Why do things go wrong (or right)? Applications of causal reasoning to verification

Hana Chockler
causaLens
and
Department of Informatics
King's College, London

causaLens

KING'S College LONDON

Humans trust Causal AI with complex decisions

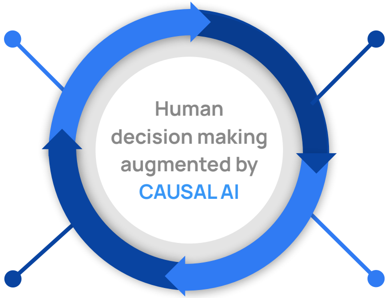
Correlation ML systems learn to perform simple predictions

But predictions are a very small element of decision making.

Causal AI is the only technology that can augment human decision making

DOMAIN EXPERTISE
Considering the wider context when making decisions

SCENARIOS
Evaluating hypothetical scenarios



CONSTRAINTS
Respecting human, legal and environmental constraints

INTERVENTIONS
Planning and designing optimal interventions

World’s First Full-Stack Causal AI Platform

We launched the World’s <u>First Causal AI Enterprise Platform</u>, which automates everything from Raw Data to Improved Business Decisions.

In-house data

External data

→

Data Cleaning & Discovery

Automatically discover valuable data

→

causaLab

Automatically build & operationalize causal models

→

decisionOS

From models to decisions

→

Decision Apps

Data from top sources

Nowcast s & forecast s

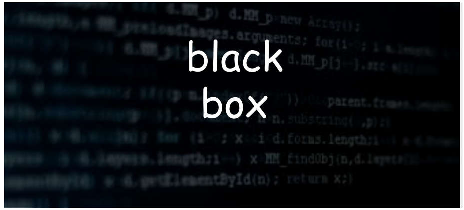
Live & adaptable Causal Models

Explainabilit y application

Business context

Optimal Intervention

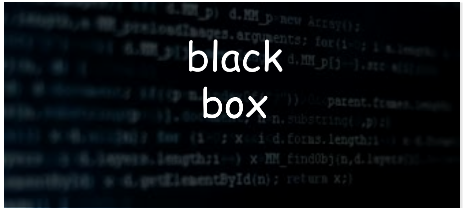
Motivation: Modern computerized systems are **huge and difficult to understand**



Motivation: Modern computerized systems are **huge and difficult to understand**

Can we understand and fix errors?

What does the system do?



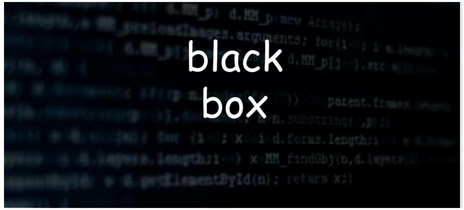
Can we be sure it is correct?

Motivation:

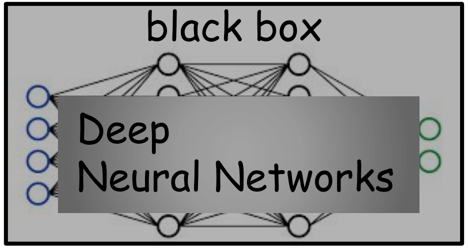
Modern computerized svstems are **huge and difficult or even impossible to understand**

Can we understand and fix errors?

What does the system do?



Can we be sure it is correct?



©Halpern & Pearl, 2001
+
©Halpern – many papers

Actual Causality

A theoretical concept from AI

Extends causal counterfactual reasoning

+

©Chockler & Halpern, 2004

Quantification of causality,

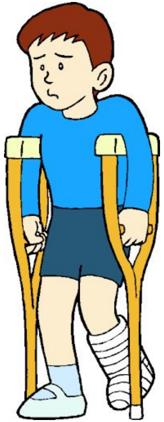
allowing to rank causes by importance

Turns out to be very useful!

<u>Intractable</u> – but there are efficient approximation algorithms and sufficient partial solutions

A-priori (type) and a-posteriori (actual) causality

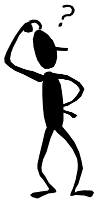
Turns out he broke his leg



There is a terrible pollution, so my next patient is likely to suffer from breathing problems

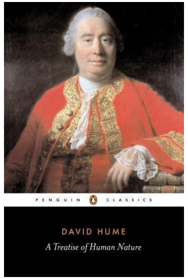
Background:

Causality

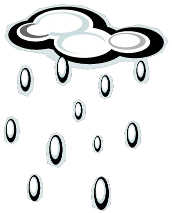


When do we say that **A** is a cause of **B**?

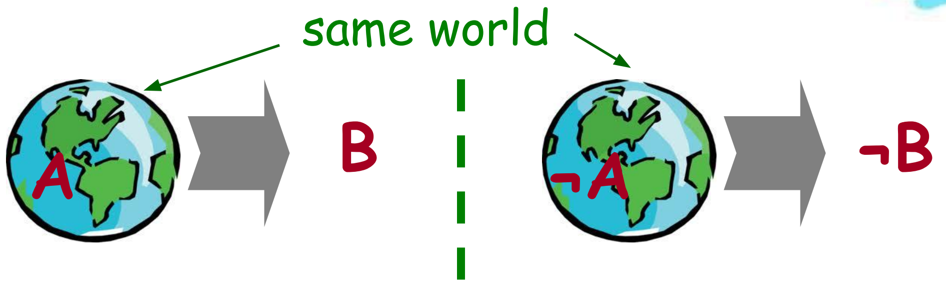
Common approach: **counterfactual causality**.



A is a **cause** of **B** if, had **A** not happened, then **B** would not have happened.



Rain is a cause of me being drenched.



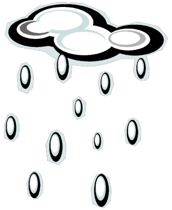
Causality

When do we say that **A** is a cause of **B**?

Common approach: **counterfactual causality**.

We need to capture more complex causal connections!

over determination



Rain or car is a cause of me being drenched?

Rain or a car causes being drenched.

Causality

When do we say that **A** is a cause of **B**?

Common approach: **counterfactual causality**.

We need to capture more complex causal connections!

preemption

10:00AM

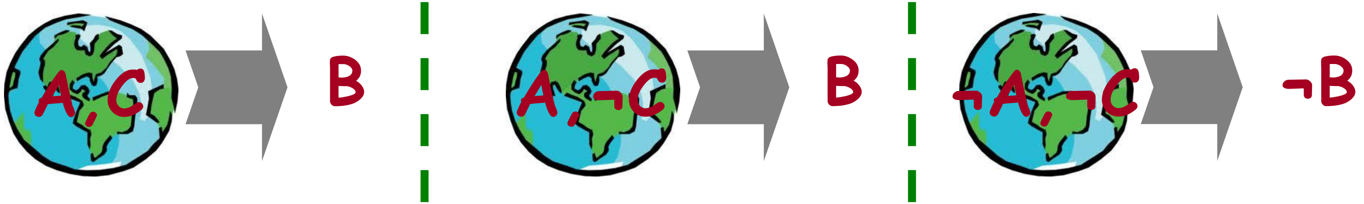
Car is a cause of me being drenched, but not the rain

©Halpern & Pearl, 2001

Actual causality

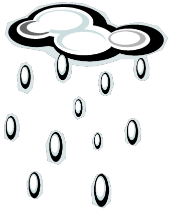
Extends the counterfactual reasoning by having expressive causal models allowing overdetermination, preemption, and complex causal structures

<u>Overdetermination:</u> A is a cause of B if there exists some contingency *C* (change in the current world)
in which B counterfactually depends on A.



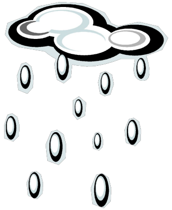
original world　　contingency　　counterfactual dependence

Illustration of overdetermination in actual causality



Rain is an actual cause of me
being drenched.

Contingency = the car

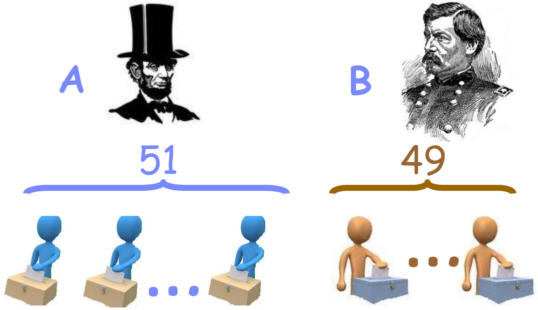


Rain is
a counterfactual cause

Responsibility: a quantitative measure of causality

Voting example

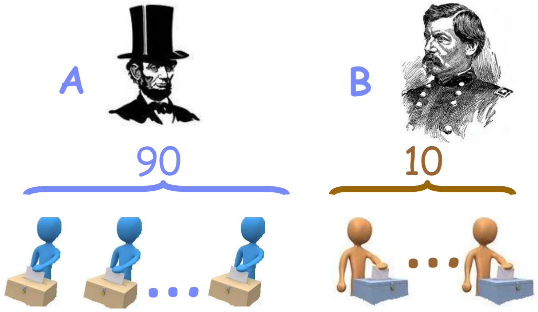
California:



Each **blue** voter is a cause of Lincoln's win

We need to distinguish between the cases!

New York:



Each **blue** voter is a cause of Lincoln's win

Responsibility: a quantitative measure of causality

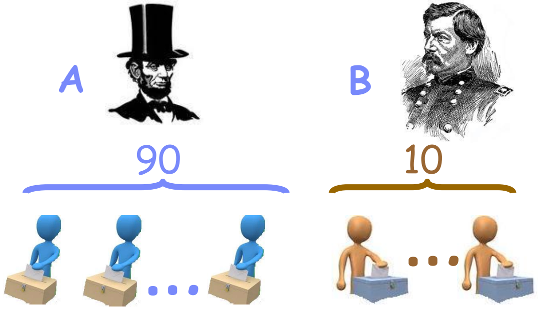
Voting example

California:

Responsibility allows to rank causes

Each blue voter is
1-responsible for
Lincoln's win

New York:



Each blue voter is
1/40-responsible for
Lincoln's win

Complexity of Computing Causality and Responsibility

Causality:

- Σ_2- complete for singleton causes
- D_2- complete in general case

Responsibility:

- $FP^{\Sigma_2[\log(n)]}$ complete.

INTRACTABLE

D_2 is the difference class of Σ_2 and Π_2

Complexity of Computing Causality and Responsibility

Causality:

- Σ_2- complete for singleton causes
- D_2- complete in general case

INTRACTABLE

Responsibility:

- $FP^{\Sigma_2[\log(n)]}$- complete.

The good news:

- There are linear-time approximation algorithms
  - Accurate on most problems
- We usually care only about highest-ranked causes
  - Polynomial to compute the exact set

Motivation :

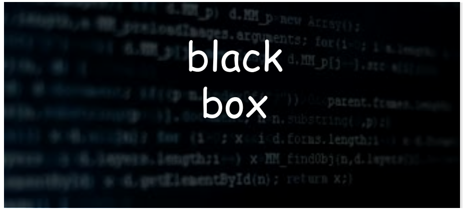
Modern computerized svstems are **huge and difficult or even impossible to understand**

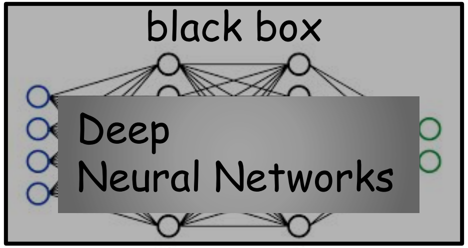
Can we understand and fix errors?

Can we be sure it is correct?

verification

What does the system do?

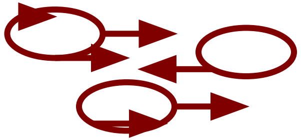




Formal Verification

?Is the system correct

A huge and difficult
to understand system M:



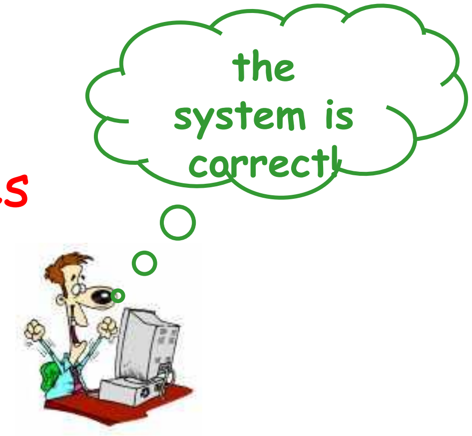
A correctness specification φ

Does M satisfy φ?

no

counterexample

yes



Formal Verification

?Is the system correct

A huge and difficult to understand system M:

A correctness specification φ

Do we actually know how to fix the system?

Do we understand the counterexample?

no

counterexample

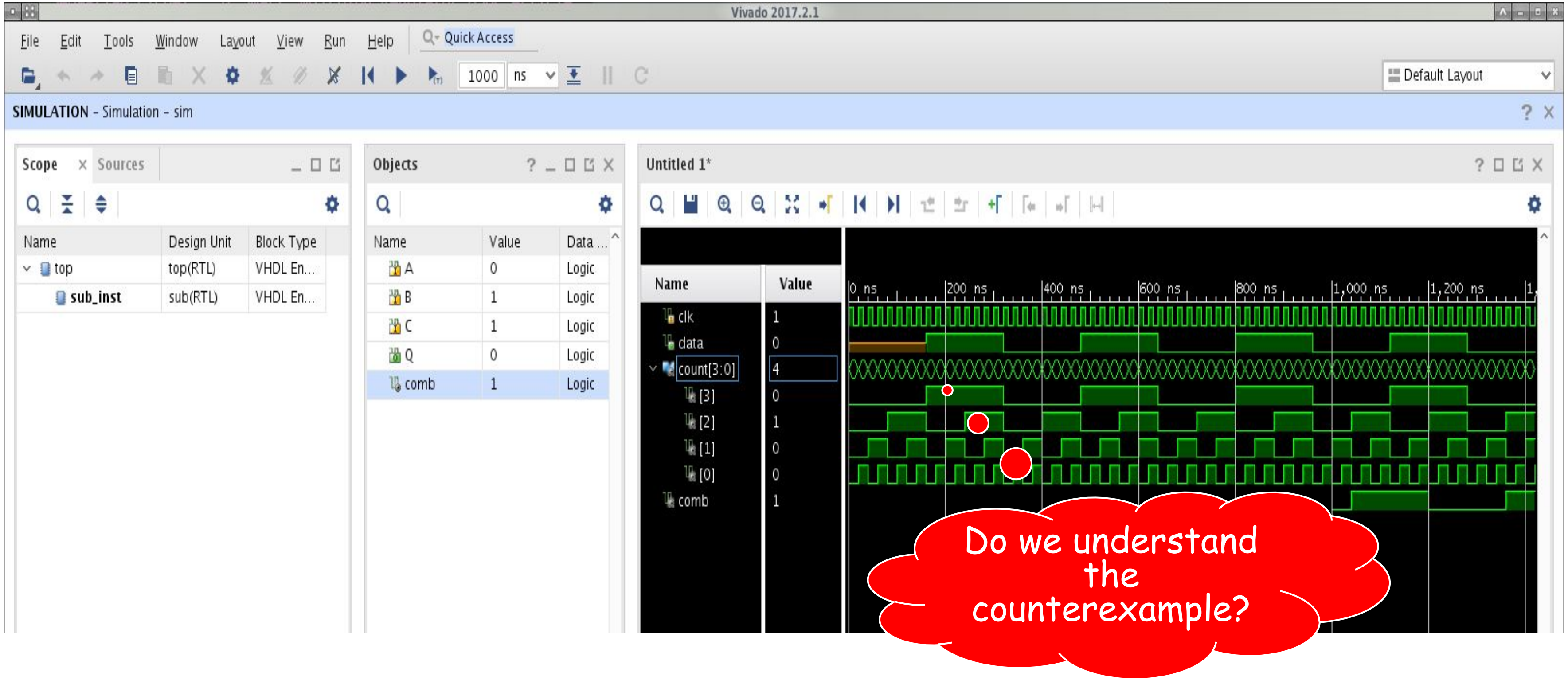
yes

the system is correct!

Counterexamples

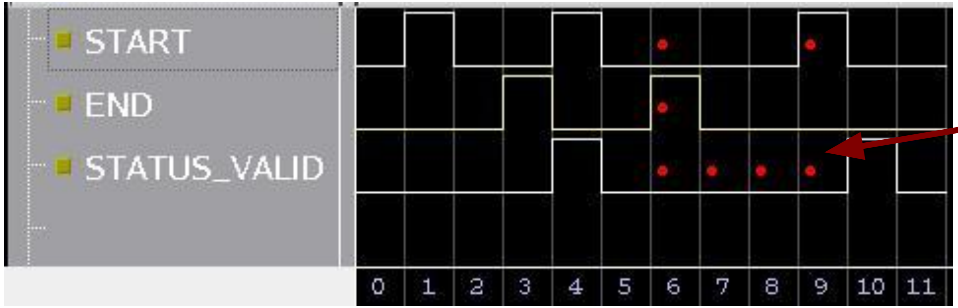
Counterexamples in hardware

A huge timing diagram that is very difficult to understand



Explaining counterexamples using causality (Red Dots) part of IBM tool

A timing diagram of a buggy hardware execution

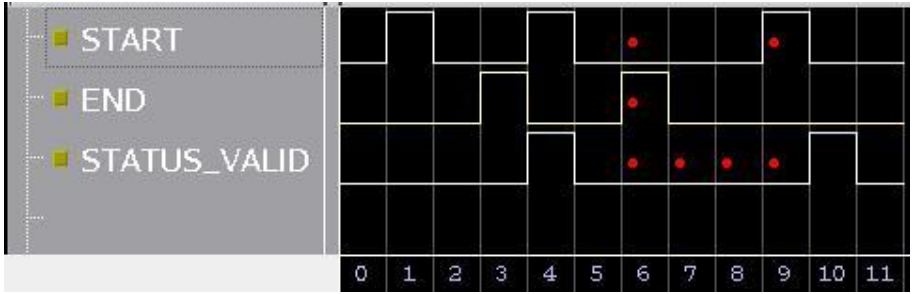


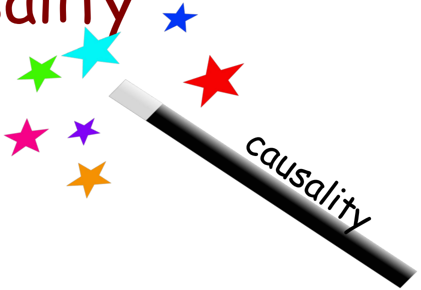
causes marked as red dots

φ = *always* ((!START *and* !STATUS_VALID *and* END) ->
next(!START Until (STATUS_VALID *and* READY))

works and is really useful!

Explaining counterexamples using causality (Red Dots)
part of IBM tool





Following this work…

Many applications of causality and responsibility to software engineering

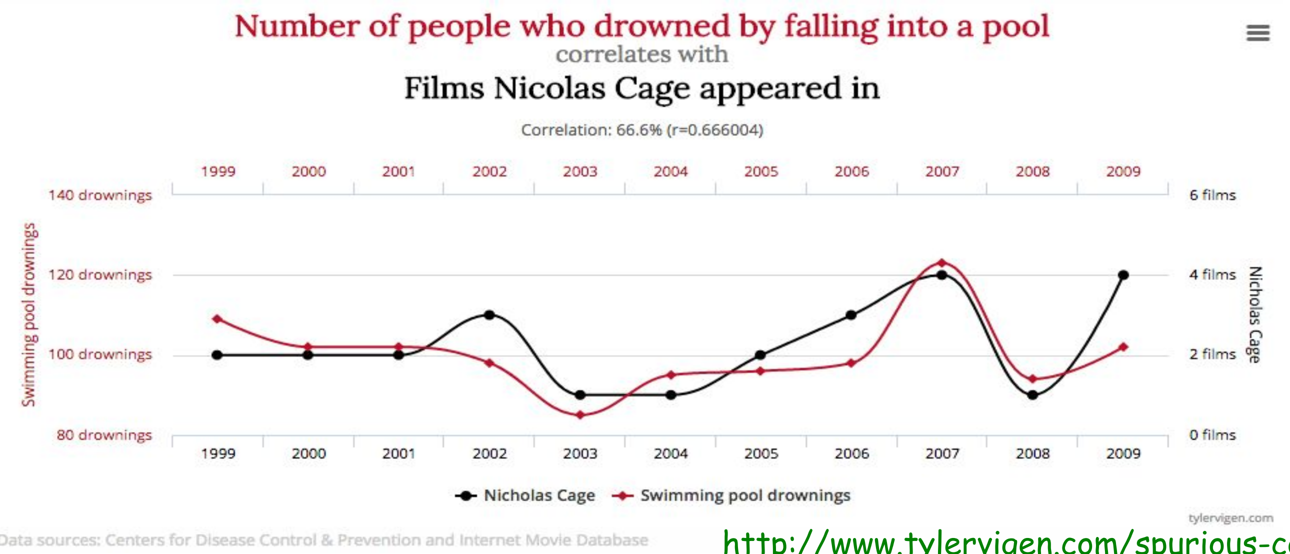
CREST workshop

ETAPS
EUROPEAN JOINT CONFERENCES ON THEORY & PRACTICE OF SOFTWARE

Ongoing work: causal debugging for software

Explanation of faults in software testing - SOA

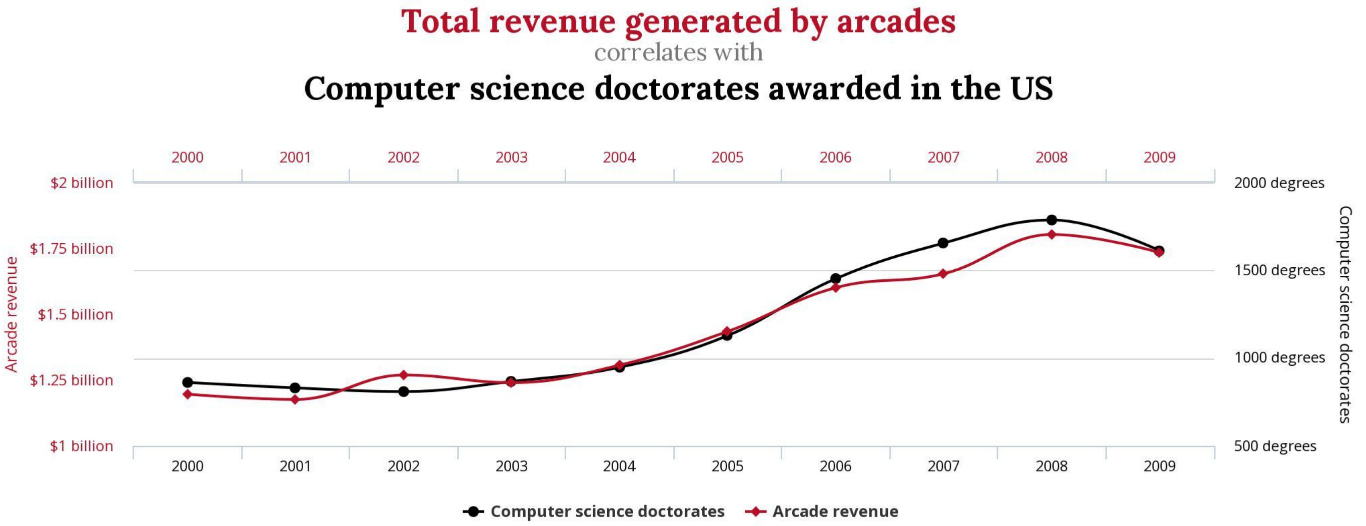
- **Statistical Analysis for Fault Localisation**
  - Looks for <u>**correlation**</u> – elements that appear more in failing traces than in passing ones are suspicious
  - Elements are ordered by their degree of suspiciousness



http://www.tylervigen.com/spurious-correlations

Explanation of faults in software testing - SOA

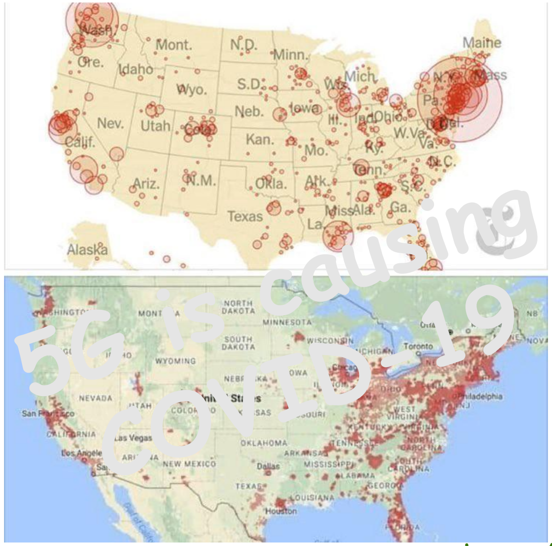
- **Statistical Analysis for Fault Localisation**
  - Looks for **correlation** – elements that appear more in failing traces than in passing ones are suspicious
  - Elements are ordered by their degree of suspiciousness



http://www.tylervigen.com/spurious-correlations

Explanation of faults in software testing - SOA

- **Statistical Analysis for Fault Localisation**
  - o Looks for **correlation** – elements that appear more in failing traces than in passing ones are suspicious
  - o Elements are ordered by their degree of suspiciousness

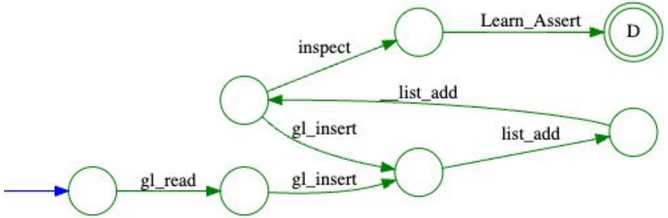


http://www.medium.com

Explanation of faults in software testing - SOA

- **Statistical Analysis for Fault Localisation**
  - Looks for **correlation** – elements that appear more in failing traces than in passing ones are suspicious
  - Elements are ordered by their degree of suspiciousness

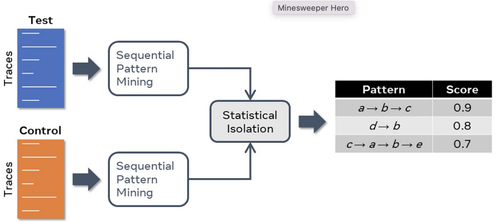
Learning the language of software errors



Recent work from Meta

Minesweeper automates root cause analysis as a first-line defense against bugs

not causal



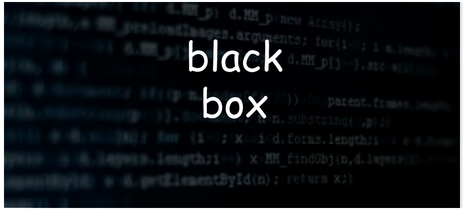
Pattern	Score
$a \rightarrow b \rightarrow c$	0.9
$d \rightarrow b$	0.8
$c \rightarrow a \rightarrow b \rightarrow e$	0.7

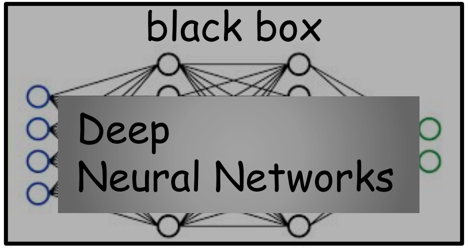
Motivation :

Modern computerized systems are huge and difficult or even impossible to understand

Can we understand and fix errors?

What does the system do?





Can we be sure it is correct?

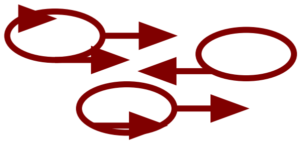
verification

Formal Verification (Model Checking)

?Is the system correct

A huge and difficult to understand system M:

verified



A correctness specification φ

Does M satisfy φ?

no

counter example

yes

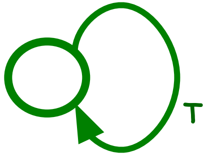
Do we really know this?

Vacuity – the main idea

Vacuous satisfaction of φ in M means that some part of φ is irrelevant in M

Printer that doesn't print

system M:



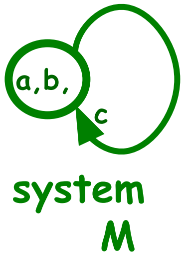
ϕ = always (req -> eventually grant)

Vacuous pass can point to problems

What is the output of vacuity check?

Vacuous satisfaction of φ in M means
that some part of φ is irrelevant in M

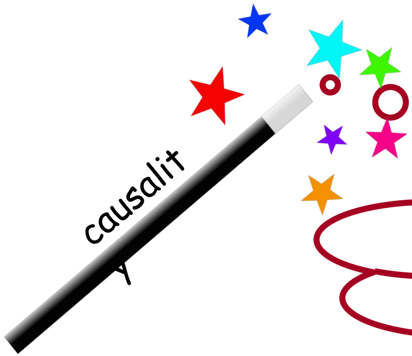
Standard vacuity checks output (some) redundant parts of ϕ



ϕ = always (a OR b OR c) → vacuity →

a is redundant:
ϕ' = always (b OR c) holds

b and c are also redundant (separately)



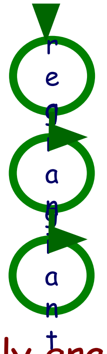
1. Compute counterfactual causes
2. Everything else is a vacuity output.
3. Can be ranked according to importance.
4. Can find non-causes.

Coverage – the main idea

Low coverage of M by φ means that some part of M is irrelevant for the satisfaction of φ

Printer that prints everything twice

system M:



ϕ = always (req -> eventually grant)

Low coverage can point to problems

What is the output of coverage check?

Low coverage of M by φ means that some parts of M are irrelevant for the satisfaction of φ

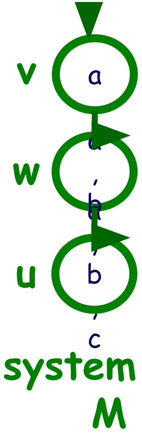
There is no standard coverage check… but if there was one…

What is the output of coverage check?

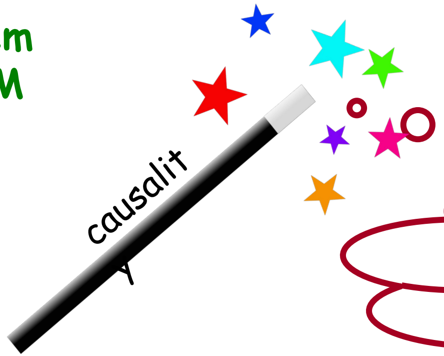
Low coverage of M by φ means that some parts
o

Impractical for huge and difficult to understand systems
(so there is a good reason it is not done)
… but can be a good idea for small critical systems

There



ϕ = al



c is completely irrelevant; a,b are causes

1. Compute counterfactual causes
2. Everything else is not covered.
3. Can be ranked according to importance.
4. Can find non-causes.

Why is verification a good application for causality?

causality

- **Interventions are always possible**
  - o An intervention amounts to a change in the value of a variable
  - o Unlike other domains, where changes can be impossible (like healthcare)

x=y → x=1

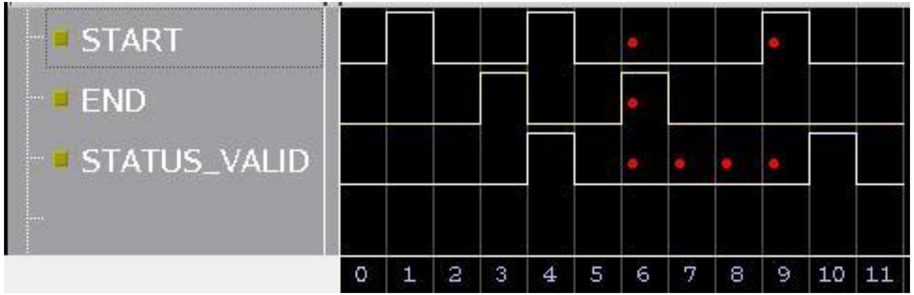
✓

Now let's replace you with the same person but without peanut allergy

✗

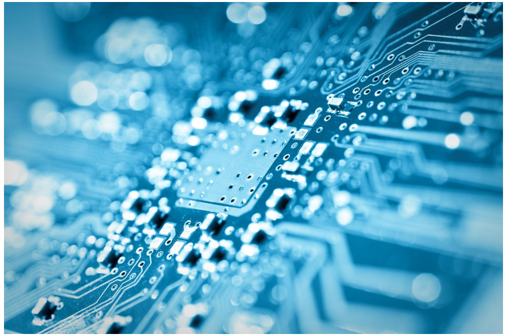
Why is verification a good application for causality?

- Interventions are always possible
- It is usually clear what the variables are and easy to calculate the equations
  - o Constructing the right model = ½ of the answer
  - o In many domains, constructing the right model is challenging
  - o An ongoing discussion in philosophy
  - o Fortunately, we are not in philosophy



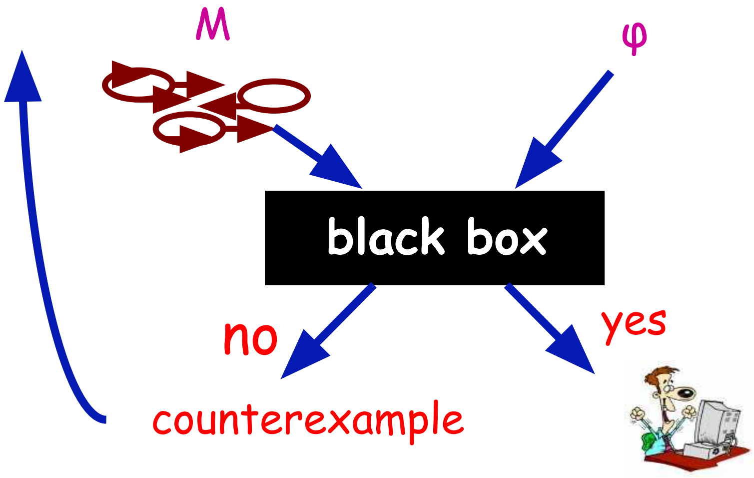
Why is verification a good application for causality?

- Interventions are always possible
- It is usually clear what are the variables and easy to calculate the equations
- The systems are deterministic and all variables are known
  - No noise, no hidden confounders
  - Not quite true for concurrent systems, but still better than in other domains



Why is verification a good application for causality?

- Interventions are always possible
- It is usually clear what are the variables and easy to calculate the equations
- The systems are deterministic and all variables are known
- The approach is agnostic to the model-checking algorithm

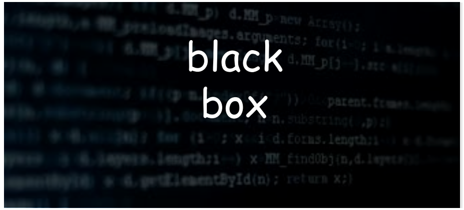


Motivation:

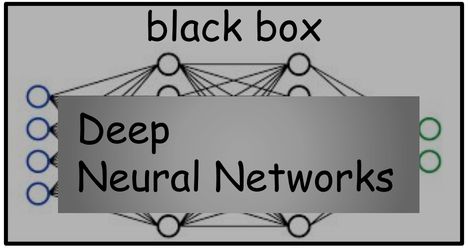
Modern computerized systems are **huge and difficult or even impossible to understand**

Can we understand and fix errors?

What does the system do?

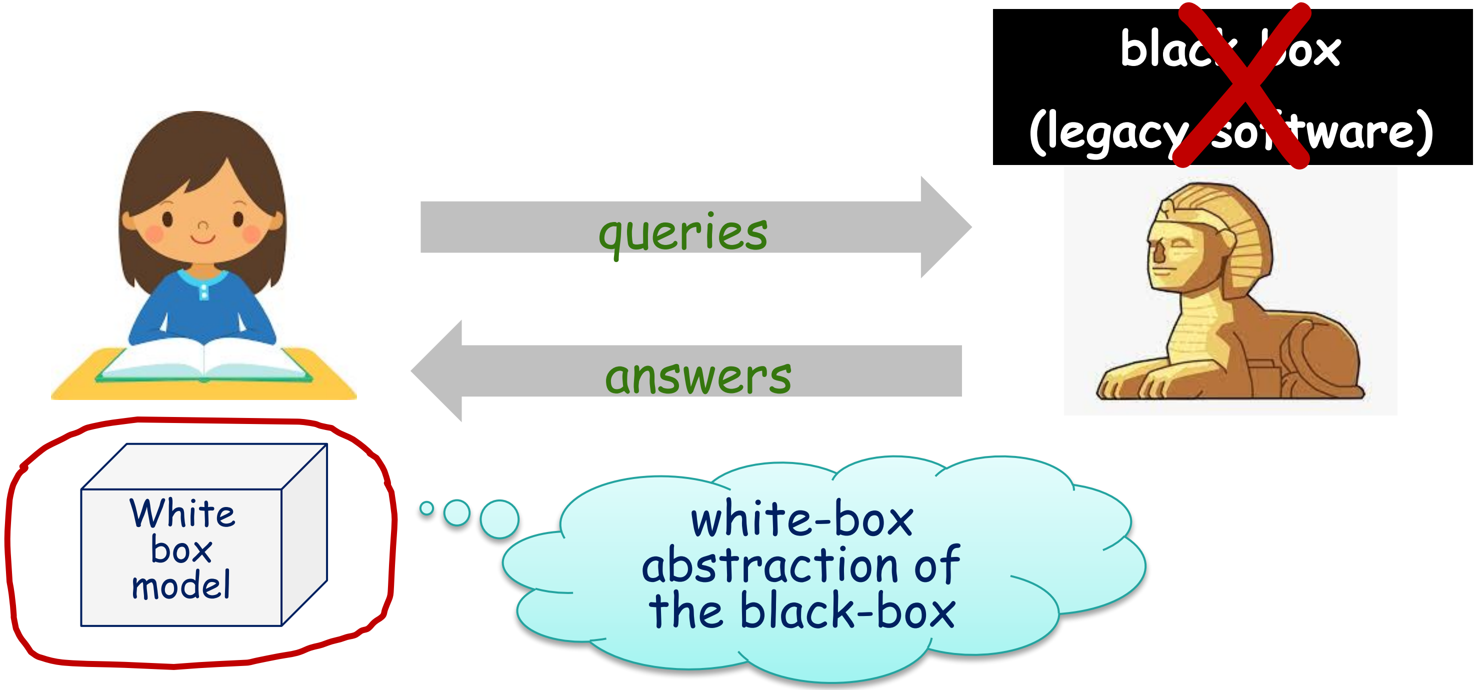


Can we be sure it is correct?



©Vaandrager (many papers)

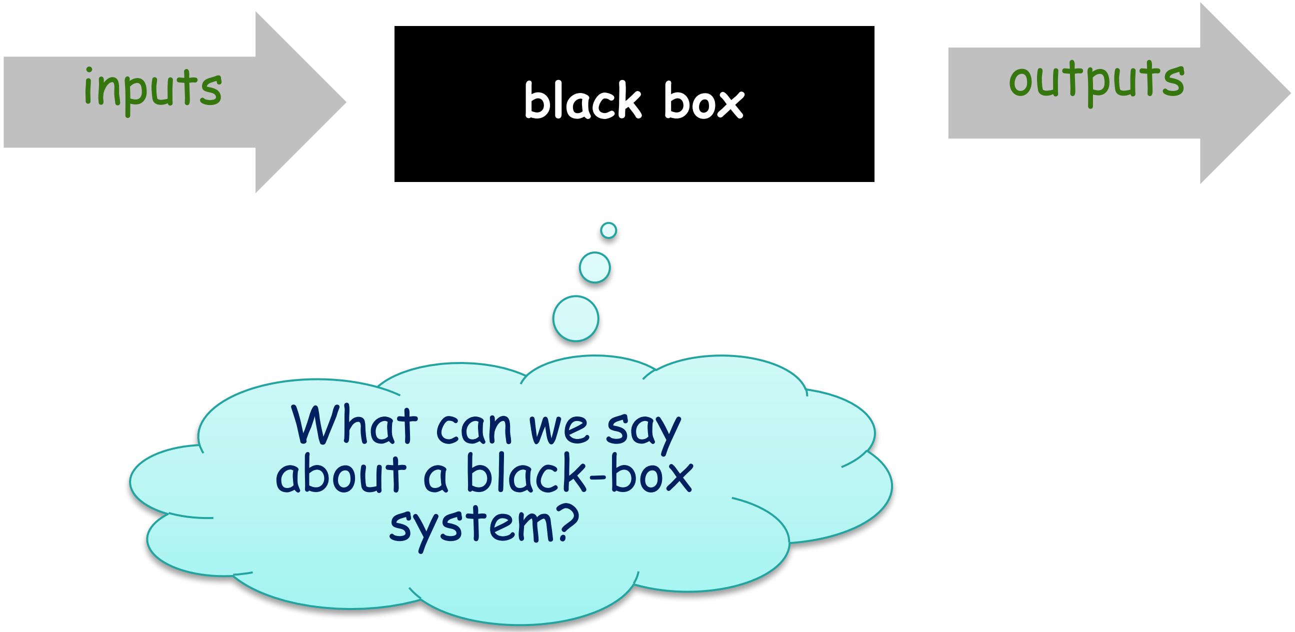
Model learning



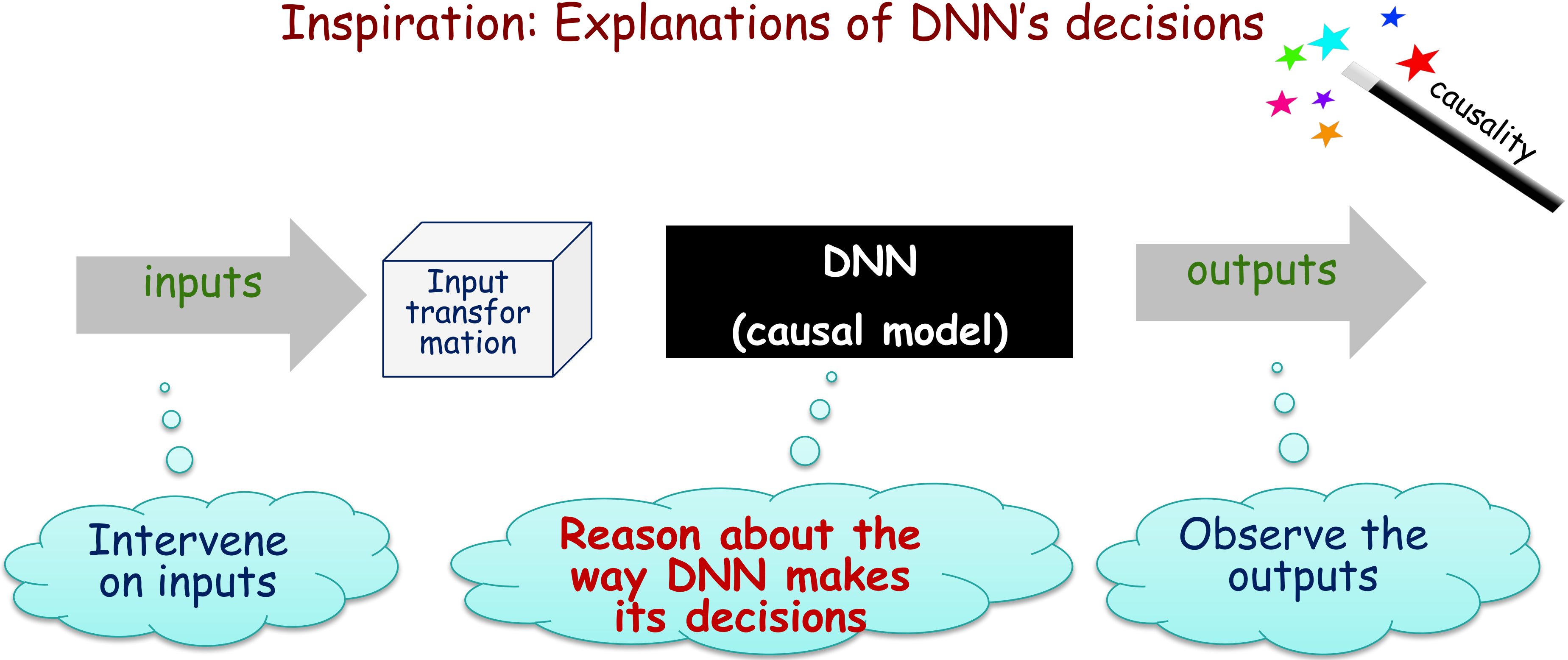
Can be viewed as a causal model

Reasoning about black-boxes

?Do we need to construct a white box at all



Inspiration: Explanations of DNN's decisions



We can reason about various properties of the system without opening the black box

Explanations for Deep Neural Network's decisions

DNN for
classifying animals

red panda

Causal explanation:
minimal, sufficient,
non-trivial subset of
the pixels of the image

Because
of this part:

Subtle misclassifications – uncovered by explanations

DNN for
classifying images

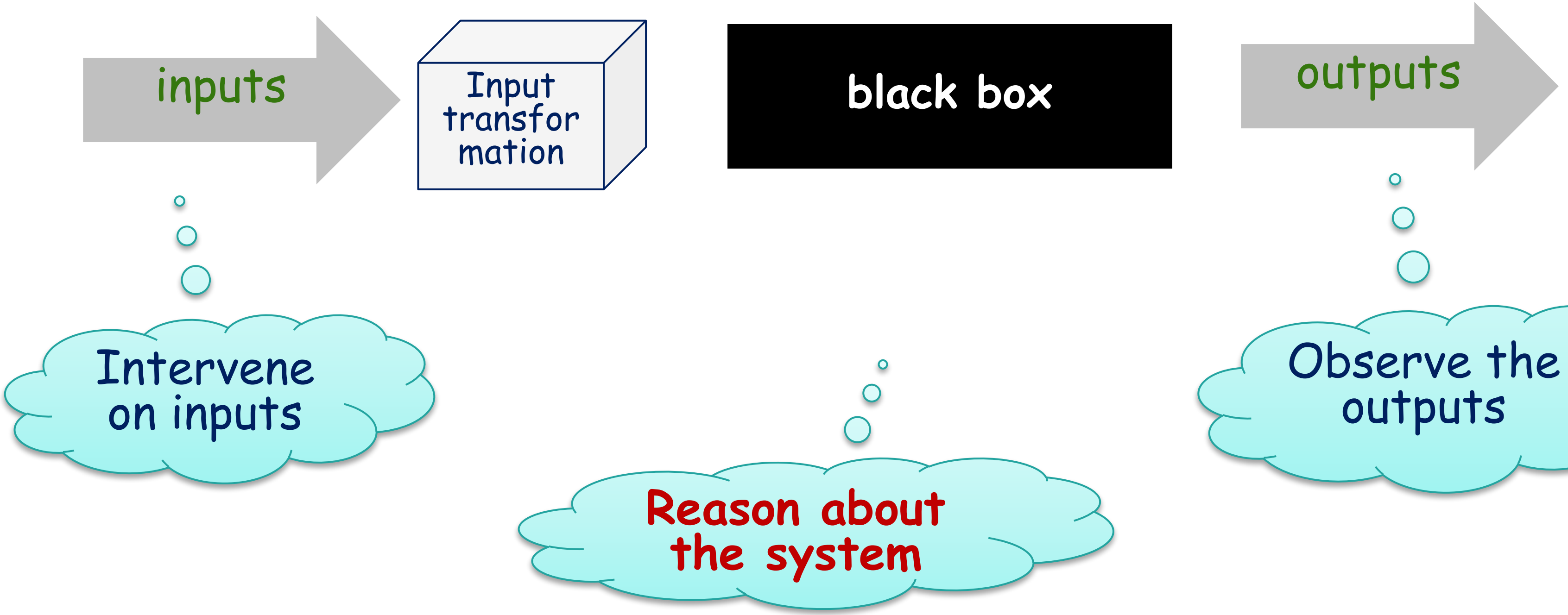
cowboy hat

seems
ok

Because
of this part: 

Explanation
uncovered
misclassification!

Can we use a similar approach to answer the question
? “?What does the system do“



Can we use a similar approach to answer the question

? “?What does the system do“

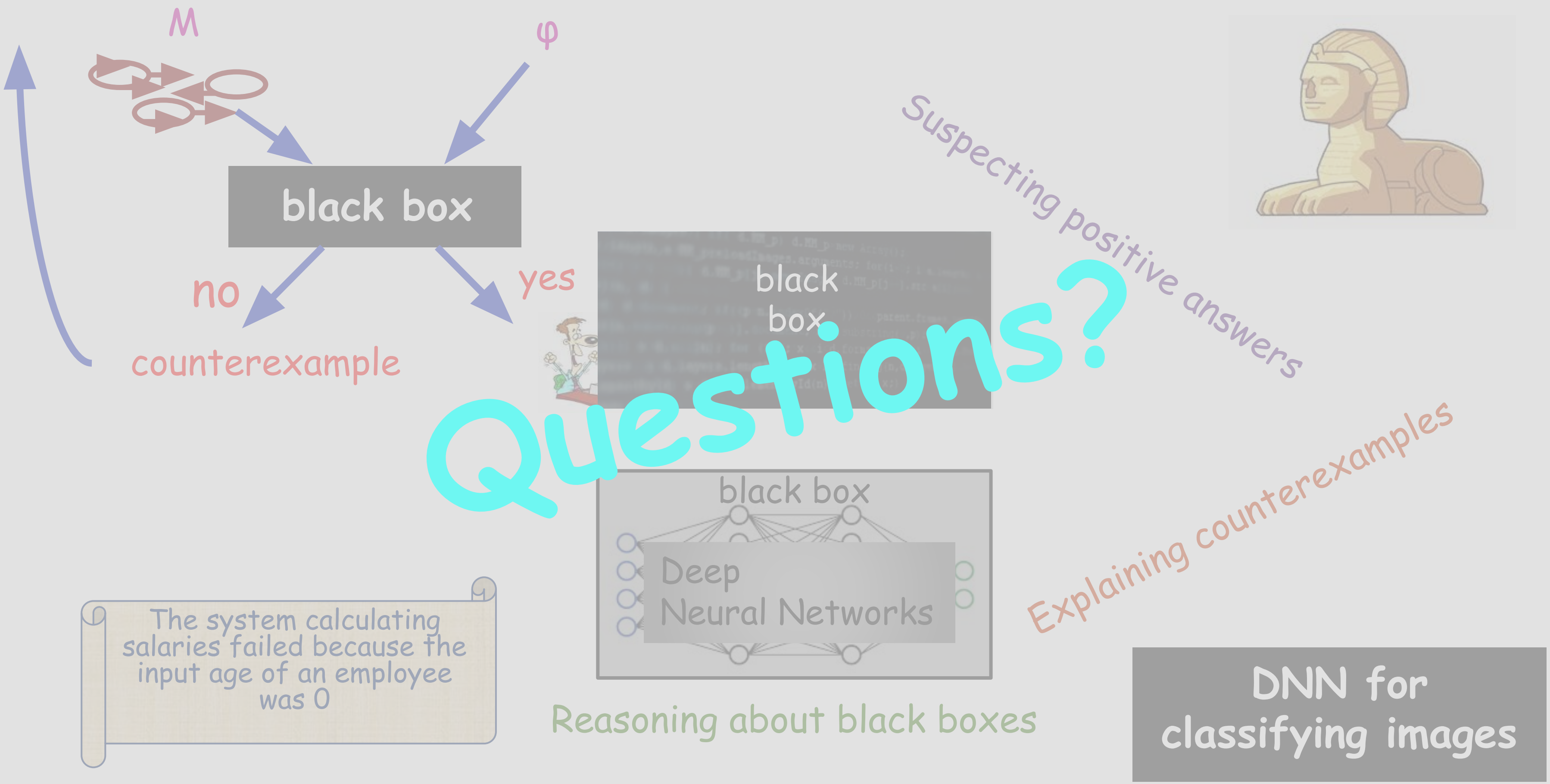
The answer will be in terms of the inputs

The system calculating salaries failed because the input age of an employee was 0

Not model learning, but can help to understand the system

Bibliography

- Chockler and Halpern. "Responsibility and Blame: A Structural-Model Approach". J. Artif. Intell. Res. 22: 93-115 (2004)
- Chockler, Halpern, Kupferman. What causes a system to satisfy a specification? ACM Trans. Comput. Log. 9(3): 20:1-20:26 (2008)
- Beer, Ben-David, Chockler, Orni, Trefler. "Explaining Counterexamples Using Causality". FMSD (2012)
- Chockler, Gurfinkel, Strichman: Beyond vacuity: towards the strongest passing formula. FMSD (2013)
- Aleksandrowicz, Chockler, Halpern, Ivrii. "The Computational Complexity of Structure-Based Causality". AAAI'14: 974-980.
- Alrajeh, Chockler, Halpern. "Combining Experts' Causal Judgments". Artif. Intell. (2020).
- Chockler, Kesseli, Kroening, Strichman. "Learning the Language of Software Errors". J. Artif. Intell. Res. 67: 881-903 (2020).
- Sun, Chockler, Huang, Daniel Kroening. "Explaining Image Classifiers Using Statistical Fault Localization". ECCV'20: 391-406.
- Chockler, Kroening, Sun. "Explanations for Occluded Images". ICCV'21: 1234-1243.
- Pouget, Chockler, Sun, Kroening. "Ranking Policy Decisions". NeurIPS'21.


M
φ
black box
no
yes
counterexample
black
box
Questions?
Suspecting positive answers
black box
Deep
Neural Networks
Explaining counterexamples
The system calculating salaries failed because the input age of an employee was 0
Reasoning about black boxes
DNN for classifying images