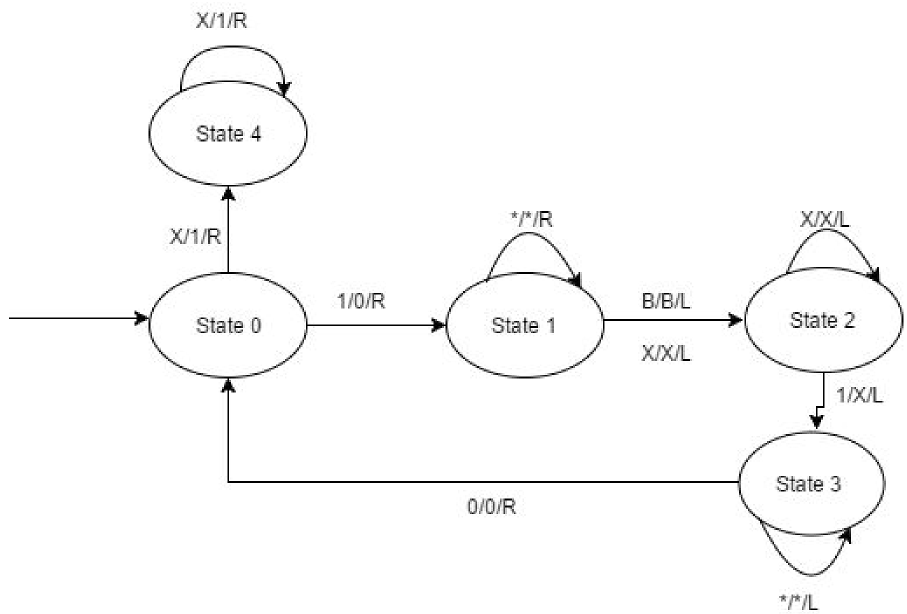
Problem Set 6 Solutions
(12.24, 12.25, 12.28, 12.32, 12.40)
COMS W1004 Fall '16

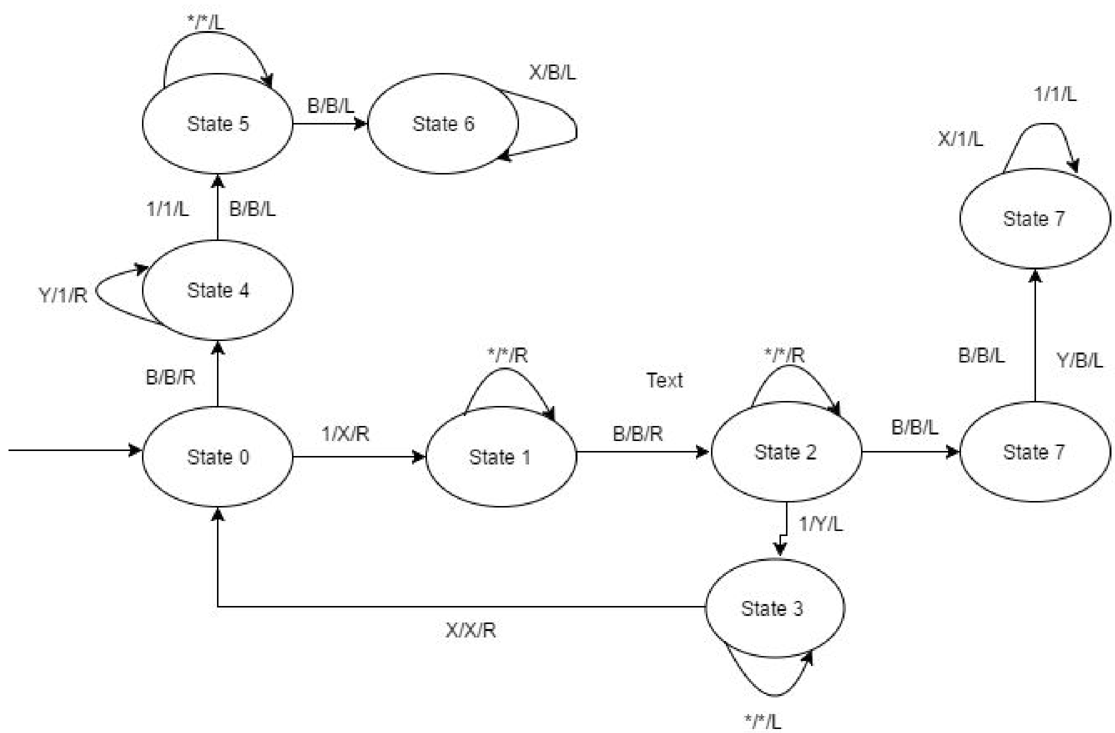
Note: all Turing machine solutions are not unique. In some cases, there might also be more optimal solutions.
(All from Schneider and Gersting)
12.24: Write a turing machine that takes any unary string of an even number of 1s and halts when the first half of the string changed to 0s.

Solution:
Instruction set (see Instruction_Sets.txt)
State diagram:



12.25: Write a turing machine that takes as input the unary representation of any two different numbers separated by a blank, and halts with the representation of the larger of the two numbers on the tape.

Solution:
Instruction set (see Instruction_Sets.txt)
State diagram:



<u>12.28:</u> Draw a state diagram for a Turing machine that increments a binary number. Thus, if the binary representation of 4 is initially on the tape,

… b 1 0 0 …

Then the output is the binary representation of 5,

… b 1 0 1 …

Or if the initial tape contains the binary representation of 7,

… b 1 1 1 …

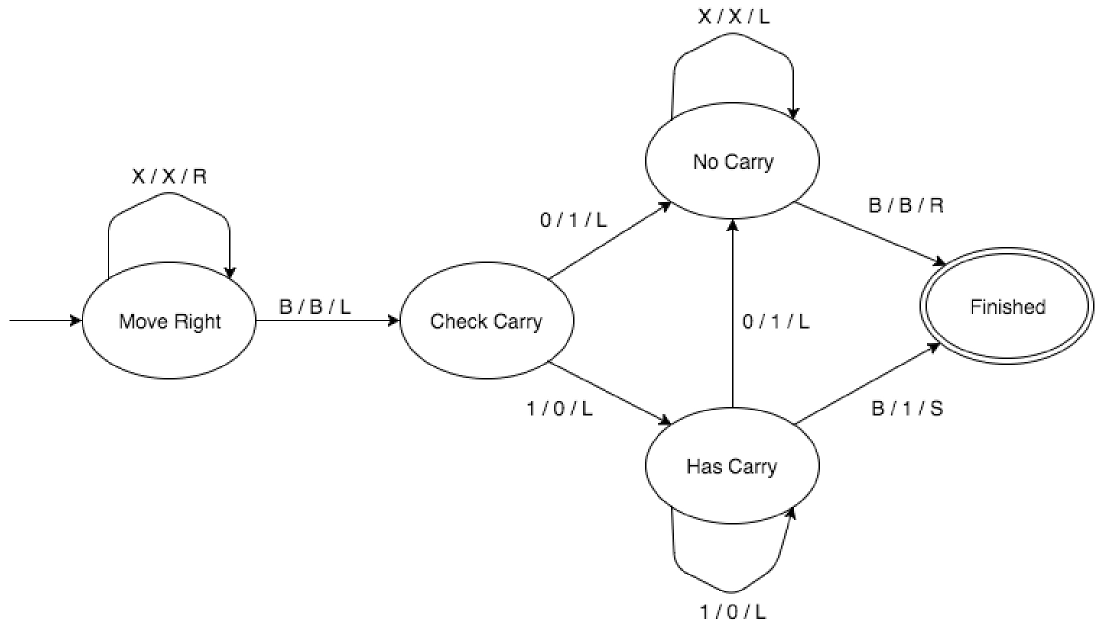
Then the output is the binary representation of 8,

… b 1 0 0 0 …

<u>Solution:</u>

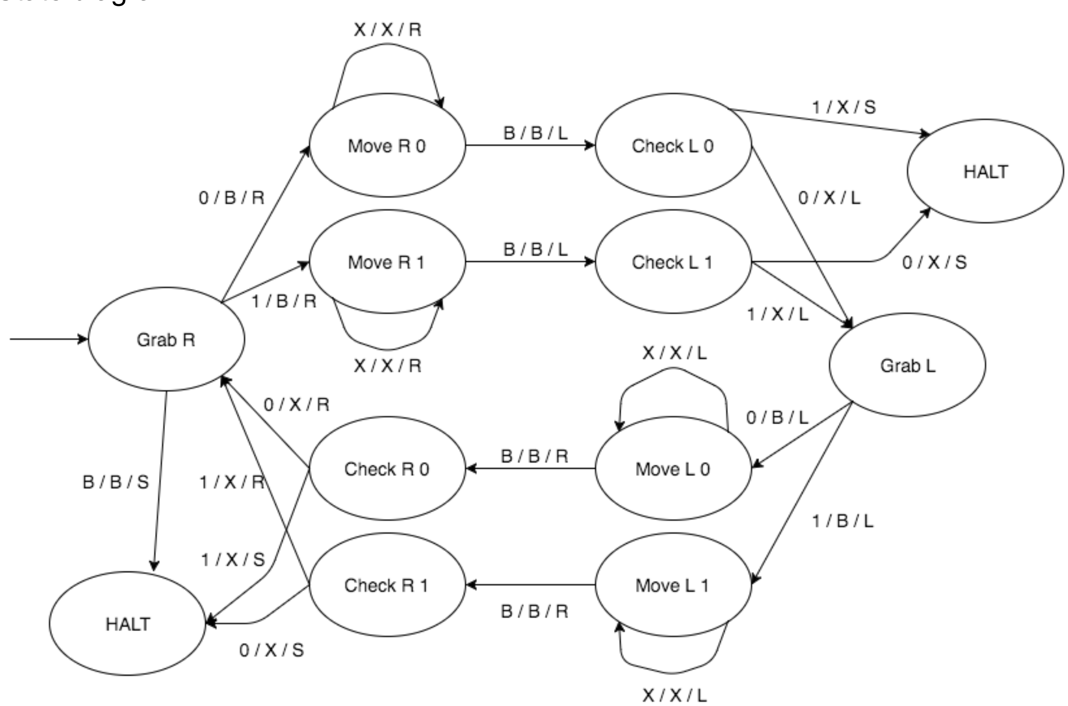
Instruction set (see Instruction_Sets.txt)

State diagram:



12.32: A palindrome is a string of characters that reads the same forward and backward, such as radar or IUPUI. Write a Turing machine to decide whether any binary string is a palindrome by halting with a blank tape if the string is a palindrome and halting with a nonblank tape if the string is not a palindrome.

Solution:
Instruction set (see Instruction_Sets.txt)
State diagram:



<u>12.40:</u> The 10-step halting problem is to decide, given any collection of Turing machine instructions, together with any initial tape contents, whether that Turing machine will halt within 10 steps when started on that tape. Explain why the 10-step halting problem is computable.

<u>Solution:</u>

Let's call the Turing machine associated with the collection of instructions defined by the problem T. The 10-step halting problem is computable because we can design a machine, S, that will simulate the first 10 steps of T. If T halts in 10 steps, then S will show 1 and halt, and if T does not halt in the first 10 steps, then S will still halt, but show 0.

In this way, we can always know whether a machine T halts in 10 steps. Therefore, the 10-step halting problem is computable.