TECHNISCHE UNIVERSITÄT DRESDEN

CENTER FOR PERSPICUOUS COMPUTING

Patrick Koopmann

# Using Abduction to Explain Missing Entailments in OWL Ontologies

ESAO 2023, 14 February, 2023

# Motivation

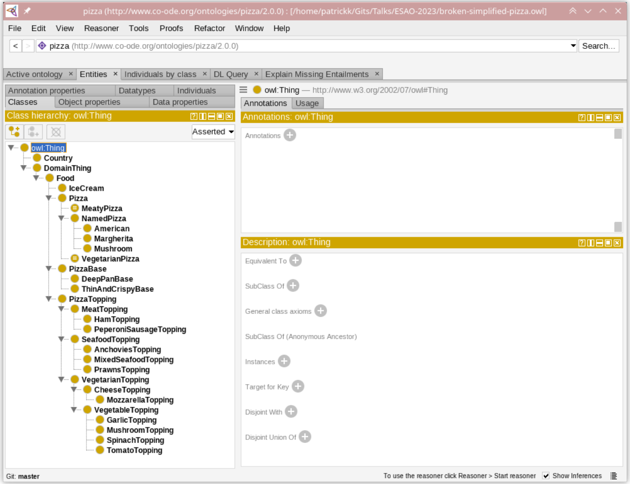
- We consider OWL ontologies based on Description Logics (DLs)
- Developing, maintaining and understanding ontologies can be challenging
  - large number of classes and axioms
  - complex interactions between axioms
- Reasoning with DLs is explainable “*by design*”
  - inferences through symbolic reasoning

# Motivation

- We consider OWL ontologies based on Description Logics (DLs)
- Developing, maintaining and understanding ontologies can be challenging
  - large number of classes and axioms
  - complex interactions between axioms
- Reasoning with DLs is explainable “*by design*”
  - inferences through symbolic reasoning

- **Vision:** Ontology tools support users like a tutor would:
  - explain inferences performed by the reasoner
  - offer different types of explanations
  - guide them to a solution to their problems
  - help them understand the reasoning

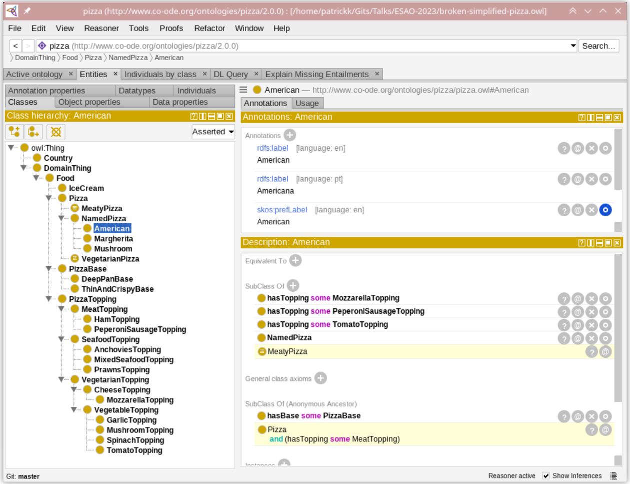
# Working with OWL Ontologies



Protégé is a popular tool for developing OWL ontologies

- Example shows a simplified version of the Pizza ontology

# Working with OWL Ontologies


pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]
File Edit View Reasoner Tools Proofs Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
DomainThing 〉 Food 〉 Pizza 〉 NamedPizza 〉 American
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×
Annotation properties Datatypes Individuals
Classes Object properties Data properties
Class hierarchy: American
Asserted
owl:Thing
Country
DomainThing
Food
IceCream
Pizza
MeatyPizza
NamedPizza
American
Margherita
Mushroom
VegetarianPizza
PizzaBase
DeepPanBase
ThinAndCrispyBase
PizzaTopping
MeatTopping
HamTopping
PeperoniSausageTopping
SeafoodTopping
AnchoviesTopping
MixedSeafoodTopping
PrawnsTopping
VegetarianTopping
CheeseTopping
MozzarellaTopping
VegetableTopping
GarlicTopping
MushroomTopping
SpinachTopping
TomatoTopping
American — http://www.co-ode.org/ontologies/pizza/pizza.owl#American
Annotations Usage
Annotations: American
Description: American
Git: master
Reasoner active
☑ Show Inferences

Annotations

rdfs:label [language: en]
American

rdfs:label [language: pt]
Americana

skos:prefLabel [language: en]
American

Equivalent To

SubClass Of

hasTopping some MozzarellaTopping

hasTopping some PeperoniSausageTopping

hasTopping some TomatoTopping

NamedPizza

MeatyPizza

General class axioms

SubClass Of (Anonymous Ancestor)

hasBase some PizzaBase

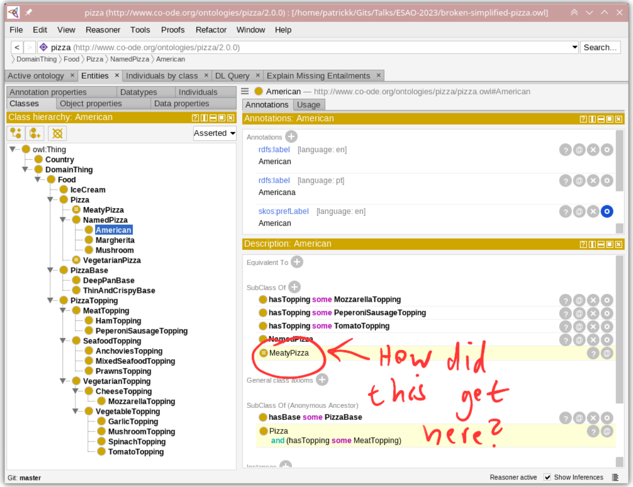
Pizza
and (hasTopping some MeatTopping)

Instances

Protégé is a popular tool for developing OWL ontologies

- Example shows a simplified version of the Pizza ontology
- Clicking on a class shows **stated** and inferred information

# Working with OWL Ontologies



pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]
File
Edit
View
Reasoner
Tools
Proofs
Refactor
Window
Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
DomainThing 〉 Food 〉 Pizza 〉 NamedPizza 〉 American
Active ontology ×
Entities ×
Individuals by class ×
DL Query ×
Explain Missing Entailments ×
Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties
Class hierarchy: American
Asserted
American — http://www.co-ode.org/ontologies/pizza/pizza.owl#American
Annotations
Usage
Annotations: American
Description: American
Git: master
Reasoner active
☑ Show Inferences

```
owl:Thing
  Country
  DomainThing
    Food
      IceCream
      Pizza
        MeatyPizza
        NamedPizza
          American
          Margherita
          Mushroom
        VegetarianPizza
      PizzaBase
        DeepPanBase
        ThinAndCrispyBase
      PizzaTopping
        MeatTopping
          HamTopping
          PeperoniSausageTopping
        SeafoodTopping
          AnchoviesTopping
          MixedSeafoodTopping
          PrawnsTopping
        VegetarianTopping
          CheeseTopping
            MozzarellaTopping
          VegetableTopping
            GarlicTopping
            MushroomTopping
            SpinachTopping
            TomatoTopping
```

Annotations

rdfs:label [language: en]
American

rdfs:label [language: pt]
Americana

skos:prefLabel [language: en]
American

Equivalent To

SubClass Of

- hasTopping some MozzarellaTopping
- hasTopping some PeperoniSausageTopping
- hasTopping some TomatoTopping
- NamedPizza
- MeatyPizza

General class axioms

SubClass Of (Anonymous Ancestor)

- hasBase some PizzaBase
- Pizza and (hasTopping some MeatTopping)

Instances

How did this get here?

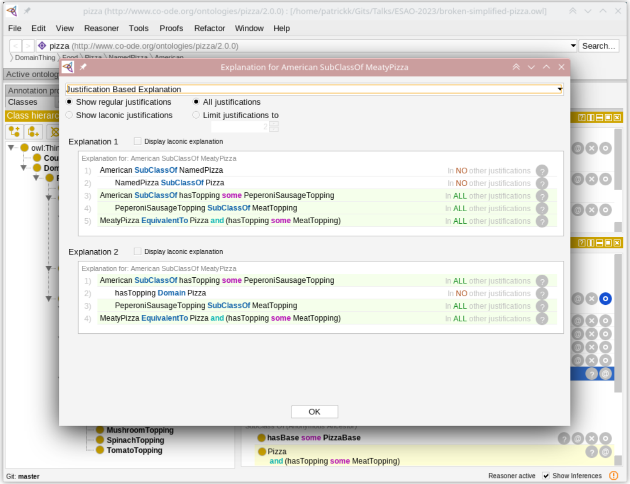
Protégé is a popular tool for developing OWL ontologies

- Example shows a simplified version of the Pizza ontology
- Clicking on a class shows **stated** and inferred information

A user might want the inferred information explained:

- to understand the mechanism of the ontology
- in case the inferred information looks incorrect

# Justifications


pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]
File Edit View Reasoner Tools Proofs Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
DomainThing 〉 Food 〉 Pizza 〉 NamedPizza 〉 American
Active ontolog
Annotation pr
Classes
Class hierarc
owl:Thin
Cou
Dom
MushroomTopping
SpinachTopping
TomatoTopping
hasBase some PizzaBase
Pizza
and (hasTopping some MeatTopping)
Git: master
Reasoner active
Show Inferences

Explanation for American SubClassOf MeatyPizza

Justification Based Explanation

● Show regular justifications ● All justifications

○ Show laconic justifications ○ Limit justifications to 2

Explanation 1 □ Display laconic explanation

Explanation for: American SubClassOf MeatyPizza

1) American SubClassOf NamedPizza — In NO other justifications

2) NamedPizza SubClassOf Pizza — In NO other justifications

3) American SubClassOf hasTopping some PeperoniSausageTopping — In ALL other justifications

4) PeperoniSausageTopping SubClassOf MeatTopping — In ALL other justifications

5) MeatyPizza EquivalentTo Pizza and (hasTopping some MeatTopping) — In ALL other justifications

Explanation 2 □ Display laconic explanation

Explanation for: American SubClassOf MeatyPizza

1) American SubClassOf hasTopping some PeperoniSausageTopping — In ALL other justifications

2) hasTopping Domain Pizza — In NO other justifications

3) PeperoniSausageTopping SubClassOf MeatTopping — In ALL other justifications

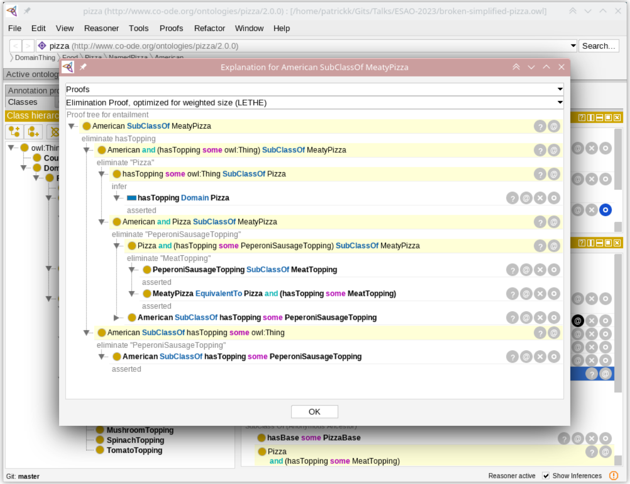
4) MeatyPizza EquivalentTo Pizza and (hasTopping some MeatTopping) — In ALL other justifications

OK

Justifications:

- minimal set of axioms sufficient for entailment
- standard functionality of Protégé

# Proofs with evee-protege


pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]
File Edit View Reasoner Tools Proofs Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
DomainThing 〉 Food 〉 Pizza 〉 NamedPizza 〉 American
Active ontolog
Annotation pr
Classes
Class hierarc
owl:Thin
Cou
Dom
MushroomTopping
SpinachTopping
TomatoTopping
SubClass Of (Anonymous Ancestor)
hasBase some PizzaBase
Pizza
and (hasTopping some MeatTopping)
Git: master
Reasoner active
☑ Show Inferences

Explanation for American SubClassOf MeatyPizza

Proofs

Elimination Proof, optimized for weighted size (LETHE)

Proof tree for entailment

```
American SubClassOf MeatyPizza
eliminate hasTopping
  American and (hasTopping some owl:Thing) SubClassOf MeatyPizza
  eliminate "Pizza"
    hasTopping some owl:Thing SubClassOf Pizza
    infer
      hasTopping Domain Pizza
      asserted
    American and Pizza SubClassOf MeatyPizza
    eliminate "PeperoniSausageTopping"
      Pizza and (hasTopping some PeperoniSausageTopping) SubClassOf MeatyPizza
      eliminate "MeatTopping"
        PeperoniSausageTopping SubClassOf MeatTopping
        asserted
        MeatyPizza EquivalentTo Pizza and (hasTopping some MeatTopping)
        asserted
      American SubClassOf hasTopping some PeperoniSausageTopping
  American SubClassOf hasTopping some owl:Thing
  eliminate "PeperoniSausageTopping"
    American SubClassOf hasTopping some PeperoniSausageTopping
    asserted
```

OK

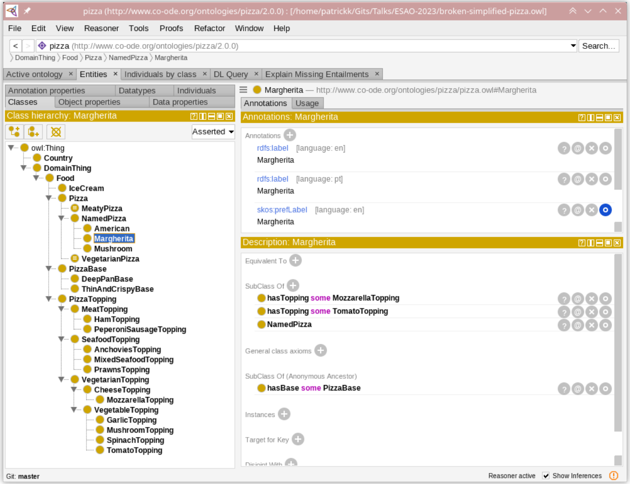
Proofs:

- Explain entailment in detail
- Plugin upports $\mathcal{EL}$, $\mathcal{ALCH}$, $\mathcal{ALCOI}$

[Alrabaa, Borgwardt, Friese, Koopmann, Mendez, Popovic; DL 2022]

`https://github.com/de-tu-dresden-inf-lat/evee`

# Missing Entailments



pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]

Annotations

rdfs:label [language: en]
Margherita

rdfs:label [language: pt]
Margherita

skos:prefLabel [language: en]
Margherita

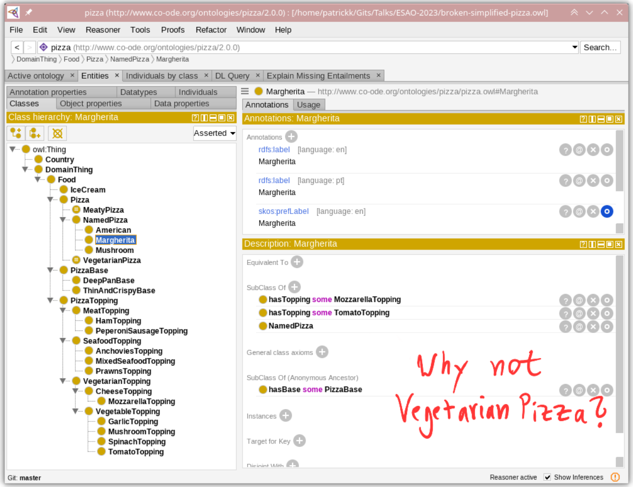
SubClass Of

- hasTopping some MozzarellaTopping
- hasTopping some TomatoTopping
- NamedPizza

SubClass Of (Anonymous Ancestor)

- hasBase some PizzaBase

# Missing Entailments


File Edit View Reasoner Tools Proofs Refactor Window Help
Search...
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×
Annotation properties Datatypes Individuals
Classes Object properties Data properties
Annotations Usage
Asserted
Reasoner active
☑ Show Inferences

pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]

pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)

〉DomainThing 〉 Food 〉 Pizza 〉 NamedPizza 〉 Margherita

Class hierarchy: Margherita

- owl:Thing
  - Country
  - DomainThing
    - Food
      - IceCream
      - Pizza
        - MeatyPizza
        - NamedPizza
          - American
          - Margherita
          - Mushroom
        - VegetarianPizza
      - PizzaBase
        - DeepPanBase
        - ThinAndCrispyBase
      - PizzaTopping
        - MeatTopping
          - HamTopping
          - PeperoniSausageTopping
        - SeafoodTopping
          - AnchoviesTopping
          - MixedSeafoodTopping
          - PrawnsTopping
        - VegetarianTopping
          - CheeseTopping
            - MozzarellaTopping
          - VegetableTopping
            - GarlicTopping
            - MushroomTopping
            - SpinachTopping
            - TomatoTopping

Margherita — http://www.co-ode.org/ontologies/pizza/pizza.owl#Margherita

Annotations: Margherita

Annotations

rdfs:label [language: en]
Margherita

rdfs:label [language: pt]
Margherita

skos:prefLabel [language: en]
Margherita

Description: Margherita

Equivalent To

SubClass Of
hasTopping some MozzarellaTopping
hasTopping some TomatoTopping
NamedPizza

General class axioms

SubClass Of (Anonymous Ancestor)
hasBase some PizzaBase

Instances

Target for Key

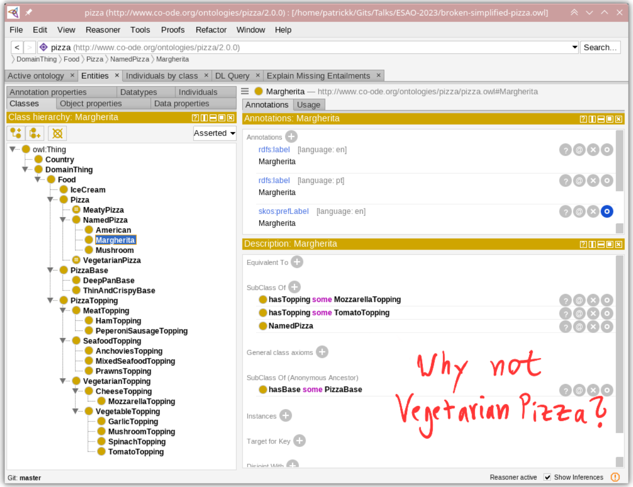
Disjoint With

Why not Vegetarian Pizza?

Git: master

- Why is Margherita not classified as VegetarianPizza?

# Missing Entailments


pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]
File
Edit
View
Reasoner
Tools
Proofs
Refactor
Window
Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
DomainThing 〉 Food 〉 Pizza 〉 NamedPizza 〉 Margherita
Active ontology ×
Entities ×
Individuals by class ×
DL Query ×
Explain Missing Entailments ×
Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties
Class hierarchy: Margherita
Asserted
Margherita — http://www.co-ode.org/ontologies/pizza/pizza.owl#Margherita
Annotations
Usage
Annotations: Margherita
Description: Margherita
Git: master
Reasoner active
☑ Show Inferences

- owl:Thing
  - Country
  - DomainThing
    - Food
      - IceCream
      - Pizza
        - MeatyPizza
        - NamedPizza
          - American
          - Margherita
          - Mushroom
        - VegetarianPizza
      - PizzaBase
        - DeepPanBase
        - ThinAndCrispyBase
      - PizzaTopping
        - MeatTopping
          - HamTopping
          - PeperoniSausageTopping
        - SeafoodTopping
          - AnchoviesTopping
          - MixedSeafoodTopping
          - PrawnsTopping
        - VegetarianTopping
          - CheeseTopping
            - MozzarellaTopping
          - VegetableTopping
            - GarlicTopping
            - MushroomTopping
            - SpinachTopping
            - TomatoTopping

Annotations

rdfs:label [language: en]
Margherita

rdfs:label [language: pt]
Margherita

skos:prefLabel [language: en]
Margherita

Equivalent To

SubClass Of
- hasTopping some MozzarellaTopping
- hasTopping some TomatoTopping
- NamedPizza

General class axioms

SubClass Of (Anonymous Ancestor)
- hasBase some PizzaBase

Instances

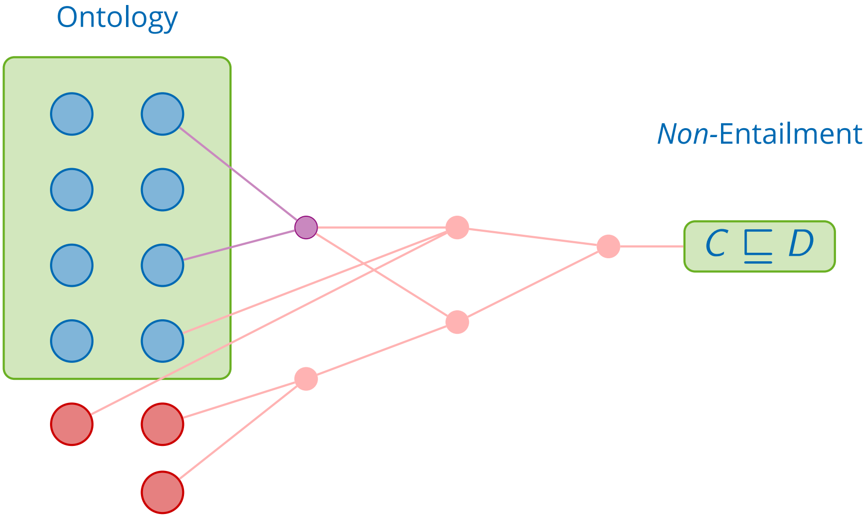
Target for Key

Disjoint With

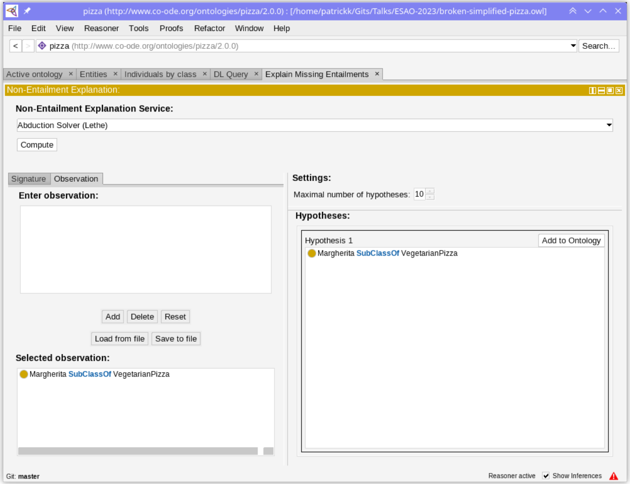
Why not Vegetarian Pizza?

- Why is Margherita not classified as VegetarianPizza?
- How can I ensure it gets classified as VegetarianPizza?

# Explaining Missing Entailments



# Abduction with `evee-protege`



pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]

Non-Entailment Explanation Service: Abduction Solver (Lethe)

Maximal number of hypotheses: 10

Selected observation:

Margherita **SubClassOf** VegetarianPizza

Hypotheses:

Hypothesis 1

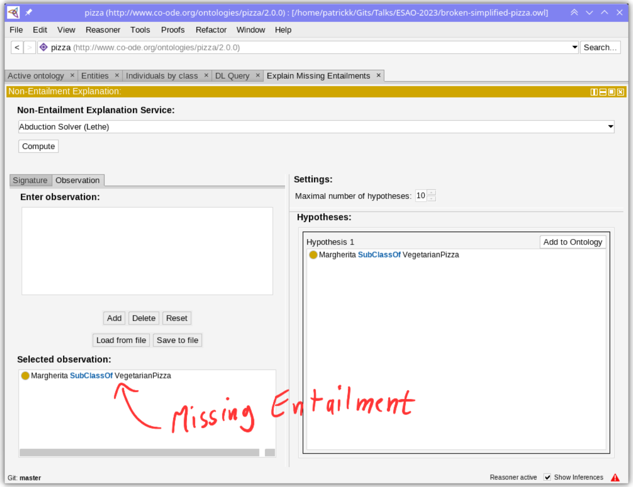
Margherita **SubClassOf** VegetarianPizza

Coming version of `evee-protege`

How to make **Margherita** become classified as **VegetarianPizza**?

*Special thanks to Tom Friese!*

# Abduction with `evee-protege`



pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]

File Edit View Reasoner Tools Proofs Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×

Non-Entailment Explanation:

**Non-Entailment Explanation Service:** Abduction Solver (Lethe)

Compute
Signature Observation

**Enter observation:**

Add Delete Reset
Load from file Save to file

**Selected observation:**

● Margherita **SubClassOf** VegetarianPizza

Missing Entailment

**Settings:**

Maximal number of hypotheses: 10

**Hypotheses:**

Hypothesis 1

● Margherita **SubClassOf** VegetarianPizza

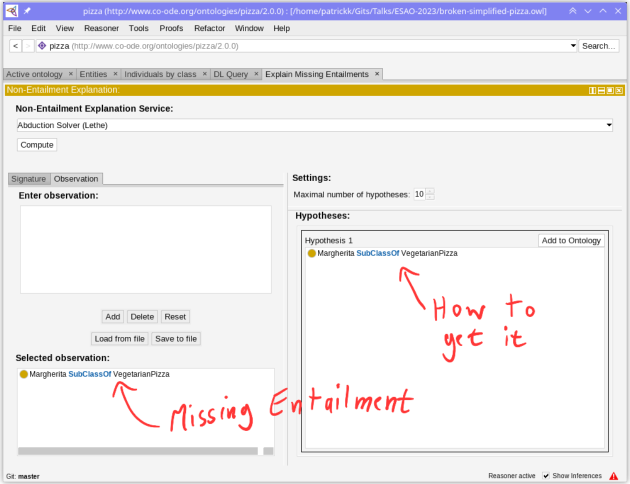
Git: **master**

Coming version of `evee-protege`

How to make **Margherita** become classified as **VegetarianPizza**?

*Special thanks to Tom Friese!*

# Abduction with evee-protege



pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]

File Edit View Reasoner Tools Proofs Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×
Non-Entailment Explanation:

Non-Entailment Explanation Service: Abduction Solver (Lethe)

Compute
Signature Observation

Enter observation:

Add Delete Reset
Load from file Save to file

Selected observation:

Margherita SubClassOf VegetarianPizza

Settings:

Maximal number of hypotheses: 10

Hypotheses:

Hypothesis 1 — Add to Ontology

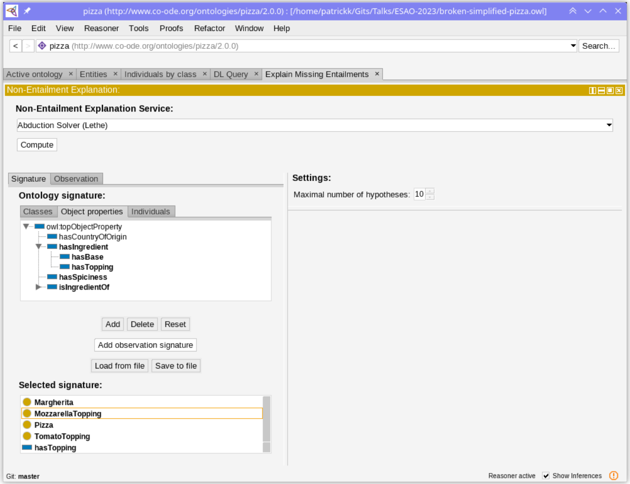
Margherita SubClassOf VegetarianPizza

Coming version of evee-protege

How to make Margherita become classified as VegetarianPizza?

*Special thanks to Tom Friese!*

# Abduction with evee-protege



pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]

pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)

Non-Entailment Explanation:

Non-Entailment Explanation Service: Abduction Solver (Lethe)

Settings:

Maximal number of hypotheses: 10

Ontology signature:

- owl:topObjectProperty
  - hasCountryOfOrigin
  - hasIngredient
    - hasBase
    - hasTopping
  - hasSpiciness
  - isIngredientOf

Selected signature:

- Margherita
- MozzarellaTopping
- Pizza
- TomatoTopping
- hasTopping

Git: master

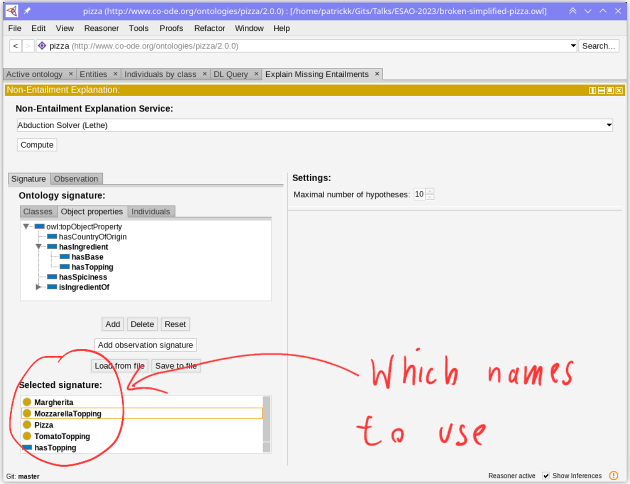
Coming version of evee-protege

How to make **Margherita** become classified as **VegetarianPizza**?

- using only relevant vocabulary

*Special thanks to Tom Friese!*

# Abduction with evee-protege


pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]
File Edit View Reasoner Tools Proofs Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×
Non-Entailment Explanation:
Compute
Signature Observation
Classes Object properties Individuals
Add Delete Reset
Add observation signature
Load from file Save to file
Git: master
Reasoner active ☑ Show Inferences
Which names to use

**Non-Entailment Explanation Service:** Abduction Solver (Lethe)

**Settings:**

Maximal number of hypotheses: 10

**Ontology signature:**

- owl:topObjectProperty
  - hasCountryOfOrigin
  - hasIngredient
    - hasBase
    - hasTopping
  - hasSpiciness
  - isIngredientOf

**Selected signature:**

- Margherita
- MozzarellaTopping
- Pizza
- TomatoTopping
- hasTopping

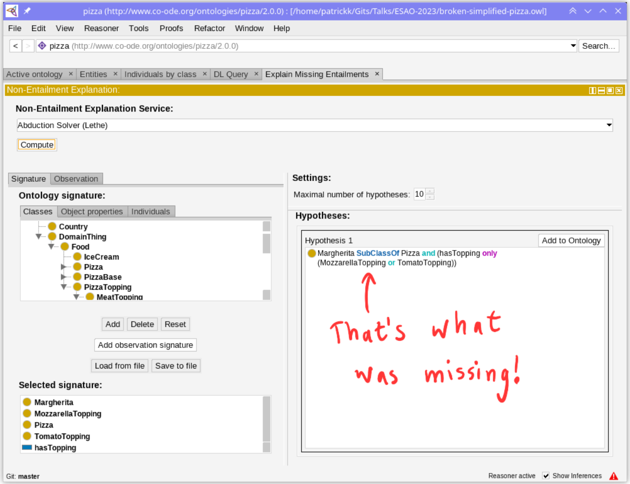
Coming version of evee-protege

How to make **Margherita** become classified as **VegetarianPizza**?

- using only relevant vocabulary

*Special thanks to Tom Friese!*

# Abduction with evee-protege



pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]

pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)

Non-Entailment Explanation:

**Non-Entailment Explanation Service:** Abduction Solver (Lethe)

**Ontology signature:**

- Country
- DomainThing
  - Food
    - IceCream
    - Pizza
    - PizzaBase
    - PizzaTopping
      - MeatTopping

**Selected signature:**

- Margherita
- MozzarellaTopping
- Pizza
- TomatoTopping
- hasTopping

**Settings:**

Maximal number of hypotheses: 10

**Hypotheses:**

Hypothesis 1

Margherita SubClassOf Pizza and (hasTopping only (MozzarellaTopping or TomatoTopping))

↑ That's what was missing!

Git: master

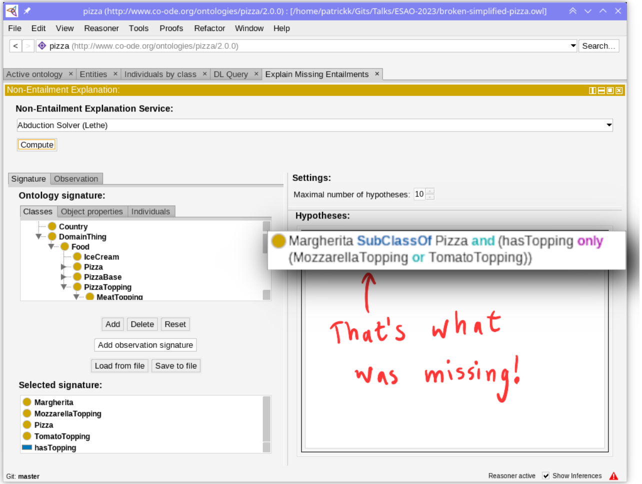
Coming version of evee-protege

How to make Margherita become classified as VegetarianPizza?

- using only relevant vocabulary
- without changing more than necessary

*Special thanks to Tom Friese!*

# Abduction with evee-protege


pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]
File Edit View Reasoner Tools Proofs Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×
Non-Entailment Explanation:
Compute
Signature Observation
Classes Object properties Individuals
Country
DomainThing
Food
IceCream
Pizza
PizzaBase
PizzaTopping
MeatTopping
Add Delete Reset
Add observation signature
Load from file Save to file
Git: master
Reasoner active
☑ Show Inferences

Non-Entailment Explanation Service:

Abduction Solver (Lethe)

Ontology signature:

Selected signature:

- Margherita
- MozzarellaTopping
- Pizza
- TomatoTopping
- hasTopping

Settings:

Maximal number of hypotheses: 10

Hypotheses:

Margherita SubClassOf Pizza and (hasTopping only (MozzarellaTopping or TomatoTopping))

That's what was missing!

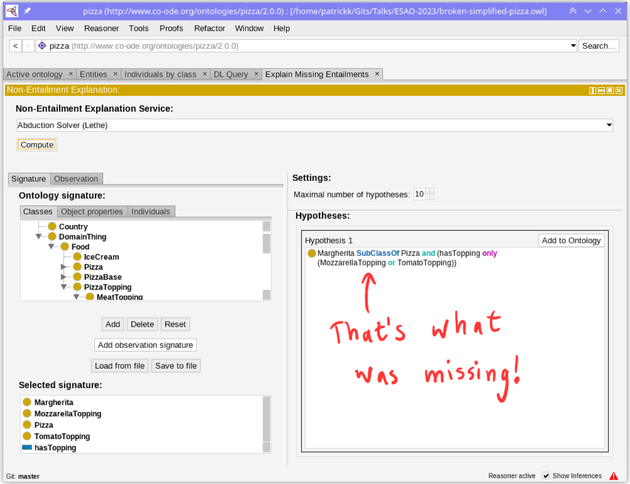
Coming version of evee-protege

How to make **Margherita** become classified as **VegetarianPizza**?

- using only relevant vocabulary
- without changing more than necessary

*Special thanks to Tom Friese!*

# Abduction with `evee-protege`



File Edit View Reasoner Tools Proofs Refactor Window Help
Search...
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×
Compute
Signature Observation
Classes Object properties Individuals
Add Delete Reset
Add observation signature
Load from file Save to file
Add to Ontology
Reasoner active ☑ Show Inferences

pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]

pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)

Non-Entailment Explanation:

**Non-Entailment Explanation Service:** Abduction Solver (Lethe)

**Ontology signature:**

- Country
- DomainThing
  - Food
    - IceCream
    - Pizza
    - PizzaBase
    - PizzaTopping
      - MeatTopping

**Selected signature:**

- Margherita
- MozzarellaTopping
- Pizza
- TomatoTopping
- hasTopping

**Settings:**

Maximal number of hypotheses: 10

**Hypotheses:**

Hypothesis 1

Margherita SubClassOf Pizza and (hasTopping only (MozzarellaTopping or TomatoTopping))

That's what was missing!

Git: master

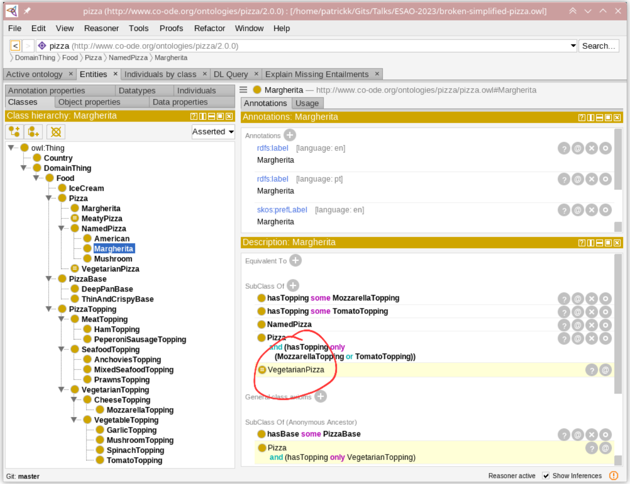
Coming version of `evee-protege`

How to make **Margherita** become classified as **VegetarianPizza**?

- using only relevant vocabulary
- without changing more than necessary

*Special thanks to Tom Friese!*

# Abduction with evee-protege


pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]
File Edit View Reasoner Tools Proofs Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
DomainThing 〉 Food 〉 Pizza 〉 NamedPizza 〉 Margherita
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×
Annotation properties Datatypes Individuals
Classes Object properties Data properties
Class hierarchy: Margherita
Asserted
Margherita — http://www.co-ode.org/ontologies/pizza/pizza.owl#Margherita
Annotations Usage
Annotations: Margherita
Description: Margherita
Git: master
Reasoner active
☑ Show Inferences

- owl:Thing
  - Country
  - DomainThing
    - Food
      - IceCream
      - Pizza
        - Margherita
        - MeatyPizza
        - NamedPizza
          - American
          - Margherita
          - Mushroom
        - VegetarianPizza
      - PizzaBase
        - DeepPanBase
        - ThinAndCrispyBase
      - PizzaTopping
        - MeatTopping
          - HamTopping
          - PeperoniSausageTopping
        - SeafoodTopping
          - AnchoviesTopping
          - MixedSeafoodTopping
          - PrawnsTopping
        - VegetarianTopping
          - CheeseTopping
            - MozzarellaTopping
          - VegetableTopping
            - GarlicTopping
            - MushroomTopping
            - SpinachTopping
            - TomatoTopping

Annotations

rdfs:label [language: en]
Margherita

rdfs:label [language: pt]
Margherita

skos:prefLabel [language: en]
Margherita

Equivalent To

SubClass Of

- hasTopping some MozzarellaTopping
- hasTopping some TomatoTopping
- NamedPizza
- Pizza and (hasTopping only (MozzarellaTopping or TomatoTopping))
- VegetarianPizza

General class axioms

SubClass Of (Anonymous Ancestor)

- hasBase some PizzaBase
- Pizza and (hasTopping only VegetarianTopping)

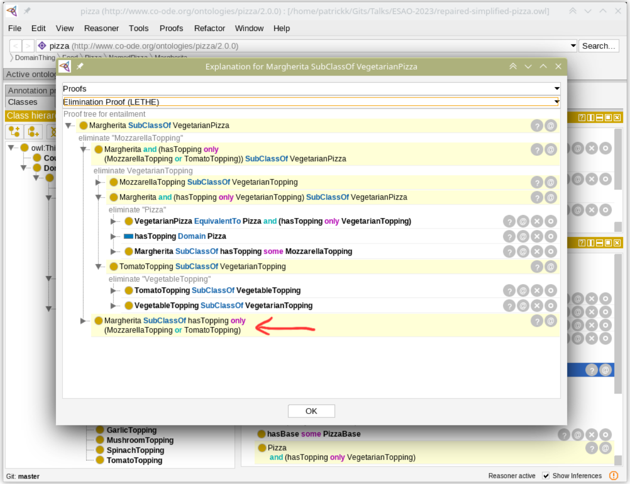
Coming version of evee-protege

How to make **Margherita** become classified as **VegetarianPizza**?

- using only relevant vocabulary
- without changing more than necessary

*Special thanks to Tom Friese!*

# Abduction with evee-protege



pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/repaired-simplified-pizza.owl]

pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)

Explanation for Margherita SubClassOf VegetarianPizza

Proofs

Elimination Proof (LETHE)

Proof tree for entailment

- Margherita SubClassOf VegetarianPizza
  - eliminate "MozzarellaTopping"
  - Margherita and (hasTopping only (MozzarellaTopping or TomatoTopping)) SubClassOf VegetarianPizza
    - eliminate VegetarianTopping
    - MozzarellaTopping SubClassOf VegetarianTopping
    - Margherita and (hasTopping only VegetarianTopping) SubClassOf VegetarianPizza
      - eliminate "Pizza"
      - VegetarianPizza EquivalentTo Pizza and (hasTopping only VegetarianTopping)
      - hasTopping Domain Pizza
      - Margherita SubClassOf hasTopping some MozzarellaTopping
    - TomatoTopping SubClassOf VegetarianTopping
      - eliminate "VegetableTopping"
      - TomatoTopping SubClassOf VegetableTopping
      - VegetableTopping SubClassOf VegetarianTopping
  - Margherita SubClassOf hasTopping only (MozzarellaTopping or TomatoTopping) ←

GarlicTopping
MushroomTopping
SpinachTopping
TomatoTopping

hasBase some PizzaBase

Pizza and (hasTopping only VegetarianTopping)

Git: master

Reasoner active ☑ Show Inferences

Coming version of evee-protege

How to make **Margherita** become classified as **VegetarianPizza**?

- using only relevant vocabulary
- without changing more than necessary

*Special thanks to Tom Friese!*

# Abduction

**Abduction** after *Charles Sanders Peirce*

*Given* knowledge $\mathcal{K}$ observation $\Phi$

*Generate* a “good” hypothesis $\mathcal{H}$ s.t. $\mathcal{K} \wedge \mathcal{H} \models \Phi$

Original idea:

- Generate a rational *(“best”)* explanation for an observed phenomenon $\Phi$
- Example: $\Phi =$“The street is wet.”, $\mathcal{H} =$“It was raining.”

# Abduction

**Abduction** after *Charles Sanders Peirce*

*Given* knowledge $\mathcal{K}$ observation $\Phi$

*Generate* a “good” hypothesis $\mathcal{H}$ s.t. $\mathcal{K} \wedge \mathcal{H} \models \Phi$

Original idea:

- Generate a rational *(“best”)* explanation for an observed phenomenon $\Phi$
- Example: $\Phi =$“The street is wet.”, $\mathcal{H} =$“It was raining.”

Contrast to deduction and induction:

- Deduction: Given $K$, generate $F$ s.t. $K \models F$ Induction: Generalize many observations

# Abduction

**Abduction** after *Charles Sanders Peirce*

*Given* knowledge $\mathcal{K}$ observation $\Phi$

*Generate* a “good” hypothesis $\mathcal{H}$ s.t. $\mathcal{K} \wedge \mathcal{H} \models \Phi$

Original idea:

- Generate a rational *(“best”)* explanation for an observed phenomenon $\Phi$
- Example: $\Phi =$“The street is wet.”, $\mathcal{H} =$“It was raining.”

Contrast to deduction and induction:

- Deduction: Given $K$, generate $F$ s.t. $K \models F$ Induction: Generalize many observations

Considered for many logical formalisms in computer science:

- propositional logic, first-order logic, logic programing, description logics, . . .

# Abduction for Rational Explanations in DLs

## Observation

$$\mathsf{street} : \mathsf{Unstable}$$

## Background Knowledge

$$\mathsf{street} : \mathsf{Street} \qquad \mathsf{canal} : \mathsf{Waterway}$$

$$\mathsf{EvaporiteFormation} \sqcap \exists \mathsf{borders}.(\mathsf{WaterWay} \sqcap \neg\exists \mathsf{lining}.\mathsf{WaterProof}) \sqsubseteq \exists \mathsf{affectedBy}.\mathsf{Dissolution}$$

$$\mathsf{EvaporiteFormation} \sqcap \exists \mathsf{affectedBy}.\mathsf{Dissolution} \sqsubseteq \forall \mathsf{above}.\mathsf{Unstable}$$

$$(\mathsf{WaterWay} \sqcup \mathsf{Street}) \sqcap \mathsf{EvaporiteFormation} \sqsubseteq \bot$$

## Hypothesis:

$$\mathsf{e} : \mathsf{EvaporiteFormation} \quad (\mathsf{e}, \mathsf{canal}) : \mathsf{borders} \quad (\mathsf{e}, \mathsf{street}) : \mathsf{above} \quad \mathsf{canal} : (\forall \mathsf{lining}.\bot)$$

# Abduction in Description Logics

Different types of abduction based on shape of observation and hypothesis

- Concept Abduction
  - Given $\mathcal{O}$, $C$, find $D$ s.t. $\mathcal{O} \models D \sqsubseteq C$
  - Generate subsumees
- TBox / ABox / Knowledge Base Abduction
  - generate terminological knowledge / assertional facts / both

# Abduction in Description Logics

Different types of abduction based on shape of observation and hypothesis

- Concept Abduction
  - Given $\mathcal{O}$, $C$, find $D$ s.t. $\mathcal{O} \models D \sqsubseteq C$
  - Generate subsumees
- TBox / ABox / Knowledge Base Abduction
  - generate terminological knowledge / assertional facts / both

How to select good hypotheses?

- Consistency: $\mathcal{K} \wedge \mathcal{H} \not\models \bot$
- Relevance: $\mathcal{H} \not\models \Phi$
- Explanatoriness: $\mathcal{K} \not\models \Phi, \mathcal{H} \not\models \Phi$
- Minimality: $\Phi$ is subset / cardinality / semantically minimal

[Elsenbroich, Kutz, Sattler; OWLED 2006]

# Abduction in Description Logics

Different types of abduction based on shape of observation and hypothesis

- Concept Abduction
  - Given $\mathcal{O}$, $C$, find $D$ s.t. $\mathcal{O} \models D \sqsubseteq C$
  - Generate subsumees
- TBox / ABox / Knowledge Base Abduction
  - generate terminological knowledge / assertional facts / both

How to select good hypotheses?

- $\boxed{\text{Consistency: } \mathcal{K} \wedge \mathcal{H} \not\models \bot}$
- Relevance: $\mathcal{H} \not\models \Phi$
- Explanatoriness: $\mathcal{K} \not\models \Phi, \mathcal{H} \not\models \Phi$
- Minimality: $\Phi$ is subset / cardinality / semantically minimal

[Elsenbroich, Kutz, Sattler; OWLED 2006]

# Abduction in Description Logics

Different types of abduction based on shape of observation and hypothesis

- Concept Abduction
  - Given $\mathcal{O}$, $C$, find $D$ s.t. $\mathcal{O} \models D \sqsubseteq C$
  - Generate subsumees
- TBox / ABox / Knowledge Base Abduction
  - generate terminological knowledge / assertional facts / both

How to select good hypotheses?

- Consistency: $\mathcal{K} \land \mathcal{H} \not\models \bot$
- Relevance: $\mathcal{H} \not\models \Phi$
- Explanatoriness: $\mathcal{K} \not\models \Phi, \mathcal{H} \not\models \Phi$
- Minimality: $\Phi$ is subset / cardinality / semantically minimal

[Elsenbroich, Kutz, Sattler; OWLED 2006]

# Abduction in Description Logics

Different types of abduction based on shape of observation and hypothesis

- Concept Abduction
  - Given $\mathcal{O}$, $C$, find $D$ s.t. $\mathcal{O} \models D \sqsubseteq C$
  - Generate subsumees
- TBox / ABox / Knowledge Base Abduction
  - generate terminological knowledge / assertional facts / both

How to select good hypotheses?

- Consistency: $\mathcal{K} \wedge \mathcal{H} \not\models \bot$
- Relevance: $\mathcal{H} \not\models \Phi$
- Explanatoriness: $\mathcal{K} \not\models \Phi, \mathcal{H} \not\models \Phi$
- Minimality: $\Phi$ is subset / cardinality / semantically minimal

[Elsenbroich, Kutz, Sattler; OWLED 2006]

# Abduction in Description Logics

Different types of abduction based on shape of observation and hypothesis

- Concept Abduction
  - Given $\mathcal{O}$, $C$, find $D$ s.t. $\mathcal{O} \models D \sqsubseteq C$
  - Generate subsumees
- TBox / ABox / Knowledge Base Abduction
  - generate terminological knowledge / assertional facts / both

How to select good hypotheses?

- Consistency: $\mathcal{K} \wedge \mathcal{H} \not\models \bot$
- Relevance: $\mathcal{H} \not\models \Phi$
- Explanatoriness: $\mathcal{K} \not\models \Phi, \mathcal{H} \not\models \Phi$
- Minimality: $\Phi$ is subset / cardinality / semantically minimal

[Elsenbroich, Kutz, Sattler; OWLED 2006]

# Selecting Good Hypotheses

$\mathsf{p1} : \mathsf{CoronaPatient}$

$\mathsf{CoronaPatient} \equiv$
$\mathsf{Patient} \sqcap \exists \mathsf{infectedWith}.\mathsf{Corona}$
$\vdots$

$\mathsf{p1} : \mathsf{Patient}$
$\mathsf{p1} : \exists \mathsf{infectedWith}.\mathsf{Corona}$

# Selecting Good Hypotheses

$\mathsf{p1} : \mathsf{CoronaPatient}$

$\mathsf{CoronaPatient} \equiv$
$\mathsf{Patient} \sqcap \exists \mathsf{infectedWith}.\mathsf{Corona}$
$\vdots$

$\mathsf{p1} : \mathsf{Patient}$
$\mathsf{p1} : \exists \mathsf{infectedWith}.\mathsf{Corona}$

$\mathsf{c1} : \mathsf{InfectedComputer}$

$\mathsf{InfectedComputer} \equiv$
$\mathsf{Vulnerable} \sqcap \exists \mathsf{inContactWith}.\mathsf{Worm}$
$\vdots$

$\mathsf{c1} : \mathsf{Vulnerable}$
$\mathsf{c1} : \exists \mathsf{inContactWith}.\mathsf{Worm}$

# Selecting Good Hypotheses

$p1 : \text{CoronaPatient}$

$\text{CoronaPatient} \equiv$
$\text{Patient} \sqcap \exists \text{infectedWith}.\text{Corona}$
$\vdots$

$p1 : \text{Patient}$
$p1 : \exists \text{infectedWith}.\text{Corona}$

$c1 : \text{InfectedComputer}$

$\text{InfectedComputer} \equiv$
$\text{Vulnerable} \sqcap \exists \text{inContactWith}.\text{Worm}$
$\vdots$

$c1 : \text{Vulnerable}$
$c1 : \exists \text{inContactWith}.\text{Worm}$

*Logics alone is not sufficient to determine good hypotheses!*

# Selecting Good Hypotheses

$\mathsf{p1} : \mathsf{CoronaPatient}$

$\mathsf{CoronaPatient} \equiv$
$\mathsf{Patient} \sqcap \exists \mathsf{infectedWith}.\mathsf{Corona}$
$\vdots$

$\mathsf{p1} : \mathsf{Patient}$
$\mathsf{p1} : \exists \mathsf{infectedWith}.\mathsf{Corona}$

$\mathsf{c1} : \mathsf{InfectedComputer}$

$\mathsf{InfectedComputer} \equiv$
$\mathsf{Vulnerable} \sqcap \exists \mathsf{inContactWith}.\mathsf{Worm}$
$\vdots$

$\mathsf{c1} : \mathsf{Vulnerable}$
$\mathsf{c1} : \exists \mathsf{inContactWith}.\mathsf{Worm}$

*Logics alone is not sufficient to determine good hypotheses!*

*We need further knowledge from the user.*

# How to Select Good Hypotheses?

Different ways to restrict solution space:

- Specify set of abducible axioms / concepts
  - Example: $\{\ \mathsf{c1} : \mathsf{Vulnerable},\ \mathsf{c1} : \exists\mathsf{inContactWith}.\mathsf{Worm},\ \ldots\ \}$
  - Requirement: every hypothesis is a subset of this set

# How to Select Good Hypotheses?

Different ways to restrict solution space:

- Specify set of abducible axioms / concepts
  - Example: $\{\,\mathbf{c1} : \mathbf{Vulnerable},\ \mathbf{c1} : \exists\mathbf{inContactWith}.\mathbf{Worm},\ \ldots\,\}$
  - Requirement: every hypothesis is a subset of this set
  - *But what if we do not know what axioms we are looking for?*

# How to Select Good Hypotheses?

Different ways to restrict solution space:

- Specify set of abducible axioms / concepts
  - Example: $\{\ \mathsf{c1} : \mathsf{Vulnerable},\ \mathsf{c1} : \exists\mathsf{inContactWith}.\mathsf{Worm},\ \ldots\}$
  - Requirement: every hypothesis is a subset of this set
  - *But what if we do not know what axioms we are looking for?*

- Specify an signature for the hypothesis
  - Example: $\{\ \mathsf{Vulnerable},\ \mathsf{inContactWith},\ \mathsf{Worm},\ \ldots\}$
  - Requirement: hypotheses can only use names from the signature

# How to Select Good Hypotheses?

Different ways to restrict solution space:

- Specify set of abducible axioms / concepts
  - Example: $\{\ \mathbf{c1} : \mathbf{Vulnerable},\ \mathbf{c1} : \exists \mathbf{inContactWith}.\mathbf{Worm},\ \ldots\ \}$
  - Requirement: every hypothesis is a subset of this set
  - *But what if we do not know what axioms we are looking for?*

- Specify an signature for the hypothesis
  - Example: $\{\ \mathbf{Vulnerable},\ \mathbf{inContactWith},\ \mathbf{Worm},\ \ldots\ \}$
  - Requirement: hypotheses can only use names from the signature
  - Often combined with constraints on the structure of axioms:
    - disallow complex concepts, specify syntactic *patterns*
    - . . . but then we may miss the right hypothesis

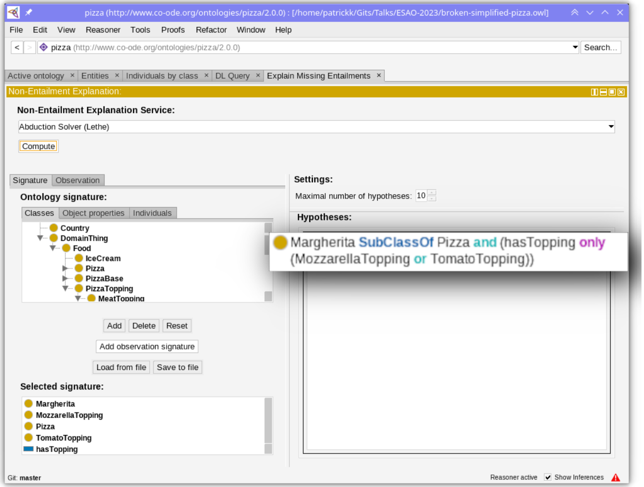
# How to Select Good Hypotheses?

Different ways to restrict solution space:

- Specify set of abducible axioms / concepts
  - Example: $\{\ \mathbf{c1} : \mathbf{Vulnerable},\ \mathbf{c1} : \exists \mathbf{inContactWith}.\mathbf{Worm}, \ldots\}$
  - Requirement: every hypothesis is a subset of this set
  - *But what if we do not know what axioms we are looking for?*

- Specify an signature for the hypothesis
  - Example: $\{\ \mathbf{Vulnerable},\ \mathbf{inContactWith},\ \mathbf{Worm}, \ldots\}$
  - Requirement: hypotheses can only use names from the signature
  - Often combined with constraints on the structure of axioms:
    - disallow complex concepts, specify syntactic *patterns*
    - ...but then we may miss the right hypothesis
  - ...or we allow arbitrary concepts within signauture
    - more flexibility
    - *solution space becomes unbounded!*

# Signature-Based Abduction with evee-protege


pizza (http://www.co-ode.org/ontologies/pizza/2.0.0) : [/home/patrickk/Gits/Talks/ESAO-2023/broken-simplified-pizza.owl]
File Edit View Reasoner Tools Proofs Refactor Window Help
pizza (http://www.co-ode.org/ontologies/pizza/2.0.0)
Search...
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×
Non-Entailment Explanation:
Compute
Signature Observation
Classes Object properties Individuals
Country
DomainThing
Food
IceCream
Pizza
PizzaBase
PizzaTopping
MeatTopping
Add Delete Reset
Add observation signature
Load from file Save to file
Git: master
Reasoner active ☑ Show Inferences

Non-Entailment Explanation Service: Abduction Solver (Lethe)

Ontology signature:

Selected signature:

- Margherita
- MozzarellaTopping
- Pizza
- TomatoTopping
- hasTopping

Settings:

Maximal number of hypotheses: 10

Hypotheses:

Margherita SubClassOf Pizza and (hasTopping only (MozzarellaTopping or TomatoTopping))

- We believe that in many application scenarios, the signature will be quite natural.
- In any case, our plugin will support the user in finding the appropriate signature.

# Complexity of Signature-Based ABox Abduction

## Hypotheses **without complex concepts**

| Description Logic | $\mathcal{EL}$ | $\mathcal{EL}_\bot$ | $\mathcal{ALCI}$ | $\mathcal{ALCF}$ |
|---|---|---|---|---|
| *deciding existence* | P-c | ExpTime-c | coNExpTime-c | undecidable |
| *worst case size* | polynomial | exponential | exponential | not computable |

## Hypotheses **with complex concepts**

| Description Logic | $\mathcal{EL}$ | $\mathcal{EL}_\bot$ | $\mathcal{ALC}$ |
|---|---|---|---|
| *deciding existence* | P-c | ExpTime-c | N2ExpTime$^{\text{NP}}$-c |
| *worst case size* | polynomial | 2-exponential | 3-exponential |
| *worst case #individuals* | polynomial | exponential | 2-exponential |

[Koopmann; IJCAI 2021]

# Complexity of Signature-Based ABox Abduction

### Hypotheses **without complex concepts**

| Description Logic | $\mathcal{EL}$ | $\mathcal{EL}_\bot$ | $\mathcal{ALCI}$ | $\mathcal{ALCF}$ |
|---|---|---|---|---|
| *deciding existence* | P-c | ExpTime-c | coNExpTime-c | undecidable |
| *worst case size* | polynomial | exponential | exponential | not computable |

### Hypotheses **with complex concepts**

| Description Logic | $\mathcal{EL}$ | $\mathcal{EL}_\bot$ | $\mathcal{ALC}$ |
|---|---|---|---|
| *deciding existence* | P-c | ExpTime-c | N2ExpTime$^{NP}$-c |
| *worst case size* | polynomial | 2-exponential | 3-exponential |
| *worst case #individuals* | polynomial | exponential | 2-exponential |

[Koopmann; IJCAI 2021]

# Complexity of Signature-Based ABox Abduction

### Hypotheses **without complex concepts**

| Description Logic | $\mathcal{EL}$ | $\mathcal{EL}_\bot$ | $\mathcal{ALCI}$ | $\mathcal{ALCF}$ |
|---|---|---|---|---|
| *deciding existence* | P-c | ExpTime-c | coNExpTime-c | undecidable |
| *worst case size* | polynomial | exponential | exponential | not computable |

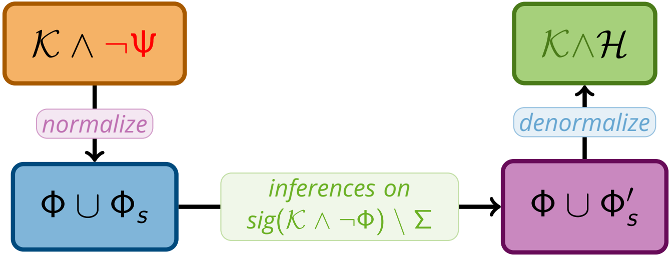
### Hypotheses **with complex concepts**

| Description Logic | $\mathcal{EL}$ | $\mathcal{EL}_\bot$ | $\mathcal{ALC}$ |
|---|---|---|---|
| *deciding existence* | P-c | ExpTime-c | N2ExpTime$^{\text{NP}}$-c |
| *worst case size* | polynomial | 2-exponential | 3-exponential |
| *worst case #individuals* | polynomial | exponential | 2-exponential |

[Koopmann; IJCAI 2021]

# Signature-Based Abduction in Practice

- We support signature-based knowledge base abduction
- We generate complex concepts, but not fresh individual names
- Our approach reduces abduction to deduction:



$\mathcal{K} \land \neg\Psi$
normalize
$\Phi \cup \Phi_s$
inferences on $sig(\mathcal{K} \land \neg\Phi) \setminus \Sigma$
$\Phi \cup \Phi'_s$
denormalize
$\mathcal{K} \land \mathcal{H}$

- Normal form ensures finitely many inferences
- Dedicated calculus ensures all relevant inferences are performed

[Koopmann, Del-Pinto, Tourret, Schmidt; KR 2020]

# Signature-Based Abduction in Practice

The output is a Boolean $\mathcal{ALCOI}\mu$ KB capturing the entire solution space

$$\mathsf{pc} : \mathsf{InfectedComputer}$$

$$\begin{aligned} & \mathsf{pc} : \mu X.\,(\exists \mathsf{pluggedTo}.\mathsf{InfectedUSBDrive} \sqcup \exists \mathsf{connectedTo}^{-}.X) \\ \vee\ & \mathsf{infected\_pc} : \mu X.\,(\exists \mathsf{connectedTo}.\{\mathsf{pc}\} \sqcup \exists \mathsf{connectedTo}.X) \end{aligned}$$

# Signature-Based Abduction in Practice

The output is a Boolean $\mathcal{ALCOI}\mu$ KB capturing the entire solution space

$$\mathsf{pc} : \mathsf{InfectedComputer}$$

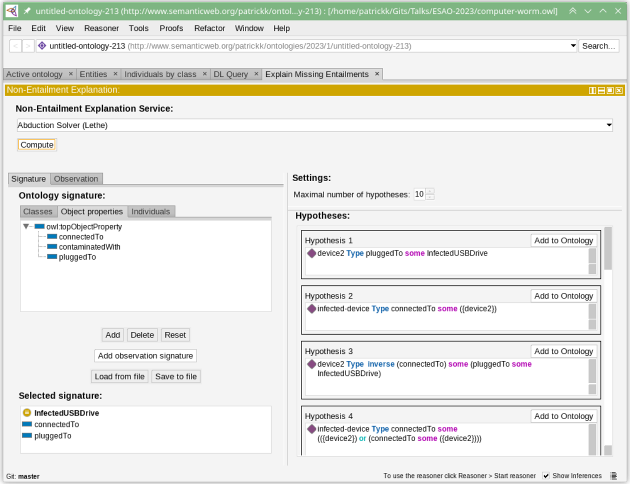
$$\begin{aligned} & \mathsf{pc} : \mu X.\,(\exists \mathsf{pluggedTo}.\mathsf{InfectedUSBDrive} \sqcup \exists \mathsf{connectedTo}^{-}.X) \\ \vee\ & \mathsf{infected\_pc} : \mu X.\,(\exists \mathsf{connectedTo}.\{\mathsf{pc}\} \sqcup \exists \mathsf{connectedTo}.X) \end{aligned}$$

Hypotheses in $\mathcal{ALCOI}$ are generated via unravelling:

1. $\mathsf{pc} : \exists \mathsf{pluggedTo}.\mathsf{InfectedUSBDrive}$
2. $\mathsf{infected\_pc} : \exists \mathsf{connectedTo}.\{\mathsf{pc}\}$
3. $\mathsf{pc} : \exists \mathsf{connectedTo}^{-}.\exists \mathsf{pluggedTo}.\mathsf{InfectedUSBDrive}$
4. $\mathsf{infected\_pc} : \exists \mathsf{connectedTo}.\exists \mathsf{connectedTo}.\{\mathsf{pc}\}$

$\vdots$

# Unravelling in evee-protege


untitled-ontology-213 (http://www.semanticweb.org/patrickk/ontol...y-213) : [/home/patrickk/Gits/Talks/ESAO-2023/computer-worm.owl]
File Edit View Reasoner Tools Proofs Refactor Window Help
untitled-ontology-213 (http://www.semanticweb.org/patrickk/ontologies/2023/1/untitled-ontology-213)
Search...
Active ontology × Entities × Individuals by class × DL Query × Explain Missing Entailments ×
Non-Entailment Explanation:
Compute
Signature Observation
Classes Object properties Individuals
Add Delete Reset
Add observation signature
Load from file Save to file
Add to Ontology
Git: master
To use the reasoner click Reasoner > Start reasoner
☑ Show Inferences

**Non-Entailment Explanation Service:**

Abduction Solver (Lethe)

**Ontology signature:**

- owl:topObjectProperty
  - connectedTo
  - contaminatedWith
  - pluggedTo

**Selected signature:**

- **InfectedUSBDrive**
- connectedTo
- pluggedTo

**Settings:**

Maximal number of hypotheses: 10

**Hypotheses:**

Hypothesis 1

device2 Type pluggedTo some InfectedUSBDrive

Hypothesis 2

infected-device Type connectedTo some ({device2})

Hypothesis 3

device2 Type inverse (connectedTo) some (pluggedTo some InfectedUSBDrive)

Hypothesis 4

infected-device Type connectedTo some (({device2}) or (connectedTo some ({device2})))

# Practicality

Evaluation in [Koopmann, Del-Pinto, Tourret, Schmidt; KR 2020]:

- Real ontologies up to 50,000 axioms
- Generated observations and signatures (4 settings)
- Timeout of 5 minutes

Results:

- Success rates: 89.5% – 96.4%
- Computation time: 2.5 – 16.9 seconds on average
- Solution size: 9.7 – 24.2 axioms on average
- Alternatives: 1.8 – 3.7 disjuncts on average
- Fixpoints operators: 0.8 – 5.3 % of cases

# Conclusion

Abduction to explain missing entailments

- generate extension of knowledge to create given entailment
- additional criteria needed to select good hypothesis
  - consistency, minimality, signature restrictions
- full signature-based abduction is challenging, but often possible in practice
- supported in coming version of `evee-protege`
  - also supports *model generation* and *connection-minimal abduction* to explain missing entailments
  - check also out `Evonne` as a more advanced tool for explaining DL reasoning
    - online demo under `https://imld.de/evonne`

# Conclusion

Abduction to explain missing entailments

- generate extension of knowledge to create given entailment
- additional criteria needed to select good hypothesis
  - consistency, minimality, signature restrictions
- full signature-based abduction is challenging, but often possible in practice
- supported in coming version of `evee-protege`
  - also supports *model generation* and *connection-minimal abduction* to explain missing entailments
  - check also out `Evonne` as a more advanced tool for explaining DL reasoning
    - online demo under `https://imld.de/evonne`

- Future directions:
  - User study
  - More refined and interactive criteria to select hypotheses
  - Practical abduction to invent *fresh individuals*

# Bibliography

M. Horridge, B. Parsia, U. Sattler. "*Laconic and precise justifications in OWL*". In: The Semantic Web—ISWC 2008 (2008).

C. Alrabbaa, S. Borgwardt, T. Friese, P. Koopmann, J. Méndez, A. Popovič. "*On the eve of true explainability for OWL ontologies: Description logic proofs with Evee and Evonne*". In: Proceedings of DL 22 (2022).

C. Elsenbroich, O. Kutz, U. Sattler. "*A Case for Abductive Reasoning over Ontologies*". In: Proceedings of OWLED (2006).

F. Wei-Kleiner, Z. Dragisic, P. Lambrix. "*Abduction framework for repairing incomplete $\mathcal{EL}$ ontologies: Complexity results and algorithms*". In: Proceedings of the AAAI Conference on Artificial Intelligence (2014).

D. Calvanese, M. Ortiz, M. Simkus, G. Stefanoni, G. "*Reasoning about explanations for negative query answers in DL-Lite*". Journal of Artificial Intelligence Research, 48, 635-669 (2013).

P. Koopmann. "*Two ways of explaining negative entailments in description logics using abduction*." Explainable Logic-Based Knowledge Representation (XLoKR 2021) (2021).

# Bibliography

M. Bienvenu. “*Complexity of Abduction in the $\mathcal{EL}$ Family of Lightweight Description Logics*“. In: Proceedings of KR (2008).

J. Du, H. Wan, H. Ma. “*Practical TBox abduction based on justification patterns*”. In: Proceedings of the AAAI Conference on Artificial Intelligence (2017).

Koopmann, P. “*Signature-based abduction with fresh individuals and complex concepts for description logics*”. In: Proceedings of IJCAR (2021).

Koopmann, P. “*Signature-Based ABox Abduction in $\mathcal{ALC}$ is Hard*”. In: Proceedings of SOQE (2021).

P. Koopmann, W. Del-Pinto, S. Tourret, R. Schmidt. “*Signature-based abduction for expressive description logics*”. In: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (2020).

F. Haifani, P. Koopmann, S. Tourret, and C. Weidenbach. “*Connection-minimal abduction in $\mathcal{EL}$ via translation to FOL*”. In 11th International Joint Conference on Automated Reasoning (2022).

# Bibliography

J. Du, K. Wang, and Y. Shen. “*A tractable approach to ABox abduction over description logic ontologies*”. In: Proceedings of the AAAI Conference on Artificial Intelligence (2014).