

Memoria Practica 1 AUTLEN
Salvador Martin Barcia, Blanca Mercado Molero

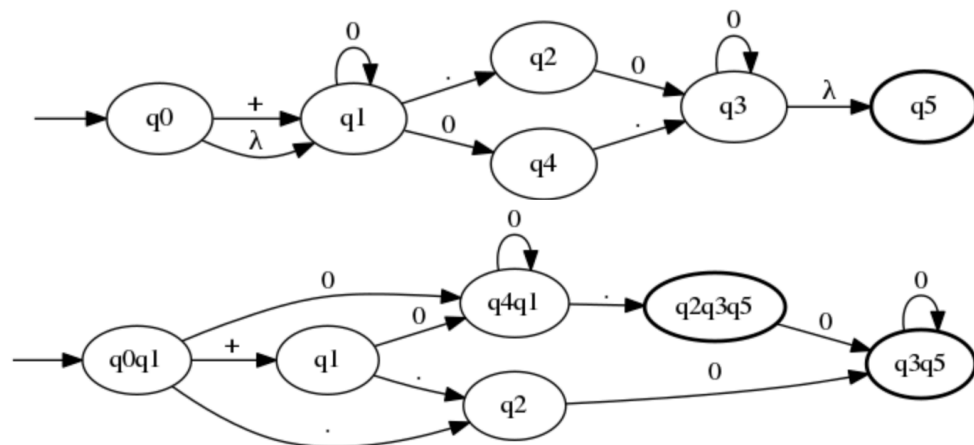
Hemos creado dos estructuras, una de estados para almacenar los estados que tendrá el nuevo AFD, la estructura contiene un "tipo" que nos dirá si el estado es inicial final normal o inicialYfinal, un array de int que contendrá los índices de los estados a los que corresponde en el AFND, un int que nos dirá cuantos estados tiene y un boolean que nos dirá si ha sido explorado ese estado. Otra de transiciones en el que tendremos el estado inicial y el final con sus respectivas longitudes de estados y el símbolo con el que se realiza la transición.

El algoritmo mete el estado inicial en una pila, se llama a una función que mira si hay transiciones lambda desde ese estado. Si las hay, se meterán en la pila y con todos los estados que haya en la pila se creará un nuevo estado. Desde ahí entramos en unos bucles anidados que irán realizando esto mismo con todos los estados que vayamos encontrando y creará nuevos estados basándose en las lambdas, o si desde un estado se puede llegar a dos o mas distintos con el mismo símbolo creando así un único nuevo estado.

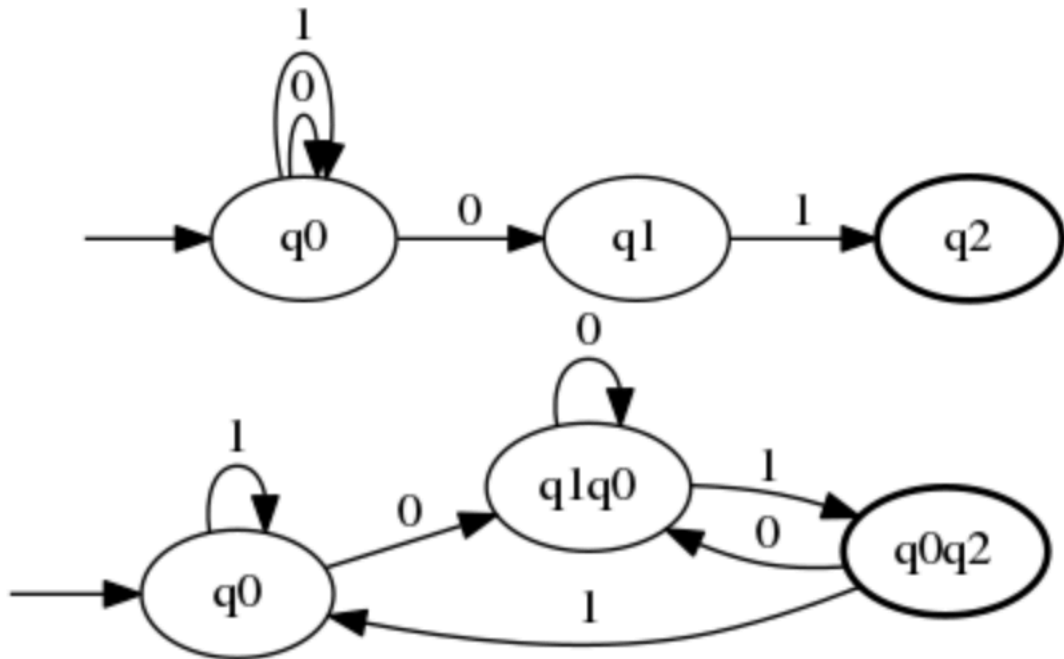
Todos estos pasaran de la pila a un array de estados y un array de transiciones. Al acabar esto pasaremos todos los símbolos, estados y transiciones al autómata.

Hemos probado el autómata con el ejemplo del enunciado y con algunos de las diapositivas, aquí dos ejemplos de la transformación.

El primero:



Y el segundo:



Codigo:

```
p_afnd= AFNDNuevo("af11", 3, 2);

AFNDInsertaSimbolo(p_afnd,"0");
AFNDInsertaSimbolo(p_afnd,"1");

AFNDInsertaEstado(p_afnd, "q0", INICIAL);
AFNDInsertaEstado(p_afnd, "q1", NORMAL);
AFNDInsertaEstado(p_afnd, "q2", FINAL);
AFNDInsertaTransicion(p_afnd, "q0", "0", "q0");
AFNDInsertaTransicion(p_afnd, "q0", "1", "q0");
AFNDInsertaTransicion(p_afnd, "q0", "0", "q1");
AFNDInsertaTransicion(p_afnd, "q1", "1", "q2");
afd = AFNDTransforma(p_afnd);
printf("Imprime afd: \n");
AFNDImprime(stdout,afd);

AFNDADot(afd);
AFNDADot(p_afnd);

AFNDElimina(afd);
AFNDElimina(p_afnd);

return 0;
```