9]^*[13579][\text{^}123456789] \quad (15)$$

debe ser cualquier sucesión de 1 o más dígitos donde el último siempre sea impar.

2. Pruebas

Se han preparado en un mismo fichero de entrada diferentes líneas para comprobar el funcionamiento de cada orden. Ante la primera orden, solo se deberá mostrar la segunda, cuarta y la décima línea (solo hay dígitos). Ante la segunda orden, solo se deberá mostrar la cuarta y decimotercera línea, el resto de líneas no tienen intención de mostrar identificadores e IPs y, si la tienen, es con errores. Por último, ante la tercera orden, solo tendrían que mostrarse las líneas 2 (contiene un par, pero también un impar), 5 (contiene un impar) y 14 (hay un impar entre dos cadenas).

ENTRADA:

```
 fermin36 24.254.24.4 ¶
27cadena  88¶
 fermin38 26.256.26.4 pepe¶
36 fermin 26.256.26.4 ¶
27 no final ¶
 fermin36 26.256.26 ¶
 fermin36 26.256.26.8.66 ¶
 fermin36 26.256.26.8. ¶
no inicio 56¶
26¶
 fermin36 26.256..26.8 ¶
 fermin36 26.2556.6.8 ¶
 fer36min 2.4.6.8¶
```

SALIDA:

hendrix01:~/ egrep '^[0-9].*[0-9]$' t.txt

27cadena 88

36 fermin 26.256.26.4

26

hendrix01:~/ 0-9][0-9]?[0-9]?[.][0-9][0-9]?[0-9]?[.][0-9][0-9]?[0-9]?[ ]*$' t.txt

fermin36 24.254.24.4

fer36min 2.4.6.8

hendrix01:~/ egrep '[0-9]*[13579][^123456789]' t.txt

27cadena 88

27 no final

fer353min 2.4.6. 8

Ejercicio 8:

1. Resumen

Para plantear este ejercicio, toma valor la relación entre los tres tipos de modificaciones a realizar, puesto que el orden de declaraciones permitirá simplificar expresiones regulares.

Para modificar sucesiones de movimientos que terminan en una coordenada vertical múltiplo de tres, se ha creado el siguiente patrón asociado a la acción de escribir "++" al inicio y final de la cadena reconocida:

$$r_1 = I\big(([H]^*V)\{3\}\big)^*[H]^*F \qquad (16)$$

Dicho patrón reconoce toda cadena que contenga 0 o más subcadenas de 3 movimientos verticales alternados con un número indefinido de movimientos horizontales, y que finalice con la cerradura de Kleene de H (permite asimismo reconocer cadenas sin movimientos verticales, considerado 0 también múltiplo de 3). Es decir, se considera a la totalidad de los múltiplos de 3.

Para modificar cadenas con un número par que no sea múltiplo de 3 de movimientos verticales, se propone el siguiente patrón con la acción correspondiente:

$$r_2 = I\big(([H]^*V)\{6\}\big)^*([H]^*V[H]^*V[H]^*)\{1,2\}F \qquad (17)$$

Esta expresión regular reconoce patrones en los que se realizan un número indefinido de veces 6 movimientos verticales, y además se realizan 1 o 2 veces otros 2 movimientos verticales. Entre estos movimientos hay un número indefinido de movimientos horizontales. Se toman todos los pares situados entre múltiplos pares de 3, es decir, la totalidad de todos los pares no múltiplos de 3.

Para modificar cadenas con un número impar no múltiplo de 3 de movimiento verticales se propone el siguiente patrón:

$$r_3 = I[H]^*V\big(([H]^*V)\{2\}\big)^*[H]^*F \qquad (18)$$

Como se puede observar este patrón reconoce toda cadena con 1+2n movimientos verticales, es decir con cualquier número impar de estos. Ahora bien, el orden de criterios indicará que, en el caso de ser una cadena con un número impar múltiplo de 3 de movimientos verticales, se aplique la regla (16).

Queda así la aplicación de la regla (18) exclusivamente para aquellas cadenas con un número impar no múltiplo de 3 de movimientos verticales, tal como se pretendía.

2. Pruebas

Para probar nuestro código, planteamos un fichero de entrada en el que se indica lo que se debería esperar en el comportamiento del programa.

ENTRADA:

```
NUMERO  PAR DE MOVIMIENTOS NO MULTPLO DE TRES IVHVHVHVF, ¶
NUMERO DE MOVIMIENTOS MULTIPLO DE TRES IHVVHVHF, ¶
NUMERO DE MOVIMIENTOS MULTIPLO DE TRES IVHVHVHVHVVF¶
CERO CONSIDERADO MULTIPLO DE TRES IHHHF ¶
NUMERO IMPAR DE MOVIMIENTOS NO MULTIPLO DE TRES IVHF¶
SUCESION EN CURSO, NO TIENE FINAL IVHVHVHVHV¶
NUMERO  PAR DE MOVIMIENTOS NO MULTPLO DE TRES IVHVHVVHVHHHHVHVVF, ¶
NUMERO IMPAR DE MOVIMIENTOS NO MULTIPLO DE TRES IVHVVVVHF¶
```

SALIDA:

```
NUMERO  PAR DE MOVIMIENTOS NO MULTPLO DE TRES +IVHVHVHHHHHHHHHHVF+, ¶
NUMERO DE MOVIMIENTOS MULTIPLO DE TRES ++IHVVHVHF++,¶
NUMERO DE MOVIMIENTOS MULTIPLO DE TRES ++IVHVHVHVHVVF++¶
CERO CONSIDERADO MULTIPLO DE TRES ++IHHHF++ ¶
NUMERO IMPAR DE MOVIMIENTOS NO MULTIPLO DE TRES -IVHF-¶
SUCESION EN CURSO, NO TIENE FINAL IVHVHVHVHV¶
NUMERO  PAR DE MOVIMIENTOS NO MULTPLO DE TRES +IVHVHVVHVHHHHVHVVF+,¶
NUMERO IMPAR DE MOVIMIENTOS NO MULTIPLO DE TRES -IVHVVVVHF-¶
```