MA/CSSE 474   Homework #4 (36 points)

Don't forget the optional survey for 5 points extra-credit.

1. (t-6) 5.9(a) *ndfsmtodfsm*

2. 5.9(c) *ndfsmtodfsm*

3. (t-6) 5.10 *DFSM for* $\neg L(M)$

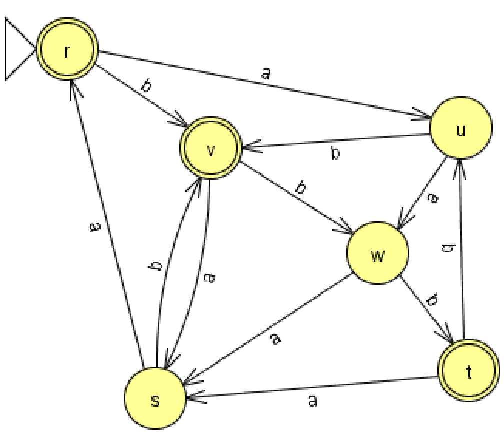
4. 5.11 *Equivalence classes of* $\approx_L$

5. (t-6) 5.11a [be sure to notice instructions (i) and (ii)]

6. (t -6) 5.11d [be sure to notice instructions (i) and (ii)]

7. (t-6) 5.12 *minimize a DFSM* Show the details of your work.
8. (t-6) not in book The DFSM pictured below is minimal, and r is its start state. Construct the canonical form representation of this DFSM as described in Section 5.8 of the Rich textbook. Each vertex in the graph will be given a new name, one of {q0, q1, …, q5}. Complete the following table to indicate which new name will be given to which state. Assume that *a* comes before *b*.

Old state name	New state name
r	
s	
t	
u	
v	
w	



Example: http://www.rose-hulman.edu/class/cs/csse474/202040/Resources/DFSM-Canonical-Form/DFSM-canonical-form.pdf

Some past questions and answers from Piazza:

In general, not for a specific problem.

When is epsilon/empty string needed for NDFSMs?

I've noticed that we sometimes use epsilon/empty string in NDFSMs to go from state to state and sometimes not.
How do we know when we need the empty string and when we don't?

Answer: It is never necessary to use include epsilon transitions when we create a NDFSM .

In fact we can easily prove that given a NDFSM that includes epsilon transitions, we can make an equivalent NDFSA that does not include any epsilon transitions.

So epsilon transitions are simply a matter of convenience, as is non-determinism itself. Use them when they make it easier to create a machine that accepts a particular language.