

1 Diseño

1.1 El evaluador de expresiones regulares

Antes de iniciar el diseño del árbol sintáctico, se debe comprobar que la expresión regular (ER) que se desea convertir a AFD es válida. Esto significa que la ER debe estar compuesta por sólo símbolos del alfabeto y de operadores de una ER. Además se debe comprobar que el orden o secuencia (sintaxis) de los símbolos de la ER, es el correcto. Un detalle importante que se debe tener en cuenta, es que, normalmente, en las ER el operador de concatenación es omitido, por lo que se debe determinar cuándo se trata de una operación de concatenación y poder agregar dicho operador a una nueva cadena que se formará y convertirá a AFD.

1.2 El algoritmo

Debido a que una expresión regular está conformada por operaciones binarias como la concatenación y la disyunción, el árbol sintáctico que represente a dicha expresión, debe ser un árbol binario. Un árbol sintáctico está constituido por nodos, donde cada nodo puede tener cero (si el nodo es una hoja), uno (si el nodo es una estrella de Kleen) o dos nodos hijos (si el nodo es una concatenación o una disyunción). Un nodo representa la raíz de un subárbol, es decir, la raíz de toda la descendencia del nodo (nodos hijos, nodos hijos de los hijos y así sucesivamente). Cada nodo debe poder almacenar la siguiente información:

- El valor o dato del nodo. Este puede ser un símbolo del alfabeto en el caso de un nodo hoja o un operador en el caso de un nodo interno.
- Si es verdad o falso que el subárbol, cuya raíz es el nodo, puede ser omitido, es decir, si todo el subárbol puede anularse.
- Las posiciones de los símbolos con los que puede iniciar la subexpresión representada por el subárbol cuya raíz es el nodo o simplemente P_{mraPos} del nodo.
- Las posiciones de los símbolos con los que puede terminar la subexpresión representada por el subárbol cuya raíz es el nodo o simplemente U_{tmaPos} del nodo.
- Obviamente, una referencia para cada uno de sus dos posibles nodos hijos: hijo izquierdo e hijo derecho.