Gradiance Online Accelerated Learning

Zayd

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

Help

Submission number:	64803
Submission certificate:	EB641652
Submission time:	2014-02-24 18:57:49 PST (GMT - 8:00)

Number of questions:	5
Positive points per question:	4.0
Negative points per question:	0.0
Your score:	20

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

a) 10111011

b) 01111010

c) 1101010

d) 11001110

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

Every string in the language {011, 10, 110}* has to be formed from zero or more uses of the strings 011, 10, and 110. A string may be used more than once.

In this simple example, we can look at a string of 0's and 1's and break it into pieces that are 011, 10, or 110 simply. Look at the first two positions. If they are 01, then the next position must be 1 and we can remove 011 from the beginning and repeat the process. If the first two positions are 10 instead, then remove these and repeat. If the first two positions are 11, then check that the next position is 0 and remove 110 from the front if so, and repeat. If so doing eventually leaves us with the empty string, then the string is in {011, 10, 110}*; otherwise it is not.

2. Find, in the list below, a regular expression whose language is the reversal of the language of this regular expression: (0+1)(1+2)*(0+2). Recall that the reversal of a language is formed by reversing all its strings, and the reversal of a string $a_1a_2...a_n$ is $a_n...a_2a_1$.

a) (0+1)(0+2)(1+2)*

b) (0+2)(0+1)(1+2)*

c) (1+2)*(0+2)(0+1)

d) (0+2)(1+2)*(0+1)

Answer submitted: **d)**

You have answered the question correctly.

Question Explanation:

A good way to form a regular expression for the reversal of the language of a concatenation of regular expressions is to reverse the order of these regular expressions, and then work on the reversal of each of these subexpressions. For the particular examples found in this question, all you then need to know is that the reversal of a union of one or more characters does not change the expression or the set it denotes. Also, if an expression is its own reversal, then its closure is also the reversal of its closure.

For example, the reversal of 0 is 0; the reversal of 0+1 is 0+1, and the reversal of (0+1)* is (0+1)*. These three expressions represent, respectively, {0}, {0,1}, and the set of all strings of 0's and 1's. Each of these languages is its own reversal.

3. h is a homomorphism from the alphabet {a,b,c} to {0,1}. If h(a) = 01, h(b) = 0, and h(c) = 10, which of the following strings is in h^{-1}(010010)?

a) *bcab*

b) *babc*

c) *cbcb*

d) *baab*

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

Suppose h(w) = 010010. As there are two 1's in 010010, there must be exactly two positions of *w* from the set {*a*, *c*}. These two positions generate two 0's, so there must be exactly two other positions of *w* that have *b*. That is, *w* consists of two *b*'s and two *a*'s or *c*'s. The first symbol of *w* cannot be *c*, or h(w) would start with 1. Similarly, the last symbol cannot be *a*.

If the first symbol of *w* is *b*, then the next must be *c*, and if the first symbol of *w* is *a*, then the next must be *b*. Similarly, the last two positions of *w* can only be *ab* or *bc*. That is, the four possible values of *w* are *abab, abbc, bcab*, and *bcbc*.

4. 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+10)*

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

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

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

Answer submitted: **d)**

You have answered the question correctly.

Question Explanation:

Expressions 1, 4, 5, and 7 all generate all strings of 0's and 1's. They are thus all equivalent. Expressions 2 and 3 are equivalent; each generates all strings of 0's and 1's such that every 1 is followed by a 0. Expression 6 is equivalent to none of the others, since it generates all strings of 0's and 1's that are either empty or end in 0.

5. 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. 01φ = φ
3. ε01 = 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*(10*) follows from the idempotent law for union.

b) 01φ = φ follows from the identity law for concatenation.

c) ε01 = 01 follows from the annihilator law for concatenation.

d) 0*1* = 1*0* is false.

Answer submitted: **d)**

You have answered the question correctly.

Question Explanation:

1. $0^*1^* = 1^*0^*$ is false.
2. $01\varphi = \varphi$ follows from the annihilator law for concatenation.
3. $\varepsilon 01 = 01$ follows from the identity law for concatenation.
4. $(0^* + 1^*)0 = 0^*0 + 1^*0$ follows from the distributive law of concatenation over union.
5. $(0^*1)0^* = 0^*(10^*)$ follows from the associative law for concatenation.
6. $01+01 = 01$ follows from the idempotent law for union.