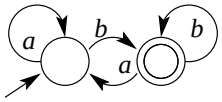
Question 9.1–1: (Solution, p 3) Consider the following finite automaton.



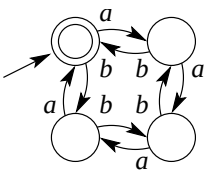
a. Check the strings that the automaton will accept.

__ *b*	__ *bab*
__ *a*	__ *bbbb*
__ *aba*	__ *baba*

b. Give an English description of the set of strings accepted by this automaton.

Question 9.1–2: (Solution, p 3) Draw a finite automaton that accepts all strings containing only *a*'s and *b*'s that begin in *a*.

Question 9.1–3: (Solution, p 3) Consider the following finite automaton.



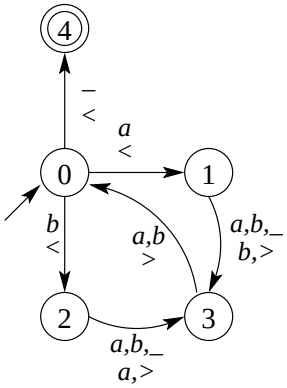
Check the strings that are within the language accepted by this finite automaton.

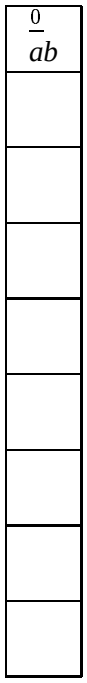
__ *ab*	__ *aabbb*
__ *bbb*	__ *bbbabb*
__ *baaa*	__ *aabaabaa*
__ *abba*	

Question 9.1–4: (Solution, p 3) Design a finite state automaton that will recognize the language of all strings containing only *a*'s and *b*'s where there are at least 3 *b*'s.

Question 9.2–1: (Solution, p 3)

Consider the following Turing machine. (Note that the underscore represents a blank on the tape.)





At right, diagram this Turing machine's computation as it goes through the string *ab*. If you run out of blanks in the table, stop.

To represent the machine's initial position in the table at right, we write "$\underline{0}\ ab\ $". This represents a tape containing "$ab\ $" (with blanks extending infinitely both ways), where the Turing machine is currently in state 0 of its finite automaton, and its head is pointing to the initial *a*.

Question 9.2–2: (Solution, p 3) Design a Turing machine that transforms a string containing only *a*'s, *b*'s, and *c*'s by replacing each letter preceding an *a* to a *b*. (Do not worry about the case when the string begins with an *a*.) Thus, *bccb* would remain unchanged while *caccaa* would change to *bacbba*. The Turing machine should always eventually enter an accepting state to terminate.

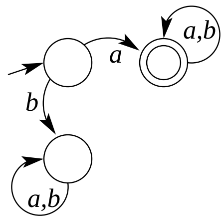
Solution 9.1–1: (Question, p 1)

a.

√ *b*	√ *bab*
__ *a*	√ *bbbb*
__ *aba*	__ *baba*

b. It accepts exactly those strings containing only *a*’s and *b*’s that end in a *b*.

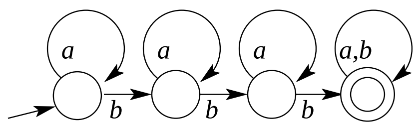
Solution 9.1–2: (Question, p 1)



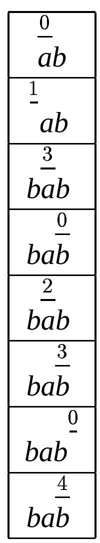
Solution 9.1–3: (Question, p 1)

√ *ab*	__ *aabbb*
__ *bbb*	√ *bbbabb*
__ *baaa*	√ *aabaabaa*
√ *abba*	

Solution 9.1–4: (Question, p 1)



Solution 9.2–1: (Question, p 2)



(At this point, the machine has nowhere to go, and so it stops.)

Solution 9.2–2: (Question, p 2)