

Teoría de Autómatas y Lenguajes Formales (2008/09)

Práctica 1

(Expresiones regulares y autómatas finitos)

Enunciado

El objetivo de esta práctica es reforzar los conceptos relacionados con las expresiones regulares, con los autómatas finitos, y con algunas de las transformaciones básicas que se pueden realizar entre estos formalismos.

Descarga el fichero [ocaml-talf.tar.gz](http://www.dc.fi.udc.es/~grana/TALF/talf_ii_2008_2009_PL1.html) y descomprímelo en un directorio propio, porque algunas de las operaciones están ya implementadas:

- Se proporcionan dos funciones, `cadena_of_string` y `cadena_of_file`, para construir valores del tipo de dato `simbolo list`.
- Se proporcionan dos funciones, `er_of_string` y `er_of_file`, para construir valores del tipo de dato `er`.
- Se proporcionan dos funciones, `af_of_string` y `af_of_file`, para construir valores del tipo de dato `af`.
- Se proporciona también la función `dibuja_af`, para poder visualizar y/o imprimir la representación gráfica de un valor de tipo `af`.
- Se proporciona la función `escaner_af : Auto.simbolo list -> Auto.af -> bool`, que verifica si una cadena de símbolos terminales es aceptada o no por un autómata finito.

Se pide implementar lo siguiente:

1. Una función `escaner_parcial_af : Auto.simbolo list -> Auto.af -> bool * (int * int) list`, que verifica si cualquier subcadena de la cadena recibida es aceptada o no por un autómata finito. En caso afirmativo, la función devolverá un par cuya primera componente será `true`, y cuya segunda componente será la lista de las posiciones (dentro de la cadena global recibida) del primer y del último símbolo de las subcadenas aceptadas. En caso de que ninguna subcadena sea aceptada, la función devolverá `(false, [])`. Valor de este apartado: 4 puntos.
2. Una función `af_of_er : Auto.er -> Auto.af`, que dada una expresión regular devuelva el autómata finito que acepta exactamente el lenguaje regular denotado por dicha expresión regular. Valor de este apartado: 4 puntos.
3. Un programa ejecutable de nombre `talfgrep`. El programa debe recibir dos parámetros en la línea de comandos: una expresión regular y el nombre de un fichero de texto. Posteriormente, durante su ejecución, el programa debe imprimir por pantalla las líneas del fichero que contienen alguna subcadena que encaja con la expresión regular recibida, precedidas del número de línea que ocupan dentro de dicho fichero, y de las posiciones (dentro de cada línea) del primer y del último símbolo de las subcadenas encajadas. Valor de este apartado: 2 puntos.

Para ello, echa primero un vistazo al fichero [manual.ps](#) o [manual.pdf](#), el cual contiene la definición de los tipos de datos involucrados, con sus correspondientes constructores. Realiza las implementaciones de

todos los apartados en un fichero de nombre `talfgrep.ml`, que es el único que debes entregar después.

Tiempo para la realización de la práctica: 3 semanas.
