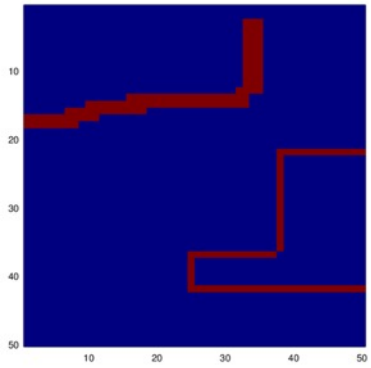
TrinityAI: On Computing Relevant Parameters of Decision Functions

Susmit Jha Patrick Lincoln

Computer Science Laboratory
SRI International

Example Decision Making System: Path planning

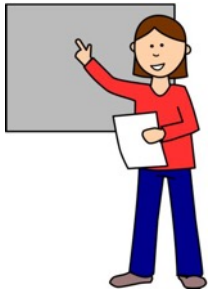
Why did a path planning algorithm (A*) pick a particular segment in the optimal path?
E.g. The optimal path goes via Bridge B because Bridge A is blocked.

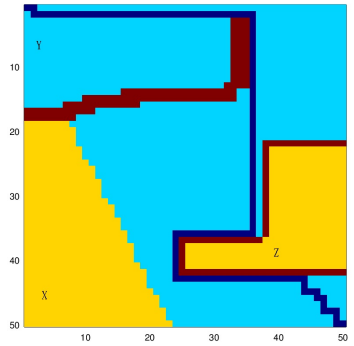


Maps

Assignments to Input
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)

A*





Does plan satisfy ϕ?
E.g. Plan goes via some segment

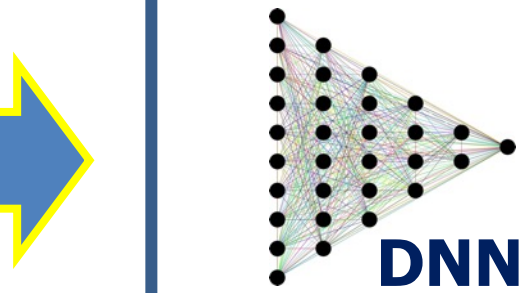
Output of evaluating ϕ
O1 = (0)
O2 = (1)

Example Decision Making System: Fairness in ML

Did race of an individual influence the ML model making decisions about loan eligibility?

-0.5	0.16	1.59	10.32	0.15	0.08	12.61	0.25	8.51	1.74
-1	0.17	1.67	10.33	0.16	0.09	11.96	0.25	7.71	1.46
-1.5	0.18	1.76	10.35	0.17	0.09	11.39	0.24	7.03	1.24
-2.5	0.19	1.92	10.38	0.19	0.11	10.40	0.24	5.95	0.92
-5	0.23	2.34	10.47	0.22	0.14	8.55	0.23	4.19	0.49
-7.5	0.27	2.73	10.55	0.26	0.17	7.32	0.21	3.21	0.29
-10	0.31	3.12	10.62	0.29	0.19	6.42	0.20	2.61	0.19
-12.5	0.35	3.49	10.70	0.33	0.22	5.73	0.19	2.21	0.13
-15	0.39	3.86	10.77	0.36	0.25	5.18	0.18	1.93	0.09
-17.5	0.42	4.24	10.85	0.39	0.29	4.72	0.16	1.72	0.07
-20	0.46	4.62	10.92	0.42	0.32	4.33	0.15	1.57	0.05

Individual Data

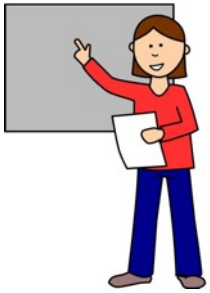


ϕ

Loan

No Loan

Does ML model satisfy ϕ?
E.g. Does an individual with income above 60K get loan?



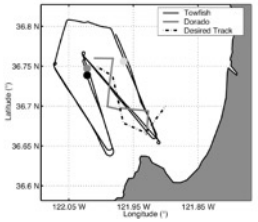
Output of evaluating ϕ
O1 = (0)
O2 = (1)

Assignments to Input
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)

Example Decision Making System: AI Co-Designer for CPS

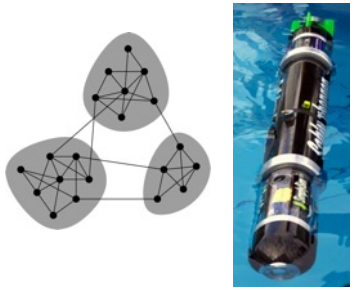
What part of the specification influenced the choice of the optimal propulsion method?

Requirement	Metric
Total system weight	2000 kg or less
Transit distance	70 km
Trackline distance	10,000km
Depth requirement	300 m
Transit hotel	Determined by design
Survey payload power	See spec sheets
Shoreside turnaround time	18 hr



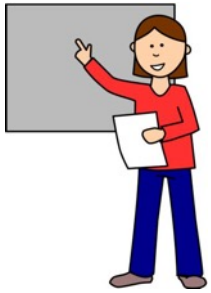
Mission Requirements

NuSymCORE



Does AI co-designer satisfy ϕ?
E.g. Why to use buoyancy for propulsion and not propellers?
`MaxVelocity < 0.1m/s and Endurance > 16 hrs`

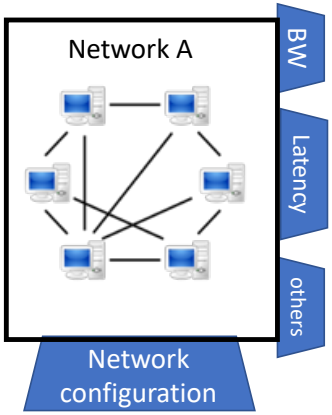
Output of evaluating ϕ
O1 = (0)
O2 = (1)



Assignments to Input
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)

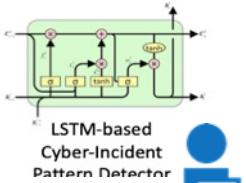
Example Decision Making System: SDN Cyber Defense Policy

What part of the state of the network make the SDN policy to block a particular IP?

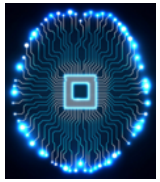


SDN with Real/Simulated Traffic

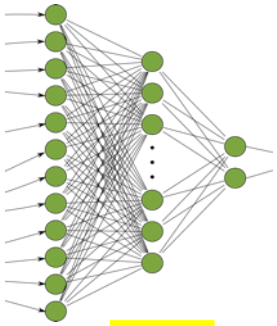
Assignments to Input
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)



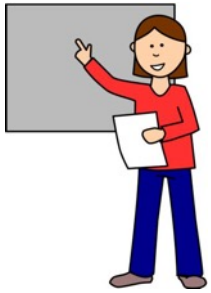
Expert Response Rules



Continuous Deep Reinforcement Learning (RL)



Cyber Defense Policy



Does the policy satisfy ϕ?
E.g. Why was traffic from a particular IP blocked?
`Past behavior of the IP, neighbor IPs`

Output of evaluating ϕ
O1 = (0)
O2 = (1)

Decision Making System

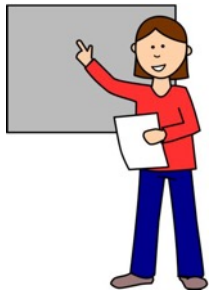
Why did the decision making system take a particular decision for a given input?

Assignments to Input
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)

Synthesizer

Complex Algorithm or ML model

Output of the algorithm or model



Oracle

Check whether the output satisfies some property ϕ of interest ?

Output of evaluating ϕ
O1 = (0)
O2 = (1)

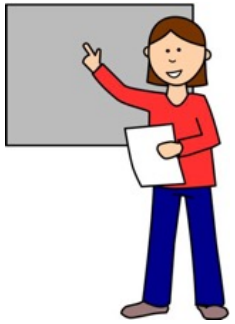
Decision Making System

Oracle Guided Formal Synthesis

$\mathcal{C}$
Concept Class

Synthesizer

Positive Witness
$x \in \phi$, if one exists, else $\perp$
Negative Witness
$x \notin \phi$, if one exists, else $\perp$
Membership: Is $x \in \phi$?
Yes / No
Equivalence: Is $f = \phi$?
Yes / No + $x \in \phi \oplus f$
Subsumption/Subset: Is $f \subseteq \phi$?
Yes / No + $x \in f \setminus \phi$
Distinguishing Input: f, $X \subseteq f$
f' s.t. $f' \neq f \wedge X \subseteq f'$, if it exists; o.w. $\perp$



Oracle

A dialogue is a sequence of (query, response) confirming to an oracle interface O

An Oracle-guided formal synthesis algorithm is a pair <L, T> where

- L is a learner, a non-deterministic algorithm mapping a dialogue to a concept c and query q
- T is an oracle/teacher, a non-deterministic algorithm mapping a dialogue and query to a response r

An Oracle-guided formal synthesis algorithm <L,T> solves a synthesis problem if there exists a dialogue between L and T that converges in the target concept $f \in C$

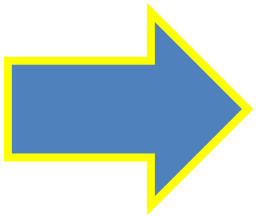
- Programs: ICSE'10 (MIP Award at ICSE'20), PLDI'11, DTTC'13, NSV'14, Acta Informatica'17
- Controllers: ICCPS'10, EMSOFT'11, IJBRA'12, FORMATS'16, FORMATS'18, Allerton'18, ACC'19
- Explanations and intent: NFM'17, RV'17, NFM'18, JAR'18, NeurIPS'18, FMSD'19

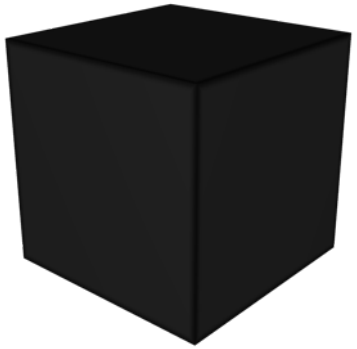
Problem Setup

A black box function with N inputs where its output depends only on a small subset of size k << N.

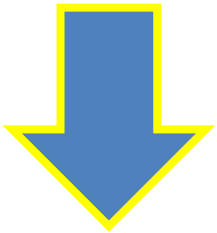
In practice: The black box produces an output and we are interested in some specific property of this output which depends on only a small subset of inputs.

Assignments to Input Variables
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)





ϕ



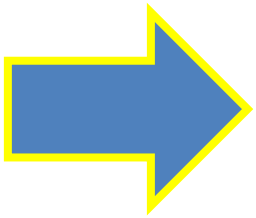
Set of possible explanations =
Set of Boolean formula with N variables =
O (Double exponential in N)

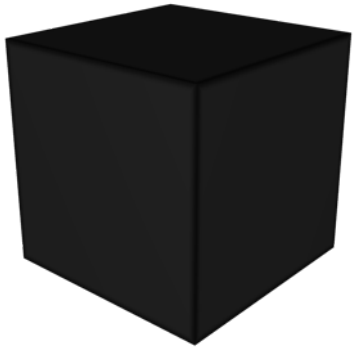
Output
O1 = (0)
O2 = (1)

Hamming Distance Based Search

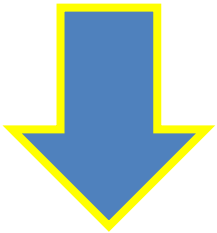
Start with a random assignment to variables

Assignments to Input Variables
m1 = (0,0,0,1,1,0,1)





ϕ

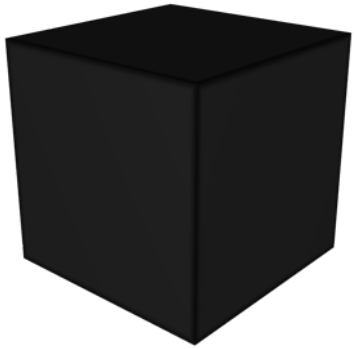


Output
O1 = (0)

Hamming Distance Based Search

Randomly sample assignments till the blackbox produces a different output, that is, O1 ≠ O2

Assignments to Input Variables
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)



ϕ

Output
O1 = (0)
O2 = (1)

Hamming Distance Based Search

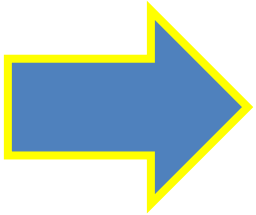
From these two assignments, do a Hamming distance based binary search to find two assignments where the blackbox produces different output and the assignments differ in exactly one input variable.

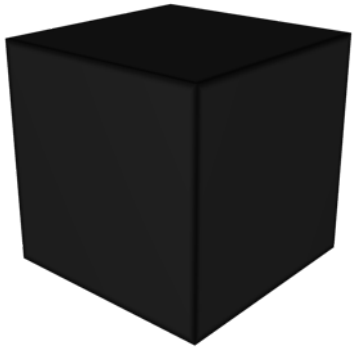
Assignments to V

m1 = (0,0,0,1,1,0,1)

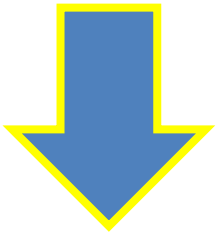
m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)





ϕ



Output

O1 = (0)

O2 = (1)

Hamming Distance Based Search

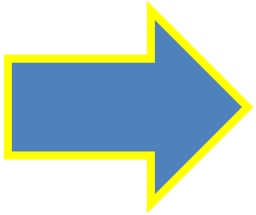
From these two assignments, do a Hamming distance based binary search to find two assignments where the blackbox produces different output and the assignments differ in exactly one input variable.

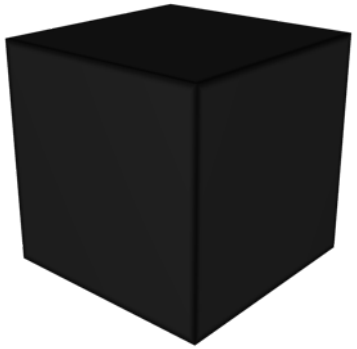
Assignments to V

m1 = (0,0,0,1,1,0,1)

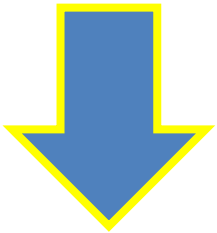
m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)





ϕ



Output

O1 = (0)

O2 = (1)

O3 = (0)

Hamming Distance Based Search

From these two assignments, do a Hamming distance based binary search to find two assignments where the blackbox produces different output and the assignments differ in exactly one input variable.

Hamming Distance = 4

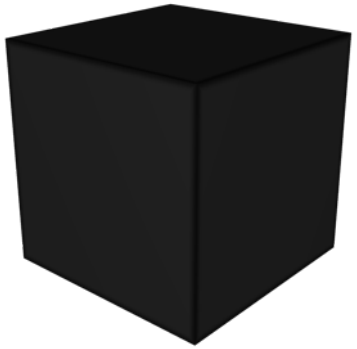
Hamming Distance = 2

Assignments to V

~~m1 = (0,0,0,1,1,0,1)~~

m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)



ϕ

Output

~~O1 = (0)~~

O2 = (1)

O3 = (0)

Hamming Distance Based Search

From these two assignments, do a Hamming distance based binary search to find two assignments where the blackbox produces different output and the assignments differ in exactly one input variable.

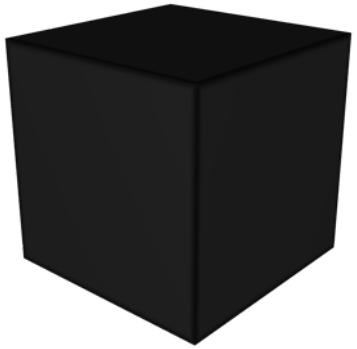
Hamming Distance = 2

Assignments to V

m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)

m4 = (0,0,1,1,1,1,0)



ϕ

Output

O2 = (1)

O3 = (0)

O4 = (0)

Hamming Distance Based Search

From these two assignments, do a Hamming distance based binary search to find two assignments where the blackbox produces different output and the assignments differ in exactly one input variable.

Hamming Distance = 2

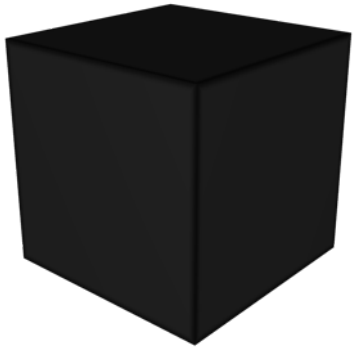
Hamming Distance = 1

Assignments to V

m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)

m4 = (0,0,1,1,1,1,0)



ϕ

Output

O2 = (1)

O3 = (0)

O4 = (0)

Hamming Distance Based Search

From these two assignments, do a Hamming distance based binary search to find two assignments where the blackbox produces different output and the assignments differ in exactly one input variable.

Hamming Distance = 2

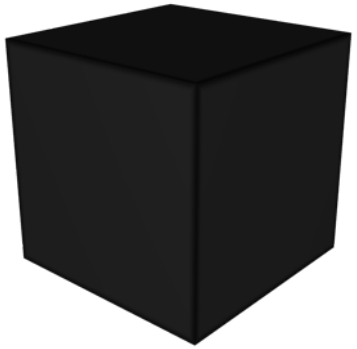
Hamming Distance = 1

Assignments to V

m2 = (0,0,1,1,0,1,0)

~~m3 = (0,0,0,1,1,1,0)~~

m4 = (0,0,1,1,1,1,0)



ϕ

Output

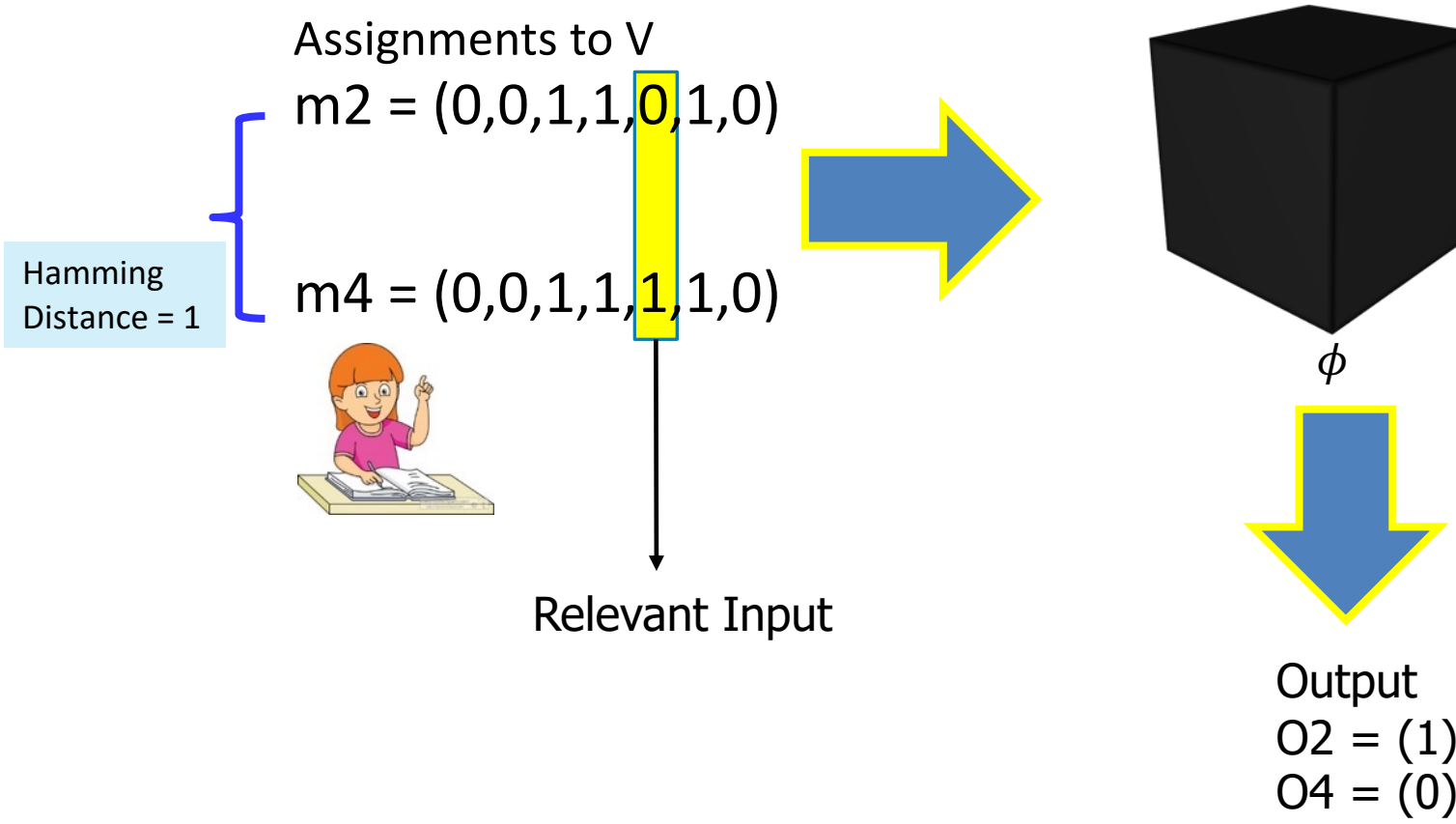
O2 = (1)

~~O3 = (0)~~

O4 = (0)

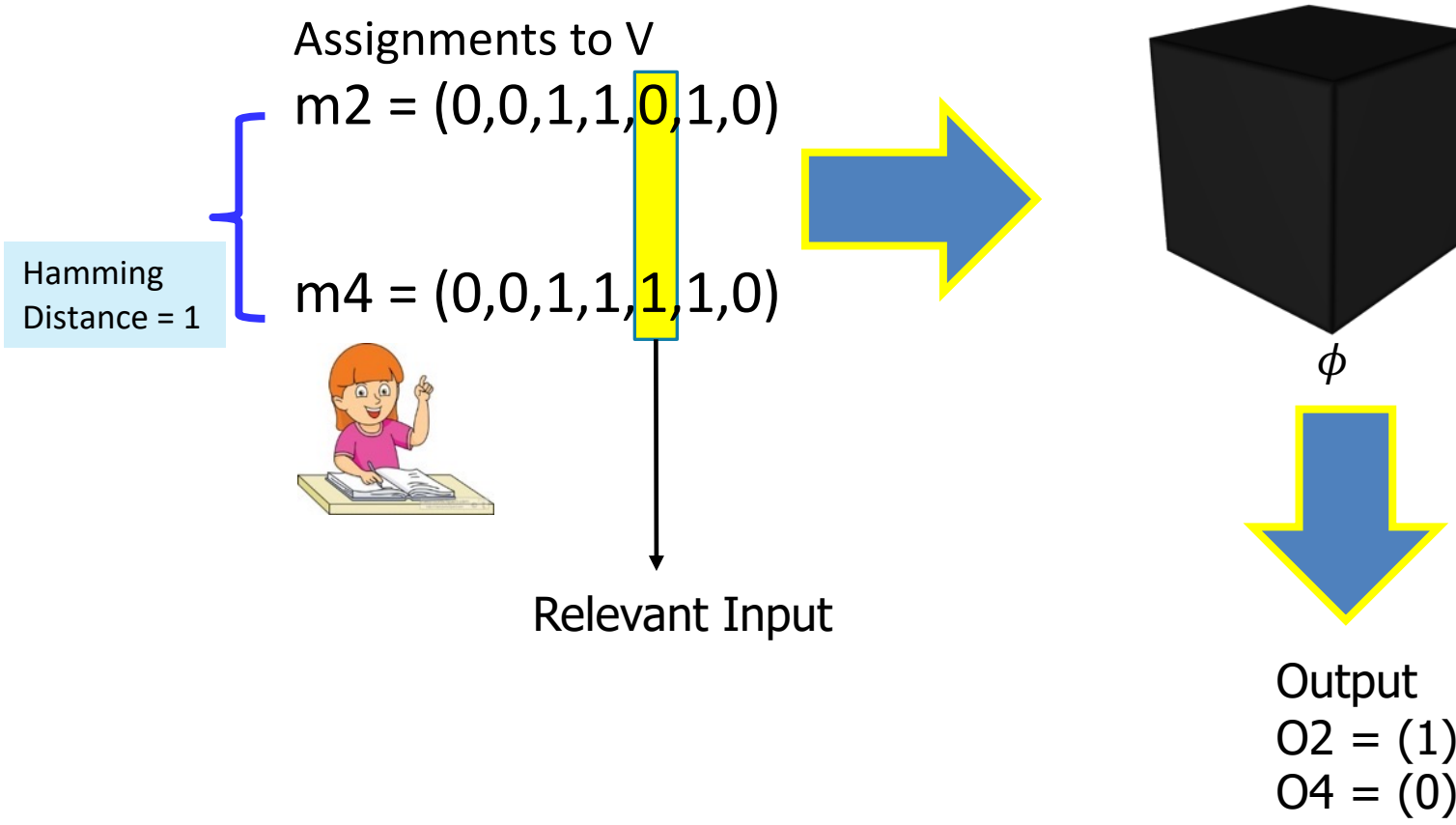
Hamming Distance Based Search

From these two assignments, do a Hamming distance based binary search to find two assignments where the blackbox produces different output and the assignments differ in exactly one input variable.



Hamming Distance Based Search

Fix relevant input to each of the two possible values and solve the problem of finding relevant inputs for N-1 inputs.

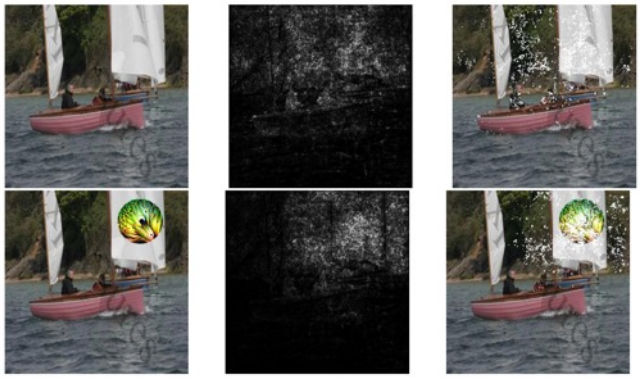


Relevant variables can be found with confidence κ in $2^{2|U|}ln(|V|/(1-\kappa))$ for κ PAC guarantee queries to the oracle.

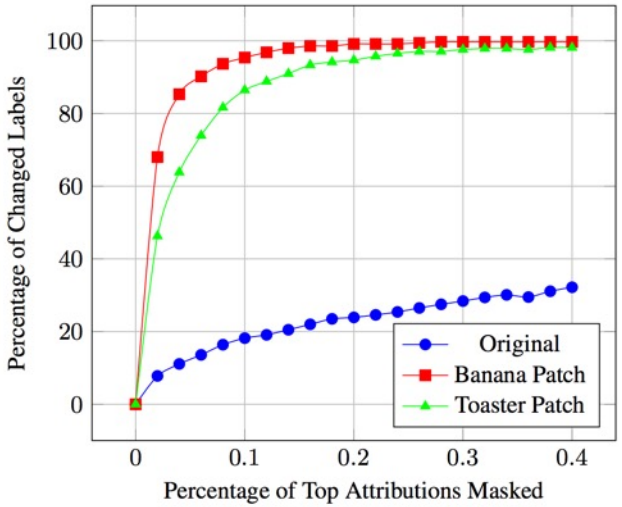
V : set of all variables U : set of relevant variables

Reactive Exploration Strategy
|V| = 96
|U| <= 2
Runtime < 5 seconds

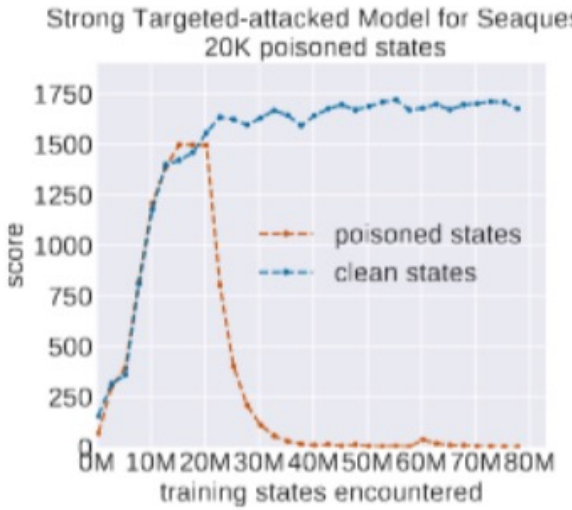
Explaining A* Planning
|V| = 2500
|U| <= 4
Runtime < 3 minutes

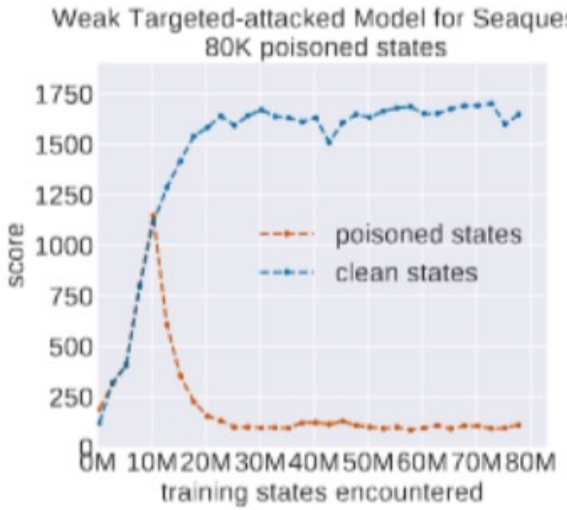


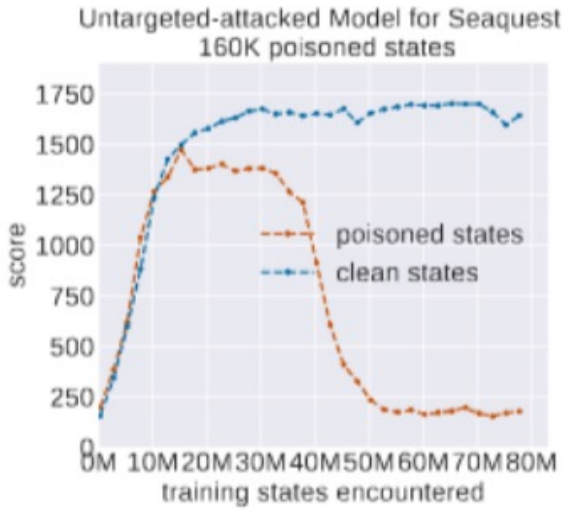
(Jha et al. NeurIPS'19: Confidence, Detecting adv attacks)

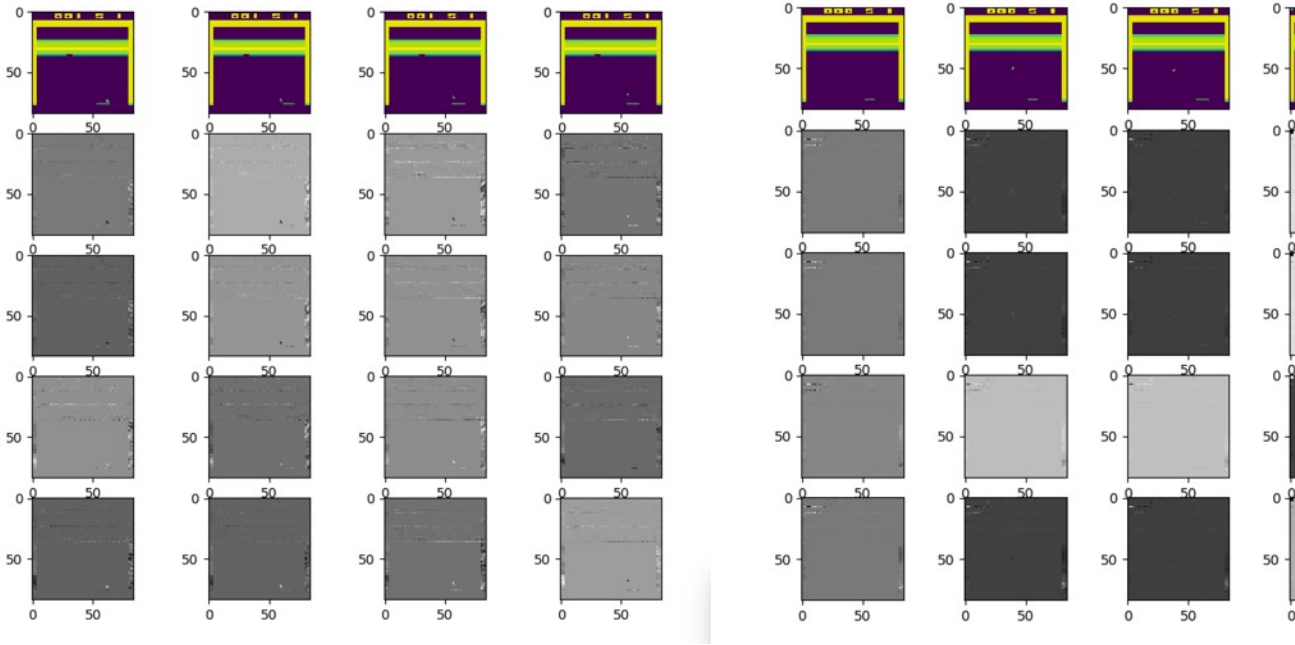


Quantitative Relevance and Trojan Detection in RL (DAC'20)





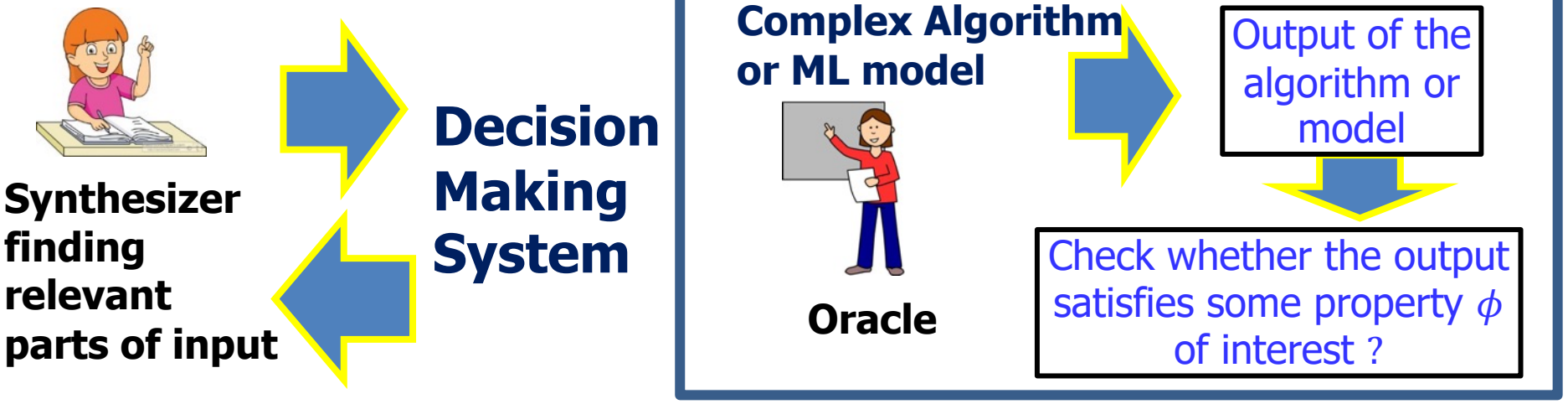




Conclusion

Part of the Trinity – a neurosymbolic AI system being built at SRI

- Ack: DARPA Assured Autonomy, DARPA Symbiotic Design of CPS, IARPA TrojAI, ARL IoBT



Why did the decision making system take a particular decision for a given input?

- Qualitative Relevant Inputs can be found using queries logarithm in the input dimension.
- Applied to a variety of applications where complex decision making algorithms need to be explained.