Informatics 1 - Computation & Logic: Tutorial 6

Computation: Non-Deterministic FSMs and Regular Expressions

Week 8: 7 - 11 November 2016

Please attempt the entire worksheet in advance of the tutorial, and bring with you all work, including (if a computer is involved) printouts of code and test results. Tutorials cannot function properly unless you do the work in advance.

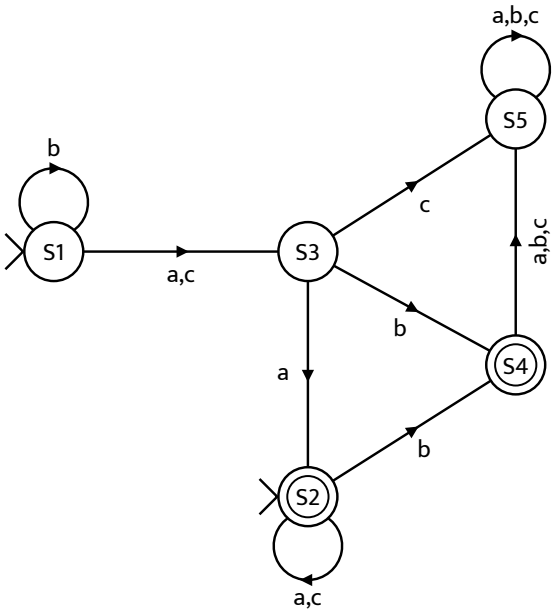
You may work with others, but you must understand the work; you can't phone a friend during the exam.

Assessment is formative, meaning that marks from coursework do not contribute to the final mark. But coursework is not optional. If you do not do the coursework you are unlikely to pass the exams.

Attendance at tutorials is **obligatory**; please let your tutor know if you cannot attend.

You may find it useful to refer to the FSM Workbench question set which accompanies this tutorial at homepages.inf.ed.ac.uk/s1020995/tutorial6.

1. Consider the finite state machine in the diagram below.

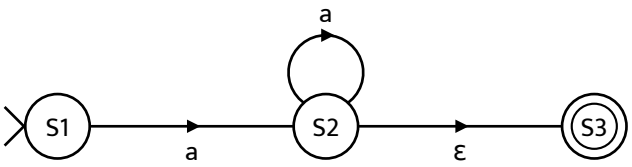


   (a) For each input sequence in the table below, record whether it is accepted by the FSM.

Input	Is Accepted?
⟨ ⟩	
b	
aa	
ba	
abaab	
acaca	
aaab	
bbbcb	
cacba	

   (b) Is the FSM deterministic? Justify your answer.

2. This NFA over the alphabet {a} uses an ε transition.

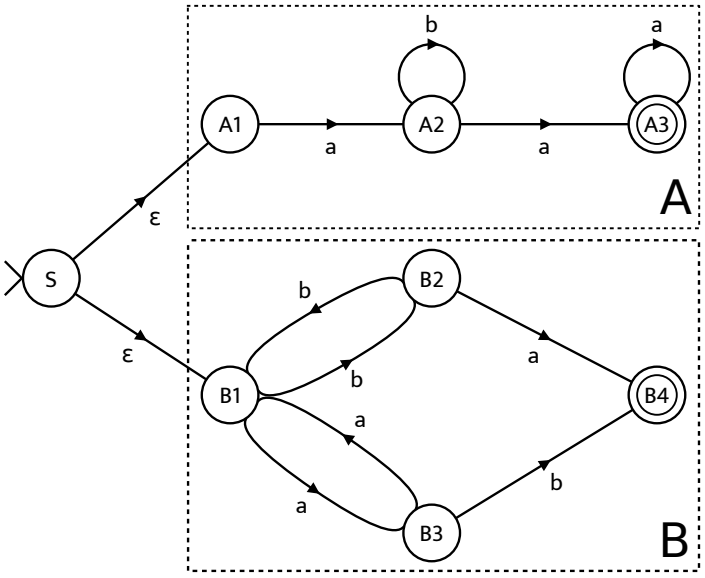


   (a) Describe the language accepted by this machine in words.

   (b) Describe the language accepted by this machine using a regular expression.

   (c) Design a deterministic machine that accepts the same language as this machine.

3. ε-transitions provide a simple way of combining FSMs. The machine below has been composed from two machines A and B, which had initial states A1 and B1.



   (a) Considering machines A and B separately, give a regular expression which describes the language they accept.

   (b) Considering the whole machine, give a regular expression which describes the language the machine accepts.

(c) L_A and L_B are the sets of inputs accepted by machines A and B. Give an expression relating L_A and L_B to L, where L is the set of input accepted by the whole machine.

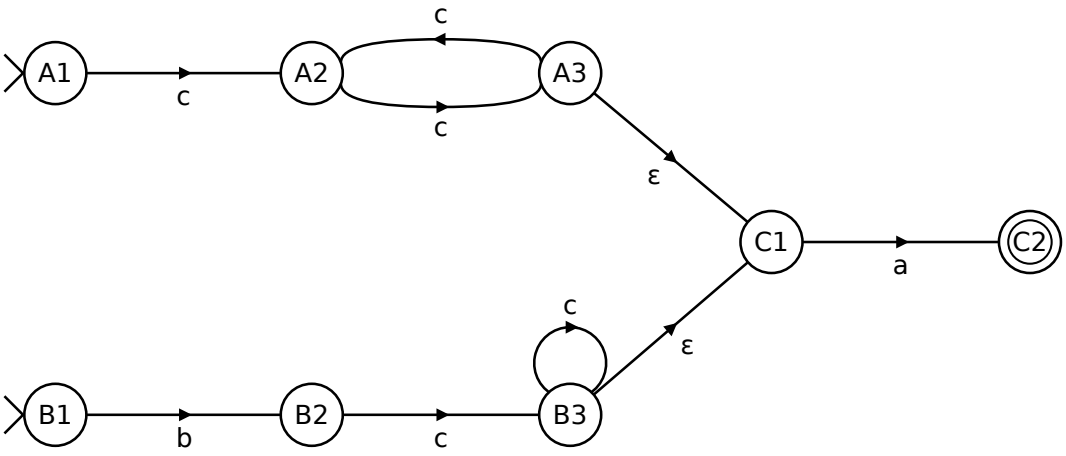
4. Consider the regular expression $\texttt{ab}(\texttt{a}|\texttt{b})^*$

   (a) Describe in words the language that the expression matches. Include two examples of strings that are matched.

   (b) Design a finite state machine that accepts that language.

   (c) Building on your answer to (b), design a finite state machine that accepts $\texttt{ab}(\texttt{a}|\texttt{b})^*\texttt{bb}^*(\texttt{aa})^*$.

5. Consider this NFA over the alphabet $\{\texttt{a}, \texttt{b}, \texttt{c}\}$.

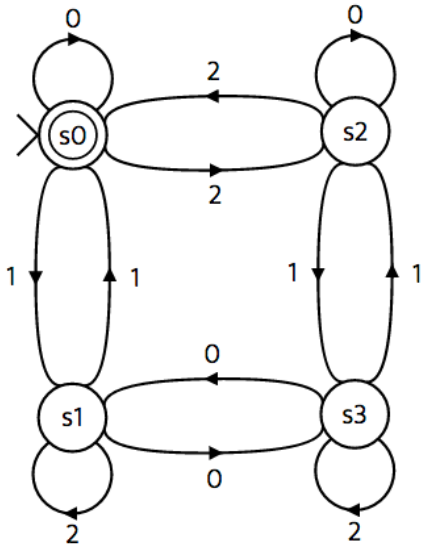


   (a) Describe, both in words and with a regular expression, the language accepted by this machine. Hint: think about the sequences that end in A3 and B3.

   (b) Design a DFA that accepts the same language.

   (c) Are there any NFAs that cannot be converted into an equivalent DFA?

6. Consider this DFA over the alphabet $\{0, 1, 2\}$. It should be familiar.



(a) Describe, in words, the language accepted by this machine. Hint: Your description in words should refer to ternary numbers.

(b) Replace each transition labelled 0 by a transition labelled ε, between the same two states. The resulting automaton is not a DFA. (Why not?)

   i. Construct an equivalent DFA.

   ii. Describe, both in words and with a regular expression, the language accepted by this machine.

(c) Next, replace each transition (of the original machine) labelled 1 by a transition labelled ε, between the same two states.

   i. Again, construct an equivalent DFA, and, ii, describe the language it accepts.

(d) Repeat the exercise replacing each transition (of the original machine) labelled 2 by a transition labelled ε, between the same two states.

   i. Construct an equivalent DFA, and, ii, describe the language it accepts

(e) BONUS QUESTION: Give a regular expression that describes the language accepted by the original machine. Test your answer using the grep utility.

This bonus question goes somewhat beyond the call of duty. Feel free not to attempt it. That said, by the end of week 7 you should have all the tools required to complete it. If you do choose to try it, I suggest you use cut and paste in some suitable editor to make, and keep track of the algebraic substitutions that are required.