Q1:

a). I would implement IDE in Scanner and Parser, because although scanner can tell the difference between strings, comments, keywords, etc., it cannot recognize regular expressions. On the contrary, a parser gives syntactic analysis on tokens. Together, these two phases can distinguish different expressions and also recognize them as well. Therefore, if IDE is implemented in the parser of a compiler, it can provide detailed information about the code, and hence make subtle distinction in colouring.

b). I would implement the parser phase, because “the principal purpose of the scanner is to simplify the task of the parser” (Scott, L. M., 2008, p27), so without scanner, parser can still recognize regular expressions and generate parse tree, but with more effort

Q2:

a).

[a-zA-Z]*(a|e|i|o|u)[a-zA-Z]*(a|e|i|o|u)[a-zA-Z]*

b).

[0-9]*(00|25|50|75)

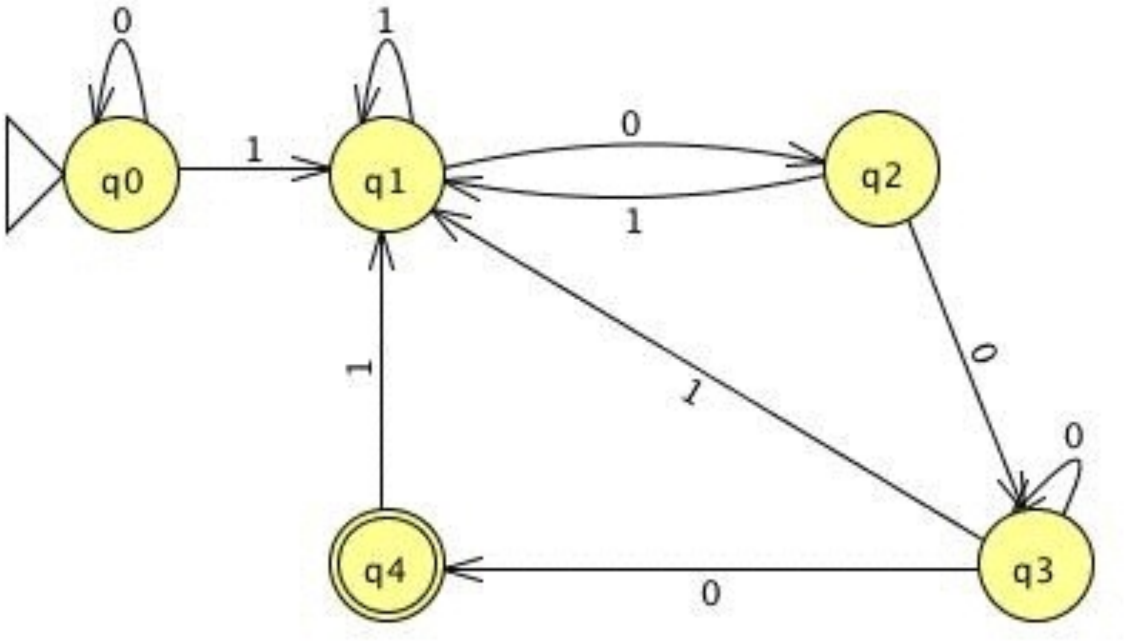
c).

(1|0|11)00*(100*|10*100*)*

Q3:

a). Finite, regular language

DFA:

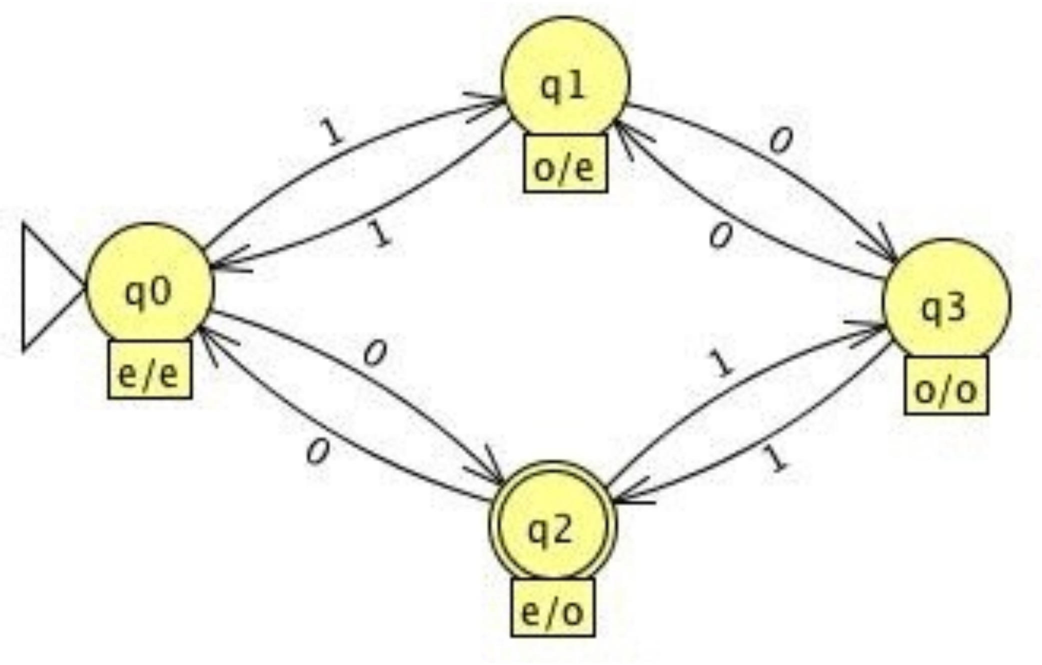


b). Infinite, not regular language

Reason: If it is regular, then we can find a finite automata that describes the language. But in this case, there is no way to use a finite automata to keep track of the numbers of 0s and 1s and compare them, which means it requires an infinite automata to recognize the language.

c). Infinite, regular language

DFA:



d). Infinite, regular language

References

Scott, L. M. (2008). *Programming Language Pragmatics*. Burlington, MA, U.S.A: Morgan Kaufmann Publishers is an imprint of Elsevier