Logical Reasoning Styles and Their Applications

Jacqueline Mitchell (CSCI 698 Teaching Video)

Three Statements

This code passed all of my tests, so it must be correct!

If x > 0, then y = x + 1 is definitely greater than 0.

The program crashed after I changed this line of code, so that must be the bug.

Q: What is the **structural** difference between these statements?

① What is being derived?

② What is being used to derive "that thing"?

① + ② govern/determine the type of reasoning being used

How are these Statements Different?

a), b), c) all are different in terms of

① What is being derived

② What is being used to derive it

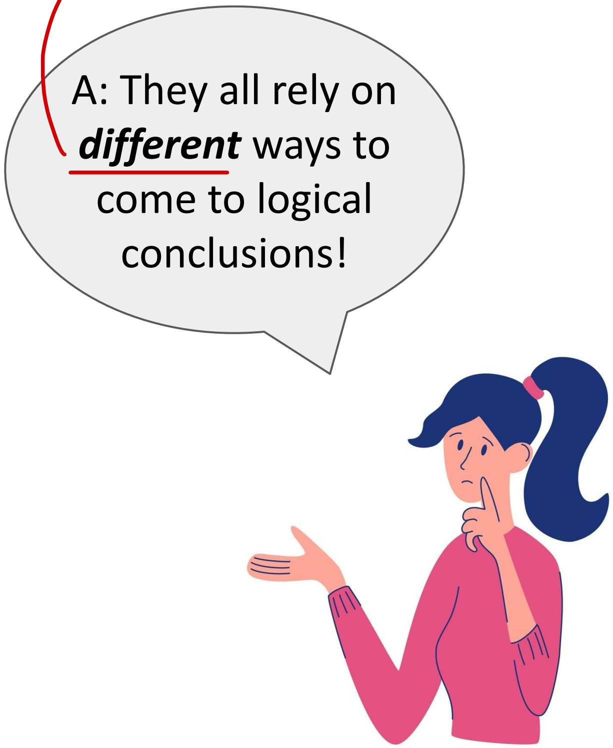
a) *This code passed all of my tests, so it must be correct!*

b) *If x > 0, then y = x + 1 is definitely greater than 0.*

+ also relies on our knowledge of math with real numbers

c) *The program crashed after I changed this line of code, so that must be the bug.*

+ also relies on our knowledge of programming and the compiler



Why is Classifying Reasoning Important?

This code passed all of my tests, so it must be correct!

If x > 0, then y = x + 1 is definitely greater than 0.

The program crashed after I changed this line of code, so that must be the bug.

Different logical reasoning styles underpin many logical processes in CS

including parts of machine learning, neurosymbolic systems, debugging, and more!

Our Roadmap

<u>Three Types of Reasoning</u>

Inductive

Deductive

Abductive

* there are other kinds too! (Check out causal/counterfactual reasoning and probabalistic reasoning)

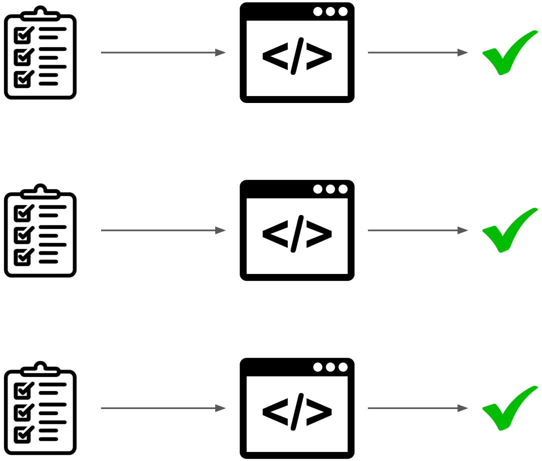
Inductive Reasoning

This code passed all of my tests, so it must be correct!

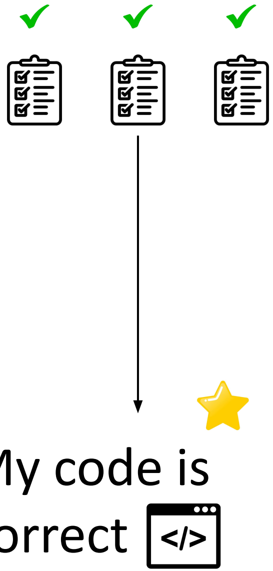
Observation 1 (test case 1 passed)

observation 2 (test case 2 passed)

observation 3 (test case 3 passed)



set of observations

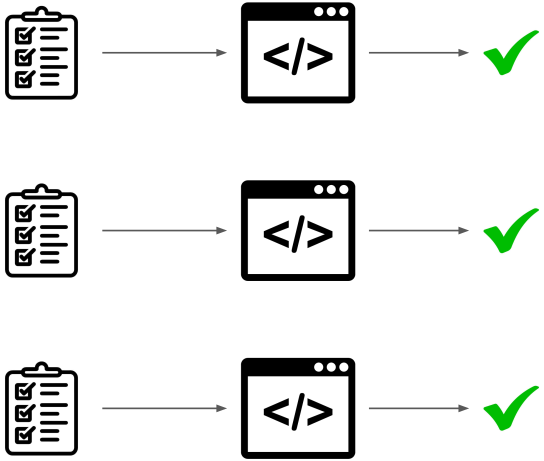


My code is correct

a general rule that we think is true

Inductive Reasoning

This code passed all of my tests, so it must be correct!

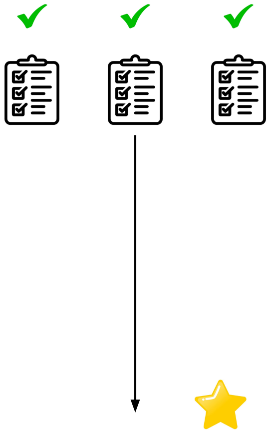


Examples

Infer

General Rule or Conclusion (***probably*** true)

* Note: we are interested in a claim about the program based on the observations

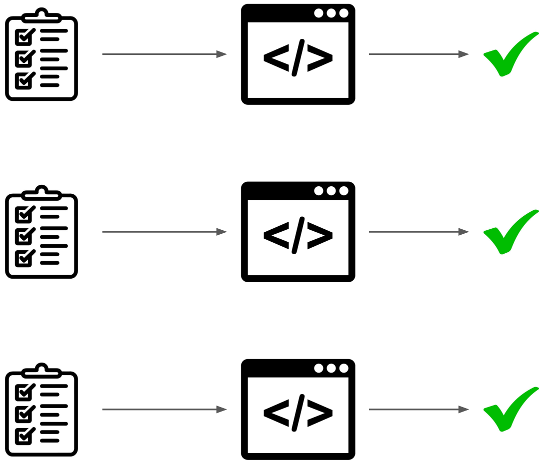


My code is correct

(based on the test cases passing, we expect the code to be correct)

Inductive Reasoning

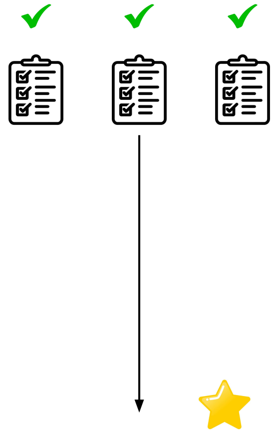
This code passed all of my tests, so it must be correct!



Examples

Infer

General Rule or Conclusion (***probably*** true)



My code is correct

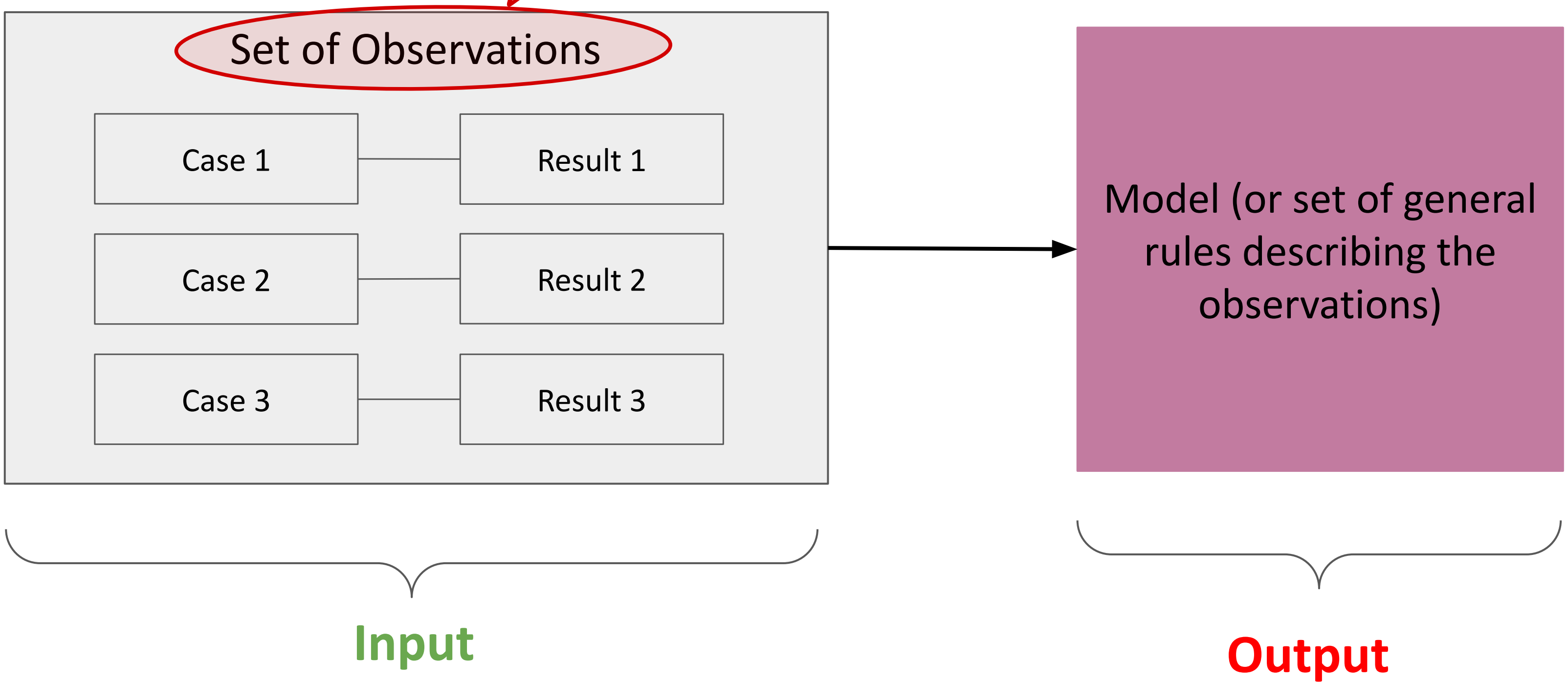
(there are no guarantees our conclusion is correct!)

Need not be true! (There could be some failing test case)

(our conclusion is just what we expect to be *likely* based on observations)

(More Generally) Inductive Reasoning

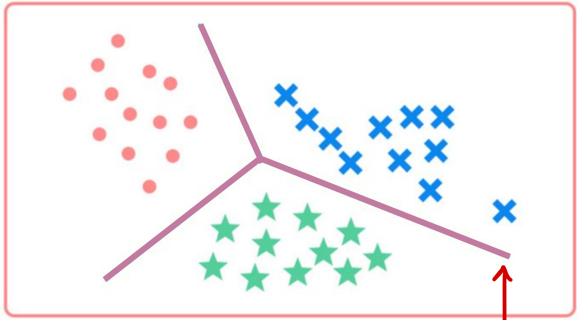
this is the key! It doesn't really matter what the observations look like



Goal: Convert some observations into rules or conclusions!

Examples of Inductive Reasoning in Computer Science

Machine Learning → Supervised Learning



Supervised learning

- note, it is possible that our model will misclassify something (classification error is an example of the fact that what we infer with induction does not necessarily hold true with respect to the ground truth of the world)

this is a model that we inferred based on the observations (labeled data)

Examples of Inductive Reasoning in Computer Science

Inductive Logic Programming

- parent(alice, bob) means alice is bob's parent

Existing Knowledge	`parent(alice, bob)` `parent(bob, charles)`
Examples	
Positive	`grandparent(alice, charles)`
Negative	`grandparent(bob, charles)`

Example of a TRUE fact (think: one class used to train a classifer with ground-truth data)

example of a FALSE fact

We infer what it means for X to be a grandparent of Y

logical and

```
grandparent(X, Y) :- parent(X, Z), parent(Z, Y)
```

(Note: in this case our rule is true, but need not be true in general)

Deductive Reasoning

If $x > 0$, then $y = x + 1$ is definitely greater than 0.

We assume that $x > 0$.

Based on this, we want to know the sign of y, where $y = x + 1$.

our problem/ premise

the knowledge we use to solve the problem

We use the mathematical properties of real numbers to infer that $y > 0$, because we're adding a positive number to another positive number

read top to bottom

Case / Logical Premise

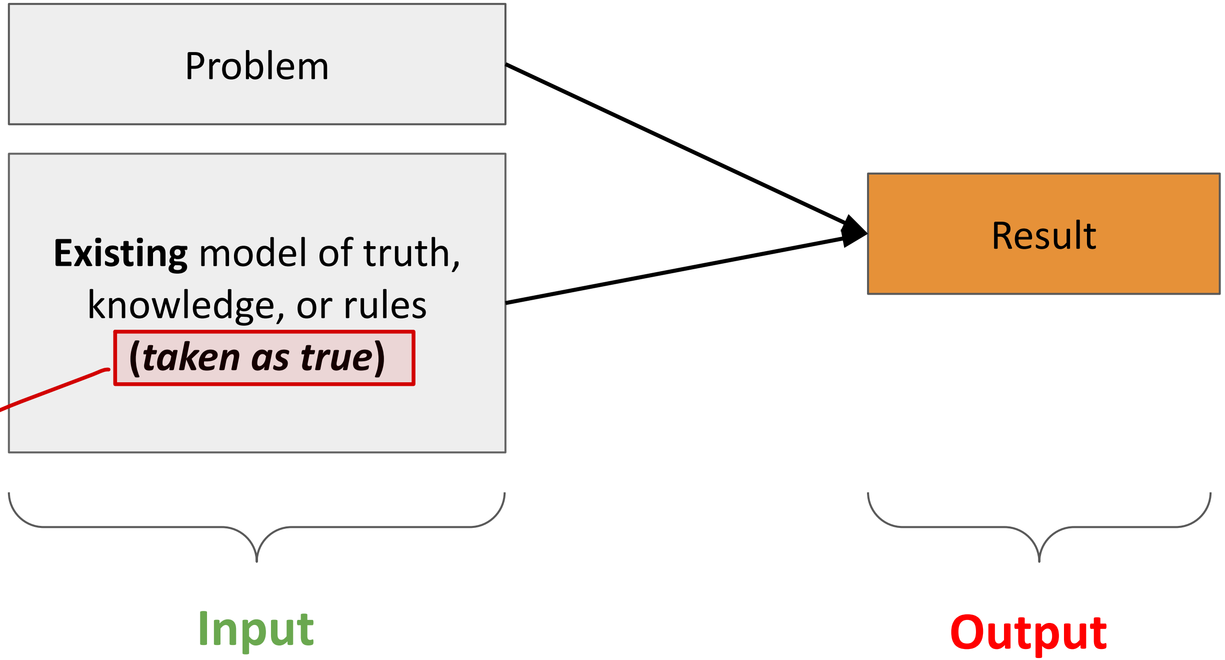
↓

Existing Knowledge or Models or Rules

↓

Answer / Conclusion

(More Generally) Deductive Reasoning



it is assumed to be true for the sake of reasoning

what the model assumes may be wrong with respect to the ground truth of the world)

(More Generally) Deductive Reasoning

Problem

Existing model of truth, knowledge, or rules (***taken as true***)

Result

Input

Output

Need not be true! (The model itself could be false)

* if the model is false, then the result or conclusion is false with respect to the ground truth of the world

↓

reasoning failure due to faulty assumptions

Examples of **Deductive** Reasoning in Computer Science

Logic Programming

```
parent(alice, bob)
parent(bob, charles)
```

the premise / our problem

```
grandparent(X, Y) :-
    parent(X, Z), parent(Z, Y)
```

the model or "rule" we assume to be true

```
grandparent(alice, charles)
```

our conclusion

* note: this is different from the previous grandparent example

Here, we are inferring a fact about alice and charles.

We ARE NOT inferring the model ("the grandparent rule")

Examples of Deductive Reasoning in Computer Science

there's a lot of interesting work on proving properties about programs. Google "formal methods and program verification" to learn more

Hoare Logic

*Hoare logic is our "model"

Program P: x := x + 1

this means: if I start with a state where x=n, then after executing program P, I end up in a state where x=n+1

Goal: Prove $\boxed{\{x=n\}\ P\ \{x = n+1\}}$

Hoare logic says, when we consider assignments, we can substitute what x is being assigned in the right-hand side

$\{x+1=n+1\}\ P\ \{x = n+1\}$

$x + 1 = n + 1 \quad \Rightarrow \quad x = n$

Hoare logic also has a rule that we can replace a condition on the left-hand side with something that it implies

$\{x=n\}\ P\ \{x = n+1\}$

We deduced our goal (with Hoare logic + arithmetic)

Abductive Reasoning

The program crashed after I changed this line of code, so that must be the bug.

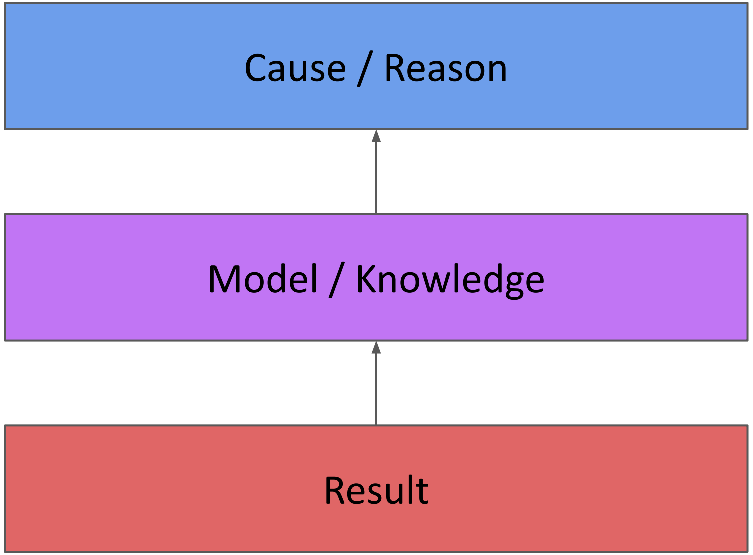
My program crashed, after I changed line 10!

our result

Based on my experience as a programmer (and the compiler), the bug must be on line 10.

the knowledge we're working with.

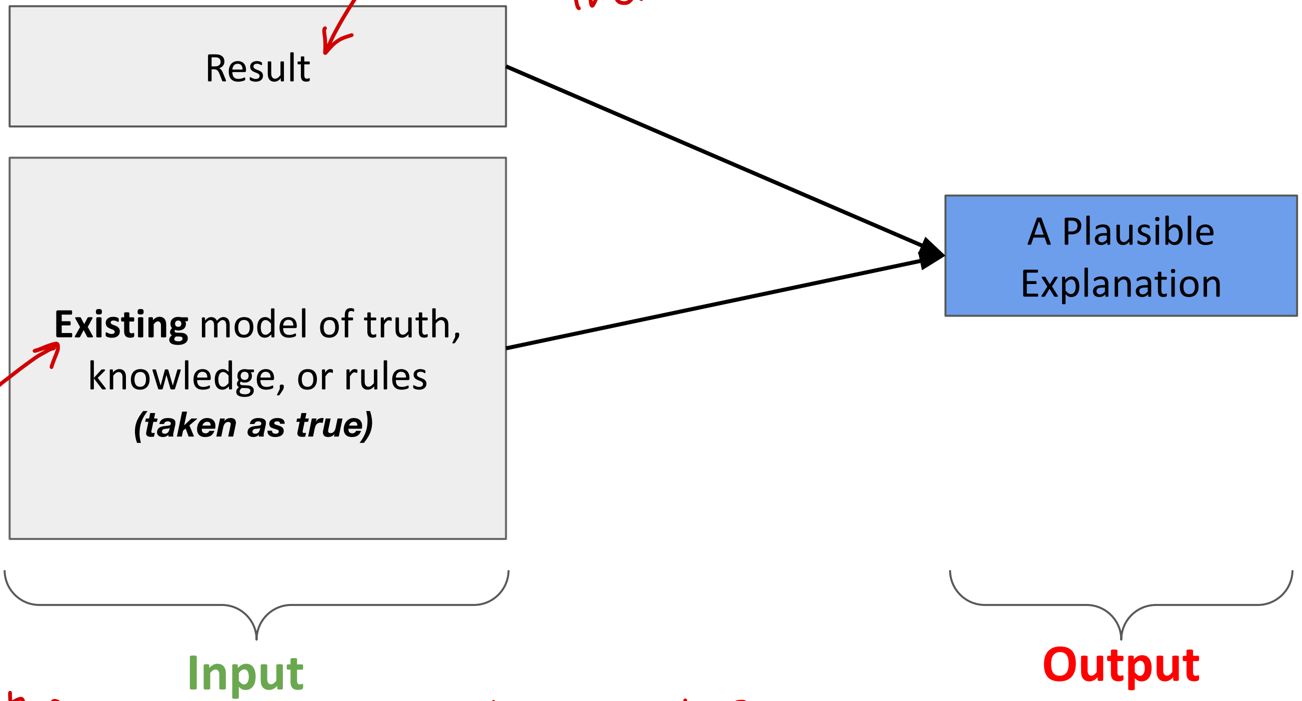
what we think is the cause



(read bottom to top)

(More Generally) **Abductive** Reasoning

some result / conclusion / phenomenon that we want to explain



again, the model need not be true! But, we assume it is the for reasoning

(More Generally) **Abductive** Reasoning

Result

Existing model of truth, knowledge, or rules *(taken as true)*

A Plausible Explanation

* It's important to remember that like deductive reasoning, the model need not be true with respect to the ground truth of the world

Input

Output

Need not be true! (The model itself could be false)

Examples of **Abductive** Reasoning in Computer Science

Debugging

it is helpful to read bottom-up

The discount code I just introduced must be causing a bug.

I infer that my new discount code introduced a bug

The failing test has to do with discount codes.

(Based on my knowledge of programming and the failing test case ...)

I added a new feature. One of my test cases now fails.

(we see that after adding a new feature, our testcase fails)

Examples of **Abductive** Reasoning in Computer Science

- alarm :- smoke
  ↳ this means from smoke, we can derive that there is an alarm

Abductive Logic Programming

Rules:

```
alarm :- smoke
smoke :- fire
```

what we are trying to explain → **Observation:** `There is an alarm.`

the set of possible explanations we allow ← **Abducible Predicates:** `{fire, rain}`

alarm ← smoke ← fire

We heard an alarm. From fire, we can infer smoke, and from smoke, we can infer alarm. So we think it is plausible that there was a fire.

Conclusion

- We learned about three kinds of reasoning:
  - Inductive [turns observations into rules/models/knowledge]
  - Deductive [derives a conclusion from a premise and existing knowledge]
  - Abductive [finds an explanation for an outcome based on existing knowledge]

- We saw examples of the various types of reasoning in computer science!