

A startup company wants to build an intelligent system that decides whether an employee should be **approved for remote work** based on certain conditions. The decision depends on the following attributes:

1. **Role** – Technical / Non-Technical
2. **Experience** – Junior / Senior
3. **Performance** – Average / Excellent
4. **Internet Quality** – Poor / Good
5. **Work Location** – Urban / Rural

The target concept is **RemoteWorkApproval**: Yes or No.

The system is trained using the following examples:

<b>Role</b>	<b>Experience</b>	<b>Performance</b>	<b>Internet</b>	<b>Location</b>	<b>Approval</b>
Technical	Senior	Excellent	Good	Urban	Yes
Technical	Junior	Excellent	Good	Urban	Yes
Non-Technical	Junior	Average	Poor	Rural	No
Technical	Senior	Average	Good	Rural	No
Technical	Senior	Excellent	Good	Rural	Yes

### **Q1. Initialization**

- a) Define the **most specific hypothesis (S )** and **most general hypothesis (G )** for this problem.
- b) Why are these initial hypotheses chosen?

### **Q2. Step-by-Step Learning**

Process the training examples **in the given order** using the Candidate Elimination Algorithm.

- a) After processing **each example**, write the updated **Specific Boundary (S)** and **General Boundary (G)**.
- b) Clearly indicate whether each example is **positive or negative** and explain the changes made.

### **Q3. Version Space Analysis**

- a) What does the **final version space** represent in this case?

b) Give **two valid hypotheses** that lie within the final version space.

#### **Q4. Prediction**

Using the final S and G boundaries, predict whether the following employee should be approved for remote work:

Role	Experience	Performance	Internet	Location
Technical	Senior	Excellent	Good	Urban

Justify your answer using the learned hypothesis boundaries.

#### **Q5. Conceptual Understanding**

- What would happen if **contradictory training examples** were added to the dataset?
- Can Candidate Elimination handle **noisy data**? Give reasons.

#### **Q6. Critical Thinking**

- Identify one **real-world limitation** of using Candidate Elimination in this scenario.
- Suggest one modification or alternative algorithm that could address this limitation.

#### **Q7: check out**

If a new attribute “**Team Dependency**” (**Low / High**) is added:

- How will the hypothesis representation change?
- What impact will this have on the size of the version space?