

Lab Two

Emily Doran

Emily.Doran1@Marist.edu

February 27, 2021

CRAFTING A COMPILER

3.3 (REGULAR EXPRESSIONS)

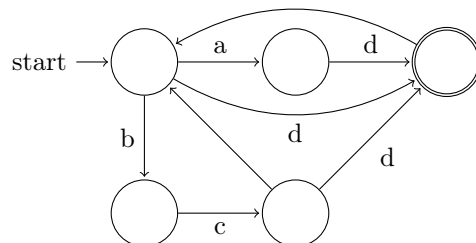
Write regular expressions that define the strings recognized by the FAs in Figure 3.33 on page 107 (Appendix).

- (1) $(ab^*a)|(ba^*b)$
- (2) $a|(a(bc|c)da)^+$
- (3) $\lambda|(ab^*c)$

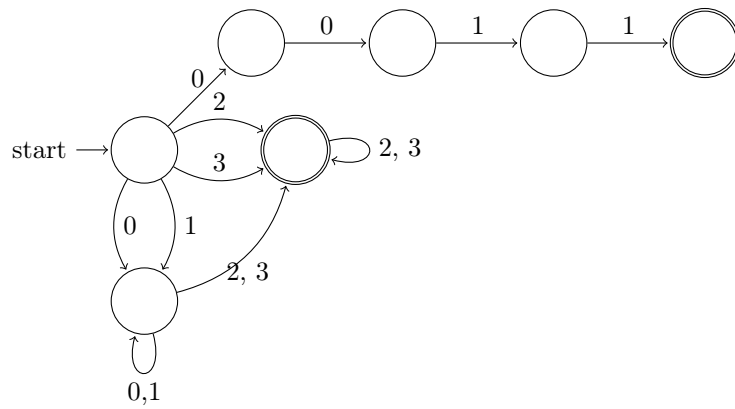
3.4 (DFAS)

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

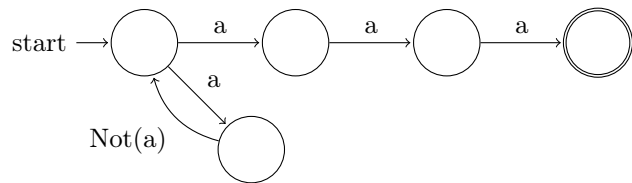
- (a) $(a|(bc)^*d)^+$



- (b) $((0|1)^*(2|3)^+)|0011$



(c) $(a\text{Not}(a))^*aaa$



DRAGON

3.3.4 (CASE INSENSITIVITY IN REGULAR EXPRESSIONS)

Most languages are *case sensitive*, so keywords can be written only one way, and the regular expressions describing their lexemes are 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.

To write a regular expression in a case-insensitive language, you have to include both the uppercase and lowercase letters for each letter in the word. For example, the regular expression for select in a case-insensitive language like SQL would look like: $[sS][eE][lL][eE][cC][tT]$

APPENDIX

FIGURE 3.3

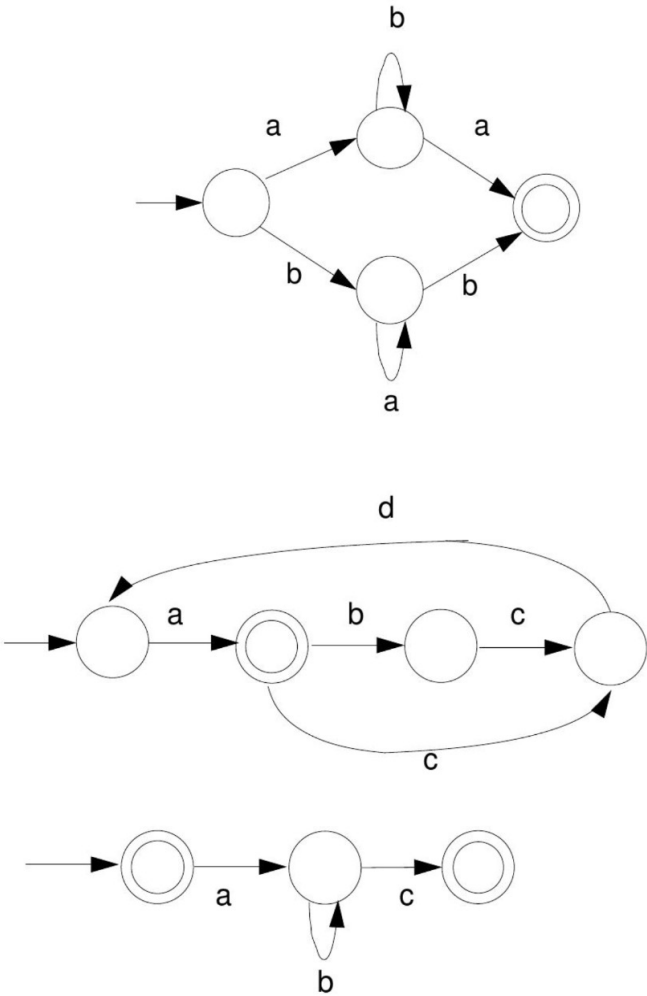


Figure 3.33: FA for Exercise 3.