Gradiance Online Accelerated Learning

Zayd

- Home Page
- Assignments Due
- Progress Report
- Handouts
- Tutorials
- Homeworks
- Lab Projects
- Log Out

Help

Submission number:	65663
Submission certificate:	BG113976
Submission time:	2014-02-28 19:00:27 PST (GMT - 8:00)

Number of questions:	4
Positive points per question:	3.0
Negative points per question:	1.0
Your score:	4

Based on Section 4.4 of HMU

1. Here is the transition table of a DFA:

	0	1
→A	E	D
*B	A	C
C	G	B
D	E	A
*E	H	C
F	C	B
G	F	E
H	B	H

Find the minimum-state DFA equivalent to the above. Then, identify in the list below the pair of equivalent states (states that get merged in the minimization process).

a) F and G

b) E and G

c) C and H

d) C and D

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

Here is the table of distinguishabilities:

A							
B	x	x					
C	x	x	x				
D			x	x			
E	x	x		x	x		
F	x	x	x		x	x	
G	x	x	x		x	x	
	H	**A**	**B**	**C**	**D**	**E**	**F**

We start by noting that accepting states B and E are distinguishable from all the other states. In this problem, each of the other x's in the table can be filled in immediately because the pair of distinguishable states goes, on either 0 or 1, to one state that is accepting and another that is not.

2. Here is the transition table of a DFA that we shall call *M*:

	0	**1**
→A	B	G
B	C	H
*C	D	G
*D	A	H
E	F	C
F	G	I
*G	H	C
*H	A	D
I	E	I

Find the minimum-state DFA equivalent to the above. States in the minimum-state DFA are each the merger of some of the states of *M*. Find in the list below a set of states of *M* that forms one state of the minimum-state DFA.

a) {C,D}

b) {G,I}

c) {A}

d) {B,E}

Answer submitted: **d)**

Your answer is incorrect.

On input 0, B goes to accepting state C, while E goes to nonaccepting state F. Moreover, E is not accessible from the start state, and so should not be included in any state of the minimum-state DFA. See Section 4.4.3 (p. 160) for the state-minimization algorithm.

Question Explanation:

First, there are several inaccessible states: E, F, and I. You can verify that there is no path from start state A to any of these states. These three states are not part of any state of the minimum-state DFA.

For the remaining six states, here is the table of distinguishabilities:

A	x				
B	x	x			
C	x	x	x		
D		x	x	x	
G	x	x	x		x
	H	**A**	**B**	**C**	**D**

We start by noting that accepting states C, D, and H are distinguishable from the other three states. Next, we can distinguish A, D, and H from B, C, and G, because the former go to nonaccepting states on input 0, while the latter three go to accepting states. Last, we can distinguish A from B, because on input 1, A goes to G while B goes to H, and we distinguished G from H on the previous round.

The correct choice is: **c)**

3. Design the minimum-state DFA that accepts all and only the strings of 0's and 1's that have 110 as a substring. To verify that you have designed the correct automaton, we will ask you to identify the true statement in a list of choices. These choices will involve:

1. The number of *loops* (transitions from a state to itself).
2. The number of transitions into a state (including loops) on input 1.
3. The number of transitions into a state (including loops) on input 0.

Count the number of transitions into each of your states ("in-transitions") on input 1 and also on input 0. Count the number of loops on input 1 and on input 0. Then, find the true statement in the following list.

a) There are two states that have two in-transitions on input 1.

b) There are two states that have no in-transitions on input 1.

c) There are two states that have two in-transitions on input 0.

d) There are three states that have one in-transition on input 1.

Answer submitted: **c)**

You have answered the question correctly.

Question Explanation:

Here is the transition table for the DFA. The intuitive meanings of the states are:

A = Nothing of 110 has been found, e.g., before any input has been read, or after you have just seen 00.
B = We've seen 1, but not 11.

C = We've seen 11 but never seen 110.
D = We've seen 110, either now or at some time in the past.

	0	**1**
→A	A	B
B	A	C
C	D	C
*D	D	D

4. Design the minimum-state DFA that accepts all and only the strings of 0's and 1's that end in 010. To verify that you have designed the correct automaton, we will ask you to identify the true statement in a list of choices. These choices will involve:

1. The number of *loops* (transitions from a state to itself).
2. The number of transitions into a state (including loops) on input 1.
3. The number of transitions into a state (including loops) on input 0.

Count the number of transitions into each of your states ("in-transitions") on input 1 and also on input 0. Count the number of loops on input 1 and on input 0. Then, find the true statement in the following list.

a) There is one state that has one in-transition on input 0.

b) There are two states that have two in-transitions on input 0.

c) There is one state that has two in-transitions on input 1.

d) There are two states that have one in-transition on input 0.

Answer submitted: **d)**

Your answer is incorrect.

To design your DFA, think of the states as remembering all you need to know about the progress made recently toward seeing tail 010. Then, ask yourself: if I have seen some prefix of the tail 010, and a 0 (or a 1) is the next input, how much of the tail have I now seen. For example, if I have seen 0, but not 010, and a 0 comes in, then I have still seen only the prefix 0. If a 1 comes in instead, then I have seen 01.

Section 1.1.1 (p. 2) gives several examples of design of automata in which states are used to remember the important aspects of the input history. The formal definition of a DFA is in Section 2.2 (p. 45), and minimizing the number of states is covered in Section 4.4.3 (p. 160).

Question Explanation:

Here is the transition table for the DFA. The intuitive meanings of the states are:

A = Nothing of the tail 010 has been found, e.g., before any input has been read, or after you have just seen 11.
B = We've just seen 0, but not 010.

C = We've just seen 01.
D = We've just seen 010 and accept.

	0	**1**
→A	B	A
B	B	C
C	D	A
*D	B	C

The correct choice is: **a)**