Lab Two - More Token Making (Now With Finite Automata)

Murray Coueslant

murray.coueslant1@marist.edu

September 20, 2019

1 The Dragon Book

1.1 Problem 3.3.4

1.1.1 Problem

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.

1.1.2 Solution

The way to express case insensitivity using pure regular expressions is to write an expression which includes each letter in the pattern in each case. So for the keyword "SELECT", the case insensitive expression would be:

$$[sS][eE][lL][eE][cC][tT]$$

Obviously, this would become grossly inefficient as the pattern grows larger or more complex. And so nowadays most regular expression engines allow the user to use a flag which instructs the engine to ignore case in the expression.

2 Crafting A Compiler

2.1 Problem 3.3

2.1.1 Problem

Write regular expressions that define the strings recognized by the FAs in Figure 2.1.

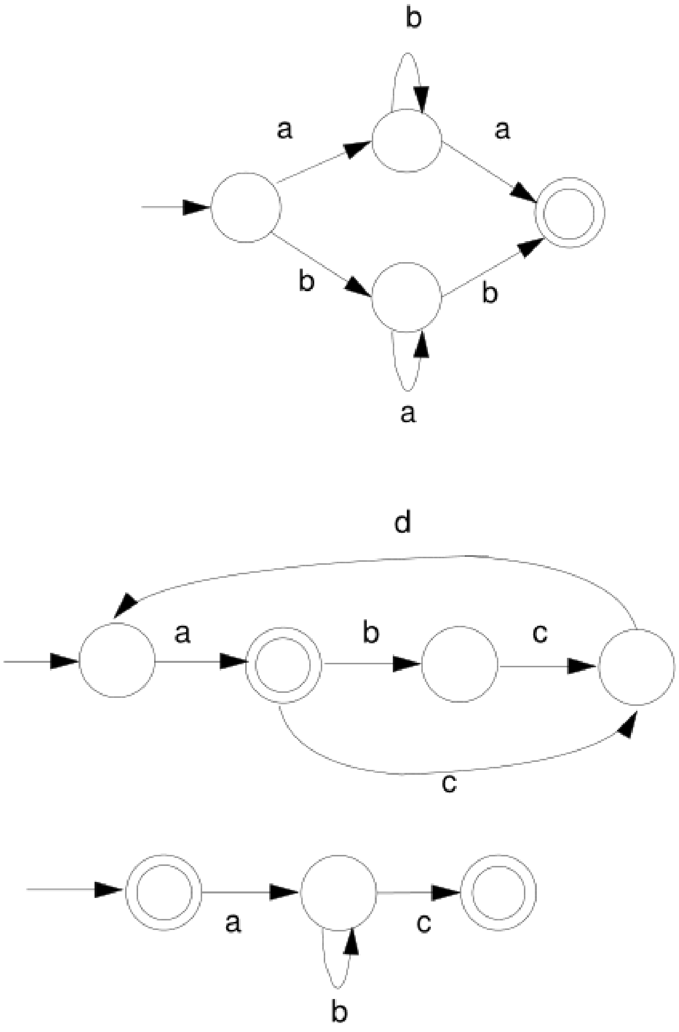

Figure 2.1: FAs for Problem 3.3.

2.1.2 Solution

The regular expressions for the above FAs are as follows:

- $(ab^*a|ba^*b)$
- $(a(bcda|cda)^*)$

- $\epsilon | ab^* c$

2.2 Problem 3.4

2.2.1 Problem

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

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

2.2.2 Solution

See the folder named DFAs next to this PDF to view images of the DFAs for this question.