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

   Solution: As per notes.

3. Clearly define the following sets.
   (a) PRIMES
   (b) SAT
   (c) 3SAT
   (d) SUBSETSUM
   (e) HAMILTONIANPATH

Solution: PRIMES is the set of all prime numbers, a subset of the natural numbers, etc.

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.

Solution:

```
public static boolean is_prime_brute(int n) {
  for (int i = 2; i < n; i++)
    if (n % i == 0)
      return false;
  return true;
}

public static boolean is_prime_sqrt(int n) {
  for (int i = 2; i < Math.sqrt(n); i++)
    if (n % i == 0)
      return false;
  return true;
}
```

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

Solution:
(a) Yes
(b) Yes
(c) No
(d) No
(e) Yes
(f) Yes
(g) No
(h) Yes

6. Explain what a decision problem is, and how decision problems relate to Turing machines.

Solution: A decision problem is a problem where the answer takes on one of two values, typically true or false. For convenience, we usually represent true by 1 and false by 0. We also usually encode instances of the problem in binary. In that case, a decision problem f has the following form.

$$f : \{0,1\}^n \to \{0,1\}$$

Decision problems are studied extensively in the context of Turing machines. For a given input, a given Turing machine can end up in three different scenarios. It can end in the accept state, it can end in the fail state, or it can end up in an infinite loop. Turing machines that never end in an infinite loop are said to decide the language they accept. In this context, accept represents an outcome of 1, and fail represents an outcome of 0 in the notation above.

7. Explain the decision problem related to PRIMES.

Solution: PRIMES is the set of all prime numbers.

$$\text{PRIMES} = \{i \mid i \in \mathbb{N}, i \text{ is prime}\}$$

The decision problem related to primes is the function f where:

$$f(i) : \mathbb{N} \to \{0, 1\} : \begin{cases} 1 & \text{if } i \text{ is prime} \\ 0 & \text{otherwise} \end{cases}$$

So, the decision problem takes as input any natural number. It outputs 1 if that number is a prime and 0 otherwise.

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) $(a \vee b) \wedge (c \vee d)$

Solution:
   (a) $(a, b) = (1, 1)$
   (b) $(a, b) = (1, 1)$
   (c) $(a, b, c, d) = (1, 1, 1, 0)$
   (d) $(a, b, c, d) = (1, 1, 1, 0)$
   (e) $(a, b, c, d) = (1, 1, 1, 0)$

9. Explain the concept of complexity in terms of Turing machines.

Solution: A Turing machine essentially consists of a (linear) tape divided into cells, and a state table. Each cell contains a single symbol. At any given point in time the machine is in a single state, and is reading the symbol in a single cell. The state table gives the rules as to what to do for the current symbol being read and the current state the machine is in. The state table will indicate a symbol to overwrite the symbol in the current cell, the neighbouring cell to which the machine should move next, and the next state the machine should enter.

Let n be the number of cells on the tape that do not initially (before the machine runs) contain the blank symbol. Call this the size of input. Now consider the

language that the Turing machine accepts, and call it L. This is the set of all possible inputs to the Turing machine in which it ends in the accept state.

For each element of L, consider the number of times we have to look up the state table before the initial input is accepted. Pick the worst of these, and call that the *worst case*. In general, L is an infinite set, so we must express the worst case in terms of n. Let $f(n)$ be the function of n expressing the number of look-ups in the worst case. The complexity of the Turing machine is $O(f(n))$.

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.

Solution: Let $f(m)$ be the number of steps it takes in the worst case to solve an input of length m for A. Let $g(n)$ be the number of steps it takes in the worst case to convert an instance of B to A. Then in the worst case, an algorithm that converts B to A and then solves A takes at most $f(m) + g(n)$ steps to complete. So, the algorithm is $O(f(m) + g(n))$.

Now, we know that f and g are polynomials, because we were told they are in the question. However, f is a polynomial in m and g in n. Can we express n in terms of a polynomial in m? We must be able to, because a Turing machine that works in polynomial time will always halt with a tape output that is of length a polynomial in the length of the input.

11. Explain what the P computational complexity class is, and give an example of a problem known to be in P.

Solution: An algorithm is said to be solvable in *polynomial time* if the number of steps required to complete the algorithm for a given input is $O(n^k)$ for some nonnegative integer k, where n is the size of the algorithm's input.

In 2002, Manindra Agrawal, Neeraj Kayal, and Nitin Saxena proved, in a paper called *PRIMES is in P*, that the problem of determining whether a given natural number was a prime or not is in P. Their algorithm is often referred to as the AKS primality test.

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.

Solution: The set 3-SAT is a subset of SAT, and SAT is in NP. Thus, 3-SAT is in NP.

We can convert an instance of SAT to an instance of 3-SAT in the following way. First convert the formula to CNF. Then convert any one-literal clause a of the SAT formula to the following 3-SAT formula:

$$(a \vee u_1 \vee u_2) \wedge (a \vee u_1 \vee \neg u_2) \wedge (a \vee \neg u_1 \vee u_2) \wedge (a \vee \neg u_1 \vee \neg u_2).$$

Convert any two-variable clause $a \vee b$ to:

$$(a \vee b \vee u_1) \wedge (a \vee b \vee \neg u_1)).$$

Any three-variable clause can be left as-is. Convert any clause with more than three variables $a \vee b \vee c \vee \ldots$ to:

$$(a \vee b \vee u_1) \wedge (c \vee \neg u_1 \vee u_2)) \wedge \ldots \wedge (i \vee \neg u_{n-4} \vee u_{n-3}) \wedge (j \vee k \vee \neg u_{n-3}).$$

Now the formula is in 3-CNF. This reduction is polynomial in time, because the output is at most three times as long as the input.

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) ϵ

Solution:

(a) Fail

(b) Fail

(c) Accept

(d) Accept

(e) Fail

(f) Accept

(g) Accept

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.

Solution:

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

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].

Solution:

State	Input	Write	Move	Next
q_0	$\sqcup$	0	R	q_1
q_0	0	0	R	q_f
q_0	1	1	R	q_f
q_1	$\sqcup$	$\sqcup$	R	q_2
q_1	0	0	R	q_f
q_1	1	1	R	q_f
q_2	$\sqcup$	1	R	q_3
q_2	0	0	R	q_f
q_2	1	1	R	q_f
q_3	$\sqcup$	$\sqcup$	R	q_0
q_3	0	0	R	q_f
q_3	1	1	R	q_f

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 $8 + 4 + 1 = 13$.

Solution:

State	Input	Write	Move	Next
q_0	$\sqcup$	0	R	q_a
q_0	0	0	R	q_0
q_0	1	1	R	q_0

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 $2 + 4 + 16 = 22$.

Solution:

State	Input	Write	Move	Next
q_0	$\sqcup$	$\sqcup$	L	q_1
q_0	0	0	R	q_0
q_0	1	1	R	q_0
q_1	$\sqcup$	$\sqcup$	R	q_4
q_1	0	$\sqcup$	R	q_2
q_1	1	$\sqcup$	R	q_3
q_2	$\sqcup$	0	L	q_0
q_2	0	0	R	q_f
q_2	1	1	R	q_f
q_3	$\sqcup$	1	L	q_0
q_3	0	0	R	q_f
q_3	1	1	R	q_f
q_4	$\sqcup$	0	R	q_a
q_4	0	0	R	q_f
q_4	1	1	R	q_f

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.

Solution:

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

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.

Solution:

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

25. List all words of length at most three in Σ^* where Σ is:
    (a) $\{0, 1\}$
    (b) $\{a, b, c\}$
    (c) $\{\}$

Solution:

(a) $\{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111\}$

(b) $\{\epsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, abc, acb, bac, bca, cab, cba\}$

(c) $\{\epsilon\}$

26. Design a Turing machine to recognise the language $\{0^n1^n \mid n \geq 0\}$.

Solution:

State	Input	Write	Move	Next
q_0	$\sqcup$	$\sqcup$	R	q_a
q_0	0	$\sqcup$	R	q_1
q_0	1	1	R	q_f
q_1	$\sqcup$	$\sqcup$	L	q_2
q_1	0	0	R	q_1
q_1	1	1	R	q_1
q_2	$\sqcup$	$\sqcup$	L	q_f
q_2	0	0	R	q_f
q_2	1	$\sqcup$	L	q_3
q_3	$\sqcup$	$\sqcup$	R	q_0
q_3	0	0	L	q_3
q_3	1	1	L	q_3

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 $w^R = 110101$.

Solution:

State	Input	Write	Move	Next
q_0	$\sqcup$	$\sqcup$	R	q_a
q_0	0	$\sqcup$	R	q_1
q_0	1	$\sqcup$	R	q_3
q_1	$\sqcup$	$\sqcup$	L	q_2
q_1	0	0	R	q_1
q_1	1	1	R	q_1
q_2	$\sqcup$	$\sqcup$	L	q_f
q_2	0	$\sqcup$	L	q_5
q_2	1	1	L	q_f
q_3	$\sqcup$	$\sqcup$	L	q_4
q_3	0	0	R	q_3
q_3	1	1	R	q_3
q_4	$\sqcup$	$\sqcup$	L	q_f
q_4	0	0	L	q_f
q_4	1	$\sqcup$	L	q_5
q_5	$\sqcup$	$\sqcup$	R	q_0
q_5	0	0	L	q_5
q_5	1	1	L	q_5

28. Design a Turing machine to recognise the language $\{a^i b^j c^k \mid i, j, k \in \mathbb{N}_0\}$

Solution:

State	Input	Write	Move	Next
q_0	$\sqcup$	$\sqcup$	R	q_a
q_0	a	a	R	q_0
q_0	b	b	R	q_1
q_0	c	c	R	q_2
q_0	$\sqcup$	$\sqcup$	R	q_a
q_0	a	a	R	q_f
q_0	b	b	R	q_1
q_0	c	c	R	q_2
q_0	$\sqcup$	$\sqcup$	R	q_a
q_0	a	a	R	q_f
q_0	b	b	R	q_f
q_0	c	c	R	q_2

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.