Finite Automata and Regular Languages – Problems

Problem 1. Regular Operations

The class of regular languages is closed under the concatenation operation. This means that if A and B are both regular languages, then $A \circ B$ is also a regular language.

Prove this by drawing a *rough, approximate* diagram for a finite automaton that recognizes the language $A \circ B$, using the finite automata for A and B as building blocks.

Answer: The accept states of A are connected to the start state of B via arrows labelled with the empty string. The accept states of A no longer become accept states; the only accept states of the overall machine are therefore those from B. So, computation accepts any strings which first pass the tests of machine A, then passes the tests of machine B.

The class of regular languages is closed under the union operation. This means that if A and B are both regular languages, then $A \cup B$ is also a regular language.

Prove this by drawing a *rough, approximate* diagram for a finite automaton that recognizes the language $A \cup B$, using the finite automata for A and B as building blocks.
Hint: This will be substantially easier if you use a ***nondeterministic*** *finite automaton.*

Answer: Create a start state. From this start state, draw two arrows labelled with the empty string. One goes to the start state of A, the other goes to the start state of B. In this way, any string which (nondeterministically!) either passes the tests of machine A OR passes the tests of machine B will succeed.

The class of regular languages is closed under the star operation. This means that if A is a regular language, then $A*$ is also.

Prove this by drawing a *rough, approximate* schematic for a finite automaton that recognizes the language $A*$, using the finite automaton for A as a building block.

To create this one, you loop the accept states of A back around to the start state of A using arrows labelled with the empty string. So any part-of-a-string which would accept is able to loop back around to the beginning and pass through the automaton again.

Problem 2. Regular Expressions

Regular languages can be written down using the regular operations, as an alternative to drawing out the entire state diagram. This is called a ***regular expression.***

For example, this is the regular expression for the regular language consisting of all strings that only consist of 0s:

0*

This is the regular expression for the regular language consisting of 01, 0101, 010101, and so on:

0∘1∘(0∘1)*   (*or, abbreviated:* 01(01)*)

Parentheses are allowed, in order to group items together.

a) Write the regular expression for the language consisting of any number of 0s, followed by any number of 1s.

0* 1*

b) Write the regular expression for the language consisting of all strings that begin and end with a 0.

0 (0 u 1)* 0

c) Write the regular expression for the language consisting of all strings which have an even number of 1s.

(0* 1 0* 1 0*)*

d) Write the regular expression for the language consisting of only the string 100101 or the string 0111000, but no other strings.

100101 u 0111000

e) Write the regular expression for the language consisting of all strings which have an even number of 1s or an even number of 0s (or both).

(0* 1 0* 1 0*)* u (1* 0 1* 0 1*)*