2	**Aim**	Demonstrate the working model and principle of candidate elimination algorithm
	Program	For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples

CONCEPT - CANDIDATE-ELIMINATION LEARNING ALGORITHM

The CANDIDATE-ELIMINTION algorithm computes the version space containing all hypotheses from H that are consistent with an observed sequence of training examples.

CANDIDATE-ELIMINATION Algorithm

Initialize G to the set of maximally general hypotheses in H
Initialize S to the set of maximally specific hypotheses in H
For each training example d, do

- If d is a positive example
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is not consistent with d
    - Remove s from S
    - Add to S all minimal generalizations h of s such that
      - h is consistent with d, and some member of G is more general than h
    - Remove from S any hypothesis that is more general than another hypothesis in S

- If d is a negative example
  - Remove from S any hypothesis inconsistent with d
  - For each hypothesis g in G that is not consistent with d
    - Remove g from G
    - Add to G all minimal specializations h of g such that
      - h is consistent with d, and some member of S is more specific than h
    - Remove from G any hypothesis that is less general than another hypothesis in G

To illustrate this algorithm, assume the learner is given the sequence of training examples from the ***EnjoySport*** task

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

CANDIDATE-ELIMINTION algorithm begins by initializing the version space to the set of all hypotheses in H;

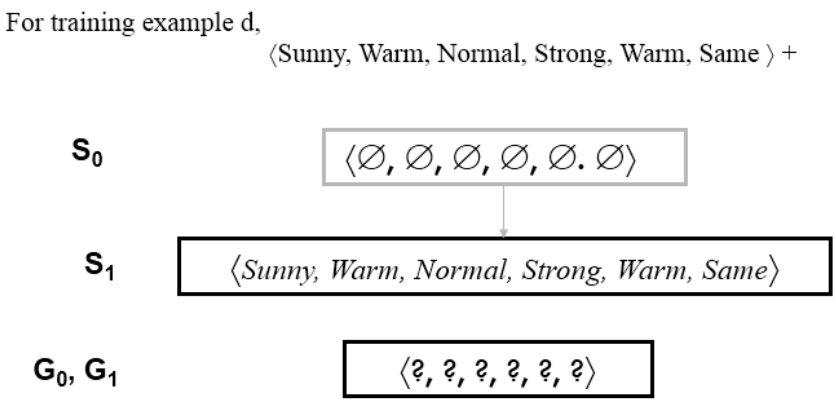
Initializing the G boundary set to contain the most general hypothesis in H

G0 ⟨?, ?, ?, ?, ?, ?⟩

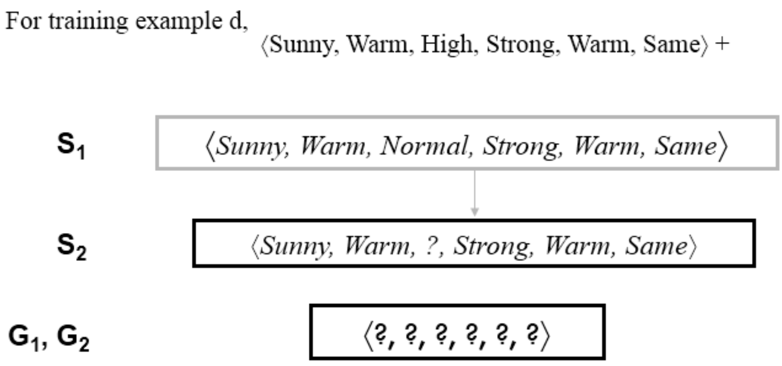
Initializing the S boundary set to contain the most specific (least general) hypothesis

S0 ⟨∅, ∅, ∅, ∅, ∅, ∅⟩

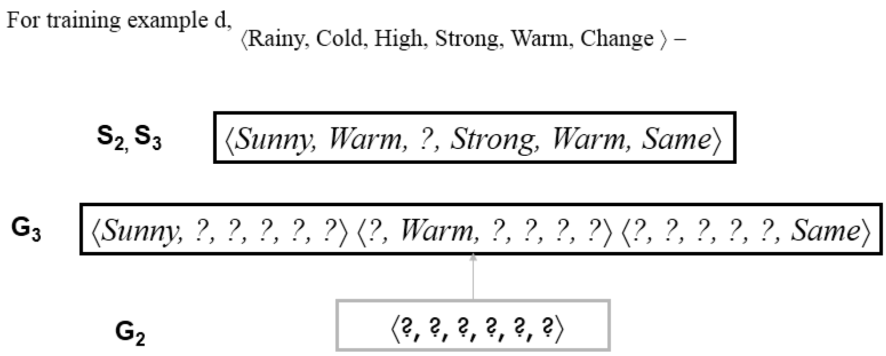
- When the first training example is presented, the CANDIDATE-ELIMINTION algorithm checks the S boundary and finds that it is overly specific and it fails to cover the positive example.
- The boundary is therefore revised by moving it to the least more general hypothesis that covers this new example
- No update of the G boundary is needed in response to this training example because G_o correctly covers this example



- When the second training example is observed, it has a similar effect of generalizing S further to S_2, leaving G again unchanged i.e., $G_2 = G_1 = G_0$



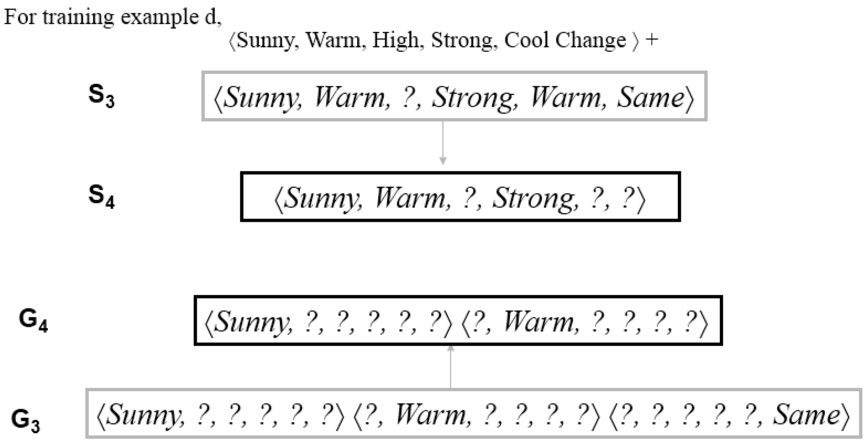
- <u>Consider the third training example</u>, this negative example reveals that the G boundary of the version space is overly general, that is, the hypothesis in G incorrectly predicts that this new example is a positive example.
- The hypothesis in the G boundary must therefore be specialized until it correctly classifies this new negative example



<u>Given that there are six attributes that could be specified to specialize G_2, why are there only three new hypotheses in G_3?</u>

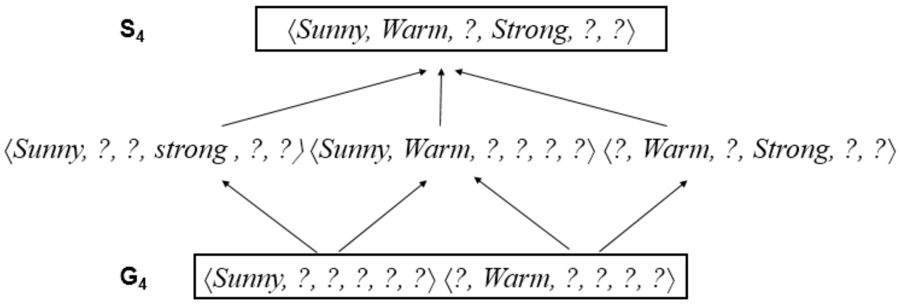
For example, the hypothesis h = (?, ?, Normal, ?, ?, ?) is a minimal specialization of G_2 that correctly labels the new example as a negative example, but it is not included in G_3. The reason this hypothesis is excluded is that it is inconsistent with the previously encountered positive examples

- <u>Consider the fourth training example</u>.



- This positive example further generalizes the S boundary of the version space. It also results in removing one member of the G boundary, because this member fails to cover the new positive example

After processing these four examples, the boundary sets S_4 and G_4 delimit the version space of all hypotheses consistent with the set of incrementally observed training examples.



Training Instances: (The below data is saved as *enjoysport.csv* file)

Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

Program:

```
import pandas as pd

data = pd.read_csv('enjoysport.csv')
concepts = data.iloc[:, :-1].values
target = data.iloc[:, -1].values
n=len(concepts[0])-1
specific_h = ['0'] * n
general_h = ['?'] * n
print("The initialization of the specific and general hypothesis ")
print(" S0:",specific_h,"\n G0:",general_h)

def learn(concepts, target):
    specific_h = concepts[0].copy()
    general_h = [["?" for _ in range(len(specific_h))] for _ in range(len(specific_h))]
    for i, h in enumerate(concepts):
        if target[i] == "yes":
            print(f"\n the {i+1} training instance is Positive \n",concepts[i])
            for x in range(len(specific_h)):
                if h[x] != specific_h[x]:
                    specific_h[x] = '?'
                    general_h[x][x] = '?'
        else:
            print(f"\nThe {i+1} training instance is Negative \n",concepts[i])
            for x in range(len(specific_h)):
                if h[x] != specific_h[x]:
                    general_h[x][x] = specific_h[x]
                else:
                    general_h[x][x] = '?'

        print(f"S{i+1}:\n", specific_h)
        print(f"G{i+1}:\n", general_h)

    general_h = [h for h in general_h if h != ['?' for _ in range(len(specific_h))]]
    return specific_h, general_h

s_final, g_final = learn(concepts, target)

print("\nThe Final Specific Hypothesis:")
print(s_final)
print("\nThe Final General Hypothesis:")
print(g_final)
```

Output:

```
The initialization of the specific and general hypothesis
S0: ['0', '0', '0', '0', '0']
G0: ['?', '?', '?', '?', '?']

The 1 training instance is Positive
['sunny' 'warm' 'normal' 'strong' 'warm' 'same']
S1:
['sunny' 'warm' 'normal' 'strong' 'warm' 'same']
G1:
[['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?']]

The 2 training instance is Positive
['sunny' 'warm' 'high' 'strong' 'warm' 'same']
S2:
['sunny' 'warm' '?' 'strong' 'warm' 'same']
G2:
[['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?']]

The 3 training instance is Negative
['rainy' 'cold' 'high' 'strong' 'warm' 'change']
S3:
['sunny' 'warm' '?' 'strong' 'warm' 'same']
G3:
[['sunny', '?', '?', '?', '?', '?'],
['?', 'warm', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', 'same']]
```

```
The 4 training instance is Positive
['sunny' 'warm' 'high' 'strong' 'cool' 'change']
S4:
['sunny' 'warm' '?' 'strong' '?' '?']
G4:
[['sunny', '?', '?', '?', '?', '?'],
['?', 'warm', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?'],
['?', '?', '?', '?', '?', '?']]

The Final Specific Hypothesis:
['sunny' 'warm' '?' 'strong' '?' '?']

The Final General Hypothesis:
[['sunny', '?', '?', '?', '?', '?'],
['?', 'warm', '?', '?', '?', '?']]
```