TDT4205 - Exercise 1

Einar Johan Trøan Sømåen (somaen)

### Task 1:

GCC 4.2.4

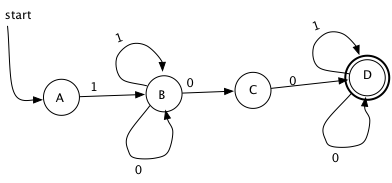
Flex 2.5.34

Bison 2.3

### Task 2:

A lexical analyzer parses the input into valid tokens (if it consists of valid lexema), while an acceptor merely determines whether the input is valid lexema for our language.

### Task 3:

1(0|1)\*00(0|1)\*

I have named my states A-D to avoid confusion with the numbers used for transition-values.

3.2: This language describes any binary number that starts with 1, and then has any number of 0s or 1s followed by a pair of 00s and then any number of 0s or 1s

Valid examples:

100

10101010100111

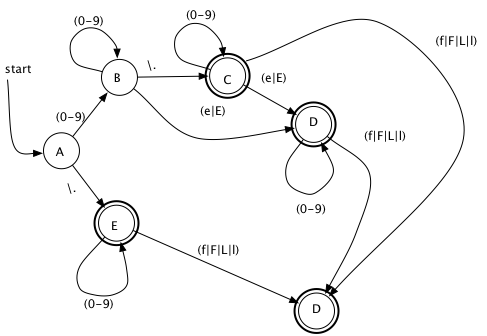
111100

11100110011010

3.3

|  |  |  |
| --- | --- | --- |
| STATE | 0 | 1 |
| A | Ø | {B} |
| B | {B,C} | {B} |
| C | {D} | Ø |
| D | {D} | {D} |

### Task 4:



### Task 5:

See attached source-code.

Note: The string regex was a nut to crack, I ended up searching on the net, and finding a solution here: <http://www.lysator.liu.se/c/ANSI-C-grammar-l.html>