CSL705: Theory of Computation

Tutorial: Regular Expressions, Right linear grammars and Pumping Lemma for Regular Languages

1. Languages like Perl and tools such as Lex use an extended form of regular expressions which include the following operators (beyond the bare-bones ones that we have discussed in class).
   (a) classes of characters in the form e.g. `[a-z]`, `[0-9]` etc?
   (b) finite iterative operators such as $r\{m, n\}$ which stand for at least m iterations and at most n iterations of r where r is a regular expression.

   (a) Prove that the languages defined by such operators are all regular.
   (b) Express these operators in terms of the bare-bones operators we have defined in class.

2. Let $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \} \cup \{+, -, .\}$.
   (a) Design a regular expression for the language of finite-length, finite-precision signed real numbers (as allowed in a language like Standard ML).
   (b) Design an automaton which will accept only such signed real numbers.
   (c) Design a right-linear grammar to generate all such signed real numbers.

3. A programming language defines an "identifier" as consisting of an alphabetic character followed by zero or more alphanumeric characters. It also defines the following keywords which cannot be used as identifiers: `if`, `then`, `elseif`, `else` and `while`.
   (a) Design individual regular expressions for identifiers, keywords, signed integers and signed real numbers in this programming language.
   (b) Design a NFA which accepts exactly all the identifiers, keywords, signed integers and signed real numbers in this programming language, such that each of these categories of strings end in different accepting states and for all other (unacceptable) strings the automaton has an error state.

4. Let G, G_1 and G_2 be right linear grammars on a non-empty finite alphabet Σ. Define the operations $\cup$, $\cap$, $\sim$, ., * on right-linear grammars such that
   (a) $\mathcal{L}(G_1 \cup G_2) = \mathcal{L}(G_1) \cup \mathcal{L}(G_1)$
   (b) $\mathcal{L}(G_1 \cap G_2) = \mathcal{L}(G_1) \cap \mathcal{L}(G_1)$
   (c) $\mathcal{L}(G_1.G_2) = \mathcal{L}(G_1).\mathcal{L}(G_1)$
   (d) $\mathcal{L}(\sim G) = \sim \mathcal{L}(G)$
   (e) $\mathcal{L}(G^*) = (\mathcal{L}(G))^*$

5. Given that a language $L \subseteq \Sigma^*$ is not regular, does it necessarily imply that its complement is also not regular? Prove or disprove the statement, *The language* $\{a^m b^n \mid m \neq n\}$ *on* $\Sigma = \{a, b\}$ *is regular.*

6. Prove that there exists no DFA which can verify addition on natural numbers. (*Hint* show that the language $\{10^m 10^n 10^{m+n} \mid m, n \geq 0\}$ is not regular.)