

## Tarea 1 - Lenguajes regulares

Profesor: Gonzalo Navarro

Auxiliar: Rodrigo Fuentes

Ayudantes: Javiera Alegría  
Gabriel Chaperón  
Matías Rojas

Fecha de entrega: 23 de Octubre, 2018

### Descripción

El objetivo de esta tarea es escribir un programa que reciba un texto y una expresión regular, y encuentre los substrings del texto que satisfagan la expresión. Para esto, considere:

- (a) Construir un AFND  $N$  que reconozca la expresión regular.
- (b) Agregar loops al estado inicial de  $N$  para buscar en texto
- (c) Convertir el AFND  $N$  a un AFD  $D$
- (d) Correr  $D$  con el texto entregado, marcando las ocurrencias del texto en que  $D$  visita algún estado final.  
Las ocurrencias marcadas en este paso son posiciones del texto en donde un segmento aceptado por la expresión regular termina. Para encontrar el inicio del segmento considere:
- (e) Construir el AFND  $N'$  obtenido de  $N$  (sin los loops agregados en el paso (b)) dando vuelta todas sus transiciones, e intercambiando el estado final con el inicial. Este autómata reconoce las palabras que reconoce  $N$ , invertidas.
- (f) Convertir el AFND  $N'$  a un AFD  $D'$ .
- (g) Por cada ocurrencia marcada del texto, correr el AFD  $D'$  en el texto en reversa desde esta, detendiéndose apenas se llegue a algún estado final. Imprimir el segmento de texto que se obtenga.  
El segmento de texto encontrado de esta manera corresponde al más corto que termina en la posición marcada y es aceptado por la expresión regular.

## Notación

El texto está construido exclusivamente con símbolos del alfabeto  $\Sigma = \{a, \dots, z, A, \dots, Z, 0, \dots, 9, \ , \backslash n\}$  (alfanuméricos, espacio y saltos de línea). Para facilitar la conversión de la expresión regular en un AFND, estas vendrán escritas en *notación prefija* utilizando los operadores regulares  $.$  (concatenación),  $|$  (unión) y  $*$  (estrella de Kleene). La definición de esta notación es:

1. Los símbolos de  $\Sigma$  son e.r. en notación prefija.
2. Si  $e_1$  y  $e_2$  son e.r. en notación prefija, entonces  $|e_1e_2$ ,  $.e_1e_2$ ,  $*e_1e_2$  son e.r. en notación prefija.

Observe que en notación prefija no son necesarios los paréntesis, por lo que no se ocupan. Por ejemplo, la notación prefija de la expresión regular  $(ab|aba)^*$  es  $*|.ab..aba$ . Observe además que se puede parsear la expresión regular en notación prefija con un simple programa recursivo.

## Entrega

- El plazo de la entrega vence el día 23 de Octubre. La entrega puede ser personal o en grupos de máximo tres personas. Debe implementar su programa en Python, C, C++ o Java.
- El programa debe recibir dos argumentos en la entrada estándar. El primero es el nombre del archivo que contiene el texto. El segundo es la expresión regular. A su vez, debe imprimir en la consola los segmentos encontrados. Por cada segmento  $x$  encontrado, debe imprimir  $[i, j] : x$  donde  $i, j$  son las posiciones del texto que delimitan  $x$ .
- Además del código fuente debe entregar un breve informe conteniendo una descripción de su programa, instrucciones de compilación y ejecución, y ejemplos de uso.

## Ejemplo

Suponga un archivo de texto *archivo.txt* que solo contiene la palabra *bab*. Entonces la ejecución del programa debe funcionar así:

```
$ ./tarea1 archivo |a..bab
$[1,1]: a
$[0,2]: bab
```