McMaster University
Department of Computing and Software
Dr. Wolfram Kahl

SE 3E03, Fall 2002
Assignment 3

Design and Selection of Programming Languages

12th September 2002

Problem 10 (Regular Expressions)

Calculate the formal languages denoted by the following regular expressions:

a) $(a \mid bb)b(b \mid aa)$

b) $(a \mid ab \mid aa)(b \mid ab \mid bb)$

c) $(C \mid M)A(C \mid S)$

d) $.(h \mid H)(t \mid T)(m \mid M)(\varepsilon \mid l \mid L)$

Problem 11

Give deterministic finite state automata accepting the languages denoted by the regular expressions in Problem 10.

Problem 12

Give deterministic finite state automata accepting the languages generated by the following regular grammars with start symbol S, uppercase letters as nonterminals, and lowercase letters as terminal symbols:

a) $S \rightarrow a$
$S \rightarrow bS$

b) $S \rightarrow a$ $\quad U \rightarrow cU$
$S \rightarrow bU$ $\quad U \rightarrow bS$

c) $S \rightarrow a$ $\quad U \rightarrow ccU$
$S \rightarrow bcU$ $\quad U \rightarrow bS$

d) $S \rightarrow a$
$S \rightarrow Sb$

e) $S \rightarrow a$ $\quad T \rightarrow Sa$
$S \rightarrow Tb$ $\quad T \rightarrow Tc$

Problem 13 (Java)

For the arithmetic expressions from Problem 9, now present a data structure definition in Java.

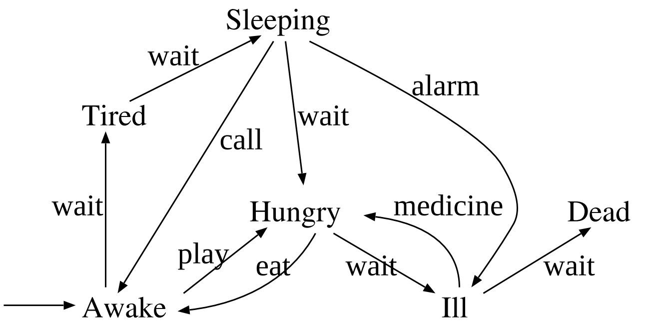
This time, arithmetic expressions may include variables, taken from some variable set $\mathcal{V}$. Evaluation then needs an *environment*, which associates numerical values with some variables. What kind of mathematical data structure is such an environment? How do you represent it in Java?

Carefully design the interface to your data structure.

Document your code using `javadoc` comments.

Problem 14 (Java Applets — Tamagochi)

The goal is to write a Java applet that implements the following finite state machine (or one of similar characteristics):



It should always display the current state of the machine, and accept input either from the keyboard or via buttons, and perform the respective state transitions.

a) What are your options for implementing this single, fixed automaton in a Java program?

b) What would be your options if you wanted to be able to load new automata at runtime?

c) Dsesign, document and implement the finite state machine, and at least a comand-line interface for interaction.

d) **For September 26:** Implement the applet.

In your home directory on the departmental machines, you (should) have a sub-directory `public_html`. If you set the permissions of your home directory, that directory, and its contents appropriately, you can use this for your own WWW presence. Write an HTML page `tamagochi.html` that contains your applet and links to source code and `javadoc` documentation.
(`chmod o+x $HOME $HOME/public_html; chmod 644 $HOME/public_html/tamagochi.html`)