CS 581: Theory of Computation
Mid-term exam

James Hook

November 1 – 8, 2005

This is a take-home exam. You may use only class notes and the text. You are not to use any other materials. You are not to discuss the exam with anyone other than the instructor during the examination period. The examination period spans from 1 to 8 November.

I will endeavor to check email regularly during the examination period.

Please return the exam within 48 hours of receipt. Please indicate the time, date and mode of checkout and the time, date and mode of submission on your exam cover-sheet.

There are five problems. All problems have equal weight.

1. Let the concatenation of strings s and t be denoted by $s \cdot t$. Let the reversal of string s be denoted by s^R. For any regular language L, let the language $pal(L)$ be the language of *palindromes* of L. That is

   $$pal(L) = \{s \cdot s^R | s \in L\}$$

   Are the regular languages closed under *pal*? Sketch a construction or a disproof.

2. Not context free

   Sipser 2.31

   Let B be the language of all palindromes over $\{0, 1\}$ containing an equal number of 0s and 1s. Show that B is not context free.

3. Queue automaton

   Sipser 3.14

   A *queue automaton* is like a push-down automaton except that the stack is replaced by a queue. A *queue* is a type allowing symbols to be written only on the left-hand end and read only at the right-hand end. Each write operation (called a *push*) adds a symbol to the left-hand end of the queue and each read operation (called a *pull*) reads and removes a symbol at the right-hand end. As with a PDA, the input is placed on a separate read-only input tape, 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.

4. Regular expression containment decidable

   Sipser 4.12

   Let $A = \{\langle R, S\rangle | R$ and S are regular expressions and $L(R) \subseteq L(S)\}$. Show that A is decidable.

5. Turing-recognizable list of deciders is incomplete.

   Sipser 4.28

   Let A be a Turing-recognizable language consisting of descriptions of Turing machines, $\{\langle M_1\rangle, \langle M_2\rangle, \ldots\}$, where every M_i is a decider. Prove that some decidable language D is not decided by any decider M_i whose description appears in A. (Hint: You may find it helpful to consider an enumerator for A.)