Homework #3: Turing Machines
Due: see Canvas

Read Chapters 3 and 4. You do not need to solve all these problems, see Canvas for details.

Written Problems

Problem 1 Give a (Sipser style) Turing machine M, describe how to convert it into a Turing machine M' so that $L(M) = L(M')$, but M' only halts with an empty tape. In other words, M' makes its tape entirely blank before halting. Beware that M might write blanks in the middle of its tape, and you may need some way to detect the left end of the tape.

Problem 2 Suppose E is an enumerator for language L, as defined in Chapter 3. Furthermore, suppose E prints strings in length order. That is, if E prints string x before printing string y, then $|x| \leq |y|$. E may repeat some strings, and it may print strings of the same length in arbitrary order. Show that L is decidable. (Hint: you might first handle the case that L is finite.)

Problem 3 Suppose M is a Turing machine, and every transition either moves right (R) or stays put (S), but no transition moves left (L). Argue that $L(M)$ is regular. Note these complications: M may halt before reading the entire input, it may take further steps when it sees blanks, and sometimes it may not halt at all.

Problem 4 Argue that if language A is recognizable, then so is language A^*. Don't use nondeterminism in your argument. (You may use multiple tapes, and you may use an "English level" TM description in the style of Sipser.)

Problem 5 Given a standard Turing machine M, argue that it can be simulated by a *FIFO queue automaton* S, as described in Problem 3.14. Your S should be deterministic.

Problem 6 Given two languages $A, B \subseteq \Sigma^*$, define their "quotient" as the language $A/B = \{x \in \Sigma^* : \exists y \in B, xy \in A\}$. That is, it is the set of strings x, such that for some string y in B, their concatenation xy is in A.

Problem 6(a). Show that if A is regular, then so is A/B. (Hint: use a DFA, modify F.)

Problem 6(b). Show that if A and B are recognizable, then so is A/B.

Problem 7 Let A and B be recognizable languages such that $A \cup B = \Sigma^*$. Show there exists a decidable language C such that $\bar{B} \subseteq C \subseteq A$. (Hint: review the proof of Theorem 4.22, which handles the special case when A and B are disjoint. It is hard to define C first, in this problem. Instead, look for its decider. That is, a TM which accepts every input string in $A - B$, rejects every input string in $B - A$, and may either accept or reject for input strings in $A \cap B$.)

JFLAP Problems

For these problems you need to submit a JFLAP Turing machine as a "jff" file. We will use JFLAP 7.1, a Java application (a "jar" file) which you may download here:

`http://cs.emory.edu/~cs424001/share/jflap/`

Each Turing machine should be deterministic and 1-tape. Note Canvas allows you to submit multiple files for an assignment.

Problem 8 (ww.jff) Design a Turing machine deciding the language $\{ww : w \in \{a,b\}^*\}$. That is: the input alphabet is $\{a,b\}$, and an input string is accepted iff it has even length and its first half equals its second half. Furthermore, to make this a bit more challenging, the tape should contain the original input string (and nothing else) when the machine halts.

Problem 9 (substring.jff) Design a Turing machine deciding the language $\{x\#y : x, y \in \{a,b\}^*$ and x is a substring of $y\}$. For example $aa\#baab$ is in the language, $aa\#abab$ is not. Note the input alphabet is $\{a,b,\#\}$, and the # simply acts as a separator.

Problem 10 (square.jff) Design a Turing machine deciding the language $\{1^n : n$ is a square$\} = \{\varepsilon, 1, 1111, 111111111, \ldots\}$. It may help to use the fact that each square is a sum of consecutive odd integers, for example 9=1+3+5. Note the input alphabet is simply $\{1\}$.