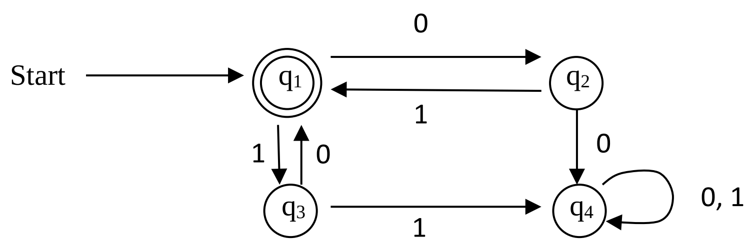
MA/CSSE 474   Homework #7   36 points   **Updated for Summer, 2018**

1. (t-9) 6.7a Show how you use the construction from the textbook.
2. 6.8
3. (t-18) Consider the DFSM M below. Use the algorithm from class to find a regular expression r such that L(R) = L(M). You should calculate all of the r_{ijk} for k=0 and k=1. For k>1, you are only required to calculate as many of the r_{ijk} as needed to do the recursive steps that the algorithm actually needs to get the answer. Be explicit about the ones that you do calculate. [This link is primarily for summer students for which there is no "in-class", but it may be helpful to winter term students as well. The proof of the "in-class" algorithm and a complete example are given in the proof of Theorem 3.4 on the bottom of p33 and on pages 34-35 from this document, taken from "introduction to *Automata Theory, Languages, and Computation* by Hopcroft and Ullman (Addison-Wesley, 1979).]



4. 6.13d Additional practice with RE→FSM
5. 6.15. Do this one if you need extra practice with the DFSM→RE algorithm. There is an error in the diagram in the book, that makes it not be a DFSM. The b-transition from q_1 to q_3 should not be there. **Remove that transition before doing the problem.**
6. (t-9) 6.18 Regular expression based on a relation and its closure. Transitive and reflexive closures are introduced in Section A.5 Closures under various operations are also mentioned on pages 17, 57, 72.
7. 6.20 True-False problems

Some past questions and answers from Piazza:

#3 Simplifying regular expressions

Any suggestions on how to simplify the regular expressions we get from using the formula? I know you can "factor" things out, but I still have very complicated expressions.

Instructor's answer:

1. factor if multiple "union terms" have common factors.
2. if r is a regular expression, (epsilon union r) * is equivalent to r*.
3. Look for anything else for which you can tell what language it is and reduce it that way.

#3 Getting r values

When we are calculating the r values to get the final R that we need, can we just give the language if we can determine it from the machine? For example, if i need to calculate r_{112} do I need to use the formula like r U r r* r kind of thing, or can I just look at the machine since it's a simple enough r value?

Instructor's answer:

I made it a very simple machine to decrease the amount of work involved in doing this problem. But the purpose of the problem is for you to show that you understand the recursive algorithm. So you must use the recursive formula explicitly when you calculate some of the r_{ijk}.

#6: Reflexive, transitive closure in 6.18

I am not sure how any element of the closure of the alphabet can be reflexive, e.g how can x E ∑* have x = xb?

Instructor's answer:

R itself is not reflexive, so it cannot contain (a, a) for any a. But to get the reflexive closure of R, we ***add*** whatever pairs are necessary in order to make the relation be reflexive.

BTW, you have to read this problem very carefully. Learning to read complex statements about sets, relations and quantifiers is one of the main points of this problem.