

# Lab 1

Exercise 3.3. Write regular expressions that define the strings recognized by the FAs in Figure 3.33 on page 107.

Answer.

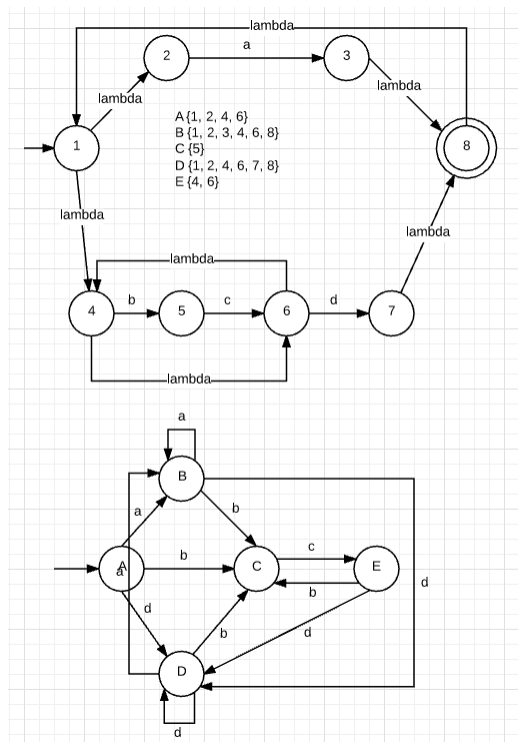
- a)  $([a|b][\ ])$

Exercise 3.4. Write DFAs that recognize the tokens defined by the following regular expressions:

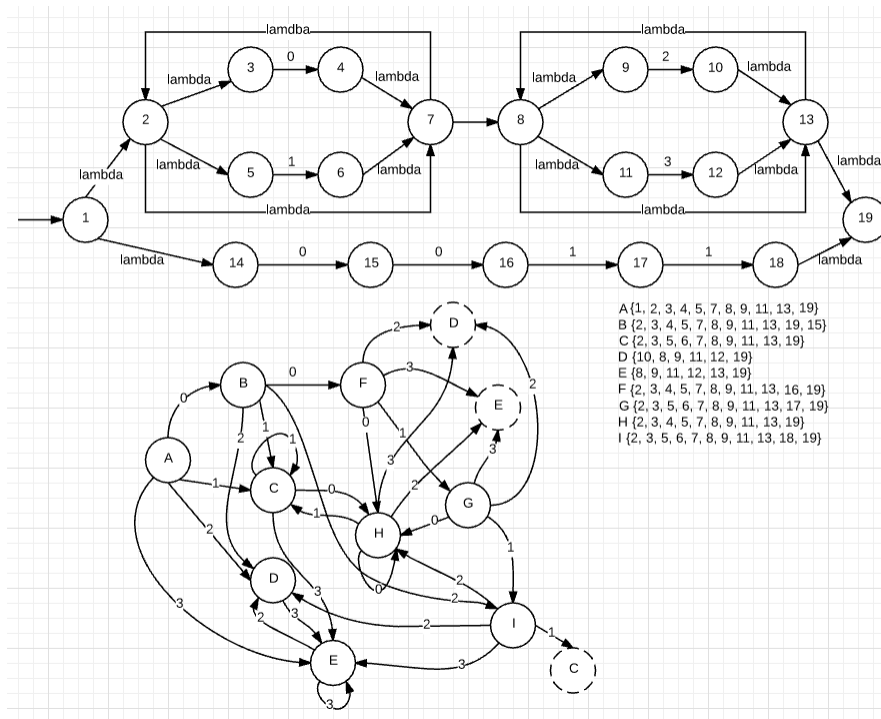
- a)  $(a|(bc)^*d)^+$
- b)  $((0|1)^*(2|3)^+)|0011$
- c)  $(aNot(a))^*aaa$

Solution.

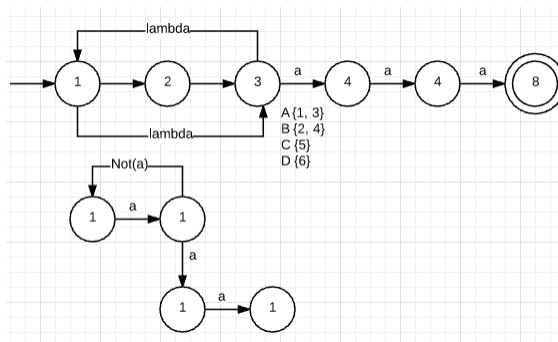
3.4a



3.4b



3.4c



Exercise 3.5. Write a regular expression that defines a C-like, fixed-decimal literal with no superfluous leading or trailing zeros. That is 0.0, 123.01, and 123005.0 are **legal**, but 00.0.001.000, and 002345.1000 are **illegal**.

Answer.

- Let DNOTZ be the set of digits from 1 to 9.
- Let D be the set of digits from 0 to 9.

Define  $(0|(DNOTZ\ D^*)\cdot(0|D^*\ DNOTZ))$ .

Define  $^(0|[1-9][0-9]^+)\cdot([0][0-9]^*[1-9]\$)$ . (Real Regex)

Exercise 3.3.4. Most languages are **case sensitive**, so keywords can be written only one way, and the regular expressions describing their lexeme is very simple. However, some languages, like SQL, are case insensitive, so a keyword can be written either in lowercase or in uppercase, or in any mixture of cases. Thus, the SQL keyword *SELECT* can also be written *select*, *Select*, or *sELeCT*, for instance. Show how to write a regular expression for a keyword in a case insensitive language. Illustrate the idea by writing the expression for "select" in SQL.

*Answer.*

$select \rightarrow [Ss][Ee][Ll][Ee][Cc][Tt]$

---