Tutorial problems: Turing machines, decidability and undecidability

1. Give implementation level and formal descriptions of a Turing machine to recognise the language \( L = \{ a^n b^n c^n \mid n \geq 0 \} \). You may assume either a single- or doubly-infinite tape.

2. Give implementation level and formal descriptions of 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.

3. Give an implementation level description of 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. (Difficult)

4. (Hopcroft et al., Exercise 8.4.3) Give implementation level descriptions of nondeterministic Turing machines — possibly a multitape machines — that recognise the following languages. Try to exploit nondeterminism to avoid iteration and keep computations short and your Turing machines small.
   (a) The set of all strings of \( a \)'s and \( b \)'s that contain two nonoverlapping occurrences of some substring of length 100, i.e.,
      \[ L = \{ wxyxz \in \{a, b\}^* \mid |x| = 100 \} \]
   (b) The language of all strings of the form \( w_1 \# w_2 \# \cdots \# w_n \), for any \( n \geq 0 \), such that each \( w_i \in \{0, 1\}^+ \) and, for some \( j \), \( w_j \) is the binary representation of the integer \( j \).

5. Prove that every context-free language is Turing-recognisable. That is, prove that every language accepted by some PDA is also accepted by some Turing machine.

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

7. Prove that every Turing-machine (with a douby-infinite tape) can be simulated by a 2-stack PDA.

8. Define a “queue machine”. Prove that if a language is accepted by some TM, and hence by some 2-stack PDA, 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.

9. Give a detailed encoding of arbitrary Turing machines in some fixed alphabet.

10. Give a detailed implementation level description of a universal Turing machine \( U \). The machine \( U \) takes as input an encoding (previous question) of a Turing machine \( M \) and a string \( w \). It should simulate the behaviour of \( M \) when started with \( w \) on its input tape, and accept (resp., reject) iff \( M \) applied to \( w \) accepts (resp., rejects).
**Hint.** Use a 3-tape machine. Store \langle M, w \rangle on the first tape, store the current tape of \( M \) on the second tape, and store the current state of \( M \) on the third tape.

11. (Sipser, Problem 4.12) Show that the language

\[
A = \{ \langle R, S \rangle \mid R \text{ and } S \text{ are regular expressions and } L(R) \subseteq L(S) \}\}
\]

is decidable.

12. (Sipser, Problem 4.13) Let \( \Sigma = \{0, 1\} \). Show that the language

\[
B = \{ \langle G \rangle \mid G \text{ is a CFG over } \Sigma \text{ and } 1^* \cap L(G) \neq \emptyset \}
\]

is decidable.

13. (Sipser, Problem 4.14) Let \( \Sigma = \{0, 1\} \). Show that the language

\[
C = \{ \langle G \rangle \mid G \text{ is a CFG over } \Sigma \text{ and } 1^* \subseteq L(G) \}
\]

is decidable.

14. (Sipser, Problem 4.24) Show that the language

\[
\text{PALDFA} = \{ \langle M \rangle \mid M \text{ is a DFA that accepts some palindrome } \}
\]

is decidable. **Hint.** Use theorems about CFLs.

15. (Sipser, Problem 5.9) Show that the language

\[
T = \{ \langle M \rangle \mid M \text{ is a TM that accepts } w^R \text{ iff it accepts } w \}
\]

is undecidable.

16. (Sipser, Problems 5.17) Show that the Post Correspondence Problem is is decidable over the unary alphabet \( \Sigma = \{1\} \).

17. (Sipser, 5.20) Prove that there exists an undecidable subset of \( \{1\} \).

18. (Sipser, 5.29) Show that both conditions of Rice’s Theorem are necessary for proving that the property \( P \) to be undecidable.

19. (Sipser, 5.33) Let

\[
S = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) = \{ \langle M \rangle \} \}
\]

Show that neither \( S \) nor \( \overline{S} \) is Turing-recognisable.

20. (Hopcroft et al., Exercise 9.3.4) We know by Rice’s theorem that none of the following problems are decidable. However, are they Turing-recognisable or not?

   (a) Does \( L(M) \) contain at least two strings?
   (b) Is \( L(M) \) infinite?
   (c) Is \( L(M) \) a context-free language?
   (d) Is \( L(M) = (L(M))^R \)?