

Nota: esta memoria resume las estructuras y el algoritmo utilizados. Para más información, consultar los comentarios del propio código.

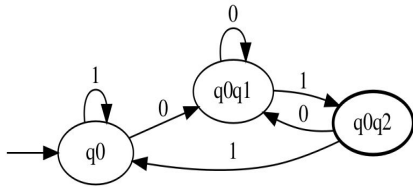
Se han utilizado 3 **estructuras**:

- **Estado:** almacena el nombre y el índice de un estado del AFND
- **EstadoCompuesto:** almacena en un solo estado de AFD un nombre compuesto por el nombre de varios estados del AFND y lo mismo con los índices de dichos estados, además del tipo del estado:
 - Final contiene algún estado final del AFND
 - Inicial si contiene el estado inicial del AFND
 - Inicial_Final si contiene ambos
 - Normal en otro caso
- **Celda:** indica el estadoCompuesto al que se transita dado un estado de partida y un símbolo, y si la celda ha sido visitada (si las transiciones para el estadoCompuesto ya han sido analizadas por el algoritmo). Esta estructura podría haberse combinado con la anterior, pero por claridad se ha dejado aparte.

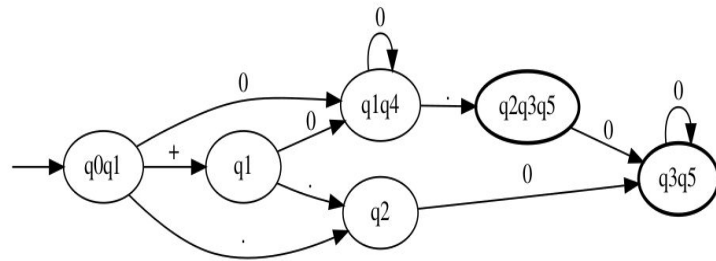
Se ha seguido el siguiente **pseudocódigo**:

- Construir matriz para transiciones del AFND
- Construir primer estado AFD
 - Partir del estado inicial de AFND
 - Buscar en matriz AFND transiciones lambda desde él
- Construir primera fila de matriz AFND
 - Para cada símbolo
 - Para cada estado que compone el de partida
 - Buscar transiciones en matriz AFND
 - Guardar nuevo EstadoCompuesto y celda en matrizAFD
 - Marcar celdas
- Mientras queden celdas sin visitar en matriz AFD
 - Estado de partida para análisis: el de la celda
 - Para cada símbolo
 - Para cada estado que compone el de partida
 - Buscar transiciones en matriz AFND
 - Buscar transiciones lambda
 - Guardar nuevo EstadoCompuesto y celda en matrizAFD
 - Marcar celdas
- Insertar símbolos, estados y transiciones en AFD
- Liberar recursos y devolver el AFD equivalente

Se han realizado **pruebas** para el autómata del enunciado y para el de la diapositiva 40 (se deja comentado en el main por si desea probarse).



Resultado para diapositiva 40



Resultado para enunciado