gradiance

Gradiance Online Accelerated Learning

Zayd

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

Help

Submission number:	61153
Submission certificate:	FG740615
Submission time:	2014-02-16 17:41:18 PST (GMT - 8:00)

Number of questions:	5
Positive points per question:	3.0
Negative points per question:	1.0
Your score:	11

Based on Sections 3.2 and 3.4 of HMU.

1. 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) 01+01 = 01 follows from the commutative law for union.

b) (0*1)0* = 0*(10*) is false.

c) (0*1)0* = 0*(10*) follows from the associative law for concatenation.

d) (0* + 1*)0 = 0*0 + 1*0 follows from the commutative law for union.

Answer submitted: **c)**

You have answered the question correctly.

2. Consider the following identities for regular expressions; some are false and some are true. You are asked to decide which and in case it is false to provide the correct counterexample.

(a) R(S+T)=RS+RT
(b) (R*)*=R*
(c) (R*S*)*=(R+S)*

(d) (R+S)*=R*+S*
(e) S(RS+S)*R=RR*S(RR*S)*
(f) (RS+R)*R=R(SR+R)*

a) (c) is false and a counterexample is: R={ab},T={b}, S={b}

b) (e) is false and a counterexample is: R={a,ε},T={b}, S={a,ε}

c) (d) is false and a counterexample is: R={a,ε},T={b}, S={a,ε}

d) (e) is false and a counterexample is: R={a},T={a}, S={b}

Answer submitted: **d)**

You have answered the question correctly.

3. Apply the construction in Figure 3.16 (p. 104) and Figure 3.17 (p. 105) to convert the regular expression (0+1)*(0+ε) to an epsilon-NFA. Then, identify the true statement about your epsilon-NFA from the list below:

a) There are 14 arcs labeled ε.

b) There are no states with more than one arc in.

c) There are 5 states with more than one arc out.

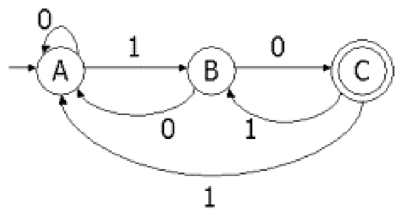
d) There are 17 arcs labeled ε.

Answer submitted: **b)**

Your answer is incorrect.

Start by applying the basis rules in Figure 3.16 (p. 104) to each of the operands, 0, 1, and ε. Remember that there are two occurrences of 0, and each needs its own automaton. Then use the recursive rules to combine automata, as in Figure 3.17 (p. 105). You need to use the rule for union twice, then the rule for closure, and finally the rule for concatenation. The entire process is outlined in Section 3.2.3 (p. 102).

4. When we convert an automaton to a regular expression, we need to build expressions for the labels along paths from one state to another state that do not go through certain other states. Below is a nondeterministic finite automaton with three states. For each of the six orders of the three states, find regular expressions that give the set of labels along all paths from the first state to the second state that never go through the third state.



Then identify one of these expressions from the list of choices below.

a) 1 represents the paths from C to B that do not go through A.

b) (10)*1 represents the paths from C to B that do not go through A.

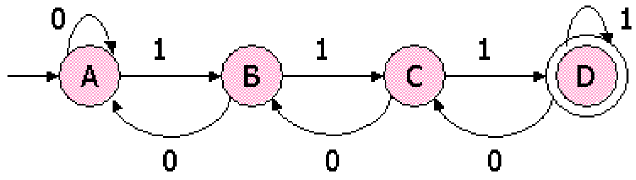
c) 1+101 represents the paths from C to B that do not go through A.

d) $(01)^+$ represents the paths from B to C that do not go through A.

Answer submitted: **b)**

You have answered the question correctly.

5. Converting a DFA such as the following:



to a regular expression requires us to develop regular expressions for limited sets of paths --- those that take the automaton from one particular state to another particular state, without passing through some set of states. For the automaton above, determine the languages for the following limitations:

1. L_{AA} = the set of path labels that go from A to A without passing through C or D.
2. L_{AB} = the set of path labels that go from A to B without passing through C or D.
3. L_{BA} = the set of path labels that go from B to A without passing through C or D.
4. L_{BB} = the set of path labels that go from B to B without passing through C or D.

Then, identify a correct regular expression from the list below. Note: there are several different regular expressions possible for each of these languages. However, each of the correct answers can be thought of as built from more limited components. For example, the regular expression **1** is the set of path labels that go from A to B without passing through any of the four states.

a) L_{BA} = 0(0+10)*

b) L_{AA} = (0+1(10)*0)*

c) L_{AB} = 0*1(01)*

d) L_{BB} = (0(00*10)*)*

Answer submitted: **a)**

You have answered the question correctly.