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

Recitation 6: Decidability and Undecidability

March 13, 2003 *Discussion Problems*

Problem 1: A Post Tag Machine is a simple machine which uses rules to manipulate a single string. We define a Post Tag Machine to be a 5-tuple,

$$(\Sigma, \Gamma, \delta, F_{acc})$$

where Σ is the input alphabet, Γ is the string alphabet (where $\Sigma \cup \{\star, \#\} \subset \Gamma$), δ is a set of rewrite rules, F_{acc} is the accept string and F_{rej} is the reject string. The rewrite rules in δ are of the form $x \rightarrow y$ where x and y are both strings in Γ^*. On input, w, the Post Tag Machine operates as follows:

- Start with the string $\star w\#$ and uses the rewrite rules in δ as follows:
- For each rewrite rule, $x \rightarrow y$, if a prefix of the machine's matches x, then remove x from the string and append y to the end of the string.
- Repeat until either:
  - The machine's string matches the string F_{acc}. In this case the machine is said to accept w.
  - The machien reaches a state in which it cannot legally apply a rewrite rule. In this case the machine is said to reject w.

Show that a Post Tag Machine is equivalent to a Turing Machine.

Problem 2: Show that the following language is decidable:

$$E_{DFA} = \{< M > | M \text{ is a DFA which accepts no strings}\}$$

Problem 3: Show that the following language is decidable:

$$EQ_{REGEXP} = \{< M, N > | M \text{ and } N \text{ are regular expressions such that } L(M) = L(N)\}$$

Problem 4: (Sipser 4.10) Show that the following language is decidable:

$$A = \{< M > | M \text{ is a DFA which doesn't accept any string with an odd number of 1s}$$

Problem 5: Show that the following language is undecidable:

$$E_{TM} = \{< M > | M \text{ is a TM which accepts no strings}\}$$

Problem 6: Show that the following language is undecidable:

$$EQ_{TM} = \{< M, N > | M \text{ and } N \text{ are TMs such that } L(M) = L(N)\}$$

Problem 7: Show that the following language is undecidable:

$$REGULAR_{TM} = \{< M > | M \text{ is a TM which recognizes a regular language}\}$$