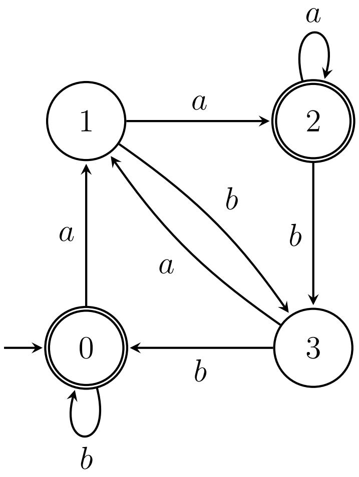
Applied Algebra

Instructions Please write your name in the upper right-hand corner of the page. Use complete sentences, along with any necessary supporting calculations, to answer the following questions.

1. Determine the input sequences that the following four-state automaton accepts. Notice that both the initial state 0 and the state 2 are acceptance states.



Solution. I claim that the accepted inputs are the following: the empty input, the single-letter input b, and all inputs whose final two letters match. You may be able to see this answer by inspection, but here is some discussion in case you are not convinced.

Since the initial state 0 is an acceptance state, the empty input is accepted. Since there is a b-loop at state 0, an input consisting of the single letter b is accepted. For the same reason, an input consisting entirely of repetitions of the letter b is accepted, but this case need not be listed separately: it is a special case of an input ending in bb.

Now consider any input that does end in bb. Since the arrows with label b all point either to state 3 or to state 0, the machine moves to one of these states when the next-to-last letter is read. When the last letter is read, the machine moves to the acceptance state 0, since the arrows leaving states 3 and 0 with label b both point to state 0. Thus every input ending bb is accepted because it lands the machine in the acceptance state 0.

Conversely, it needs to be shown that every input of length at least 2 that lands the machine in state 0 does end in *bb*. Evidently the last letter of the input is *b*, because the only arrows that point to state 0 have label *b*. These arrows come from either state 3 or state 0 itself, so the next-to-last input letter must also be *b* (since the only way to get to either state 3 or state 0 is via an arrow with label *b*). There is no constraint on the third-to-last input letter, however, for you can get to state 0 from itself either by the sequence *bbb* (staying always at state 0) or by the sequence *abb* (going through states 1 and 3).

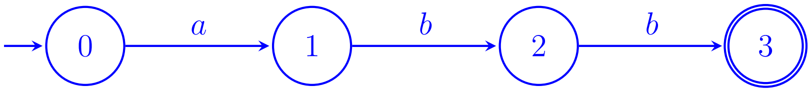
A completely parallel argument (notice the symmetry in the diagram) shows that every input sequence that ends in *aa* lands the machine in state 2 and is therefore accepted, and every sequence that lands the machine in state 2 does end in *aa*.

As a check, observe from the diagram that every input sequence ending in *ab* lands the machine in state 3 and is therefore not accepted, and every input sequence ending in *ba* lands the machine in state 1 and is therefore not accepted.

In summary, an input sequence of 2 or more letters is accepted if and only if the sequence ends in either *aa* or *bb*.

2. Construct a four-state automaton that accepts the input sequences of letters *a* and *b* ending *abb* (and does not accept any other input sequences).

Solution. The skeleton of the machine must look like the following.



Now it is necessary to fill in the rest of the arrows.

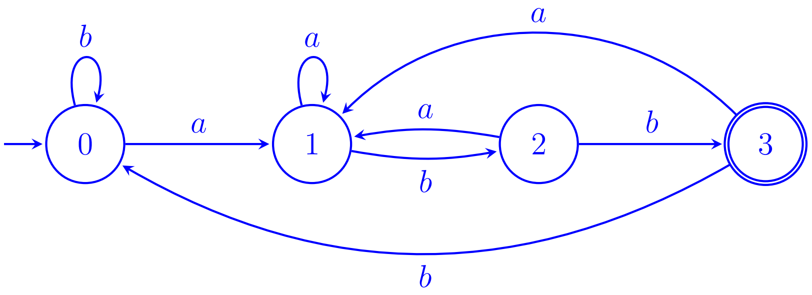
The *b*-arrow leaving state 0 cannot point to state 1, for then the input *bbb* would be accepted. Similarly, this arrow cannot point to state 2 or to state 3. Thus there must be a *b*-loop at state 0.

The *a*-arrow leaving state 1 cannot point to state 2, for then the input *aab* would be accepted. Similarly, this arrow cannot point to state 3. The *a*-arrow leaving state 1 cannot point to state 0 for a more subtle reason: if it did, then the input *aabb* would end in state 0 (we already know that there is a b-loop at state 0), so this input would not be accepted, contrary to the statement of the problem. The only possibility remaining is that there is an a-loop at state 1.

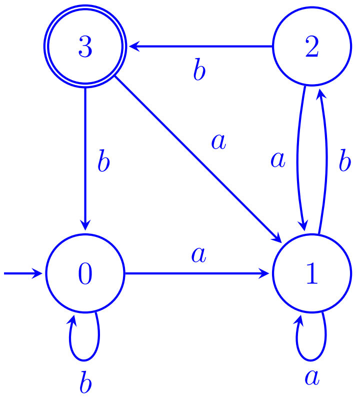
Similar reasoning shows that the a-arrows leaving states 2 and 3 must point to state 1.

The b-arrow leaving state 3 cannot point to state 1, for then the input $abbbbb$ would be accepted; similarly, this arrow cannot point to state 2 or to state 3. Therefore this arrow points to state 0.

Here is the final state diagram.



Instead of drawing the state diagram along a line, you could fold it up into a square. Or you could describe the machine by a transition table, with state 3 designated as the acceptance state. These answers are illustrated below.



	a	b
0	1	0
1	1	2
2	1	3
3	1	0