## Problem Set 7

Due: Wednesday, April 1, 2009 at 12:30 in Elaine Wilson office (SC 3229). If the door is locked, slide your solutions under the door.
Version: $\mathbf{1 . 1}$

Submission instructions: Submit each problem on a separate sheet of paper, put your name on each sheet, and write your discussion section time and day (e.g. Tuesday 10am) in the upper righthand corner. These details may sound picky, but they make the huge pile of homeworks much easier to grade quickly and more importantly, since we return them in the discussion sections, easier for you to get them back.

Also, write on each exercise the name/netid of your group members.
(Q1) Building Recursive Automata.
[Category: Construction, Points: 10]
The language of Boolean expressions is defined as the language of the grammar $\mathcal{G}=(\mathcal{V}, \Sigma, \mathcal{R}, \mathrm{B})$ where $\mathcal{V}=\{B\}, \Sigma=\{0,1,(),, \vee, \neg\}$ and $B$ has the following rules

$$
\mathrm{B} \rightarrow 0|1|(\mathrm{B} \vee \mathrm{~B}) \mid(\neg B) .
$$

A Boolean expression evaluates to 0 or 1 in the natural sense: 0 is false, 1 is true, $\vee$ represents the boolean disjunction (or) and $\neg$ represents the boolean negation (not). For example, the expression $(\neg(0 \vee 1))$ evaluates to 0 , and the expression $(\neg((\neg 1) \vee(\neg 1)))$ evaluates to 1 .
(a) Let $L$ be the following language over the alphabet $\Sigma^{\prime}=\Sigma \cup\{=\}$.

$$
L=\{\langle e x p\rangle=b \mid\langle e x p\rangle \text { is a boolean expression that evaluates to } b\} .
$$

(Eg. The word " $(\neg(0 \vee 1))=0$ " is in $L$ but the word $(\neg(0 \vee 1))=1$ is not in $L$.) Construct a recursive automaton for $L$ and briefly describe why it works.
(b) For the string $w=(0 \vee 1)=1$ show the run of your automaton on it, including stack contents at each point of the run (i.e., list the sequence of configurations in the accept trace for $w$ for your RA). Use the formal definition given in the lecture notes.
(Q2) Recursive Automata with Finite Memory.
[Category: Proof, Points: 10]
You are working on a computer, which has a limited stack size of (say) 5. You know this means that you can have a call depth of at most 5 recursive calls.
(a) Argue that the language accepted by any RA on this machine is regular.

More formally, given a RA

$$
\mathrm{M}=\left(\mathcal{M}, \text { main, }\left\{\left(Q_{m}, \Sigma \cup \mathcal{M}, \delta_{m}, q_{0}^{m}, F_{m}\right) \mid m \in \mathcal{M}\right\}\right),
$$

describe an NFA $\mathrm{D}=\left(Q^{\prime}, \Sigma, \delta^{\prime}, q_{0}, F^{\prime}\right)$ that will accept the same language.
(b) What is $Q^{\prime}$ ?
(c) What is $q_{0}$ ? What is $F^{\prime}$ ?
(d) What is $\delta^{\prime}$ ?
(Hint: Think about using configurations of the RA in your construction. See the class notes for the formal definition of what is configuration. It might also be useful for you the review the concept of acceptance for a RA.)

