Memoria práctica 1 AUTLEN

Borja Pérez Bernardos y Alejandro Pascual Pozo

1. Estructuras creadas

1.1 int_list

Implementación estándar de un array list de enteros.

```
struct _IntList {
    int *values;
    int size;
    int max_size;
};
```

1.2 state_list

Almacena tres listas: una de listas de subestados, una de estados objetivos de transición y otra de símbolos de transición. Estos tres tipos de listas utilizan la estructura int_list.

Cada estado tiene asociado un índice i. En substates[i] encontramos los estados no deterministas que le corresponden y en transitions_targets[i] y transitions_symbols[i] sus transiciones.

```
struct _StateList {
    IntList **substates;
    IntList **transitions_targets;
    IntList **transitions_symbols;
    int size;
    int max_size;
};
```

2. Pseudocódigo del algoritmo implementado

```
estados.añadir(clausuraLambda(afnd.estadoInicial))
for estado in estados
    for símbolo in afnd.símbolos
        siguienteEstado = transiciónConSímbolo(afnd, estado, símbolo)
        if siguienteEstado not in estados
            estados.añadir(siguienteEstado)
        estado.añadirTransición(siguienteEstado)
return new AFND(estados)
```

3. Pruebas realizadas

Se han realizado pruebas con diferentes autómatas, comprobando que la salida es la misma para la transformación determinista que para la original.

Se adjuntan los ficheros necesarios para la compilación y ejecución de una serie de tests automatizados. Para ello se deberán ejecutar los siguientes comandos:

make

make run_tests

La salida debería ser comprobada de manera automática, indicándose al final que los tests se han ejecutado correctamente. Si se desea, es posible revisar los resultados de cada una de las pruebas, ya que también son impresos en la consola.

4. Correcto uso de la memoria

Hemos comprobado que no se produzcan pérdidas de memoria al utilizar nuestro programa. Se puede comprobar ejecutando los comandos:

make

make debug

O para el caso de los tests:

make

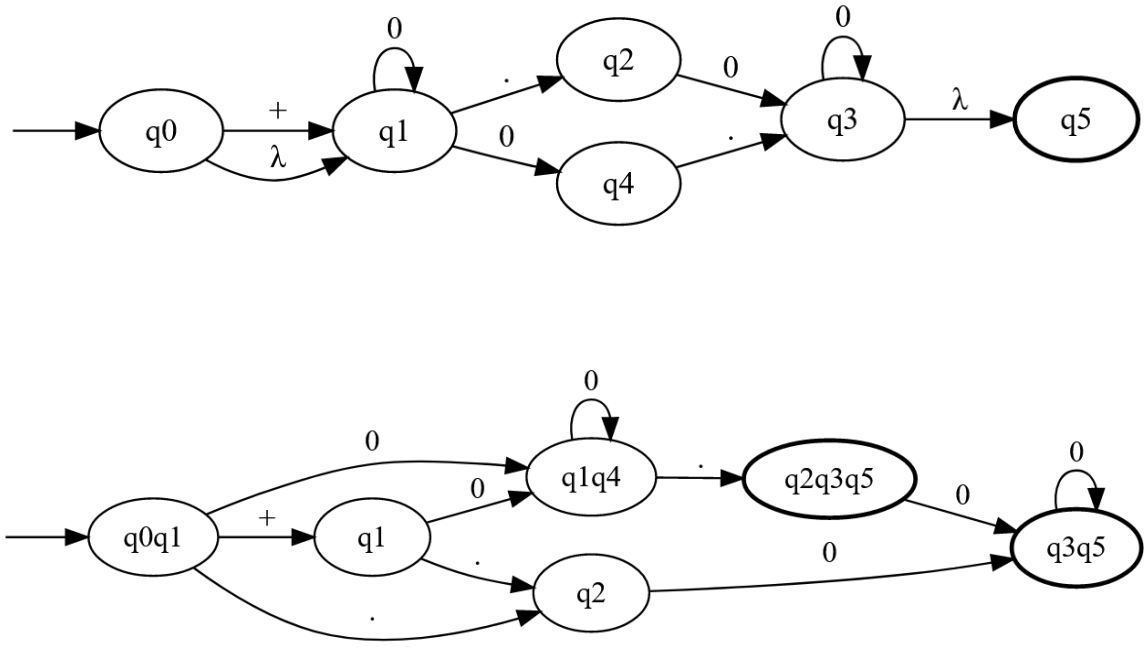
make debug_tests

Salida de Valgrind para los tests:

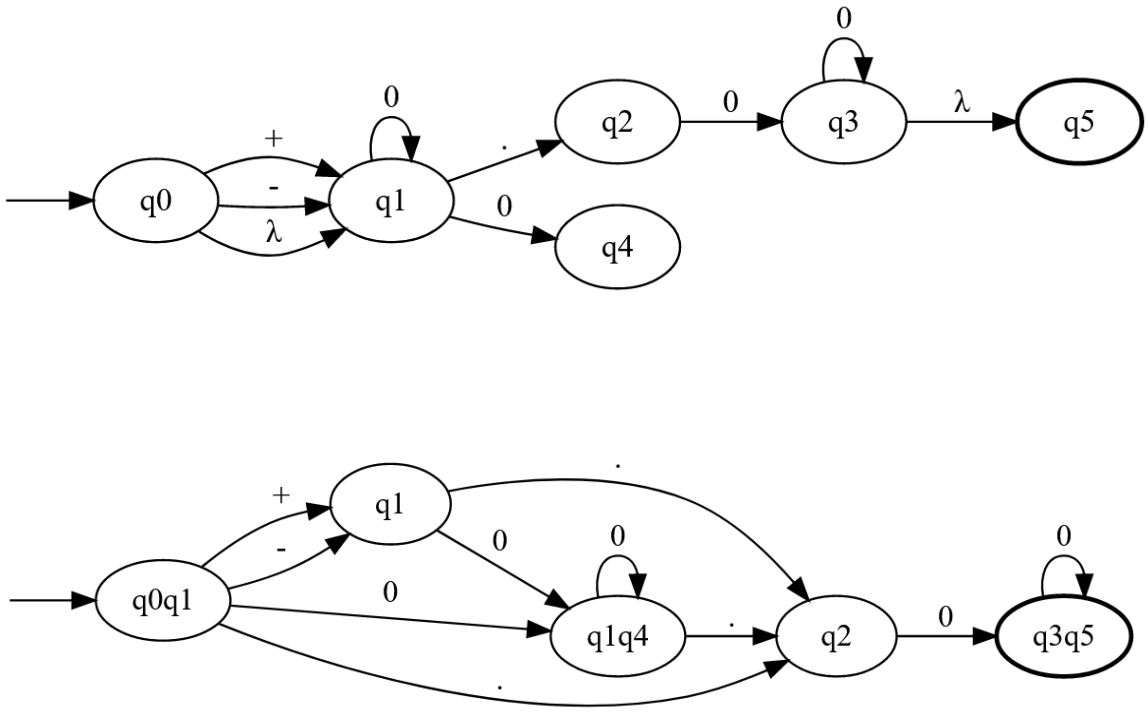
```
Todos los tests se han ejecutado correctamente.
==2912== HEAP SUMMARY:
==2912==     in use at exit: 0 bytes in 0 blocks
==2912==   total heap usage: 2,556 allocs, 2,556 frees, 777,815 bytes
allocated
==2912==
==2912== All heap blocks were freed -- no leaks are possible
==2912==
==2912== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
==2912== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

5. Autómatas generados en las pruebas

5.1 Autómata del enunciado



5.2 Autómata de las diapositivas



5.3 Autómata que reconoce el lenguaje