Problem Set 3

204213 Theory of Computation

Due: 15 January 2010

1. Using the construction in the proof of the theorem: Every regular language is a FA language, construct finite automata accepting these languages.

   (a) $a^*(ab \cup ba \cup \varepsilon)b^*$

   (b) $((a \cup b)^*(\varepsilon \cup c)^*)^*$

   (c) $((ab)^* \cup (bc)^*)ab$

2. Let L,$L' \subseteq \Sigma^*$. Define the following languages.
   1. Pref(L)=$\{w \in \Sigma^* : x = wy$ for some $x \in L, y \in \Sigma^*\}$ (the set of prefixes of L).
   2. Suf(L)=$\{w \in \Sigma^* : x = yw$ for some $x \in L, y \in \Sigma^*\}$ (the set of suffixes of L).
   3. Max(L)=$\{w \in L$:if $x \neq \varepsilon$ then $wx \notin L\}$.
   Show that if L is accepted by some finite automaton, then so is each of the following.

   (a) Pref(L)

   (b) Suf(L)

   (c) Max(L)

3. Apply the construction in our proof to obtain regular expressions corresponding to each of the finite automata in the book on page 84. Simplify the resulting regular expression as much as you can.

4. Consider the alphabet $\Sigma = \{a, b, (,), \cup, ^*, \emptyset\}$. Construct a context-free grammar that generate all strings in Σ^* that are regular expression over $\{a, b\}$.

5. A program in a real programming language, such as C or Pascal, consists of statements, where each statement is one of several types:

(a) assignment statement, of the form `id` $:= E$, where E is any arithmatic expression (generated by the grammar described in class).

(b) conditional statement, of the form, say, `if` $E < E$ `then statement`, or a `while` statement of the form `while` $E < E$ `do statement`.

(c) compound statement, that is, many statements preceded by a `begin`, followed by an `end`, and seperated by a ";".

Give a context-free grammar that generates all possible statements in the simplified programming language described above.