Problem Set 5

Fall 09

Due: Thursday Dec 3 at 11:00 AM in class (i.e., Room 103 Talbot Lab)
Please follow the homework format guidelines posted on the class web page:
`http://www.cs.uiuc.edu/class/fa09/cs373/`

1. [**Points**: 15]

   (a) For each of the following PCP problems, either find a solution or prove that a solution does not exist.

   i.
   $$\begin{bmatrix} 11 \\ 101 \end{bmatrix} \begin{bmatrix} 11 \\ 11011 \end{bmatrix} \begin{bmatrix} 110 \\ 1 \end{bmatrix}$$

   ii.
   $$\begin{bmatrix} 10 \\ 1 \end{bmatrix} \begin{bmatrix} 10 \\ 01 \end{bmatrix} \begin{bmatrix} 01 \\ 10 \end{bmatrix} \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

   iii.
   $$\begin{bmatrix} 1110 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0111 \end{bmatrix}$$

   (b) Prove that PCP is undecidable even if we restrict its alphabet to two symbols, for example $\Sigma = \{0, 1\}$.

   (c) Prove that PCP is decidable if we restrict its alphabet to one symbol, for example $\Sigma = \{0\}$.

2. [**Points**: 15]

   Consider the alphabet $\Sigma = \{a, b, c, d\}$. Let $R = \{(ab, \varepsilon), (ba, \varepsilon), (cd, \varepsilon), (dc, \varepsilon)\}$. Starting with a string $w \in \Sigma^*$, we can convert w to some other string w' by applying the following rule:
   - If $(x, y) \in R$ or $(y, x) \in R$, and x is a substring of w, i.e. $w = w_1 x w_2$, then $w' = w_1 y w_2$.

   The single player *Group Game* is started by some string w in Σ^*. At each round the player changes the string by applying the above rule. The game ends after a finite number of rounds. We say the pair of strings (w, w') is a *good pair* if, when the player starts the game with string w, he can finish it with string w'.

   Given the pair (w, w') as input, is it decidable whether it is a good pair? Prove your answer.

3. [**Points**: 10]

   Let $A_1 \subset \{0,1\}^*$ and $A_2 \subset \{0,1\}^*$ be Turing-recognizable languages such that $A_1 \cup A_2 = \{0,1\}^*$ and $A_1 \cap A_2 \neq \emptyset$. Prove that $A_1 \leq_m (A_1 \cap A_2)$, where $A \leq_m B$ means that language A is mapping-reducible to language B.

4. [**Points**: 15] Let A be Turing-recognizable, but not Turing-decidable. Consider $A' = \{0w \mid w \in A\} \cup \{1w \mid w \notin A\}$. For both A' and its complement, are they Turing-decidable or Turing-recognizable? Prove your point.

5. [**Points**: 10] Determine whether each language is decidable and prove your answer without using Rice's theorem.

   (a) $A = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$

   (b) $A = \{\langle M \rangle \mid$ given a blank input$M$ uses at most 20 cells of tape $\}$

   (c) $A = \{\langle M \rangle \mid L(M)$ is finite and Turing-decidable $\}$

   (d) $A = \{\langle M \rangle \mid L(M)$ contains some string that is a palindrome $\}$

6. [**Points**: 15] For each of the following either prove that the language is undecidable or present an algorithm to solve the problem, that halts on all inputs. M_i is the i-th Turing Machine in some fixed enumeration of all Turing Machines.

   (a) Given i and w, decide whether M_i only writes to the cells initially occupied by w

   (b) Given i and w, decide whether M_i ever moves its head to the left on input w

   (c) Given i and j, decide whether $L(M_i) = L(M_j)^c$, where A^c is the complement of A

7. [**Points**: 20] Consider the following Register Machine model which has:

   (a) A finite number of (arithmetic) registers: $R0, R1, R2, \ldots$

   (b) An infinite number of available memory locations (each location can be accessed by its index)

   (c) An instruction set (defined below)

   The contents of each register and memory location is a non-negative integer not bigger than K, for some fixed integer K. The instruction set is:

   (a) $ADD\ Rx, Ry, Rz$ – add contents of reg Rx and reg Ry, putting result into Rz

   (b) $LOADC\ Rx, NUM$ – place constant NUM into reg Rx

   (c) $LOAD\ Rx, M$ – put contents of memory location M into Rx

   (d) $LOADI\ Rx, M$ – (load indirect) put $value(value(M))$ into Rx

   (e) $STORE\ Rx, M$ – store contents of Rx into location M

   (f) $JUMP\ Rx, M$ – if value of Rx is 0 then jump to instruction at location M, else continue normal execution

   (g) $HALT$ – halt the execution

   The machine starts with program instructions written in contiguous block of memory, starting at location 0; a string w written in a block of memory at location M; the value of register $R0$ is M, all other registers are set to 0.
   The machine begins by executing instruction at memory location 0. After executing current instruction and modifying the contents of registers and memory, the machine proceeds to the next instruction. The next instruction is the adjacent instruction in the next memory location (with bigger index), or the instruction located at the memory address specified in the second parameter of $JUMP$ instruction, if $JUMP$ was executed.
   Note that program is located in memory and thus machine can even alter itself during execution.

The Register Machine M accepts the string w, if it halts with a non-zero value in register $R0$. Prove that this machine is equivalent to a Turing machine. That is, a language is Turing recognizable if and only if there exists a Register Machine, as defined above, that can recognize it.