6.045J/18.400J: Automata, Computability and Complexity T.A. Matthew Lepinski

Recitation 5: Turing Machines

March 6, 2003 *Discussion Problems*

Problem 1: Define the following words, phrases and symbols.

1. Turing Machine
2. $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$
3. (Turing) Recognizable Language
4. (Turing) Decidable Language
5. (Recursively) Enumerable Language

Problem 2: Each of the following statements is either True or False.

1. A Turing Machine has a single start state, but may have many Accept states.
2. Unlike a Finite State Machine, a Turing Machine does not Halt when it reaches the end of it's input.
3. It is not possible to make a Turing Machine with only a single state.
4. The Tape alphabet, Γ, must be different than the input alphabet, Σ.
5. The blank, , is a special symbol which is used to mark the end of a Turing Machine's input. A Turing Machine can read the blank symbol, but cannot write it to the tape.
6. The Turing Machine's Read/Write Head can *never* be in the same position during two consecutive steps in the computation.

Problem 3: (Sipser 3.7) It is stated in the text that Hilbert's 10^{th} Problem (determining if a multivariable polynomial has an integer root) is undecidable. Here is a "pseudocode" description of an algorithm that takes in a polynomial, p, as input and determines if p has an integer root.

- For each possible setting of the variables, $x_1, x_2, \ldots, x_k$ to integer values do:
  1. Evaluate p on $x_1, x_2, \ldots, x_k$.
  2. If p evaluates to 0, then accept, otherwise go to the next setting of $x_1, x_2, \ldots, x_k$.
- If none of the integer settings of $x_1, x_2, \ldots, x_k$ caused you to accept, then reject.

Why does this algorithm not contradict the fact that Hilbert's 10^{th} Problem is undecidable.

Problem 4: Make a Finite State Control for a Turing Machine that recognizes the language:

$$L = \{s \in \{0,1\}^* | \text{s contains more zeros than ones}\}$$

Additionally, define Σ and Γ for your Turing Machine.

Problem 5: (Sipser 3.16) Show that a language is decidable if and only if there is exists an enumerator enumerates the language in lexicographic order.