Hello Lex

The FSL file be found in ./lexer/lexer.fsl

Question 1

We have all things for ex1 in the folder called lexer

Read the specification hello.fsl !

Done

What are the regular expressions involved ?

Digits

Which semantic values are they associated with ?

Character conversion

Question 2

Generate the lexer out of the specification using a command prompt.

Done

Which additional file is generated during the process?

We get a .fsi and a .fs file.

How many states are there by the automaton of the lexer?

9 states

Question 3

Program can be found as lexer.exe if you want to test that it works.

Question 4

Done

Question 5

```
  | ['-']?['0'-'9']+('.'['0'-'9']*)?   { LexBuffer<char>.LexemeString lexbuf }
  // Match floats
```

BCD

Exercise BCD 2.1

- (a)

```
\b42\b
```

- (b)

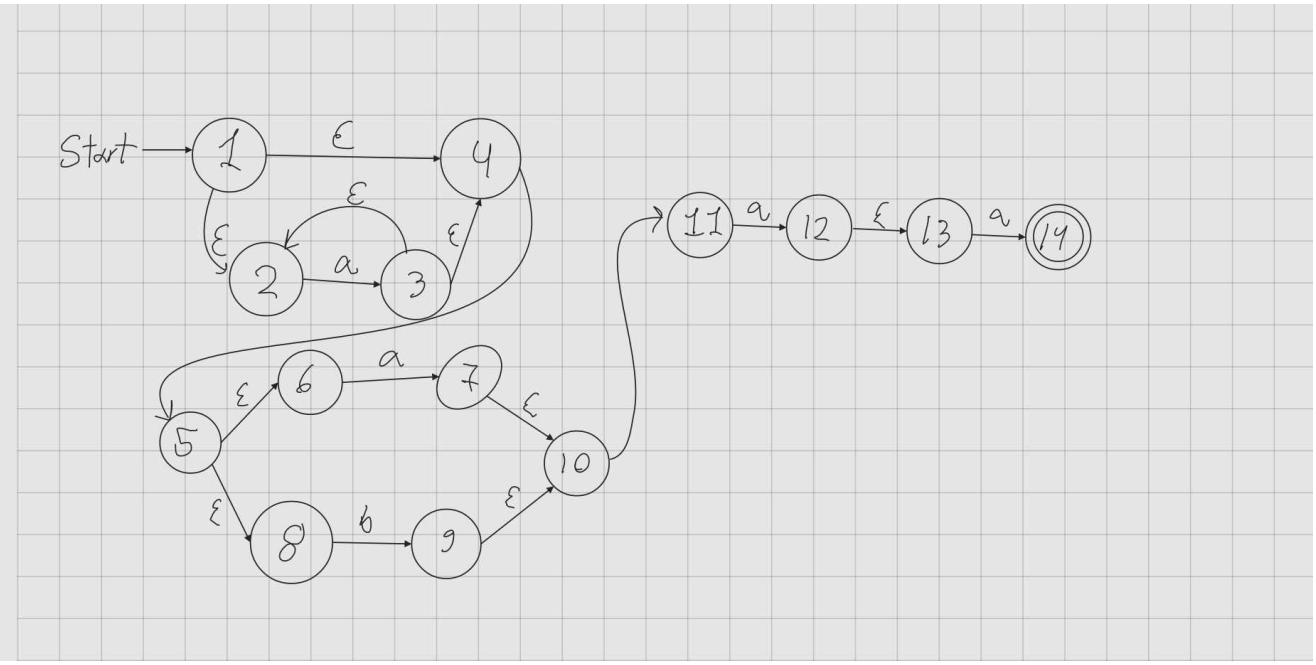
```
^(?!.*\b42\b)\d+$
```

- (c)

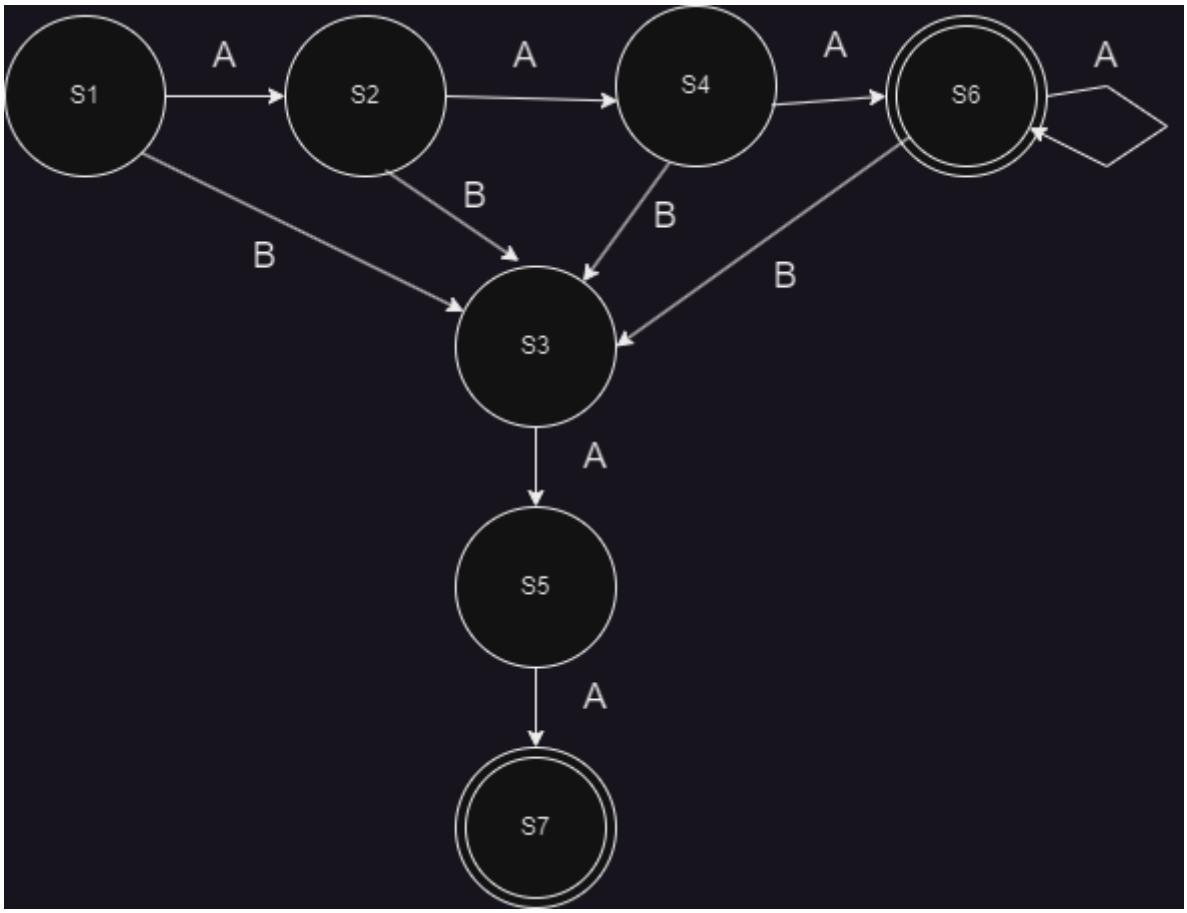
```
^([4-9][3-9]+)|\d{3,}$
```

Exercise BCD 2.2

NFA



DFA



PLC

Exercise 2.4 & 2.5

```
(* Ex 2.4 - assemble to integers *)
(* SCST = 0, SVAR = 1, SADD = 2, SSUB = 3, SMUL = 4, SPOP = 5, SSWAP = 6; *)
let sinstrToInt (list: sinstr list) =
    let rec aux sl il =
        match sl with
        | [] -> List.rev il
        | SCstI x :: xs -> aux xs ( x :: 0 :: il)
        | SVar x :: xs -> aux xs (x :: 1 :: il)
        | SAdd :: xs -> aux xs (2 :: il)
        | SSub :: xs -> aux xs (3 :: il)
        | SMul :: xs -> aux xs (4 :: il)
        | SPop :: xs -> aux xs (5 :: il)
        | SSwap :: xs -> aux xs (6 :: il)
    aux list []

let assemble instrs = sinstrToInt instrs

let list = scomp e1 [];;

(* Output the integers in list inss to the text file called fname: *)

let intsToFile (inss : int list) (fname : string) =
    let text = String.concat " " (List.map string inss)
```

```
    System.IO.File.WriteAllText(fname, text);;

intsToFile (assemble list)"is1.txt"
```

Exercise 3.2

A

```
^a?(b*(ba)*)*$|^(ab)*a?b*$
```

B