HW 7 – Regular Expression

CS 421 – Fall 2012
Revision 1.0

Assigned October 23, 2012
Due October 30, 2012, 11:59 pm
Extension 48 hours (20% penalty)

1 Change Log

1.0 Initial Release.

2 Turn-In Procedure

Answer the problem below, save your work as a PDF (either scanned if handwritten or converted from a program), and hand in the PDF. Your file should be named `hw7.pdf`.

3 Objectives and Background

The purpose of this HW is to test your understanding of regular expression and distinguish the differences between regular grammar and other grammar.

4 Problems

1. (20 points) Decide if the following descriptions of languages are regular languages. If they are not, give a small statement and suggest why they are not regular languages. If they are, write **the regular expression of the complement** of the language.

   1. $L_1 = \{0^n 1^m | 0 \leq n \leq m \leq 2n\}$

   Solution: The language cannot be regular, because it cannot be accepted by a deterministic finite state automaton (basically because you have to count the number of 0's and then the number of 1's and finite state automata can't do that.) To see this, suppose we had a deterministic finite state automaton that accepted this language. Let k be the number of states it has. Now, consider the string $0^{k+1}1^{k+1}$. This is a string in L_1. If we start in the DFA's start state and run an execution on the first 0^{k+1}, we must end a state from which we can reach an accepting state in $k+1$ steps, but not in any fewer. However, since we have only k states, every loop-free path has length at most $k-1$. Therefore, if we can reach an accepting state by executing 1^{k+1}, it must contain at least one loop, of length $j \geq 1$. Removing that loop then givens a a path labeled by 1's of length $(k+j)-j < k+1$, and thus we would be able to accept $0^{k+1}1^{(k+1)-j}$, which is not in L_1.

   2. $L_2 = \{w | w \text{ contains at least one } 1\}$ where $\Sigma = \{0, 1\}$

   Solution: 0^*

   3. $L_3 = \{w | w\text{'s length is even}\}$ where $\Sigma = \{0, 1\}$

   Solution: $(0 \vee 1)((00 \vee 01 \vee 10 \vee 11)^*)$

   4. $L_4 = \{w | w \text{ contains even number of } 0 \text{ or } w \text{ contains odd number of } 1\}$ where $\Sigma = \{0, 1\}$

Solution: The complement of L_4 is the set of all strings of 0's and 1's that have an even number of ones and an odd number of 0's. Let

$$
\begin{aligned}
\text{two1s\_even0s} = \quad & ((00)^*1(00)^*1(00)*) \vee (0(00)^*10(00)^*1(00)*) \vee \\
& (0(00)^*1(00)^*10(00)*) \vee ((00)^*10(00)^*10(00)*) \\
\text{two1s\_odd0s} = \quad & (0(00)^*1(00)^*1(00)*) \vee (0(00)^*1(00)^*1(00)*) \vee \\
& (0(00)^*1(00)^*1(00)*) \vee (0(00)^*1(00)^*1(00)*) \\
\text{even1s\_even0s} = \quad & (00)^* \vee (\text{two1s\_even0s})^* \vee \\
& ((\text{two1s\_even0s})^*\text{two1s\_odd0s}(\text{two1s\_even0s})^*\text{two1s\_odd0s}(\text{two1s\_even0s})^*)^*
\end{aligned}
$$

The language we need is given by

$$(\text{even1s\_even0s})^*0(\text{even1s\_even0s})^*$$

.

5. $L_5 = \{0^{n^2} | \text{n is natural number}\}$

Solution: The language cannot be regular, because it cannot be accepted by a deterministic finite state automaton (basically because you have to count the number of 0's). To see this, suppose we had a deterministic finite state automaton that accepted this language. Let k be the number of states it has. If $k = 1$, then either that state is an accepting state, or it is not. If it is, then the automaton accepts all strings of 0's and 1's, or it isn't, in which case it accepts nothing. L_5 is neither of those languages, so we we must have $k > 1$. Now, consider the string 0^{k^2}. Since it is in our language, there must exist a path from the start state to an accepting state lableled by 0's of length k^2. Since $k > 1$, $k^2 > k$. Therefore, the path must contain a loop of length j where $1 \leq j \leq k$. Removing that loop, we then have a path of 0's of length $k^2 - j$. Thus 0^{k^2-j} must also be accepted. But 0^{k^2-j} cannot be in L_5 since $k^2 > k^2 - k > k^2 - (2k-1) = (k-1)^2$.