CSL253 - Theory of Computation

Tutorial 7

Team Members

1. Sanjani Kumari - 12341890
2. Hirannya Mhaisbadwe - 12340950
3. Amathul Azeez Sarah- 12340210
4. Aditya Jha- 12340090

Question 4

Let $T = \{(i, j, k) \mid i, j, k \in \mathbb{N}\}$. We want to show that T is countable.

Proof: A set is *countable* if its elements can be put into a one-to-one correspondence with the natural numbers $\mathbb{N}$. Since $\mathbb{N}$ is countable, and the Cartesian product of a finite number of countable sets is also countable, it follows that $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ is countable.

Therefore, T, being the set of all ordered triples of natural numbers, is countable.

Alternatively, consider arranging all triples (i, j, k) in order of the sum $s = i + j + k$. For example:

- When $s = 0$: only one triple $\rightarrow (0, 0, 0)$
- When $s = 1$: $(1, 0, 0),\ (0, 1, 0),\ (0, 0, 1)$
- When $s = 2$: $(2, 0, 0),\ (1, 1, 0),\ (1, 0, 1),\ (0, 2, 0),\ (0, 1, 1),\ (0, 0, 2)$
- And so on...

For each natural number s, there are only finitely many triples (i, j, k) such that $i + j + k = s$.

Now, list all such triples in increasing order of s, and within each group, use any fixed order (e.g., lexicographic order). This process lists every element of T one by one, without missing any.

Since we can list all elements of T in a sequence, it follows that T is countable.

Questions 19

A useless state in a pushdown automaton is never entered on any input string. Consider the problem of determining whether a pushdown automaton has any useless states. Formulate this problem as a language and show that it is decidable for DFA.

Solution:
Let

$$L = \{\langle D \rangle \mid D \text{ is a DFA and } D \text{ has at least one useless state}\}$$

Goal: Show that L is decidable.
Definition: A state in a DFA is *useless* if it is not reachable from the start state on any input string.
Decision Procedure:
Given an encoding $\langle D \rangle$ of a DFA

$$D = (Q, \Sigma, \delta, q_0, F),$$

we construct a Turing machine M that decides L as follows:

1. **Build a reachability graph:**
   - Each node corresponds to a state in Q.
   - From each state q_i, for each symbol $a \in \Sigma$, follow $\delta(q_i, a)$.
2. **Perform BFS or DFS** starting from the initial state q_0 to find all reachable states.
3. **Mark all reachable states**. Let $R \subseteq Q$ be the set of reachable states from q_0.
4. **Check for useless states**: If $R \neq Q$, then there is at least one state in $Q \setminus R$ that is not reachable — hence a useless state.
5. **Accept or Reject**:
   - If $Q \setminus R \neq \emptyset$, accept (i.e., D has at least one useless state).
   - Else, reject.

Conclusion: This procedure halts and correctly decides whether a DFA has a useless state. Hence, L is a decidable language.