CIT3130 Theory of Computation

Tutorial problems: Turing machines

(All references are to IALC by Hopcroft et al.)

1. Modify the proof (Section 8.1) that the hello-world problem is undecidable to provide a direct proof that the halting problem is undecidable.

2. Prove that the following two problems are undecidable by reduction from either the hello-world problem or the halting problem.
   (a) Given a program and an input, does the program ever produce any output?
   (b) Given two C programs and some input, do the two programs produce the same output for the given input?

3. Construct a Turing machine to recognise the language \( L = \{ a^n b^n c^n \mid n \geq 0 \} \).

4. Construct a Turing machine to compute the sum of integers \( m \) and \( n \). The input should have the form \( 1^{m+1}01^{n+1} \), with the head on the leftmost 1, and the output should have the form \( 1^{m+n+1} \), with the head again on the leftmost 1.

5. Construct a Turing machine to compute Ackermann’s function, defined by \( A(0, n) = n + 1, A(m + 1, 0) = A(m, 1), A(m + 1, n + 1) = A(m, A(m + 1, n)) \). The input and output should have the same form as in the previous question.

6. Prove that every context-free language is recursively enumerable. That is, prove that every language accepted by some PDA is also accepted by some TM.

7. Exercises 8.4.2 and 8.4.3.

8. (a) Construct a 2-stack machine to accept the language \( L_{ab} = \{ a^n b^n c^n \mid n \geq 0 \} \).
   (b) Construct a 2-stack machine to accept the language \( L_{ww} = \{ ww \mid w \in \{a, b\}^* \} \).

9. Define a “queue machine”. Prove that if a language is accepted by some TM, and hence by some 2-stack machine, then it is also accepted by some queue machine. **Hint** There is a standard representation of a queue using two stacks with \( O(1) \) amortised time for put and get operations. For this exercise we need to be able to represent two stacks using a single queue.

10. Exercises 9.1.1 and 9.1.2.

11. Exercise 9.2.1 to 9.2.6.