Introducción a las Expresiones Regulares

Teoría de Autómatas y Lenguajes Formales
Alejandro Viloria Lanero (aviloria@infor.uva.es)
Universidad de Valladolid

Las expresiones regulares se utilizan para hacer búsquedas contextuales y modificaciones sobre textos. A pesar de que las expresiones regulares estén muy extendidas por el mundo de Unix, no existe un lenguaje estándar de expresiones regulares. Más bien se puede hablar de diferentes dialectos. Existen por ejemplo dos representantes del conocido programa grep, egrep y fgrep. Ambos usan expresiones regulares con capacidades ligeramente diferentes. Perl se puede calificar como el lenguaje con la sintaxis de expresiones regulares más desarrollado. Por suerte todos estos dialectos siguen los mismos principios y en el momento que se han entendido, el resto es sencillo.

1 Introducción

Para empezar, ubiquemos el problema por medio de un pequeño ejemplo: Supongamos que tenemos la siguiente lista de teléfonos de una empresa:

Tlfno. Nombre Despacho
...
...
3412 Bob 123
3834 Jonny 333
1248 Kate 634
1423 Tony 567
2567 Peter 435
3567 Alice 535
1548 Kerry 534
...

Se trata de una empresa con 500 personas y los datos están almacenados en un fichero ASCII normal. Los registros de personas cuyo teléfono comience con un 1, trabajan en el edificio 1. ¿Quién trabaja en el edificio 1?

Una expresión regular puede responder a eso:
grep '^1' phonelist.txt
o
egrep '^1' phonelist.txt
o
perl -ne 'print if (/^1/)' phonelist.txt

En palabras normales, esto significa: Busca todas las líneas que comiencen con un 1. El símbolo "^" es el encargado de indicar que sólo se busquen los números 1 que se encuentren al principio de la línea.

2 Lenguajes y Expresiones Regulares

Con respecto al ejemplo anterior, podemos identificar los siguientes conceptos:
- *Alfabeto*: Conjunto de caracteres que aparecen en el fichero. Para generalizar nos referiremos siempre a todo el juego ASCII de caracteres.
- *Lenguaje Universal*: Conjunto de todas las posibles secuencias de caracteres de ese alfabeto.

El Lenguaje Universal es demasiado amplio y no permite ningún tipo de restricciones a la hora de definir las secuencias de caracteres. Por eso nos interesa definir *lenguajes* más restrictivos que nos permitan localizar solamente aquellas cadenas de texto (secuencias de caracteres del alfabeto) que nos interesan.

Una Expresión Regular nos sirve para definir lenguajes, imponiendo restricciones sobre las secuencias de caracteres que se permiten en el lenguaje que estamos definiendo. Por tanto una Expresión Regular estará formada por el conjunto de caracteres del alfabeto original, más un pequeño conjunto de caracteres extra (*meta-caracteres*), que nos permitirán definir estas restricciones.

El conjunto de meta-caracteres que está más extendido es el siguiente:

Nombre	**Carácter**	**Significado**
Cierre	'*' (asterisco)	El elemento precedente debe aparecer 0 o más veces
Cierre positivo	'+' (símbolo de la suma)	El elemento precedente debe aparecer 1 o más veces
Comodín	'.' (punto)	Un carácter cualquiera excepto salto de linea
Condicional	'?' (interrogante)	Operador unario. El elemento precedente es opcional
OR	'|' (barra vertical)	Operador binario. Operador OR entre dos elementos. En el lenguaje aparecerá o uno u otro.
Comienzo de línea	^ (ángulo superior)	Comienzo de línea
Fin de línea	$ (símbolo del dólar)	Fin de línea
	[…] (caracteres entre corchetes)	Conjunto de caracteres admitidos
	[^...] (caracteres entre corchetes)	Conjunto de caracteres no admitidos
Operador de rango	'-' (guión)	Dentro de un conjunto de caracteres escrito entre corchetes, podemos especificar un rango (ej., [a-zA-z0-9])
	(…) (elementos entre paréntesis)	Agrupación de varios elementos
Carácter de escape	\ (barra inversa)	Debido a que algunos de los caracteres del alfabeto coinciden con meta-caracteres, el carácter de escape permite indicar que un meta-carácter se interprete como un símbolo del alfabeto
Salto de línea	'\n' (barra inversa + n)	Carácter de salto de línea
Tabulador	'\t' (barra inversa + t)	Carácter de tabulación

3 Ejemplos de Expresiones Regulares

Volviendo al ejemplo inicial, vemos como todos los números de teléfono se encuentran al comienzo de la línea. Para extraer aquellos que empiezan por '1' tan solo es necesario definir la expresión regular:

- ER := '^1'

Si quisiésemos extraer todos los que comenzasen por 1 ó 2:

- ER:= '^[12]'

Si quisiésemos extraer a todos excepto a los que comenzasen por 1 ó 2:

- ER:= '^[^12]'

Si quisiésemos extraer a todos los empleados que ocupan el despacho 123 ó el 124:

- ER:= '(123|124)$'

Si quisiésemos extraer a todos los empleados que ocupan un despacho que comienza por 3 (los de la tercera planta):

- ER:= '3[0-9]*$'

4 Herramientas que permiten la búsqueda por Expresiones Regulares en los entornos UNIX

- **grep, egrep**: Son las dos herramientas básicas a la hora de realizar la localización de patrones que encajan con la expresión regular indicada en ficheros de texto. La diferencia es sutil. grep es el comando básico, mientras que egrep permite la utilización de varias expresiones regulares de forma simultánea y es considerablemente más rápido. Al igual que numerosas herramientas de los entornos UNIX, están diseñadas para actuar como filtros: reciben sus datos de la entrada estándar (consola), y escriben sus resultados en la salida estándar. Para alterar este comportamiento debemos utilizar los operadores de redirección del sistema operativo y/o los de tuberías.

- sed, awk: son dos herramientas más avanzadas que nos van a permitir manipular los ficheros de forma automática utilizando expresiones regulares. Ambas serán explicadas en sesiones sucesivas.

5 Referencias

Para más información acerca de los comandos indicados, podéis consultar la página del manual del sistema operativo:
$> man grep

o bien podéis dirigiros a algunas de estas páginas:
http://iie.fing.edu.uy/~vagonbar/unixbas/expreg.htm
http://www.linuxfocus.org/Castellano/July1998/article53.html
http://www.ciberdroide.com/misc/novato/curso/regexp.html
http://bulmalug.net/body.phtml?nIdNoticia=770