Some Practice Problems about Turing Machines and Decidability

1. Design a Turing machine whose language is \( \{ ww \mid w \in \{a,b,c\}^* \} \).

2. Design a Turing machine that computes the function \( f(m,n) = 2m + 3n \).

3. Design a deterministic TM for the subset of \( 1^* \) whose strings have contain a prime number of ones. (This is Linz, 3rd edition, p.243, problem 5(e).)

4. Design a non-deterministic TM for the subset of \( 1^* \) whose strings have a composite (i.e., non-prime) number of ones. (Linz, 3rd edition, p.266, problem 6.)

5. (a) Are the recursive languages closed under union? Prove it. (b) What about the recursively enumerable languages?

6. (a) Are the recursive languages closed under complement? Prove it. (b) What about the recursively enumerable languages?

7. (a) Are the recursive languages closed under intersection? Prove it. (b) What about the recursively enumerable languages?

8. Show that it is partially decidable, but not decidable, whether a given TM accepts the string \( 11111 \). (Alternatively, show that the set of (encodings of) TM's whose languages include \( 11111 \) is recursively enumerable, but not recursive.)

9. Show that it is not decidable whether two TM's accept the same language; that is, there is no always-halting TM that, when presented with the encodings for arbitrary Turing machines, \( M \) and \( M' \), can determine whether \( L(M) = L(M') \). (Linz (2nd ed.), p. 321, problem 6.)

10. Show that the question of whether an arbitrary Turing machine accepts at least two strings is partially decidable.

Revised: 2014-12-03