

Memoria Práctica 1 - AUTLEN

En esta práctica nos encargamos de crear un algoritmo que convierte autómatas finitos no deterministas (AFND) en autómatas finitos deterministas (AFD).

Las principales diferencias entre ellos son que los AFND pueden tener varias transiciones con el mismo símbolo a estados diferentes y admite transiciones lambda, con cadenas vacías, lo que en los AFD no está aceptado. Nos basaremos en estas diferencias entre ambos autómatas para realizar esta conversión.

Para la realización de la práctica hemos utilizado la librería proporcionada en la práctica, añadiendo además nuevas estructuras y funciones que nos han ayudado. Éstas las hemos modulado en varios ficheros, donde hemos guardado sus diferentes estructuras, estado y estructura.

- El fichero estado.c se encarga de crear la estructura estado con un id, un nombre, el tipo de estado que es y una matriz de transiciones de tamaño, `numero_simbolos*numero_estados_base`, donde se guardará a que estados va ese estado con los diferentes símbolos. Esos estados los guardaremos mediante una codificación en la que se colocará un 1 en el estado al que va con un cierto símbolo.
- El fichero estructura.c se encarga de crear la estructura básica de nuestro conversor. En ella encontramos todos los estados existentes, tanto los obtenidos del AFND, como los nuevos generados para el AFD. También tendrá los estados finales que se vayan encontrando y el inicial, y la cantidad de estados, estados finales, y estados de base, que son esos estados que nos dan en el AFND, con los que se combinarán para generar los demás estados.

Proceso llevado a cabo:

Primero sacaremos toda la información necesaria del AFND y la introduciremos en nuestras estructuras. Esta información la sacaremos mediante las funciones de la librería proporcionada.

Una vez tenemos la información básica nos disponemos a introducir las transiciones, para cada estado vemos con qué símbolo va a qué estado e introducimos un 1 en la posición de ese estado para ese símbolo y así con todas las transiciones de todos los estados.

Debemos tener también en cuenta que si el estado al que transitas tiene una transición lambda también llegarías a ese estado con el mismo símbolo. Para tenerlo en cuenta usamos la función que devuelve en todos los estados en los que estás cuando estás en cierto estado (`AFND * AFNDCierraLTransicion (AFND * p_afnd);`).

Al querer generar un AFD sabemos que no podemos ir a varios estados con una misma transición, por lo que cada una de esa codificación de transiciones que hemos creado para cada estado, pasa a ser la codificación de un nuevo estado, que crearemos.

Antes de buscar nuevos estados comprobamos si el inicial tiene transición lambda y en ese caso se genera un nuevo estado de tipo INICIAL que será combinado.

Crearemos todos los estados de codificación nueva, cuyo nombre es la concatenación de los estados simples que lo componen y con sus respectivas transiciones, que se obtendrán

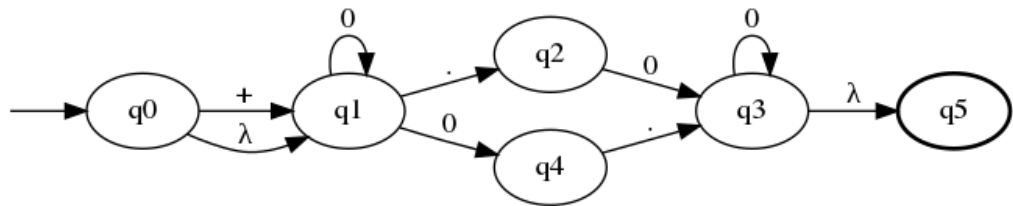
al realizar un and con las transiciones de los estados básicos que han creado este nuevo estado combinado. Esto lo realizaremos hasta que todos los estados combinados necesarios sean creados.

Tendremos finalmente todos los estados posibles a necesitar en el AFD, pero no todos, para ver cuales de ellos necesitaremos para generar el nuevo autómata realizamos una búsqueda en anchura desde el estado inicial viendo sus transiciones a los estados siguientes y estos estados son los que usaremos para generar nuestro nuevo autómata determinista, mediante las funciones de la librería ofrecida.

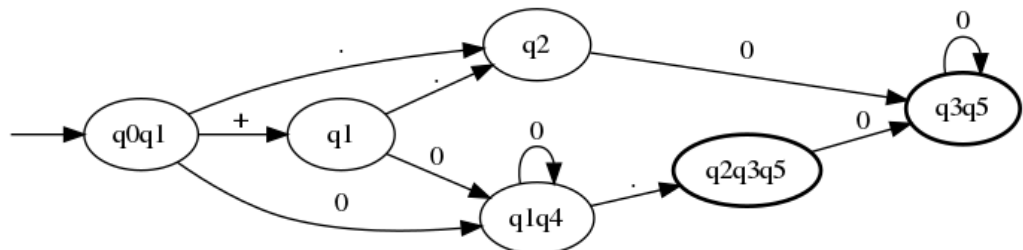
Pruebas realizadas:

Para probar esta práctica hemos creado 5 autómatas a partir del main de prueba dado, incluido este. Para crear estos autómatas, tras ejecutar el fichero.c del ejemplo en cuestión, generamos las imagenes de los autómatas con **dot afnd_p1.dot -o images/ejemplo4_AFND.png -Tpng**, siendo **afnd_p1.dot** siempre el fichero generado para el afnd y **afd_p1.dot** el del afnd. Basta con cambiar el número del ejemplo para crear el resto de las imágenes.

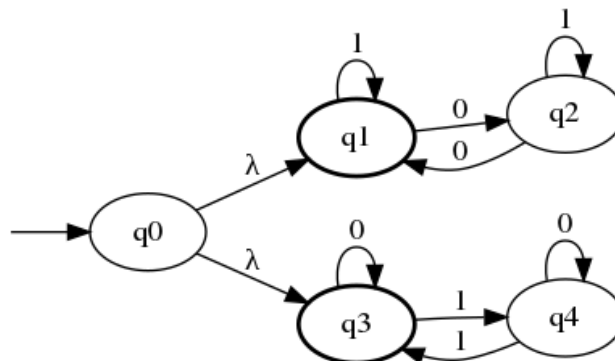
A continuación se muestran estos ejemplos de afnd , con sus respectivas traducciones afd:
Ejemplo0_AFND:



Ejemplo0_AFD:



Ejemplo1_AFND:

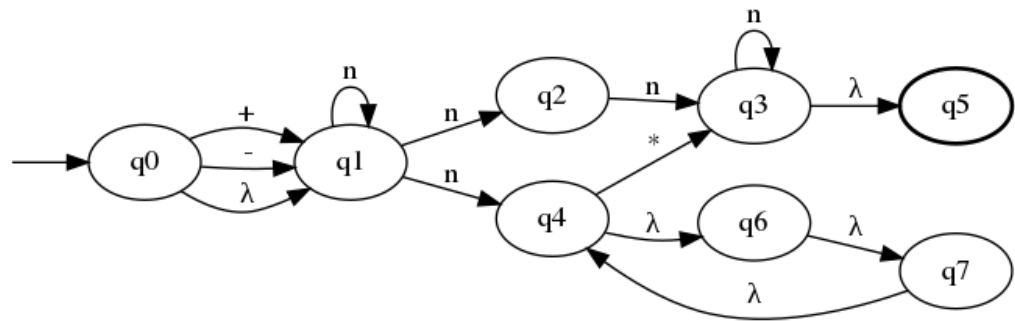


Ejemplo1_AFD:

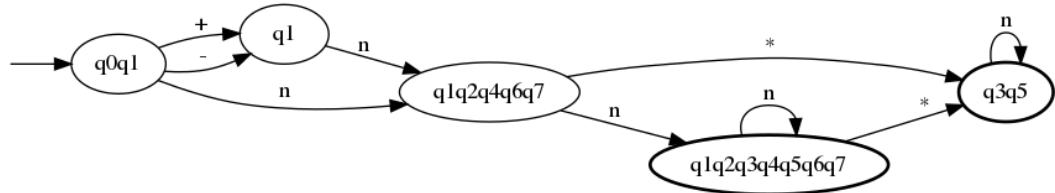
Al generar el AFD de este ejemplo nos ha salido un segmentation fault que no hemos sido capaces de arreglar. Creemos que es porque el resultado es un estado único final con transiciones de 0 y 1 a sí mismo.

Esperamos que este error no sea muy grave, ya que el resto de los autómatas se han generado correctamente.

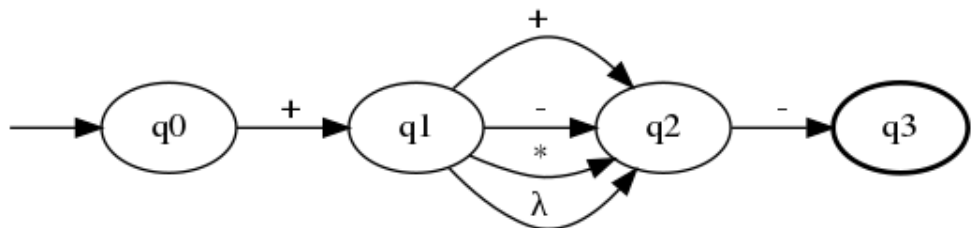
Ejemplo2_AFND:



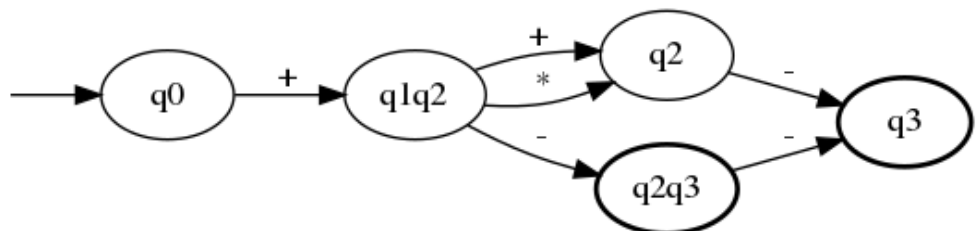
Ejemplo2_AFD:



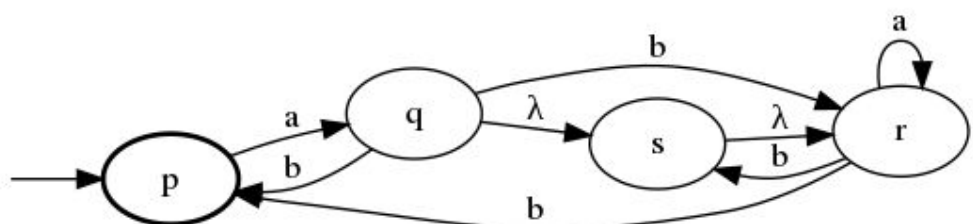
Ejemplo3_AFND:



Ejemplo3_AFD:



Ejemplo4_AFND:



Ejemplo4_AFD:

