1. A red-black automaton is a normal non-deterministic finite-state automaton, except that it has two types of transitions — red transitions and black ones. Just as in the case of normal finite-state automata, a red-black automaton accepts an input string if, starting in its initial state, it can follow transitions labelled by symbols in the input and upon exhausting the input, it finishes in one of its final states. An added constraint is that the colour of transitions followed must alternate between red and black, starting with a red one.

The following red-black automaton (red transitions represented by dashed lines) accepts strings starting with an $a$ and alternate $b$’s and $a$’s ($(ab)^*a + (ab)^*$).

\[
\begin{array}{c}
    a \\
    \xrightarrow{r} \\
    \xrightarrow{b} \quad \xrightarrow{b} \quad \xrightarrow{a} \quad \xrightarrow{b} \\
    \xrightarrow{b} \\
    \xrightarrow{a} \\
    \xrightarrow{b}
\end{array}
\]

(a) Formalise the notion of a red-black automaton.

(b) Formalise the configuration of a red-black automaton. **Hint:** You must also remember what colour transition you must next take.

(c) Define $C_0$, the initial configuration, and the set $C_F$, the set of final configurations.
(d) Define \( C \xrightarrow{a} C' \), meaning that configuration \( C' \) can be reached from configuration \( C \) reading symbol (not string) \( a \).

(e) Define \( C \Rightarrow C' \), meaning that configuration \( C' \) can be reached from configuration \( C \) with string \( s \).

(f) Hence or otherwise define the language accepted by a red-black automaton.

(g) Give a construction to show that languages accepted by red-black automata are regular.

**Hint:** Use the configurations as the new states.

2. (a) Prove that the language \( \{s1^n = 1^{n+1} \mid n \in \mathbb{N}\} \) is not regular.

(b) Consider the following problem:

**Intermediate Substring:**
Given a Turing Machine \( M \), input \( i \) and string \( s \), does the \( s \) ever appear on the tape of \( M \) when started with input \( i \)?

Show that **Intermediate Substring** is undecidable.

**Hint:** Show that a Turing Machine can be converted into one that writes a particular symbol only upon termination and then use reducibility.

(c) What class of languages do Turing Machines with a finite tape accept?

3. (a) Consider the following decision problem:

**Repeated Language (RL)**

**Given:** A finite language \( L \) over alphabet \( \Sigma (\varepsilon \notin L) \) and a string \( s \in \Sigma^* \).

**Question:** Is \( s \in L^* \)?

Show that \( RL \in NP \).

(b) Let \( V \) be a set of 3-value domain variables, each of which can take a value in the set \( N = \{0, 1, 2\} \). A literal in 3-value domain logic is a statement of the form \( v = n \) or \( v \neq n \), where \( v \in V \) and \( n \in N \).
A clause is a set of literals, satisfiable if the variables in $V$ can be assigned values such that at least one of the literals is satisfied. A set of clauses is said to be satisfiable if the variables can be assigned such that each of the clauses is satisfiable. Consider the following decision problem:

**3-Value Domain (3VD)**

**Given:** A set of clauses $C$ over 3-value domain variables $V$.

**Question:** Is $C$ satisfiable?

By reducing 3SAT to 3VD, show that 3VD is NP-complete.