



Trabajo Practico Número 4

Teoría de la Computación

ULISES C. RAMIREZ - LS00704
ulisescolrez@gmail.com

6 de Mayo del 2018

1 Expresiones regulares

Escribir expresiones regulares para los siguientes conjuntos de caracteres, o explique las razones por las que no se pueden escribir expresiones regulares.

Cabe destacar que voy a estar escribiendo las expresiones separadas con espacios, para que estas tengan el funcionamiento adecuado estos no deberán ser introducidos a la hora de utilizarlas. Esta medida la tomo con el fin de poder “separar en terminos” para así poder explicar el funcionamiento de cada una de las expresiones escritas.

1.1 Todas las cadenas de letras minúsculas que comiencen y finalicen con a

$\backslash \mathbf{b} \ a \ [a-z]^+ \ a \ \backslash \mathbf{b}$

Esta expresión regular se puede entender de la siguiente manera:

- $\backslash \mathbf{b}$: esta haciendo dos cosas,
 - la primera de ellas es que permite delimitar la palabra.
 - la segunda es que hace referencia al inicio de la misma.
- \mathbf{a} : significa el carácter \mathbf{a} literal, con eso implicamos que es necesario que el carácter en cuestión este en **minúsculas**, por lo que busca dicho carácter en la palabra.
- $[a-z]^+$: significa que se buscan una o mas combinaciones de los caracteres de la \mathbf{a} hasta la \mathbf{z} .
- \mathbf{a} : significa el carácter literal \mathbf{a} , en **minúsculas**.
- $\backslash \mathbf{b}$: aquí esto tiene la función de delimitación de la palabra que cumple las condiciones anteriores.

1.2 Todas las cadenas de letras minúsculas que comiencen o finalicen con a

$\backslash \mathbf{b} \ a \ [a-z]^+ \ \backslash \mathbf{b} \ | \ \backslash \mathbf{b} \ [a-z]^+ \ a \ \backslash \mathbf{b}$

Ésta expresión regular se puede entender de la siguiente manera: En principio podemos mencionar que se puede subdividir en dos partes, las cuales van a estar separadas por una condición de disyunción la cual va a permitir que podamos tomar como primer proposición a la **palabra que inicia con la letra a (Inicia con la letra a = P)** y como segunda proposición a la **palabra que termina con a (Termina con la letra a = Q)**.

P OR Q

De esta manera podemos darle a P su valor dentro de la expresion regular descrita en 1.2 de la siguiente manera:

$$P = \backslash b \ a \ [a-z]^+ \ \backslash b$$

Igualmente podemos definir a Q como:

$$Q = \backslash b \ [a-z]^+ \ a \ \backslash b$$

A continuación se describe a P, que ya se define como “Palabra que inicia con a”:

- **\b**: delimitador de palabra, ademas de determinar el inicio de la misma.
- **a**: caracter literal a (**minúscula**).
- **[a-z]^+**: combinacion de una o mas repeticiones de caracteres desde la a hasta la z, con la aclaración de que estos caracteres serán obligatoriamente, **minúsculas**
- **\b**: delimitador de palabra.

Finalmente se describe a Q, que está definido como “Palabra que termina con a”

- **\b**: delimitador de palabra, ademas de determinar el inicio de la misma.
- **[a-z]^+**: combinacion de una o mas repeticiones de caracteres desde la a hasta la z, con la aclaración de que estos caracteres serán obligatoriamente, **minúsculas**
- **a**: caracter literal a (**minúscula**).
- **\b**: delimitador de palabra.

1.3 Todas las cadenas de digitos que no contengan ceros al principio

`\b(?:0)\d+\b`

Se puede describir esta expresión regular de la siguiente manera:

- `\b(?:0)`: esta parte de la expresión se refiere a la parte de no iniciar una cadena con ceros, se la puede descomponer como sigue.
 - `\b`: aquí se tiene al delimitador de la palabra, este permite que no se haga uso del caracter “^”(acento circunflejo) que indica el inicio de linea solamente, sino que con este podremos evaluar a cada palabra por separado.
 - `(?:0)`: ésta expresión se conoce como *negative lookahead* y busca las coincidencias que no tengan lo especificado en la expresión, en este caso, busca que no sea un 0.
- `\d+\b`:
 - `\d+` (*decimal*): ésta parte de la expresión se refiere a que todos los caracteres deben ser numéricos entre 0 y 9, y estos pueden aparecer 1 o mas veces en el texto.
 - `\b`: ésta expresión actúa como un delimitador, de ésta manera los numeros pueden estar separados por espacios, y mezclados con caracteres alfanuméricos sin que la expresión los incluya.

1.4 Todas las cadenas de dígitos que representen números pares

`\b\d*[02468]\b`

Se puede describir la expresión de la siguiente manera:

- `\b`: delimitador, este va a permitir que se evalúe cada palabra de cada línea de manera separada.
- `\d*`: expresión que representa a 0 o mas decimales.
- `[02468]`: serie de numeros que puede seguir, de manera individual o mediante alguna combinación, luego de los decimales del punto anterior.
- `\b`: delimitador, permitirá que se puedan diferenciar las palabras de una línea al estar separadas por espacios, saltos de línea, etc.

1.5 Todas las cadenas de dígitos tales que todos los números 2 se presenten antes que todos los 9

$\backslash \mathbf{b} \quad [0 1 2 3 4 5 6 7 8]^+ \quad [0 1 3 4 5 6 7 8 9]^+ \quad \backslash \mathbf{b}$

Se puede describir la expresión de la siguiente manera:

- $\backslash \mathbf{b}$: delimitador, permite que se evalúe cada palabra de cada línea de manera separada.
- $[0—1—2—3—4—5—6—7—8]^+$: expresión que permite determinar si hay una o más ocurrencias de los dígitos descritos.
- $[0—1—3—4—5—6—7—8—9]^+$: expresión que permite determinar si hay una o más ocurrencias de los dígitos descritos.
- $\backslash \mathbf{b}$: delimitador que permite determinar el final de una palabra.

1.6 Todas las cadenas de a y b que no contengan 3 b consecutivas

La consigna no queda clara, voy a dejar esto incompleto hasta hablar con algún responsable de la cátedra

1.7 Todas las cadenas de a y b que no contengan ni un número impar de a o un número impar de b

La consigna no queda clara, se va a tomar el mismo criterio que el tomado en 1.6

1.8 Todas las cadenas de a y b que no contengan un número par de a y un número par de b

La consigna no queda clara, se va a tomar el mismo criterio que el tomado en 1.6 y 1.7

1.9 Todas las cadenas de a y b que contengan tantas a como b

La consigna no queda clara, se va a tomar el mismo criterio que el tomado en 1.6, 1.7 y 1.8

2 Pasar a español las siguientes expresiones regulares

- $(a|b)^* a (a|b|&)$: buscar por 0 o más coincidencias del caracter **a** o **b** seguidos por un caracter **a**, y luego seguido de un caracter **a** o un caracter **b** o un caracter **&**.
- $(A|B|...|Z)(a|b|...|z)^*$:
- $(aa|b)^*(a|bb)^*$: Buscar por la coincidencia de dos letras a o una letra b 0 o multiples veces luego buscar por la coincidencia de una letra a o dos letras b 0 o multiples veces.
- $(1|2|3|4|5|6|7|8|9|A|B|C|D|E|F)^+(X|x)$: Buscar una secuencia que contenga una o mas apariciones de **1** o **2** o **3** o **4** o **5** o **6** o **7** o **8** o **9** o **A** o **B** o **C** o **D** o **E** o **F**, seguido de un caracter que puede ser una **X** o una **x**

3 Trabajar con automatas finitos para los items mencionados en 2

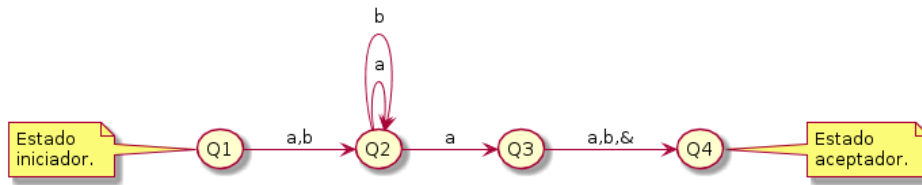


Figure 1: Automata finito para la expresión regular $(a|b)^*a(a|b|&)$

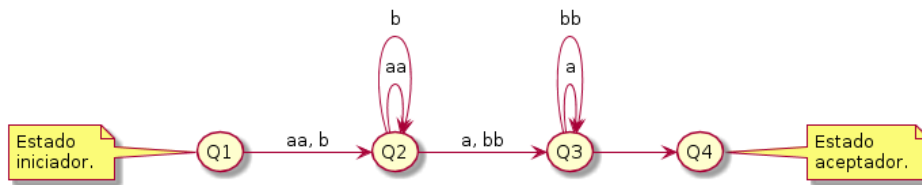


Figure 2: Automata finito para la expresión regular $(aa|b)^*(a|bb)^*$

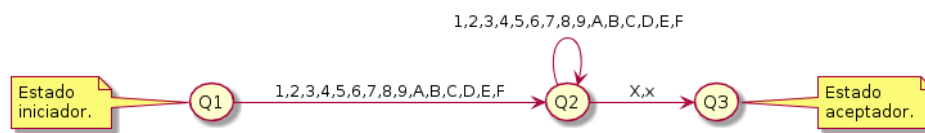


Figure 3: Automata finito para la expresión regular $(1|2|3|4|5|6|7|8|9|A|B|C|D|E|F)+(X|x)$