CSE 431 Spring 2006
Assignment #1

Due: Friday, April 7, 2006

Reminder Sign up for the *cse431@cs* mailing list.

Reading assignment: Read Chapter 3 of Sipser's text.

Problems:

1. Give implementation-level descriptions (not a full formal descriptions) of Turing machines that decide the following languages over the alphabet $\{0, 1\}$:

   (a) $\{x \in \{0, 1\}^* \mid x$ has equal numbers of 0's and 1's$\}$

   (b) $\{0^a 1^b 0^c \mid b > 0$ and $c = a \bmod b\}$.

2. Let k-PDA be a pushdown automaton that has k stacks. Thus a 0-PDA is an NFA and a 1-PDA is a conventional PDA. Show that 2-PDAs are as powerful as Turing machines by simulating a Turing machine tape with two stacks.

3. A Turing machine with a 2-dimensional tape is like a 1-tape TM except that it marked with a 2-dimensional grid of cells that are all blank, except for the input which is given in the cells under the read head and immediately to its right. Additional changes are that

   - the transition function δ, is $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, U, D\}$ where U and D indicate moves *up* and *down* one cell.
   - there is no end of the tape.

   Give an implementation level description of how an ordinary 1-dimensional Turing machine can simulate a 2-dimensional one.

4. Show that both the Turing-recognizable languages and the decidable languages are closed under concatenation.

5. (Bonus) A *queue automaton* is like a push-down automaton except that the stack is replaced by a queue. A *queue* is a tape allowing symbols to be written only on the left-hand end and read only at the right-hand end. Each write operation (we'll call it a *push*) adds a symbol to the left-hand end of the queue and each read operation (we'll call it a *pull*) reads and removes a symbol at the right-hand end. As with a PDA the input is place on a separate read-only input tapes, and the head on the input tape can move only from left to right. The input tape contains a cell with a blank symbol following the input, so that the end of the input can be detected. A queue automaton accepts its input by entering a special accept state at any time. Show that a language can be recognized by a deterministic queue automaton iff the language is Turing-recognizable.