Name____________________

CS503
Homework #6
Solutions

#1. M is the Turing machine:

δ	B	a	b	c
q_0	q_1,B,R			
q_1	q_2,B,L	q_1,a,R	q_1,c,R	q_1,c,R
q_2		q_2,c,L		q_2,b,L

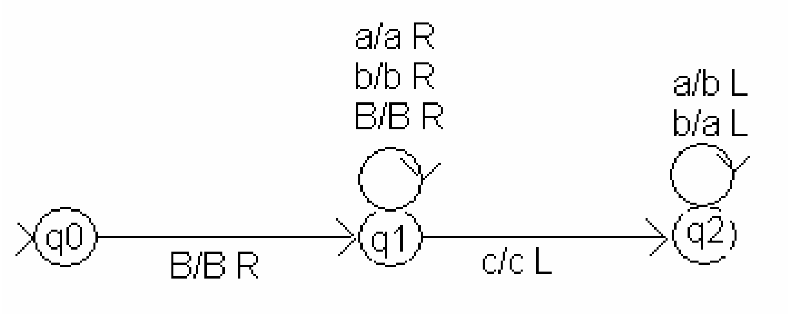
a) Trace the computation of *a a b c a*

q_0BabcaB
→ B$q_1$abcabB
→ Ba$q_1$bcabB
→ Ba$q_1$bcabB
→ Bab$q_1$cabB
→ Ba$q_2$bcabB
→ B$q_2$aacabB
→ q_2BbacabB
The T.M. halts.

b) Trace the computation of *b c b c*

q_0BababB
→ B$q_1$ababB
→ Ba$q_1$babB
→ Bab$q_1$abB
→ Babaq_1bB
→ Bababq_1B
→ BababBq_1B
The T.M. will never halt. It will keep moving right, replacing B's with B's and staying in state q_1.

c) Draw the graph for M



d) What does M do?

M replaces all of the a's with b's and all of the b's with a's before the first c in a string with a c in it, and halts. If there is no "c", it computes forever.

#2. a) Construct a Turing machine with alphabet {0,1} to compute f(n) = 2n. Represent numbers in unary notation; that is, 0 is represented by a *1* on the tape, 1 by *11*, 2 by *111*. (So if n = 3, you would be left with seven 1's on the tape etc.). Have your Turing machine halt in the configuration: q_f B f(n) B. Show a computation for f(3).

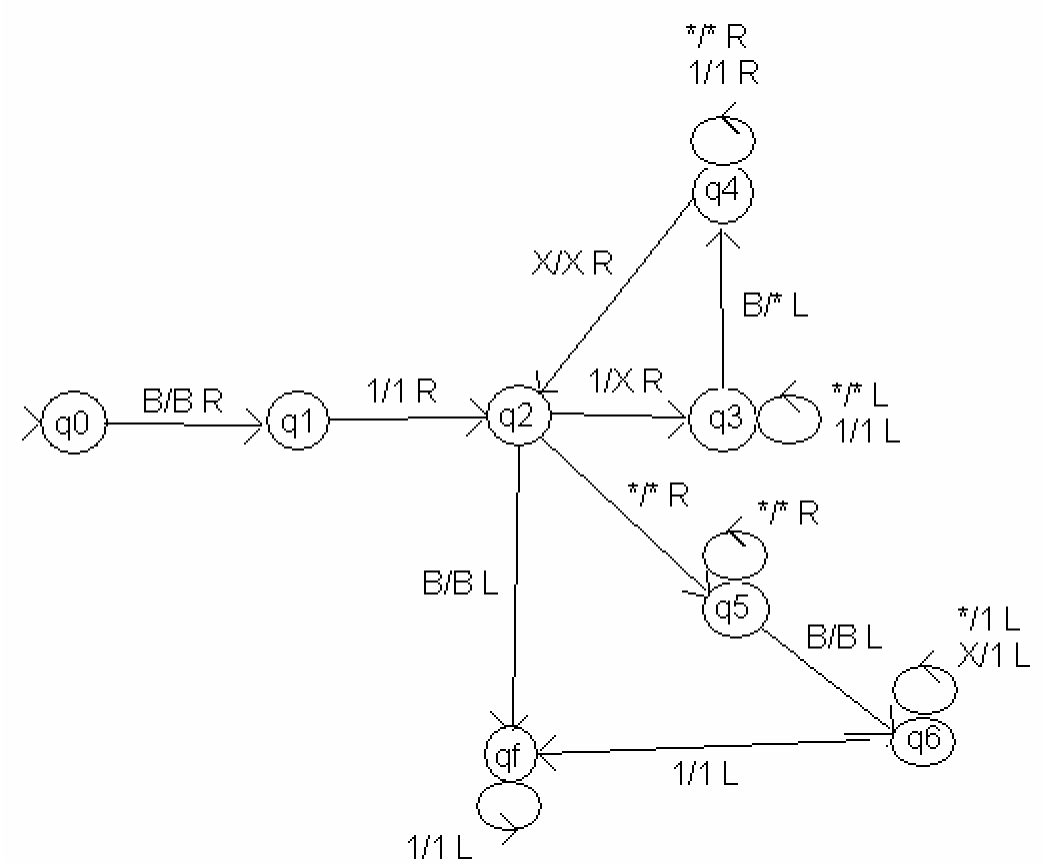
The biggest problem here was that many of your TMs didn't work for n = 0. Otherwise fine.

See http://www/cs.wpi.edu/~kal/courses/cs503/module8/Turing_Machine.doc

b) Construct a Turing machine with alphabet {0,1} to compute f(n) = n *monus* m defined by:

$$n \text{ monus } m = \begin{cases} n - m & \text{if } n \geq m \\ 0 & \text{otherwise} \end{cases}$$

Show computations for 3 monus 1 and 1 monus 3.

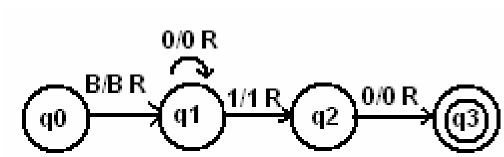


3. Create a Turing machine to accept the language: a(a U b)* b

Here's one solution:

M = ({q_0,q_1,q_2,q_3}, {a,b}, {a,b,B}, δ, q_0,{q_3})
δ (q_0,a) = (q_1,a,R) ***Check that first symbol is an a***
δ (q_1,a) = (q_1,a,R) ***Skip over a's***
δ (q_1,b) = (q_1,b,R) ***Skip over b's***
δ (q_1,B) = (q_1,B,L) ***When B found, check symbol to left***
δ (q_2,B) = (q_3,b,R) ***If it's "b" accept (halt in a final state)***

#4. Given the following Turing machine,



a) What is L(M)

0*10(1U0)*

b) Show R(M) using the encodings of Section 11.5 (discussed in class)

0001011101101110110011010110101100 11011011101101100 111010111101011000

#5. Construct a Turing machine in words (i.e, describe its moves without actually writing all the transitions) that determines whether a string over {0,1} is the encoding of a Turing machine.

The encoding of a TM looks like this:

000 en(qi) 0 en(x) 0 en(qj) 0 en(y) 0 en(d where d = L,R) 00 … 000
where all these en()'s are sequences of one or more 1's

So your Turing machine (part of the Universal TM) needs to check for this format. You should have something like::

1. **Does it begin with 000?**
   a. **If no, reject (loop forever)**
   b. **Otherwise go to Step 2**
2. **Does the tape look like (some 1's followed by a 0)^5 ?**
   a. **If no, reject (loop forever)**
   b. **If yes, go to step 3**
3. **Is there another 0 (so there are two 0's in a row signifying the end of the transition)**
   a. **If yes, is there yet another 0?**
      i. **Yes (so three 0's in a row signifying the end of all transitions), then accept (halt)**
      ii. **No, then there is another transition. Go to Step 2.**
   b. **If no, loop forever**
   c.

All of these steps could be done with (many) transitions in the universal TM, U.