

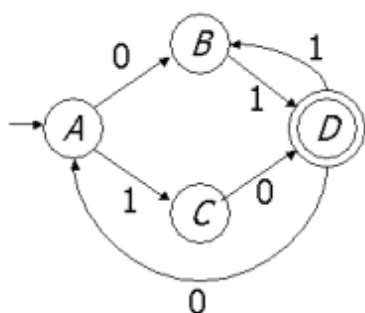
Feedback — Week 2: Regular Expressions

[Help Center](#)

You submitted this homework on **Sat 3 Oct 2015 7:34 PM CEST**. You got a score of **5.00** out of **5.00**.

Question 1

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.

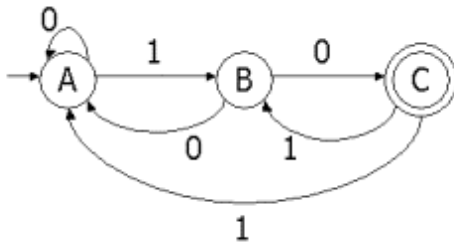
It helps to write down these expressions first, and then look for an expression that defines all the paths from A to D.

Your Answer	Score	Explanation
<input type="radio"/> $((01+10)(11)^*0)^*(01+10)$		
<input checked="" type="radio"/> $(01+10)(11+0(01+10))^*$	✓ 1.00	
<input type="radio"/> $((01+10)0+11)^*(01+10)(11)^*$		
<input type="radio"/> $(01+10)(0(01+10))^*(11)^*$		
Total	1.00 / 1.00	

Question 2

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.

Your Answer	Score	Explanation
<input type="radio"/> $0+010$ represents the paths from B to C that do not go through A.		
<input checked="" type="radio"/> $0(0+10)^*$ represents the paths from B to A that do not go through C.	✓ 1.00	
<input type="radio"/> 0^*1 represents the paths from A to B that do not go through C.		
<input type="radio"/> 0 represents the paths from B to A that do not go through C.		
Total	1.00 / 1.00	

Question 3

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

Your Answer	Score	Explanation
<input type="radio"/> 1^*0^*1		

☐ $(0+1)^*1^+1$

☐ $(0^*1)^?$

☒ $(0+1)^*1^+$ ✓ 1.00 This expression is correct. $(0+1)^*$ generates all the strings of 0's and 1's, including the empty string, and 1^+ generates one or more 1's. Any string that ends in 1 can be written as a tail of one or more 1's, generated by 1^+ , preceded by some string of 0's and 1's.

Total 1.00 / 1.00

Question 4

Convert the regular expression $(0+1)^*(0+\epsilon)$ to an epsilon-NFA, using the algorithm described in the videos. Then, identify the true statement about your epsilon-NFA from the list below:

Your Answer	Score	Explanation
<input checked="" type="radio"/> There are 14 arcs labeled ϵ .	✓ 1.00	
<input type="radio"/> There are 3 states with more than one arc out.		
<input type="radio"/> There are 6 states with more than one arc out.		
<input type="radio"/> There are 5 states with more than one arc out.		
Total	1.00 / 1.00	

Question 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:

- $0^*1^* = 1^*0^*$
- $01\emptyset = \emptyset$
- $\epsilon 01 = 01$
- $(0^* + 1^*)0 = 0^*0 + 1^*0$
- $(0^*1)0^* = 0^*(10^*)$
- $01+01 = 01$

Identify the correct statement from the list below.

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

Your Answer	Score	Explanation
<input checked="" type="radio"/> $01+01 = 01$ follows from the idempotent law for union.	✓ 1.00	
<input type="radio"/> $(0^*1)0^* = 0^*(10^*)$ follows from the idempotent law for union.		
<input type="radio"/> $(0^*1)0^* = 0^*(10^*)$ follows from the distributive law of concatenation over union.		
<input type="radio"/> $0^*1^* = 1^*0^*$ follows from the associative law for concatenation.		
Total	1.00 / 1.00	