

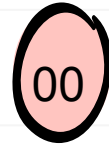


Axiom Pinpointing

Justifying consequences in Automated Reasoning

M. Fareed Arif

April 27 2020



Research Theme

Research Theme

My research interest lies in applying **formal verification** to ensure the **safety** and **security** of Computational Systems

Research Theme

[Postdoctoral Researcher]

VERDICT: Cyber Resilience Verifier



[DASC19]

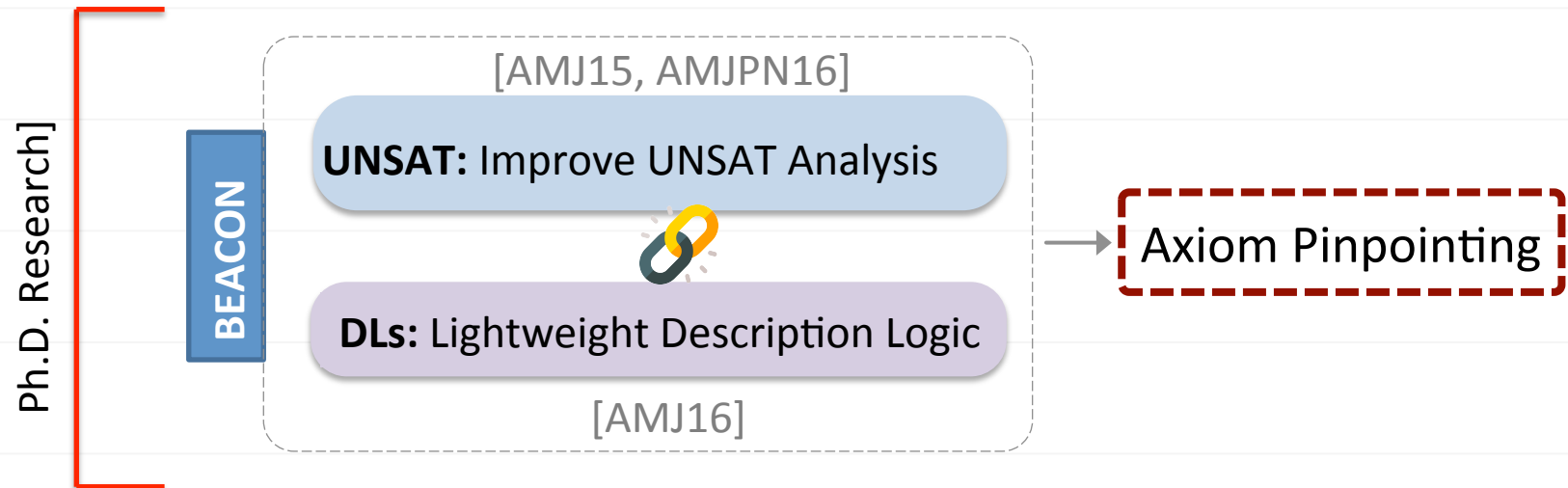
PHOENIX: PLTL Synthesizer

[USENIX security20, FMCAD20]

HDR: Hardware Model Checker

[HDR Competition 2020]

Research Theme

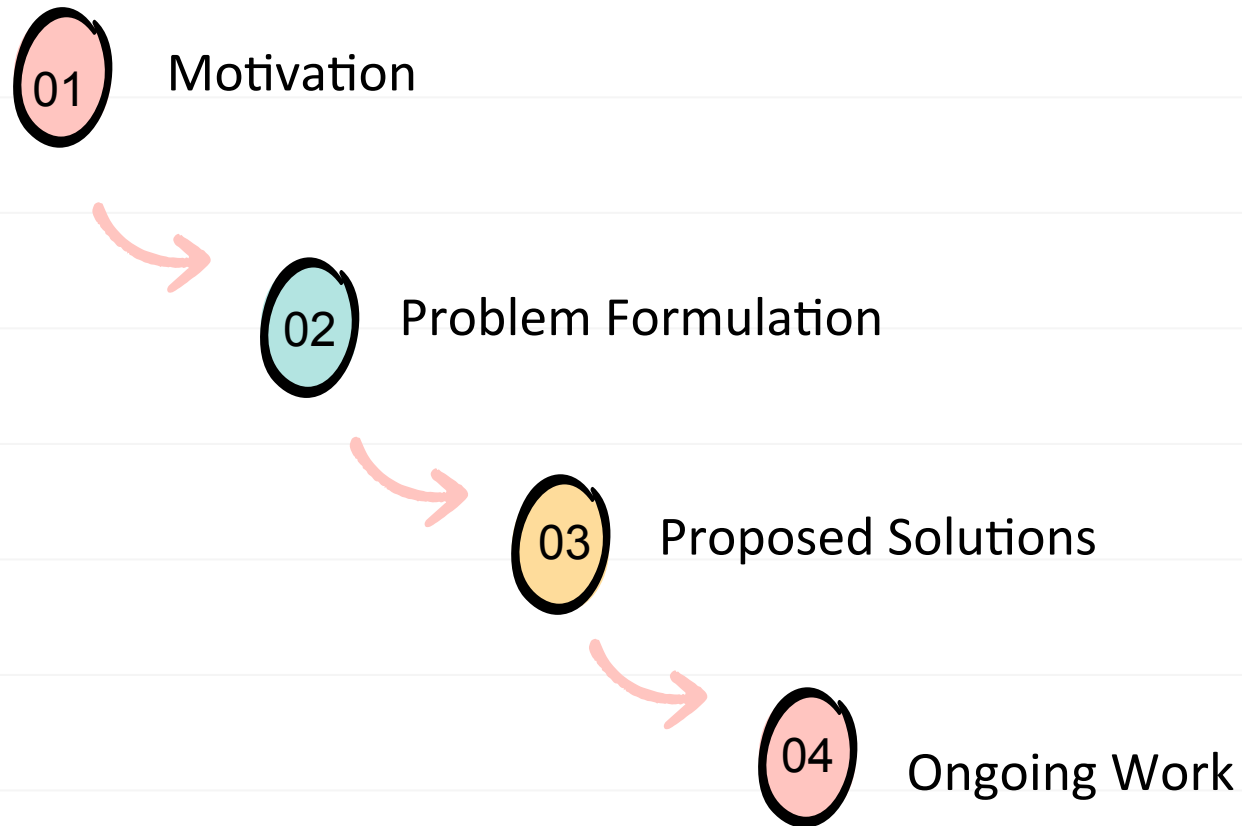




Axiom Pinpointing

Justifying consequences using
Automated Reasoning

Agenda of the Talk



Motivation

Intelligent applications need to represent and handle knowledge effectively

Motivation

Intelligent applications need to represent and handle knowledge effectively

A Knowledge Representation Language, say \mathcal{K} , provides formal semantics and reasoning methods for driving an implicit consequence, called axiom α from explicitly represented elements.

$$\mathcal{K} \models \alpha$$

Motivation

A Knowledge Representation Language, say \mathcal{K} , provides formal semantics and reasoning methods for driving an implicit consequence, called axiom α from explicitly represented elements.

$$\mathcal{K} \models \alpha$$

Understand the error and repair to rectify the error

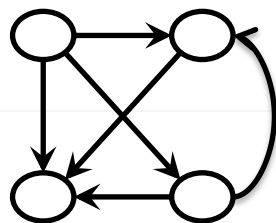
Debugging Ontologies

SNOMED-CT is a medical ontology used by U.S Federal Government systems
For the electronic exchange of clinical health information

$\text{AmpOfFinger} \sqsubseteq \text{AmpOfHand}$



Amputation of finger is an Amputation of a Hand [BP08]



Ontology O



Unexpected Behavior \mathcal{B}

Does O entails \mathcal{B} ?



Bug Explanation and Repair

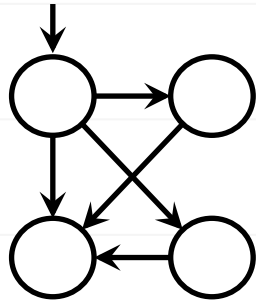
Motivation

A Knowledge Representation Language, say \mathcal{K} , provides formal semantics and reasoning methods for driving an implicit consequence, called axiom α from explicitly represented elements.

$$\mathcal{K} \models \alpha$$

If **Consequence** is a property of an interest then **explaining** it enhances both the correctness and acceptance of an intelligent computational system

Symbolic Model Checking



Abstract Model \mathcal{M}



Requirements \mathcal{R}

Does \mathcal{M} meets \mathcal{R} ?



Proof Explanation

Proof explanation in Symbolic Model Checking [GWG17]

Pinpointing lines in the source code responsible for an error [JM11]

Axiom Pinpointing:

Identifying the axioms in a \mathcal{K} that are responsible for a given **consequence** α

Identified axioms are called justification

A single **consequence** can have multiple justifications

Finding not only one but all justification-based explanations

Problem Relevance

Axiom pinpointing is a well-studied problem across several domains

Different research communities know **justification** by different names

[RR87]



MESCs

[MM14]



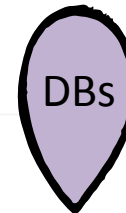
MUPS
MinAs

[SC03, BP07]



MUSes

[BL03, LS08]



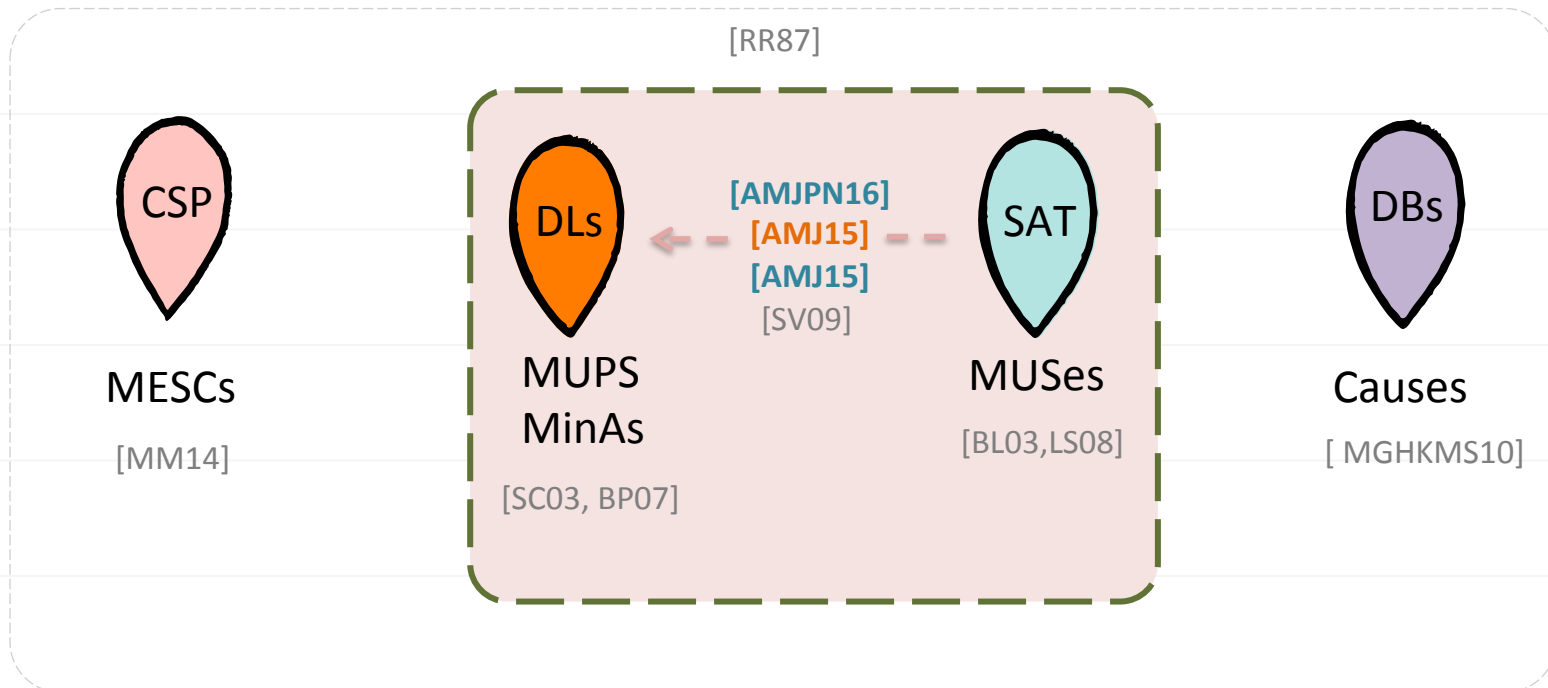
Causes

[MGHKMS10]

Problem Relevance

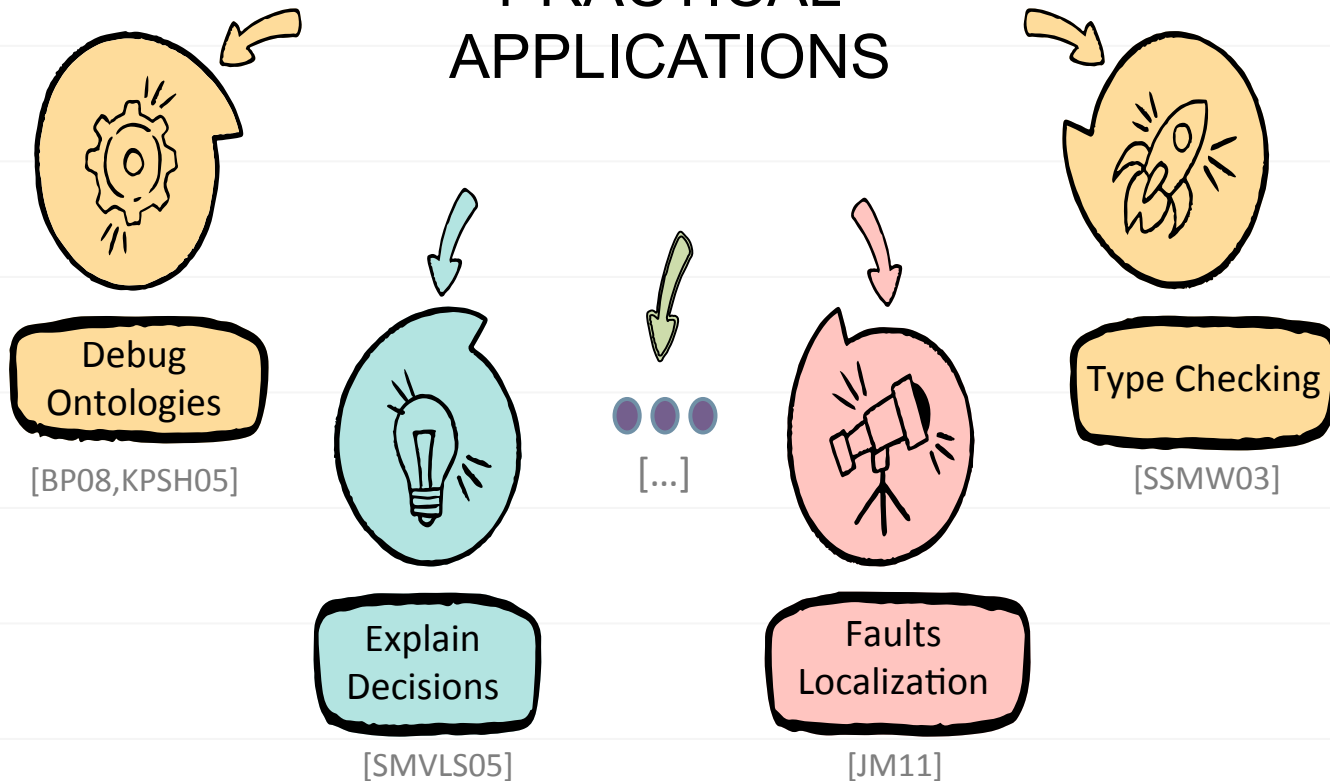
Axiom pinpointing is a well-studied problem across several domains

Different research communities know **justification** by different names



Axiom pinpointing is well-studied problem across several domains and has many interesting applications

PRACTICAL APPLICATIONS

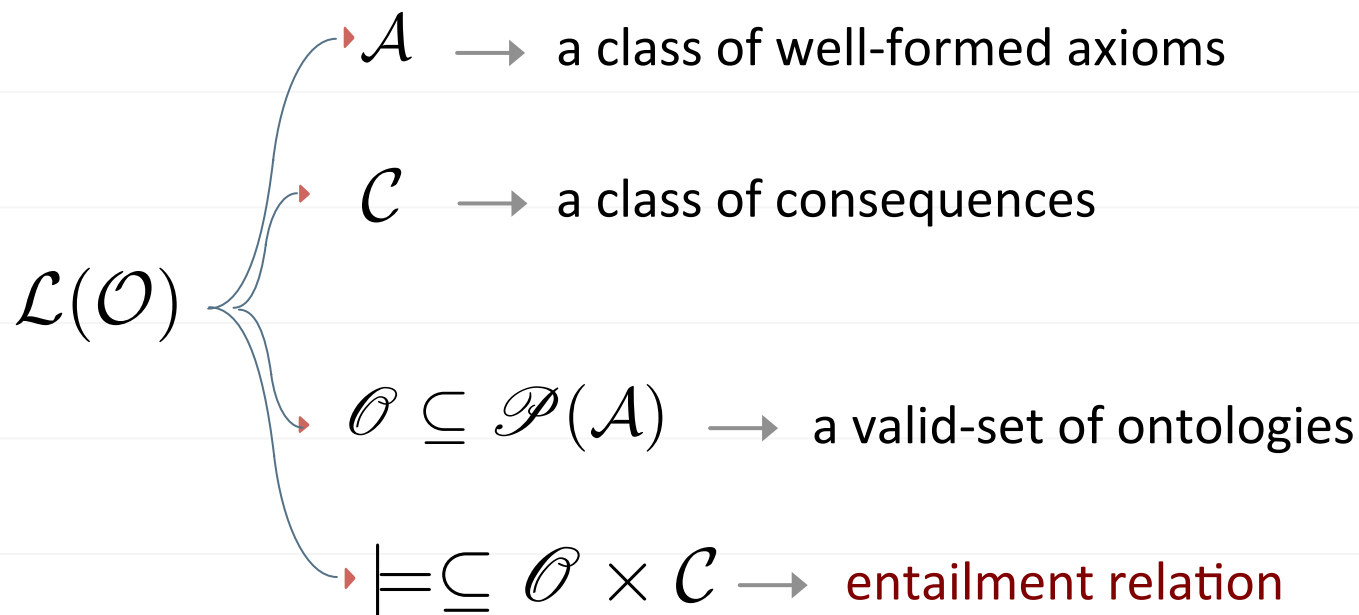




Problem Formulation

A Simple Ontology Language

A simple Ontology Language $\mathcal{L}(\mathcal{O})$ that contains four components



We only consider a monotonic relation:

$\mathcal{O}, \mathcal{O}' \in \mathcal{O}$ and $c \in \mathcal{C}$ if $\mathcal{O} \models c$ and $\mathcal{O} \subseteq \mathcal{O}'$, then $\mathcal{O}' \models c$

A Simple Ontology Language:

Given an ontology $\mathcal{O} \in \mathcal{O}$ and a consequence $c \in \mathcal{O}$

Entailment:

Decide $\mathcal{O} \models c$ holds?

Justification: $\mathcal{O} \models c$

A minimal subset Ontology $\mathcal{M} \subseteq \mathcal{O}$ that meets two conditions:

1. $\mathcal{M} \models c$
2. For every $\mathcal{M}' \subsetneq \mathcal{M}$, $\mathcal{M}' \not\models c$

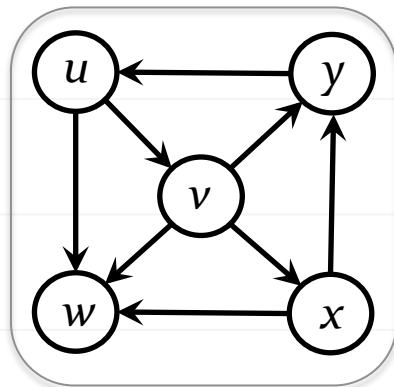
Graph Ontology Example:

$\mathcal{V} \rightarrow$ a countable set of vertices

$\mathcal{A} = \mathcal{C} = \{(v, w) | v, w \in \mathcal{V}\} \rightarrow$ edges

$\mathcal{O} \subseteq \mathcal{P}(\mathcal{A}) \rightarrow$ a finite set of graphs

$\models \rightarrow$ any two reachable vertices in a graph



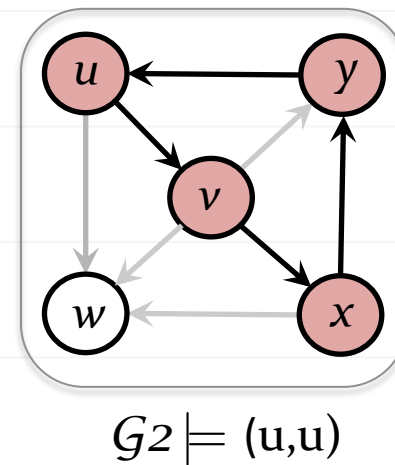
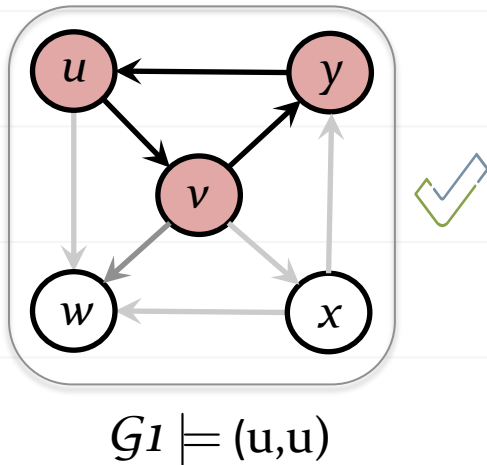
$\mathcal{G} \in \mathcal{O}$ is a finite graph, such that

$\mathcal{G} \models (u, x)$ but $\mathcal{G} \not\models (w, u)$

Axiom Pinpointing:

A justification is a **minimal sub-ontology** that still entails the consequence

Find one or more justification in ontology \mathcal{G} for a **consequence**:
 u is reachable from u .



Over-constrained Formulas in SAT:

Minimal Unsatisfiable Subset [MUS]:

$\mathcal{M} \subseteq \mathcal{F}$ is minimally unsatisfiable subformula of \mathcal{F} if and only if
 \mathcal{M} is unsatisfiable and $\forall \mathcal{M}' \subsetneq \mathcal{M} \mathcal{M}'$ is satisfiable

Minimal Correction Subset [MCS]:

$\mathcal{C} \subseteq \mathcal{F}$ is minimally correction subformula of \mathcal{F} if and only if
 $\mathcal{F} \setminus \mathcal{C}$ is satisfiable and $\forall \mathcal{C}' \subsetneq \mathcal{C} \mathcal{F} \setminus \mathcal{C}'$ is unsatisfiable



Proposed Solutions

Black-box Method:

Compute one justification by removing an axiom at a time

Data: Ontology \mathcal{O} , consequence c
Result: A justification \mathcal{M} for c w.r.t. \mathcal{O}

- 1 $\mathcal{M} \leftarrow \mathcal{O}$
- 2 **for** $\alpha \in \mathcal{O}$ **do**
- 3 **if** $\mathcal{M} \setminus \{\alpha\} \models c$ **then**
- 4 $\mathcal{M} \leftarrow \mathcal{M} \setminus \{\alpha\}$
- 5 **Return** \mathcal{M}

There is a relation between the order of axioms selected for removal and the computed justification

Enumeration over Justifications

Increase our understanding about the derivation by finding more than one or all justifications

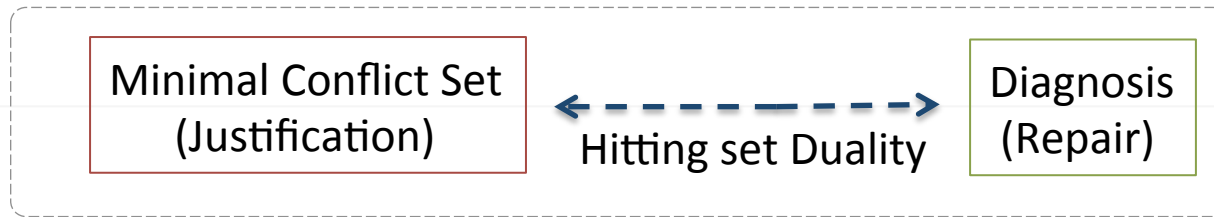
To find all Justifications, one can simply enumerate all sub-ontologies

Problem: Exponential many entailment checks, one for each subset of \mathcal{O}

The question is: can we do better?

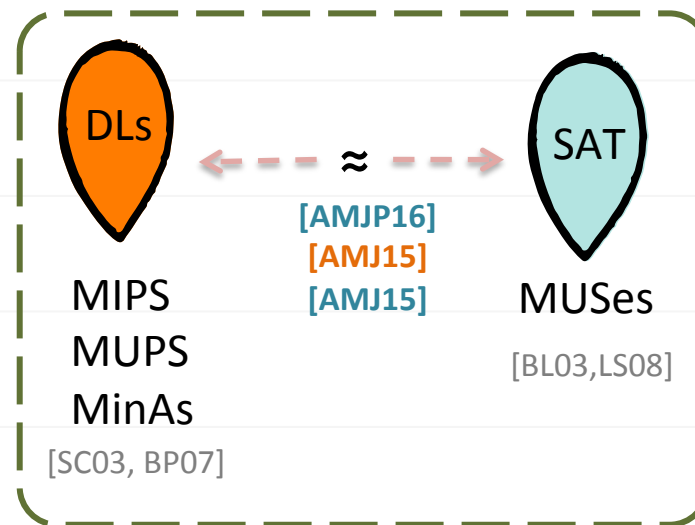
Contribution

[RR87]



[AMJ15]

MUSes are justifications and MCSes is a repairs



Optimizations:

[AMJ15]

If a set $\mathcal{O}' \subseteq \mathcal{O}$ is such that $\mathcal{O}' \models c$, then ignore all strict superset of \mathcal{O}' because minimality condition fails for superset

If $\mathcal{O}' \not\models c$ then ignore subset of \mathcal{O}' because subsets fail to entail c

Glass-box Method:

Modified Tableau-based reasoning method for standard reasoning can track the relevant traces but are inefficient

Tableau-based pinpointing approaches does not behave well in practice

Glass-box Method:

Modified Tableau-based reasoning method for standard reasoning can track the relevant traces but are inefficient

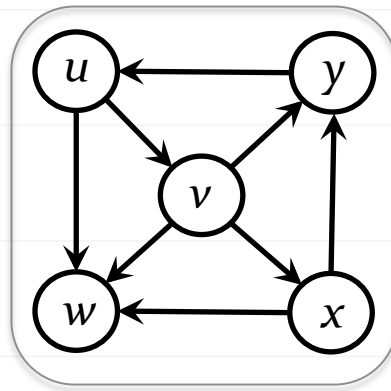
Tableau-based pinpointing approaches does not behave well in practice

A propositional formula encodes the execution of a consequence-based algorithm

Consequence-based Axiom Pinpointing:

[SV09]

A Horn clause simulates each rule possible application and a propositional variable represents a conclusion

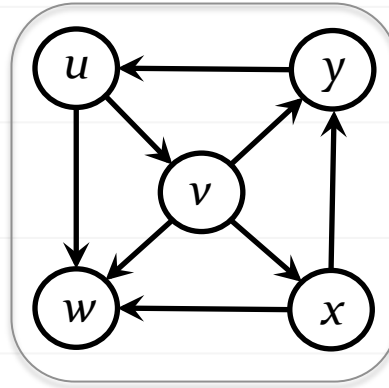


$$\mathcal{F}_{\mathcal{G}} = \mathcal{X}(u,v) \wedge \mathcal{X}(v,y) \wedge \mathcal{X}(x,w) \wedge \mathcal{X}(y,u) \\ \mathcal{X}(u,w) \wedge \mathcal{X}(v,w) \wedge \mathcal{X}(x,y) \wedge \mathcal{X}(v,x)$$

Consequence-based Axiom Pinpointing:

[SV09]

A Horn clause simulates each rule possible application and a propositional variable represents a conclusion



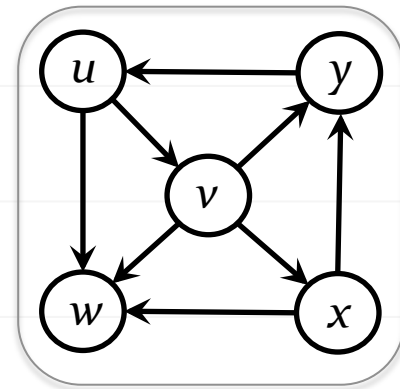
Constraints encode the edges of the example graph

Graph Ontology Example:

A consequence-based algorithm contains only one rule:

$$\{(X, Y), (Y, Z)\} \Rightarrow \{(X, Z)\}$$

$$\mathcal{F}_{\mathcal{H}} = \begin{array}{l} x_{(u,v)} \wedge x_{(v,y)} \rightarrow x_{(u,y)} \\ x_{(u,v)} \wedge x_{(v,w)} \rightarrow x_{(u,w)} \\ x_{(u,v)} \wedge x_{(v,x)} \rightarrow x_{(u,x)} \\ x_{(u,v)} \wedge x_{(v,y)} \rightarrow x_{(u,y)} \\ x_{(u,x)} \wedge x_{(x,w)} \rightarrow x_{(u,w)} \\ x_{(u,x)} \wedge x_{(x,y)} \rightarrow x_{(u,y)} \\ x_{(u,y)} \wedge x_{(y,u)} \rightarrow x_{(u,u)} \end{array}$$

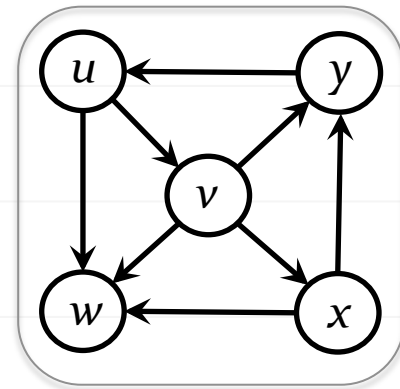


Graph Ontology Example:

A consequence-based algorithm contains only one rule:

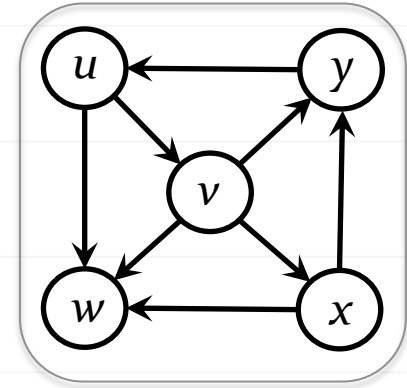
$$\{(X, Y), (Y, Z)\} \Rightarrow \{(X, Z)\}$$

Constraints encode all the reachable paths in the example graph



Graph Ontology Example:

$$\mathcal{F} = \mathcal{F}_{\mathcal{H}} \wedge \mathcal{F}_{\mathcal{G}}$$



$$\mathcal{F}_{\mathcal{H}} = x_{(u,v)} \wedge x_{(v,y)} \rightarrow x_{(u,y)} \wedge$$

$$x_{(u,v)} \wedge x_{(v,w)} \rightarrow x_{(u,w)} \wedge x_{(u,x)} \wedge x_{(x,w)} \rightarrow x_{(u,w)}$$

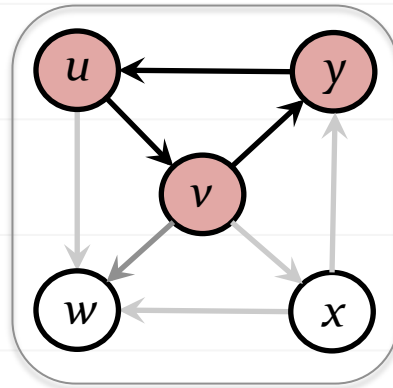
$$x_{(u,v)} \wedge x_{(v,x)} \rightarrow x_{(u,x)} \wedge x_{(u,x)} \wedge x_{(x,y)} \rightarrow x_{(u,y)}$$

$$x_{(u,v)} \wedge x_{(v,y)} \rightarrow x_{(u,y)} \wedge x_{(u,y)} \wedge x_{(y,u)} \rightarrow x_{(u,u)}$$

$$\mathcal{F}_{\mathcal{G}} = \begin{array}{l} x_{(u,v)} \wedge x_{(v,y)} \wedge x_{(x,w)} \wedge x_{(y,u)} \\ x_{(u,w)} \wedge x_{(v,w)} \wedge x_{(x,y)} \wedge x_{(v,x)} \end{array}$$

Graph Ontology Example:

$$\mathcal{F} = \mathcal{F}_{\mathcal{H}} \wedge \mathcal{F}_{\mathcal{G}}$$

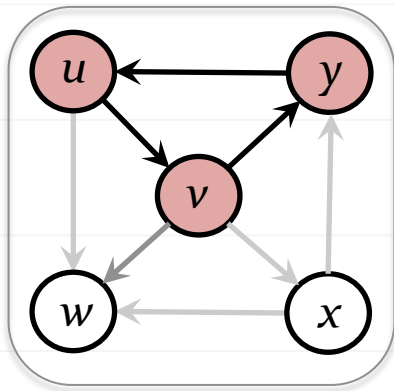


$$\mathcal{G} \models (u, u) \quad \leftarrow \text{dashed} \approx \text{dashed} \rightarrow \quad \mathcal{F} \models x_{(u,u)}$$

Graph Ontology Example:

[AMJ15]

$$\mathcal{G} \models (u, u)$$



$$\mathcal{F} \models x(u, u)$$

$$\mathcal{F} = \mathcal{F}_{\mathcal{H}} \wedge \mathcal{F}_{\mathcal{G}}$$

$$x(u, v) \wedge x(v, y) \rightarrow x(u, y)$$

$$x(u, y) \wedge x(y, u) \rightarrow x(u, u)$$

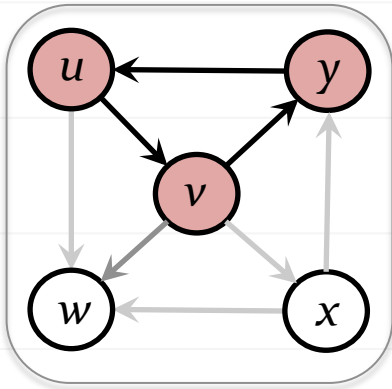
$$x(u, v) \quad x(v, y) \quad x(y, u)$$

$$\mathcal{F} \cup \{\neg x(u, u)\}$$

Graph Ontology Example:

[AMJ15]

$$\mathcal{G} \models (u, u)$$



$$\mathcal{F} \models x(u, u)$$

$$\mathcal{F} = \mathcal{F}_{\mathcal{H}} \wedge \mathcal{F}_{\mathcal{G}}$$

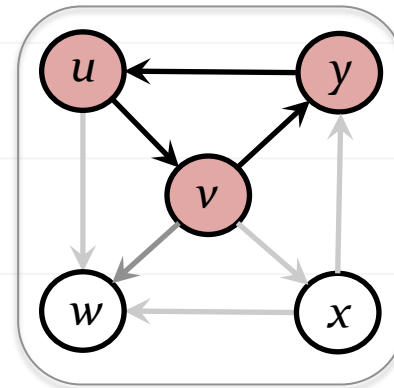
$$x(u, v) \wedge x(v, y) \rightarrow x(u, y)$$

$$x(u, y) \wedge x(y, u) \rightarrow x(u, u)$$

$$x(u, v) \quad x(v, y) \quad x(y, u)$$

$$\mathcal{F} = \mathcal{F}_{\mathcal{H}} \wedge \mathcal{F}_{\mathcal{G}} \wedge \neg x(u, u)$$

Graph Ontology Example:



$$\mathcal{G}_0 = \mathcal{F}_{\mathcal{H}}$$

$$\begin{aligned} \mathcal{G}_1 &= \{x_{(u,v)}\} & \mathcal{G}_2 &\models \{x_{(x,w)}\} & \mathcal{G}_5 &\models \{x_{(u,w)}\} \\ \mathcal{G}_2 &\models \{x_{(v,y)}\} & \mathcal{G}_4 &\models \{x_{(y,u)}\} & \mathcal{G}_6 &\models \{x_{(v,w)}\} \\ \mathcal{G}_7 &\models \{x_{(x,y)}\} & \mathcal{G}_8 &\models \{x_{(v,x)}\} \end{aligned}$$

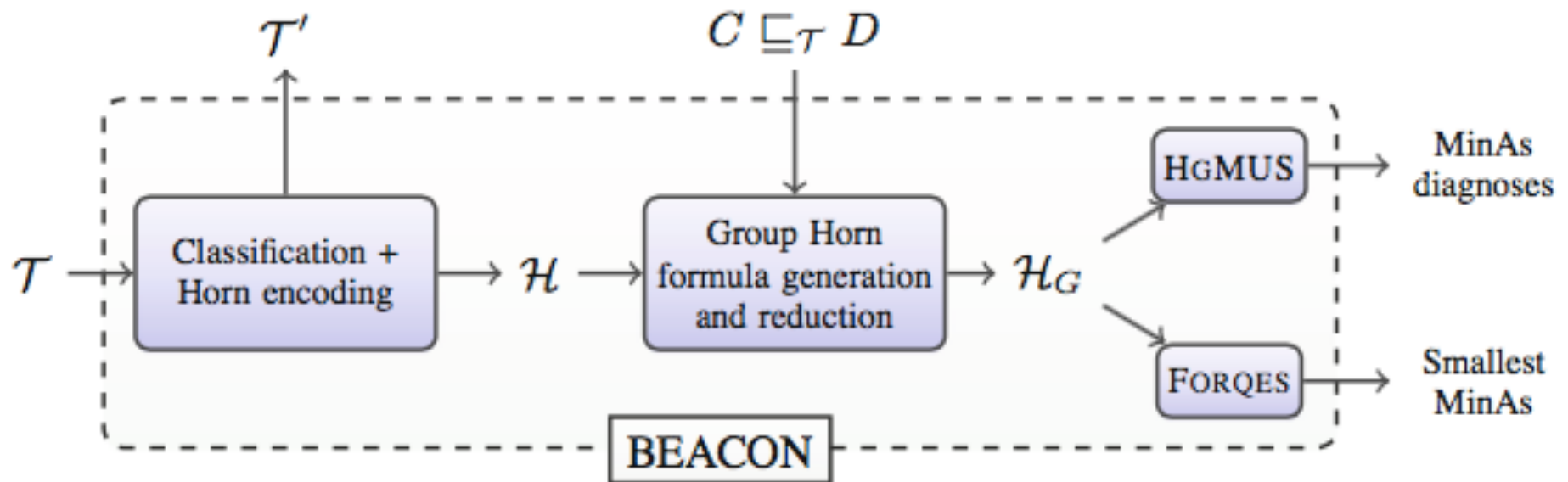
$$\mathcal{F} = \mathcal{G}_0 \cup \mathcal{G}_1 \cup \dots \cup \mathcal{G}_8 \setminus \{x_{(u,u)}\}$$

$$\mathcal{M} = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_4\}$$

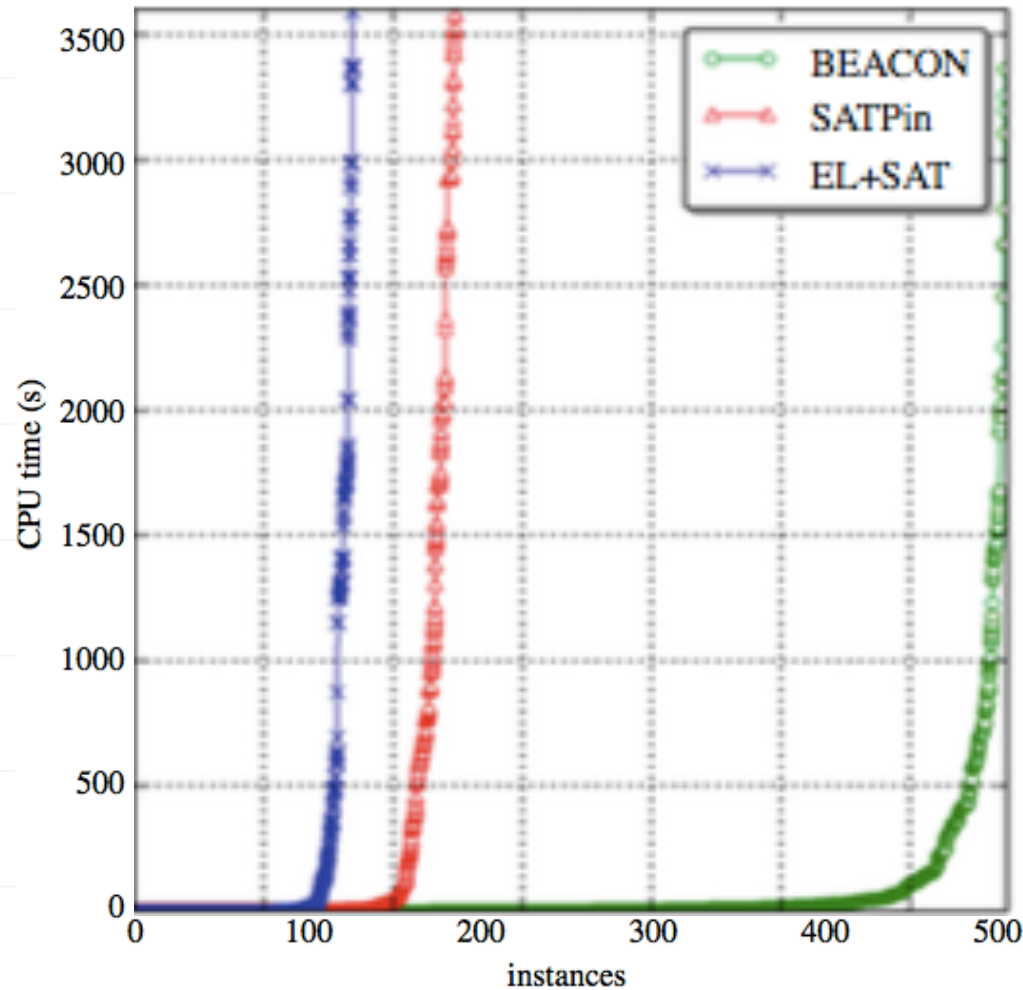
Hitting set Duality

$$\begin{aligned} \mathcal{S}_1 &= \{\mathcal{G}_1\} \\ \mathcal{S}_2 &= \{\mathcal{G}_2\} \\ \mathcal{S}_3 &= \{\mathcal{G}_4\} \end{aligned}$$

[AMJ15]



Performance:



Medical Ontologies

- ☐ SNOMED-CT
- ☐ GENE
- ☐ GALEN
- ☐ NCI



Ongoing Work

List of applications

1. Debugging Medical Ontologies [FS08]
2. Requirement analysis [JB17]
3. Equivalence checking [OGNLV10]
4. Counter-example Guided Abstraction Refinement[CGJLV00]
5. Proof-based abstraction refinement [MA03]
6. Boolean function bi-decomposition [CM11]
7. Circuit error Diagnosis [HL99]
8. Type Debugging in Haskell [SSW03]
9. Proof explanation in Symbolic model checking[GWG17]

Domain of Constraints:

1. Axioms of an Ontology
2. Boolean Formulas
3. Temporal logic formula
4. Transition state Predicates

Domain agnostic MUS enumeration Tool



University of Iowa

Computational Logic Centre

M. Fareed Arif

April 27 2020