Lecture #4: Regular Operations and Regular Expressions

What To Do Before the Lecture

1. Watch the videos for Lecture #4 —noting that they will probably be understandable if you play them at double speed. If you do not have time for this then look at the "Key Concepts" document that is found, immediately after the videos for this lecture, on the course web site, instead.

2. ***Print*** and read through the rest of this document and — if you have time — try to solve the problems! These should help you to check that you understand the regular operations, as well as the definitions of a regular language and the language of a regular expression. They should also help you to solve problems involving regular expressions — including designing a regular expression for a given language, deciding whether a string of symbols *is* a regular expression over an alphabet, and — when given a regular expression — deciding (and proving) whether a given string belongs to the language of this regular expression, and describing its language.

   The supplemental material also includes material that is for interest only. This includes proofs of claims, about relationships between regular operations and regular languages, as well as relationships between regular expressions and regular languages, that were stated in the preparatory material for this lecture. It also includes additional information about algorithms that can be used to solve problems concerning regular expressions, and how regular expressions are used in software.

Problems To Be Solved

Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ —- so that this language does not include any of the symbols

$$\lambda, \emptyset, \Sigma, (,), \cup, \circ, ^\star$$

and let

$$\Sigma_{\text{regexp}} = \Sigma \cup \{\lambda, \emptyset, \text{“}\Sigma\text{”}, (,), \cup, \circ, ^\star\} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \lambda, \emptyset, \text{“}\Sigma\text{”}, (,), \cup, \circ, ^\star\}.$$

Consider the string

$$\omega = ((((\Sigma)^\star \circ \mathtt{a}) \circ (\Sigma)^\star) \circ (\mathtt{a} \circ (\Sigma)^\star)) \in \Sigma^\star_{\text{regexp}}.$$

1. Using the definition of a ***regular expression***, prove that ω is a regular expression over Σ.

   Note: A later problem asks you to produce a ***parse tree*** for ω. Solving that problem, first, might help you to solve this one — but it is also possible that you can solve this problem *without* creating a parse tree.

   Solution:

Here is *another* way to write this.

2. Consider the string $\mu = \texttt{abaca} \in \Sigma^\star$. Use the definition of the language of a regular expression, and the solution for the first problem (as needed), to prove that μ is in the language of ω.

   Solution:

3. Describe (in clear and simple English) the language of the regular expression ω. Was the solution for the first problem helpful, as you tried to solve this one? If it was, then please say how.

The Language of the Regular Expression ω:

Was the Solution for the First Problem Helpful, Here? If so, How?

4. Produce a ***parse tree*** for ω — showing enough work so that a reader can see how you got this.

   Solution:

5. Describe how this parse tree could be used to produce a ***nondeterministic finite automaton*** with the same language as ω.

   Solution:

6. Apply the process that you have described (possibly making simplifications to nondeterministic finite automata, along the way) to produce a nondeterministic finite automaton with the same language as ω.

   Then discuss how this could be useful when you wish to decide whether a given string belongs to the language of ω.

   Note: This is less important than the other problems in this presentation, and you will not be asked to do this on assignments or tests.

7. Once again, let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and let L be the set of all strings in $\Sigma^\star$ that include an even number of copies of the symbol "a" — that is, the language

$$L = \{\mu \in \Sigma^\star \mid \text{the number of copies of "a" in } \mu \text{ is divisible by } 2\}.$$

Design a regular expression, over Σ, whose language is L.

Strategy for the Discovery of a Regular Expression for a Given Language:

Application To This Example: