The main reference for this problem set is Michael Sipser's Theory of Computation [2]. However, the first problem comes from Douglas Hofstadter's book *Gödel, Escher, Bach* [1].

1. Take the time to try to solve this problem yourself. The problem concerns strings of characters. The only characters allowed/available are M, U and I.

   You start with the string MI and the goal is to use the four rules below to convert it into the string MU. In the rules, the letter x denotes any string. You can use the rules any number of times and in any order, just so long as you get to MU.

| | Rule | Example | Explanation |
|---|---|---|---|
| 1 | xI to xIU | MI to MIU | Append U to the end of a string ending in I. |
| 2 | Mx to Mxx | MIU to MIUIU | Double the string after the M. |
| 3 | xIIIy to xUy | MUIIIU to MUUU | Replace any III with a U. |
| 4 | xUUy to xy | MUUU to MU | Remove any UU. |

2. Explain of the following terms in the context of complexity theory.
   (a) set
   (b) tuple
   (c) language
   (d) decision problem
   (e) indicator function
   (f) map
   (g) function
   (h) one-to-one map
   (i) onto map
   (j) bijection
   (k) invertible map
   (l) one-way function
3. Clearly define the following sets.
   (a) PRIMES
   (b) SAT
   (c) 3SAT
   (d) SUBSETSUM
   (e) HAMILTONIANPATH
4. Describe two different algorithms the check if a number is a prime. The algorithms should accept a single positive integer as input, and output true if the number is prime and false otherwise.

5. Determine which of the following are in PRIMES (without Google).
   (a) 2
   (b) 3
   (c) 4
   (d) 10
   (e) 11
   (f) 13,109
   (g) 100,827
   (h) 102,203
6. Explain what a decision problem is, and how decision problems relate to Turing machines.
7. Explain the decision problem related to PRIMES.
8. Determinethe following expressions are in SAT and 3SAT.
   (a) $a \vee b$
   (b) $a \wedge b$
   (c) $((a \wedge b) \vee (\neg b \wedge c)) \vee \neg d$
   (d) $(a \wedge b) \vee (c \wedge d)$
   (e) $(a \vee b) \wedge (c \vee d)$
9. Explain the concept of complexity in terms of Turing machines.
10. Explain why if we can solve decision problem $A$ in polynomial time, and we can convert decision problem $B$ to problem $A$ in polynomial time, then we can solve problem $B$ in polynomial time too.
11. Explain what the P computational complexity class is, and give an example of a problem known to be in P.
12. Explain what the NP computational complexity class is, and give an example of a problem known to be in NP.
13. Explain what the NP-hard computational complexity class is, and give an example of a problem known to be in NP-hard.
14. Explain what the NP-complete computational complexity class is, and give an example of a problem known to be in NP-complete.
15. Explain how and where passwords are stored on a typical Linux system, and outline the authentication mechanism used for logins.
16. Explain what the inputs and outputs of the SHA256 algorithm look like.
17. Prove that 3-SAT is NP-complete. You may assume that SAT is NP-complete.

18. Consider the following Turing Machine.

| State | Input | Write | Move | Next |
|---|---|---|---|---|
| $q_0$ | $\sqcup$ | $\sqcup$ | L | $q_a$ |
| $q_0$ | 0 | 0 | R | $q_0$ |
| $q_0$ | 1 | 1 | R | $q_1$ |
| $q_1$ | $\sqcup$ | $\sqcup$ | L | $q_f$ |
| $q_1$ | 0 | 0 | R | $q_1$ |
| $q_1$ | 1 | 1 | R | $q_0$ |

Determine what happens when the Turing Machine is run with the following inputs initially on the tape.

(a) 0001

(b) 0111

(c) 0110

(d) 0101010001

(e) 00000000000000111

(f) 00

(g) $\epsilon$

19. Give the state table for a Turing Machine that appends a parity bit to a tape with a string of consecutive 0's and 1's.

20. Construct a Turing Machine to compute the sequence $0 \sqcup 1 \sqcup 0 \sqcup 1 \sqcup 0 \sqcup \ldots$, that is, 0 blank 1 blank 0 blank, etc [3].

21. Give the state table for a Turing Machine that multiplies a string of consecutive 0's and 1's by 2. The machine should treat the initial contents of the tape as a natural number written in binary form, with the least significant bit at the end. That is, if the contents of the tape are 01101, then the right-most 1 represents the number 1, the middle 1 represents the number 4 and the left-most 1 represents the number 8. Then the number on the tape is $8 + 4 + 1 = 13$.

22. Give the state table for a Turing Machine that multiplies a string of consecutive 0's and 1's by 2. The machine should treat the initial contents of the tape as a natural number written in binary form, with the most significant bit at the end. That is, if the contents of the tape are 01101, then the right-most 1 represents the number 16, the middle 1 represents the number 4 and the left-most 1 represents the number 2. Then the number of the tape is $2 + 4 + 16 = 22$.

23. Give the state table for a Turing Machine that adds 1 to a string of consecutive 0's and 1's, where the least significant digit is on the right of the input.

24. Give the state table for a Turing Machine that subtracts 1 to a string of consecutive 0's and 1's, where the least significant digit is on the right of the input.

25. List all words of length at most three in $\Sigma^*$ where $\Sigma$ is:
    (a) $\{0, 1\}$
    (b) $\{a, b, c\}$
    (c) $\{\}$
26. Design a Turing machine to recognise the language $\{0^n 1^n \mid n \geq 0\}$.
27. Design a Turing machine to recognise the language $\{ww^R \mid w \in \{0, 1\}^*\}$ where $w^R$ is $w$ reversed. For example, when $w = 101011$ then $w^R = 110101$.
28. Design a Turing machine to recognise the language $\{a^i b^j c^k \mid i, j, k \in \mathbb{N}_0\}$

## References

[1] Douglas R. Hofstadter. *Godel, Escher, Bach: An Eternal Golden Braid.* Basic Books, Inc., New York, NY, USA, 1979.

[2] Michael Sipser. *Introduction to the Theory of Computation.* International Thomson Publishing, 3rd edition, 1996.

[3] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.