Find-S Algorithm:

1. **Initialization**

   The algorithm starts with the most specific hypothesis, where all attribute values are set to the most specific possible values.

2. **Updating Hypothesis**

   As the algorithm encounters positive examples, it generalizes the hypothesis by replacing specific attribute values with more general ones.

3. **Output**

   The final hypothesis represents the most general concept that covers all the positive examples.

FIND-S Algorithm

1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x

   For each attribute constraint ai in h

   If the constraint ai is satisfied by x

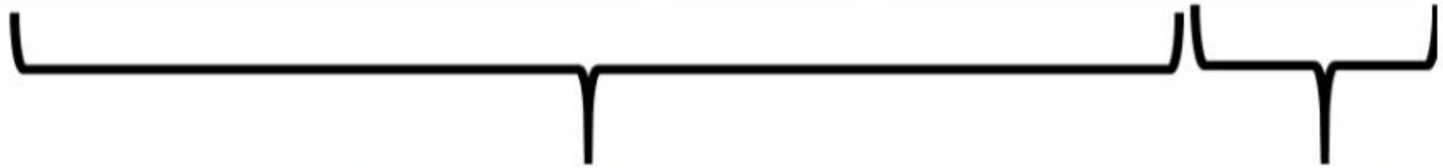
   Then do nothing

   Else

   replace ai in h by the next more general constraint that is satisfied by x
3. Output hypothesis h

Seed prediction Hypothesis:

EXAMPLE	COLOR	TOUGHNESS	FUNGUS	APPEARANCE	POISONOUS
1.	GREEN	HARD	NO	WRINKELD	YES
2.	GREEN	HARD	YES	SMOOTH	NO
3.	BROWN	SOFT	NO	WRINKLED	NO
4.	ORANGE	HARD	NO	WRINKLED	YES
5.	GREEN	SOFT	YES	SMOOTH	YES
6.	GREEN	HARD	YES	WRINKLED	YES
7.	ORANGE	HARD	NO	WRINKLED	YES



ATTRIBUTES ON WHICH THE CONCEPT DEPENDS ON

CONCEPT

Dynamic Implementation:

Find-S Algorithm

Color	Toughness	Fungus	Appearance	Poisonous (Yes/No)

Add Row | FIND HYPOTHESIS

FINAL HYPOTHESIS

Find-S Algorithm

Green	Hard	No	Wrinkled	Yes
Green	Hard	Yes	Smooth	No
Brown	Soft	No	Wrinkled	No

Add Row | FIND HYPOTHESIS

FINAL HYPOTHESIS

Green, Hard, No, Wrinkled

Find-S Algorithm

Green	Hard	No	Wrinkled	Yes
Green	Hard	Yes	Smooth	No
Brown	Soft	No	Wrinkled	No
Orange	Hard	No	Wrinkled	Yes
Green	Soft	Yes	Smooth	Yes
Green	Hard	Yes	Wrinkled	Yes
Orange	Hard	No	Wrinkled	Yes

Add Row | FIND HYPOTHESIS

FINAL HYPOTHESIS

?, ?, ?, ?

Static Implementation:

Find-S Algorithm

Color	Toughness	Fungus	Appearance	Poisonous
Green	Hard	No	Wrinkled	Yes
Green	Hard	Yes	Smooth	No
Brown	Soft	No	Wrinkled	No
Orange	Hard	No	Wrinkled	Yes
Green	Soft	Yes	Smooth	Yes
Green	Hard	Yes	Wrinkled	Yes
Orange	Hard	No	Wrinkled	Yes

FIND HYPOTHESIS

FINAL HYPOTHESIS

?, ?, ?, ?

Step 1:Initialize h to the most specific hypothesis in H

H=<Ø, Ø, Ø, Ø>

Step 2: Iteration

S1: x1=< Green , Hard, No , Wrinkled >
h1 =<Green , Hard, No , Wrinkled>

S2: x2=< > // not consider because algorithm ignore –ve example
h2=< >

S3: x3=< > //
h3=<>

S4: x4=<Orange ,Hard , No , Wrinkled>
h4=< ? , Hard, No ,Wrinkled> //replace h attribute if it does not match with the constraint

S5: x5=<Green , Soft , Yes , Smooth>
h5=< ? , ? , ? , ? >

S6: x6=<Green , Hard , Yes , Wrinkled>
h6=< ? , ? , ? , ? >

S7: x7=<Orange , Hard , No , Wrinkled>
h7=< ? , ? , ? , ? >

THEREFORE FINAL HYPOTHESIS IS:

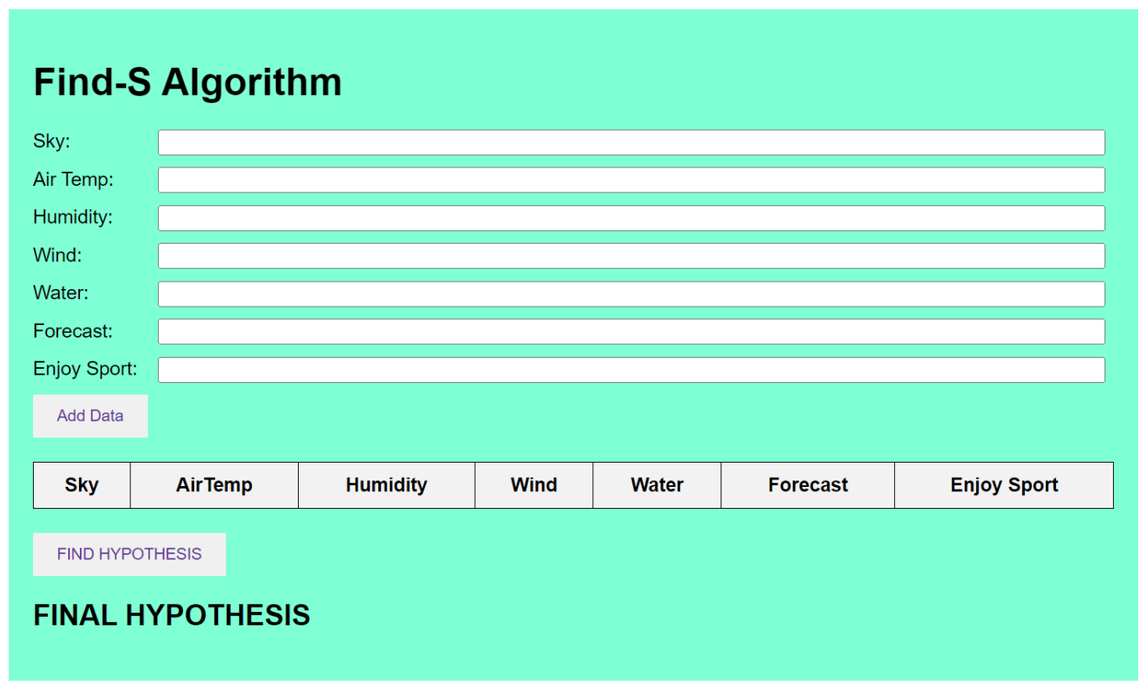
<? ,? ,? ,? ,?>

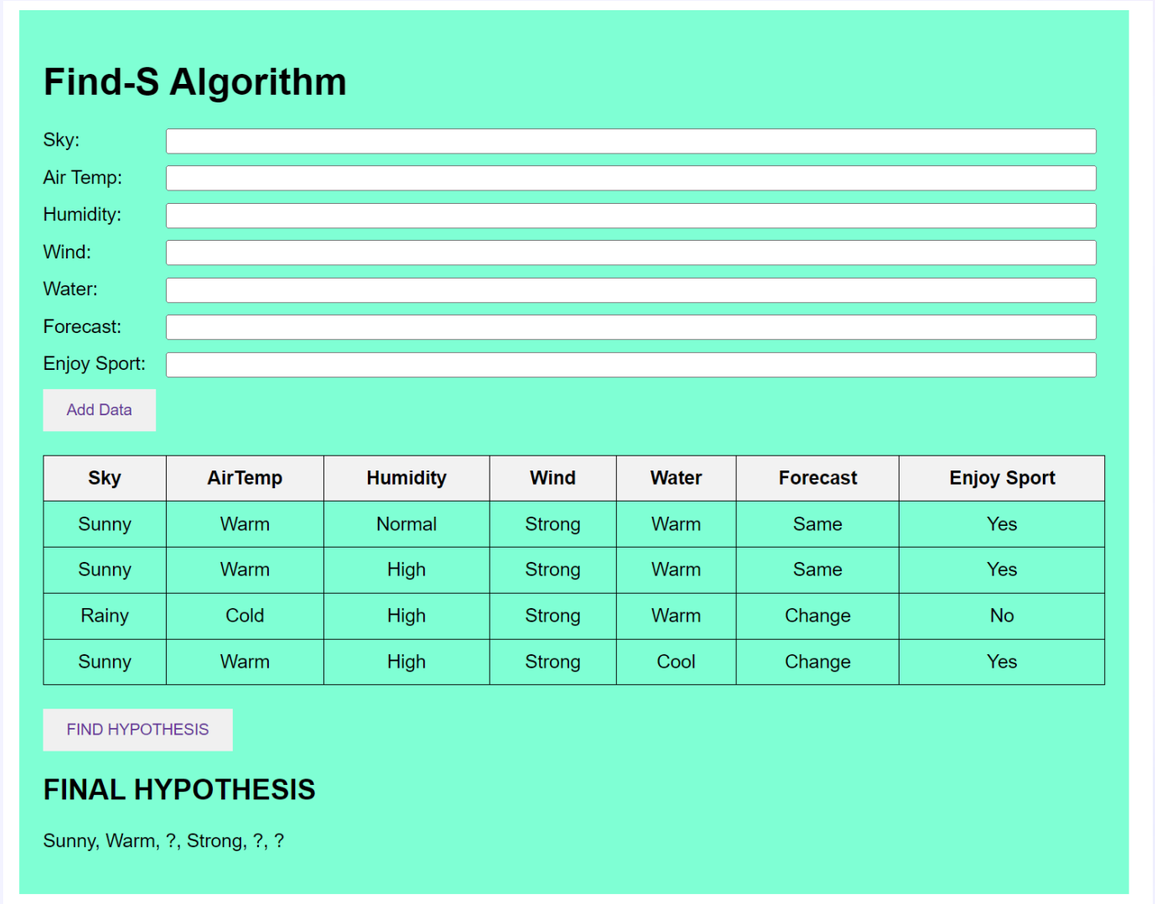
This means it doesn't specifically depends on any attribute to prove that the seed is poisonous or not

Enjoy Sports Hypothesis:

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

Dynamic implementation:





Static Implementation:

Find-S Algorithm

Sky	AirTemp	Humidity	Wind	Water	Forecast	Enjoy Sport
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

FIND HYPOTHESIS

FINAL HYPOTHESIS

Sunny, Warm, ?, Strong, ?, ?

Step 1:Initialize h to the most specific hypothesis in H

H=<Ø, Ø, Ø, Ø>

Step 2: Iteration

S1: x1=< Sunny , warm , Normal , Strong , Warm , Same >
h1 =< Sunny , warm , Normal , Strong , Warm , Same >

S2: x2=< Sunny , warm , High , Strong , Warm , Same >
h2=< Sunny , warm , ? , Strong , Warm , Same >

S3: x3=< > //- ve example ignore
h3=< >

S4: x4=< Sunny , warm , High , Strong , Cool, Change >
h4=< Sunny , warm , ? , Strong , ? , ? >

THEREFORE FINAL HYPOTHESIS IS:

< Sunny , warm , ? , Strong , ? , ? >

This means when the hypothesis matches with sunny , warm , strong attribute we can conclude that it is a enjoy sport day irrespective of other attribute

Saturday Time Table Prediction using Find-S & Candidate Elimination Algorithm

Time table used

	9.00-10.00	10.00-10.50	11.10-12.00	12.00-1.00	TIME TABLE FOLLOWED
MONDAY	EJ	ML	AI	DEVOP	YES
TUESDAY	AI	EJ	DAP	SE&T	NO
WEDNESDAY	EJ	MA/DAP LAB	MA/DAP LAB	MA/DAP LAB	YES
THURSDAY	EJ	DAP	AI	SE&T	NO
FRIDAY	DEVOP	ML	EJ	SE&T	YES

Find-S Algorithm

Find-S Algorithm

	9.00-10.00	10.00-10.50	11.10-12.00	12.00-1.00	TIME TABLE FOLLOWED
MONDAY	EJ	ML	AI	DEVOP	YES
TUESDAY	AI	EJ	DAP	SE&T	NO
WEDNESDAY	EJ	MA/DAP LAB	MA/DAP LAB	MA/DAP LAB	YES
THURSDAY	EJ	DAP	AI	SE&T	NO
FRIDAY	DEVOP	ML	EJ	SE&T	YES
SATURAY					

FIND HYPOTHESIS

THE HYPOTHESIS FOR SATURDAY TIME TABLE WOULD BE

?, ?, ?, ?

Candidate Elimination Algorithm

Candidate Elimination Algorithm

	9.00-10.00	10.00-10.50	11.10-12.00	12.00-1.00	TIME TABLE FOLLOWED
MONDAY	EJ	ML	AI	DEVOP	YES
TUESDAY	AI	EJ	DAP	SE&T	NO
WEDNESDAY	EJ	MA/DAP LAB	MA/DAP LAB	MA/DAP LAB	YES
THURSDAY	EJ	DAP	AI	SE&T	NO
FRIDAY	DEVOP	ML	EJ	SE&T	YES
SATURDAY					

SPECIFIC HYPOTHESIS: ?, ?, ?, ?

GENERAL HYPOTHESIS: None

PREDICTED SATURDAY TIMETABLE: Any, Any, Any, Any

Run Candidate Elimination

Advantages and Disadvantages of Algorithm in Programming

Advantages of Find-S Algorithm in Predictive Modeling

Simplicity

The Find-S algorithm is relatively straightforward and easy to implement.

Efficiency

The algorithm can quickly converge to the most specific hypothesis that covers all positive examples.

Interpretability

The learned hypothesis is easy to understand and can provide insights into the data.

Flexibility

The Find-S algorithm can be applied to a wide range of predictive modeling problems.

Advantages and Disadvantages of Algorithms

Limitations and Considerations of Find-S Algorithm

Sensitivity to Noise

The Find-S algorithm can be sensitive to noisy or irrelevant data, which can lead to overly specific or inaccurate hypotheses.

Handling Continuous Attributes

The algorithm is primarily designed for discrete, categorical attributes, and may not perform as well with continuous data.

Bias Towards Specific Hypotheses

The algorithm's tendency to converge to the most specific hypothesis may not always be the most appropriate for the problem at hand.

Lack of Uncertainty Handling

The Find-S algorithm does not provide any information about the uncertainty or confidence in the predicted hypotheses.