Gradiance Online Accelerated Learning

Zayd

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

Help

Submission number:	61151
Submission certificate:	BI530982
Submission time:	2014-02-16 18:01:32 PST (GMT - 8:00)

Number of questions:	7
Positive points per question:	3.0
Negative points per question:	1.0
Your score:	13

Based on Sections 3.1 and 4.1 of HMU.

1. Consider the languages.

(a) $\{0^{2n}1^n \mid n > 0\}$
(b) $\{0^{5n}1^n \mid n > 0\}$
(c) {w | w a string of 0's and 1's such that when interpreted in reverse as a binary integer it is a multiple of 5}
(d) $\{0^n1^n|n>0\}$
(e) {w | w a string of 0's and 1's such that its length is a perfect square}
(f) {w | w string of 0's and 1's such that when interpreted as a binary integer it is not a multiple of 5}
(g) {w | w a string of 0's and 1's such that its length is not a perfect cube}
(h) {w | w a string of 0's and 1's such that the number of 0's is not equal to twice the number of 1's}

Which is a regular language?

a) (d)
b) (a)
c) (h)
d) (f)

Answer submitted: **c)**

Your answer is incorrect.

First use the property that the complement of a regular language is a regular language (Theorem 4.5, p. 135) and then apply the pumping lemma (Section 4.1, p. 128) on the complement to prove that it is not a regular language. The string to pump is $0^{n!}1^{n!}$. Notice that the pumped string consists of 0's only, and it is of length between 1 and n. As a result, we can always find some number of times to pump this string that will turn it into $0^{2n!}1^{n!}$.

2. Which of the following strings is NOT in the Kleene closure of the language {011, 10, 110}?

a) 011011110

b) 01110011

c) 1010110

d) 10111011

Answer submitted: **d)**

You have answered the question correctly.

3. Here are seven regular expressions:

1. (0*+10*)*
2. (0+10)*
3. (0*+10)*
4. (0*+1*)*
5. (0+1)*
6. (0+1*0)*
7. (0+1*)*

Determine the language of each of these expressions. Then, find in the list below a pair of equivalent expressions.

a) (0+1)* and (0+1*)*

b) (0+1*0)* and (0*+1*)*

c) (0+1)* and (0+1*0)*

d) (0*+10*)* and (0+1*0)*

Answer submitted: **a)**

You have answered the question correctly.

4. Which among the following languages is not regular (cannot be defined by a regular expression or finite automaton)?

a) L={x | x=$a^m(bc^k)^n$, n, m, k positive integers}

b) L={x | x=ca^mcb^n, n, m positive integers}

c) L={x | x=$(a^2b^2c^2)^n$, n a positive integer}

d) L={x | x=$(ab^2 c)^n$, n a positive integer}

Answer submitted: **a)**

You have answered the question correctly.

5. Identify from the list below the regular expression that generates all and only the strings over alphabet {0,1} that end in 1.

a) (0+1)*10?

b) (0*1)?

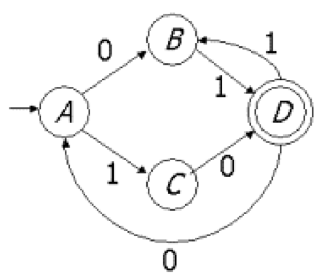
c) (0+1)*1+

d) (0+1)*10*

Answer submitted: **c)**

You have answered the question correctly.

6. Here is a finite automaton:



Which of the following regular expressions defines the same language as the finite automaton? Hint: each of the correct choices uses component expressions. Some of these components are:

1. The ways to get from A to D without going through D.
2. The ways to get from D to itself, without going through D.
3. The ways to get from A to itself, without going through A.

It helps to write down these expressions first, and then look for an expression that defines all the paths from A to D.

a) ((01+10)0)*(01+10)(11)*

b) ((01+10)(11)*0)*(01+10)(11)*

c) (01+10)(11*+0(01+10))*

d) ((01+10)(11)*0)*(01+10)

Answer submitted: **c)**

Your answer is incorrect.

The language of this expression includes 01111, which is not accepted by the automaton. Remember that 11* means one or more 1's, while (11)* means an even number of 1's. This exercise is intended to explore the techniques for converting from automata to regular expressions that are contained in Section 3.2.1 (p. 93). Remember that you can choose the order in which you number states as you like, in order to make your calculations for this problem as simple as possible.

7. In this question you are asked to consider the truth or falsehood of six equivalences for regular expressions. If the equivalence is true, you must also identify the law from which it follows. In each case the statement R = S is conventional shorthand for "L(R) = L(S)." The six proposed equivalences are:

1. $0^*1^* = 1^*0^*$
2. $01\varphi = \varphi$
3. $\varepsilon 01 = 01$
4. $(0^* + 1^*)0 = 0^*0 + 1^*0$
5. $(0^*1)0^* = 0^*(10^*)$
6. $01+01 = 01$

Identify the correct statement from the list below.

Note: we use φ for the empty set, because the correct symbol is not recognized by Internet Explorer.

a) $(0^* + 1^*)0 = 0^*0 + 1^*0$ follows from the commutative law for union.

b) $0^*1^* = 1^*0^*$ follows from the associative law for concatenation.

c) $0^*1^* = 1^*0^*$ is false.

d) $(0^* + 1^*)0 = 0^*0 + 1^*0$ follows from the associative law for concatenation.

Answer submitted: **c)**

You have answered the question correctly.