# CSL-253 THEORY OF COMPUTATION
# tutorial 7 Q9,Q25

**Team Members**

AMAN KUMAR - 12340190
JASWINDER SINGH - 12340970
K. AMANDEEIP - 12341060

**Question 9**

Show that the following is decidable:

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

## solution:

We will use a basic fact from set theory:

$$X \subseteq Y \iff X \cap \overline{Y} = \varnothing$$

In our case, this means:

$$L(R) \subseteq L(S) \iff L(R) \cap \overline{L(S)} = \varnothing$$

So, to decide whether $L(R) \subseteq L(S)$, we can check whether $L(R) \cap \overline{L(S)}$ is empty.

- If $L(R) \cap \overline{L(S)} = \varnothing$, then $L(R) \subseteq L(S)$.
- If $L(R) \cap \overline{L(S)} \neq \varnothing$, then $L(R) \not\subseteq L(S)$.

**Claim:** $A$ is decidable.

1. As $R$ and $S$ are regular languages, we can construct DFAs for both.
2. Take the complement DFA of $S$, i.e., $L(\overline{S})$.
3. Construct the intersection of DFA for $R$ and DFA for $\overline{S}$. This is valid because regular languages are closed under intersection.
4. Just check the emptiness of the DFA obtained in step 3.

Emptiness of a DFA is decidable, as any DFA has a finite number of starting states and a finite number of total states. So there must be a finite number of paths. Therefore, we can determine in finite time whether we can reach any final state or not. If we can reach a final state, the language is non-empty; otherwise, it is empty.

So, this problem is decidable.

**Question 25**

Show that the following is decidable:

$$C = \{\langle G, x\rangle \mid G \text{ is a CFG that generates some string } w, \text{ where } x \text{ is a substring of } w\}.$$

## solution;

We are given a context-free grammar (CFG) $G$ and a string $x$, and we need to determine whether $G$ generates some string $w$ such that $x$ is a substring of $w$. That is, $C = \{\langle G, x\rangle \mid \exists w \in L(G),\ x \text{ is a substring of } w\}$.

To show that $C$ is decidable, we can use the following algorithm:

1. Construct a new CFG $G'$ that generates all strings over the alphabet of $G$ that contain $x$ as a substring.
2. Construct the intersection of $L(G)$ and $L(G')$.
3. Check whether the resulting language is empty.

**Step 1: Constructing $G'$**

Let $\Sigma$ be the alphabet of $G$. Construct a regular expression:

$$R = \Sigma^* x \Sigma^*$$

This regular expression represents all strings over $\Sigma$ that contain $x$ as a substring. Since regular expressions can be converted into finite automata, and finite automata can be converted into CFGs, we can construct a CFG $G'$ such that $L(G') = L(R) = \Sigma^* x \Sigma^*$.

**Step 2: Intersection**

Since context-free languages are not closed under intersection with each other, but they *are* closed under intersection with regular languages, we can construct a new CFG $G''$ such that:

$$L(G'') = L(G) \cap L(G')$$

This is again a context-free language.

**Step 3: Emptiness Test**

There is a known algorithm to decide whether a CFG generates the empty language. Apply this to $G''$:

- If $L(G'') = \emptyset$, then reject (i.e., $x$ is not a substring of any string in $L(G)$).
- Otherwise, accept (i.e., $x$ is a substring of some string in $L(G)$).

**Conclusion:**

Since all the steps above are computable (constructing $G'$, intersecting CFG with regular language, and checking emptiness of a CFG), the language $C$ is decidable.