Explainable Answer-Set Programming

Research Project

Cape-KR 2024

Tobias Geibinger

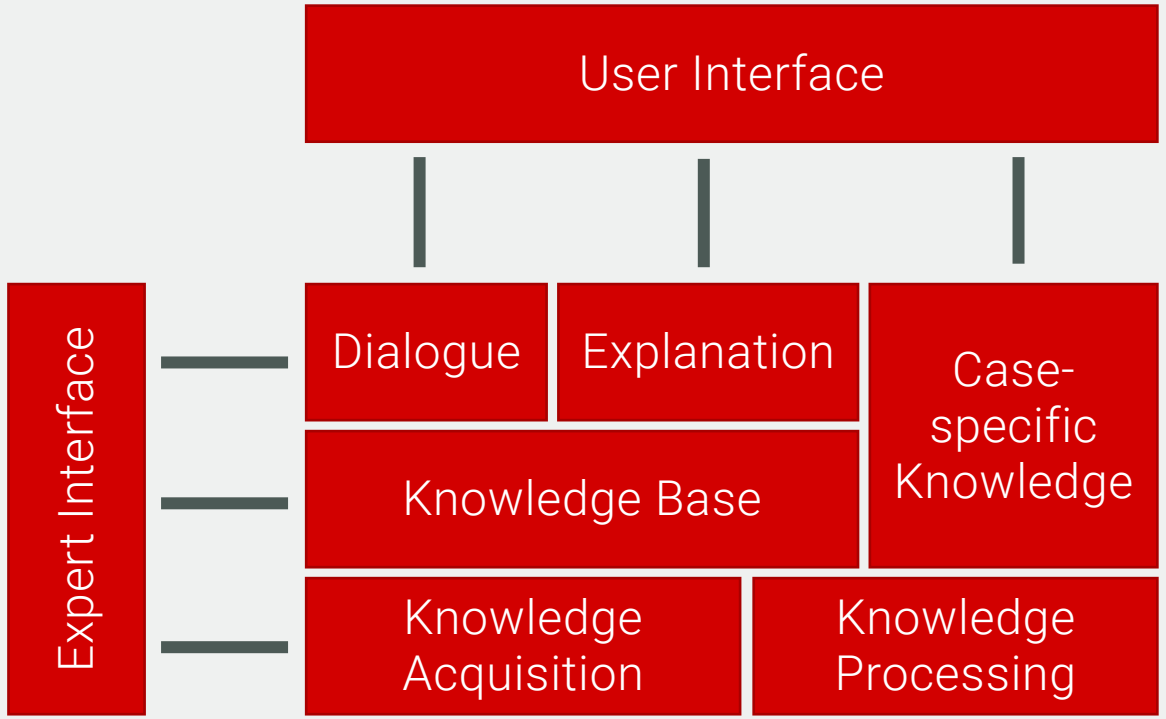
TU WIEN Informatics

kbs Knowledge-Based Systems Group

logics

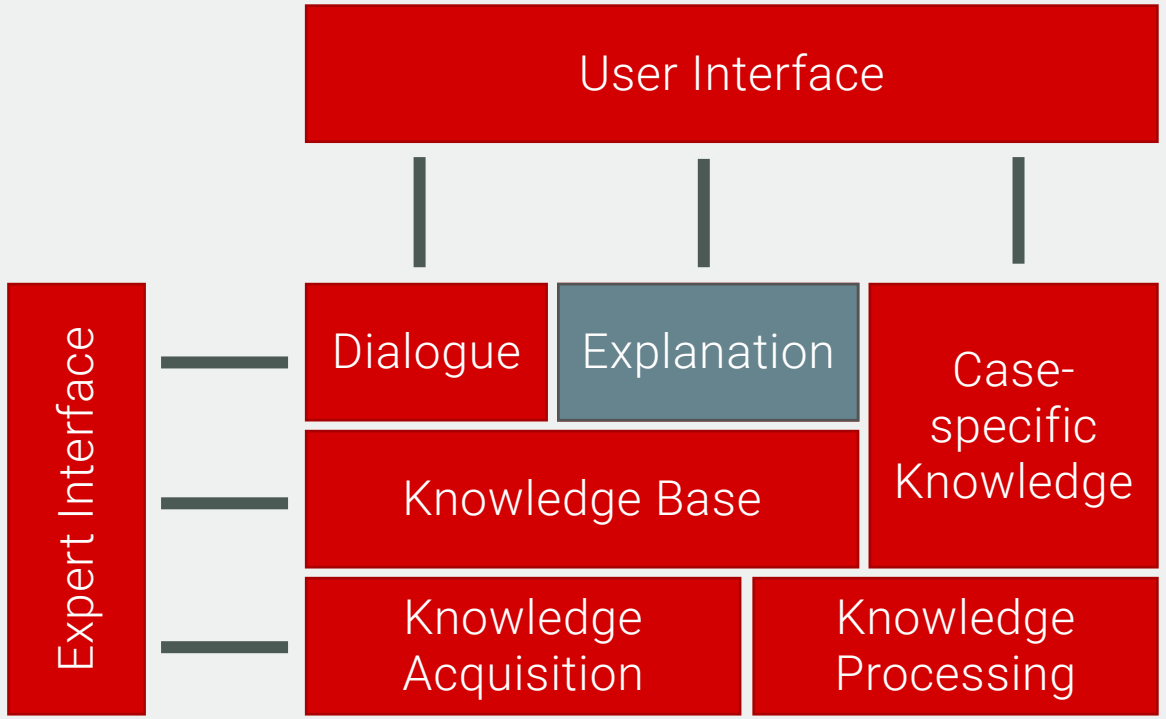
Introduction

Historical Architecture of Symbolic AI Systems



Introduction

Historical Architecture of Symbolic AI Systems



Motivation

Motivation

- We study **Answer-set Programming (ASP)**

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**
- **Efficient solvers** exist

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**
- **Efficient solvers** exist
- It's **rule-based nature** makes it attractive for **critical domains**

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**
- **Efficient solvers** exist
- It's **rule-based nature** makes it attractive for **critical domains**
- It has been utilised for:

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**
- **Efficient solvers** exist
- It's **rule-based nature** makes it attractive for **critical domains**
- It has been utilised for:

Medicine

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**
- **Efficient solvers** exist
- It's **rule-based nature** makes it attractive for **critical domains**
- It has been utilised for:

Medicine

Scheduling

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**
- **Efficient solvers** exist
- It's **rule-based nature** makes it attractive for **critical domains**
- It has been utilised for:


Medicine


Scheduling


Planning

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**
- **Efficient solvers** exist
- It's **rule-based nature** makes it attractive for **critical domains**
- It has been utilised for:


Medicine


Scheduling


Planning


Logistics

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**
- **Efficient solvers** exist
- It's **rule-based nature** makes it attractive for **critical domains**
- It has been utilised for:


Medicine


Scheduling


Planning


Logistics


Pathfinding

Motivation

- We study **Answer-set Programming (ASP)**
- ASP is a popular **declarative problem-solving paradigm**
- **Efficient solvers** exist
- It's **rule-based nature** makes it attractive for **critical domains**
- It has been utilised for:


Medicine


Scheduling


Planning


Logistics


Pathfinding

Provided solutions still need to be **explained**

Answer-set Programming (ASP)

Basics

Answer-set Programming (ASP)

Basics

ASP programs are finite sets of rules:

Answer-set Programming (ASP)

Basics

ASP programs are finite sets of rules:

$$a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n$$

Answer-set Programming (ASP)

Basics

ASP programs are finite sets of rules:

$$a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n$$

$a_1, \ldots, a_k, b_{k+1}, \ldots, b_m$ and $b_{m+1}, \ldots, b_n$ are **atoms**

Answer-set Programming (ASP)

Basics

default negation

ASP programs are finite sets of rules:

$$a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n$$

$a_1, \ldots, a_k, b_{k+1}, \ldots, b_m$ and $b_{m+1}, \ldots, b_n$ are **atoms**

Answer-set Programming (ASP)

Basics

default negation

ASP programs are finite sets of rules:

$$a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n$$

$a_1, \ldots, a_k, b_{k+1}, \ldots, b_m$ and $b_{m+1}, \ldots, b_n$ are **atoms**

Atoms can be **ground** or have **variables**:

Answer-set Programming (ASP)

Basics

default negation

ASP programs are finite sets of rules:

$$a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n$$

$a_1, \ldots, a_k, b_{k+1}, \ldots, b_m$ and $b_{m+1}, \ldots, b_n$ are **atoms**

Atoms can be **ground** or have **variables**:

$$p$$

Answer-set Programming (ASP)

Basics

default negation

ASP programs are finite sets of rules:

$$a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n$$

$a_1, \ldots, a_k, b_{k+1}, \ldots, b_m$ and $b_{m+1}, \ldots, b_n$ are **atoms**

Atoms can be **ground** or have **variables**:

$$p \qquad\qquad color(\mathtt{C})$$

Answer-set Programming (ASP)

Basics

default negation

ASP programs are finite sets of rules:

$$a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n$$

$a_1, \ldots, a_k$, $b_{k+1}, \ldots, b_m$ and $b_{m+1}, \ldots, b_n$ are **atoms**

instantiated during **grounding**

Atoms can be **ground** or have **variables**:

$$p \qquad color(\mathtt{C})$$

Answer-set Programming (ASP)

Basics

default negation

ASP programs are finite sets of rules:

$$a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n$$

$a_1, \ldots, a_k$, $b_{k+1}, \ldots, b_m$ and $b_{m+1}, \ldots, b_n$ are **atoms**

instantiated during **grounding**

Atoms can be **ground** or have **variables**:

$$p \qquad color(\mathtt{C})$$

An **interpretation** I is a set of ground atoms, which **satisfies** a rule if:

Answer-set Programming (ASP)

Basics

default negation

ASP programs are finite sets of rules:

$$a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n$$

$a_1, \ldots, a_k$, $b_{k+1}, \ldots, b_m$ and $b_{m+1}, \ldots, b_n$ are **atoms**

instantiated during **grounding**

Atoms can be **ground** or have **variables**:

$$p \qquad color(\mathtt{C})$$

An **interpretation** I is a set of ground atoms, which **satisfies** a rule if:

whenever $b_{k+1}, \ldots, b_m \in I$ and $b_{m+1}, \ldots, b_n \notin I$, then $a_i \in I$ for some $(1 \leq i \leq k)$

Answer-set Programming (ASP)

Basics

Answer-set Programming (ASP)

Basics

I is an **answer set** of program P if it is a **minimal model** of the Gelfond-Lifschitz reduct

Answer-set Programming (ASP)

Basics

I is an **answer set** of program P if it is a **minimal model** of the Gelfond-Lifschitz reduct

$$P^I := \{ \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m \quad | \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n \in P, \; b_{m+1}, \ldots, b_n \notin I \quad \}$$

Answer-set Programming (ASP)

Basics

I is an **answer set** of program P if it is a **minimal model** of the Gelfond-Lifschitz reduct

$$P^I := \{ \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m \quad | \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n \in P, \; b_{m+1}, \ldots, b_n \notin I \quad \}$$

Intuition: Assuming everything not in I is **false**, the rest is **stable** w.r.t. P

Answer-set Programming (ASP)

Basics

I is an **answer set** of program P if it is a **minimal model** of the Gelfond-Lifschitz reduct

$$P^I := \{ \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m \quad | \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n \in P, \; b_{m+1}, \ldots, b_n \notin I \quad \}$$

Intuition: Assuming everything not in I is **false**, the rest is **stable** w.r.t. P

Example:

$$\begin{aligned} warhawk(\mathsf{X}) &\leftarrow republican(\mathsf{X}), not\ pacifist(\mathsf{X}) \\ pacifist(\mathsf{X}) &\leftarrow quaker(\mathsf{X}), not\ warhawk(\mathsf{X}) \\ republic(\mathsf{nixon}) &\leftarrow \\ quaker(\mathsf{nixon}) &\leftarrow \end{aligned}$$

Answer-set Programming (ASP)

Basics

I is an **answer set** of program P if it is a **minimal model** of the Gelfond-Lifschitz reduct

$$P^I := \{ \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m \quad | \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n \in P, \; b_{m+1}, \ldots, b_n \notin I \quad \}$$

Intuition: Assuming everything not in I is **false**, the rest is **stable** w.r.t. P

Example:

$$\begin{aligned} warhawk(\mathsf{X}) &\leftarrow republican(\mathsf{X}), not\ pacifist(\mathsf{X}) \\ pacifist(\mathsf{X}) &\leftarrow quaker(\mathsf{X}), not\ warhawk(\mathsf{X}) \\ republic(\mathsf{nixon}) &\leftarrow \\ quaker(\mathsf{nixon}) &\leftarrow \end{aligned}$$

Answer sets:

$$I_1 = \{ republican(\mathsf{nixon}), quaker(\mathsf{nixon}), warhawk(\mathsf{nixon}) \}$$

Answer-set Programming (ASP)

Basics

I is an **answer set** of program P if it is a **minimal model** of the Gelfond-Lifschitz reduct

$$P^I := \{ \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m \quad | \quad a_1 \vee \ldots \vee a_k \leftarrow b_{k+1}, \ldots, b_m, not\ b_{m+1}, \ldots, not\ b_n \in P, \; b_{m+1}, \ldots, b_n \notin I \quad \}$$

Intuition: Assuming everything not in I is **false**, the rest is **stable** w.r.t. P

Example:

$$\begin{aligned} warhawk(\mathsf{X}) &\leftarrow republican(\mathsf{X}), not\ pacifist(\mathsf{X}) \\ pacifist(\mathsf{X}) &\leftarrow quaker(\mathsf{X}), not\ warhawk(\mathsf{X}) \\ republic(\mathsf{nixon}) &\leftarrow \\ quaker(\mathsf{nixon}) &\leftarrow \end{aligned}$$

Answer sets:

$I_1 = \{ republican(\mathsf{nixon}), quaker(\mathsf{nixon}), warhawk(\mathsf{nixon}) \}$

$I_2 = \{ republican(\mathsf{nixon}), quaker(\mathsf{nixon}), pacifist(\mathsf{nixon}) \}$

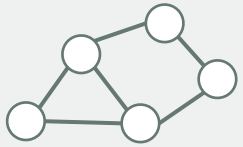
Answer-set Programming (ASP)

Methodology

Answer-set Programming (ASP)

Methodology

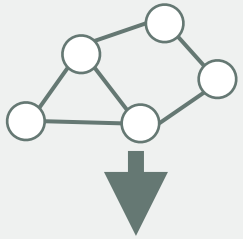
Combinatorial Problem



Answer-set Programming (ASP)

Methodology

Combinatorial Problem



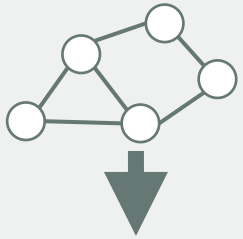
Facts

$node(1), node(2), node(3),$
$node(4), node(5), edge(1,2),$
$edge(2,3), edge(2,5), edge(3,5),$
$edge(4,5), edge(1,4)$

Answer-set Programming (ASP)

Methodology

Combinatorial Problem



Facts

$node(1), node(2), node(3),$
$node(4), node(5), edge(1,2),$
$edge(2,3), edge(2,5), edge(3,5),$
$edge(4,5), edge(1,4)$

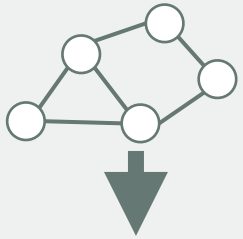
ASP Encoding

$$r(X) \vee g(X) \vee b(X) \leftarrow node(X)$$
$$\leftarrow edge(X,Y), r(X), r(Y)$$
$$\leftarrow edge(X,Y), g(X), g(Y)$$
$$\leftarrow edge(X,Y), b(X), b(Y)$$

Answer-set Programming (ASP)

Methodology

Combinatorial Problem



Facts

$node(1), node(2), node(3),$
$node(4), node(5), edge(1,2),$
$edge(2,3), edge(2,5), edge(3,5),$
$edge(4,5), edge(1,4)$

ASP Encoding

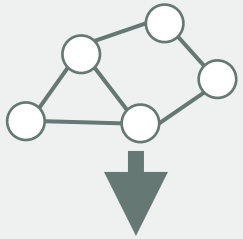
$r(X) \vee g(X) \vee b(X) \leftarrow node(X)$
$\leftarrow edge(X,Y), r(X), r(Y)$
$\leftarrow edge(X,Y), g(X), g(Y)$
$\leftarrow edge(X,Y), b(X), b(Y)$

ASP Solver

Answer-set Programming (ASP)

Methodology

Combinatorial Problem



Facts

$node(1), node(2), node(3),$
$node(4), node(5), edge(1,2),$
$edge(2,3), edge(2,5), edge(3,5),$
$edge(4,5), edge(1,4)$

ASP Encoding

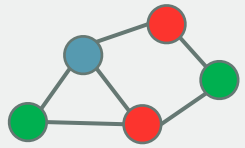
$r(X) \vee g(X) \vee b(X) \leftarrow node(X)$
$\leftarrow edge(X,Y), r(X), r(Y)$
$\leftarrow edge(X,Y), g(X), g(Y)$
$\leftarrow edge(X,Y), b(X), b(Y)$

ASP Solver

Answer set

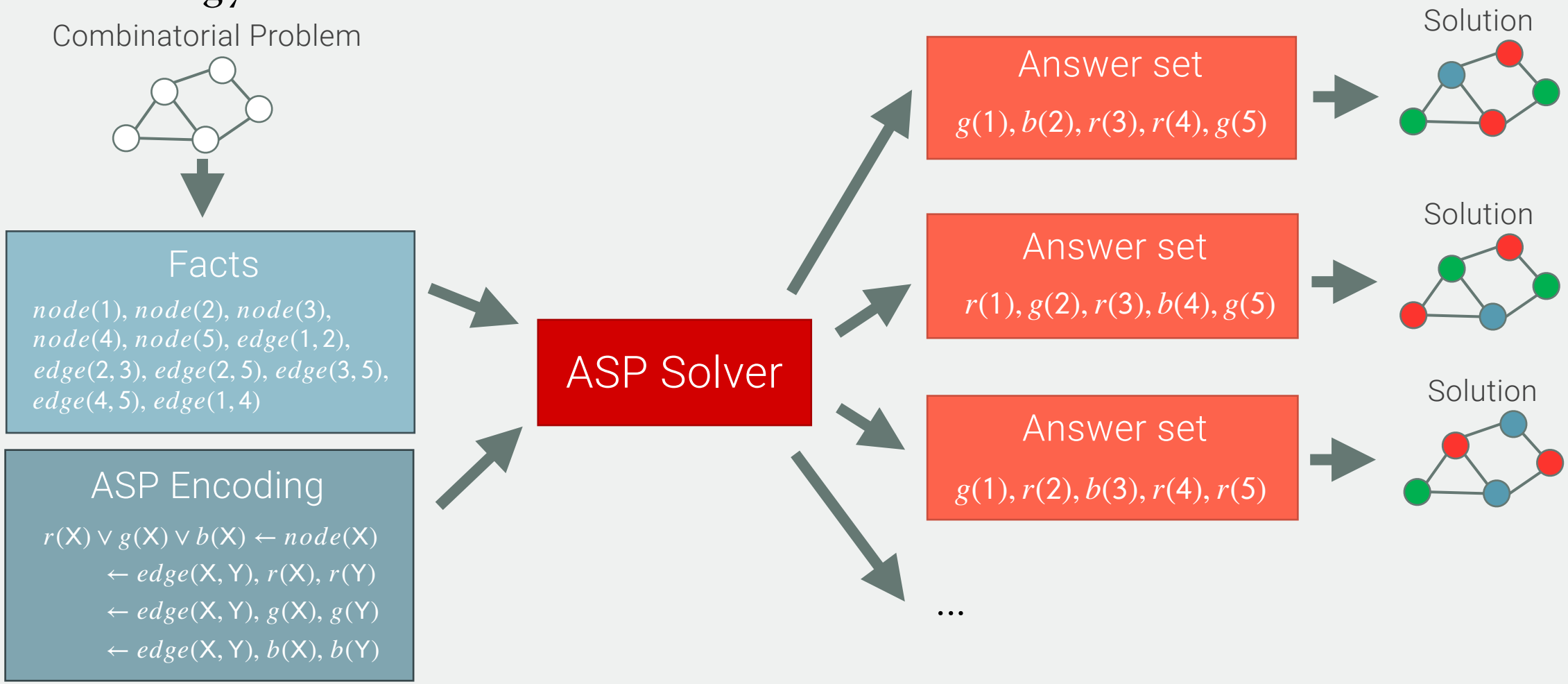
$g(1), b(2), r(3), r(4), g(5)$

Solution



Answer-set Programming (ASP)

Methodology



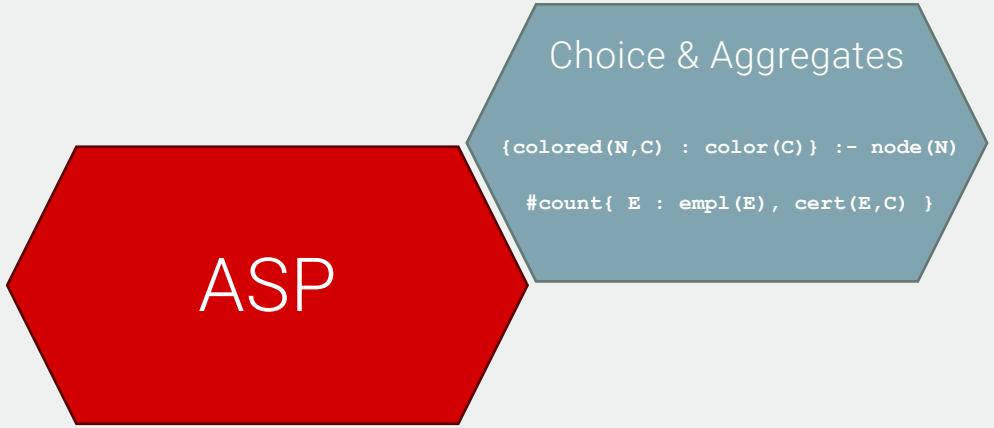
Answer-set Programming

Extensions

Answer-set Programming

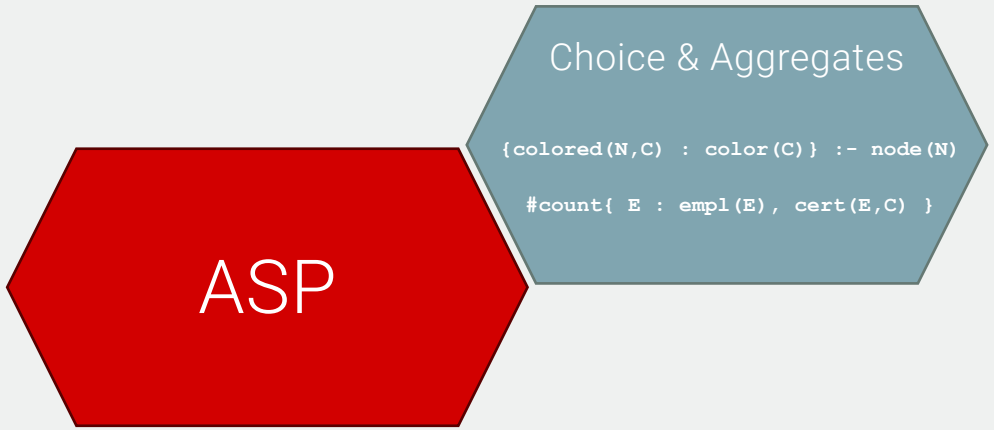
Extensions

- Choice rules & Aggregates



Answer-set Programming

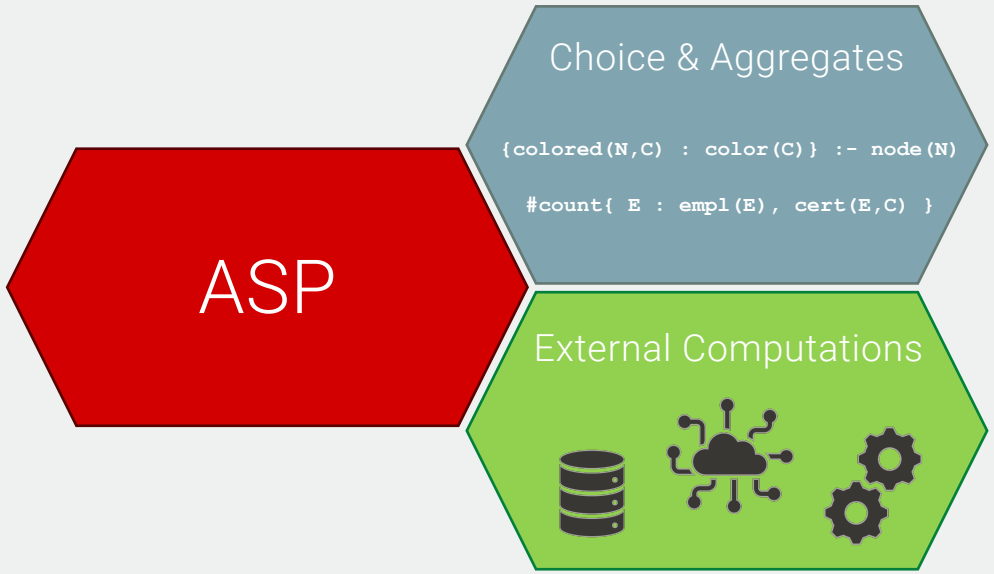
Extensions



- Choice rules & Aggregates
  - Syntactic sugar
  - but very useful

Answer-set Programming

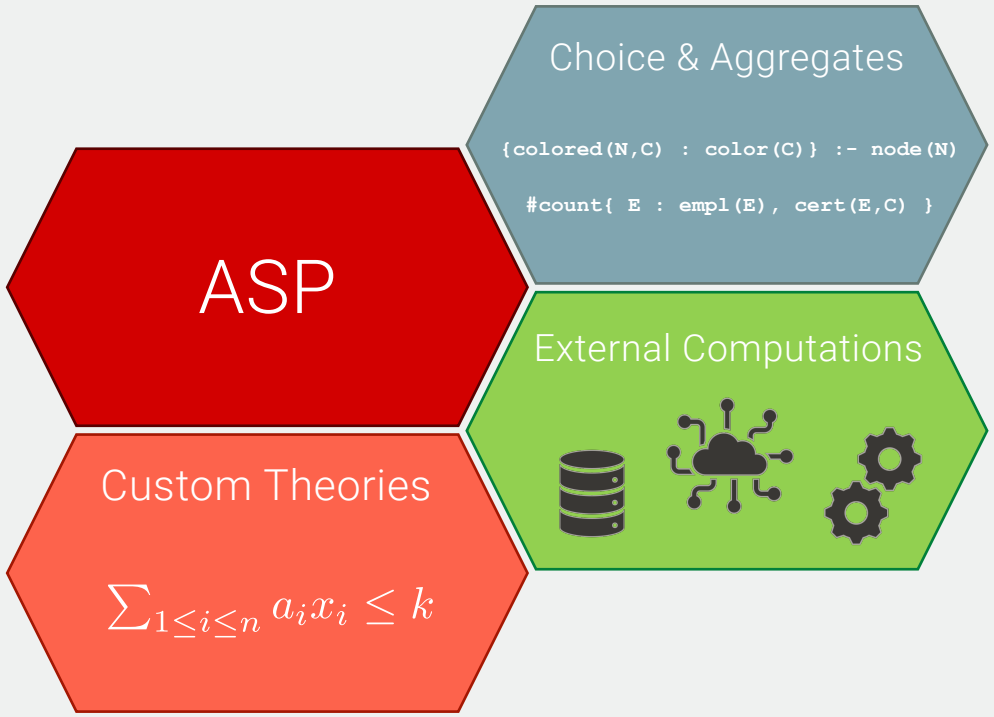
Extensions



- Choice rules & Aggregates
  - Syntactic sugar
  - but very useful
- HEX programs / clingo

Answer-set Programming

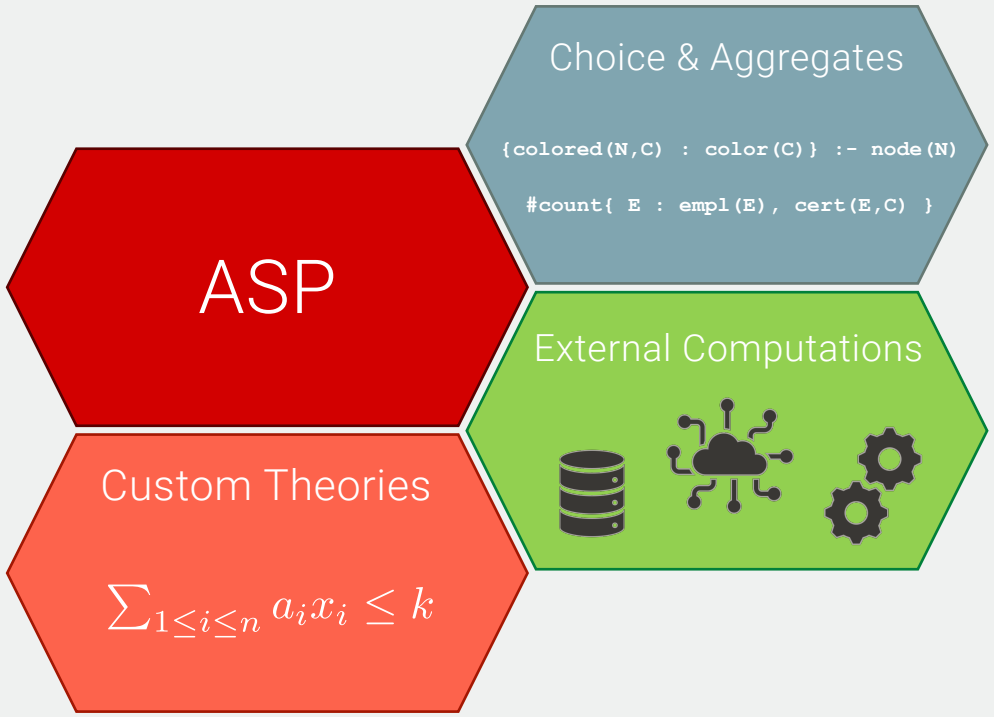
Extensions



- Choice rules & Aggregates
  - Syntactic sugar
  - but very useful
- HEX programs / clingo
- ASP modulo Theories

Answer-set Programming

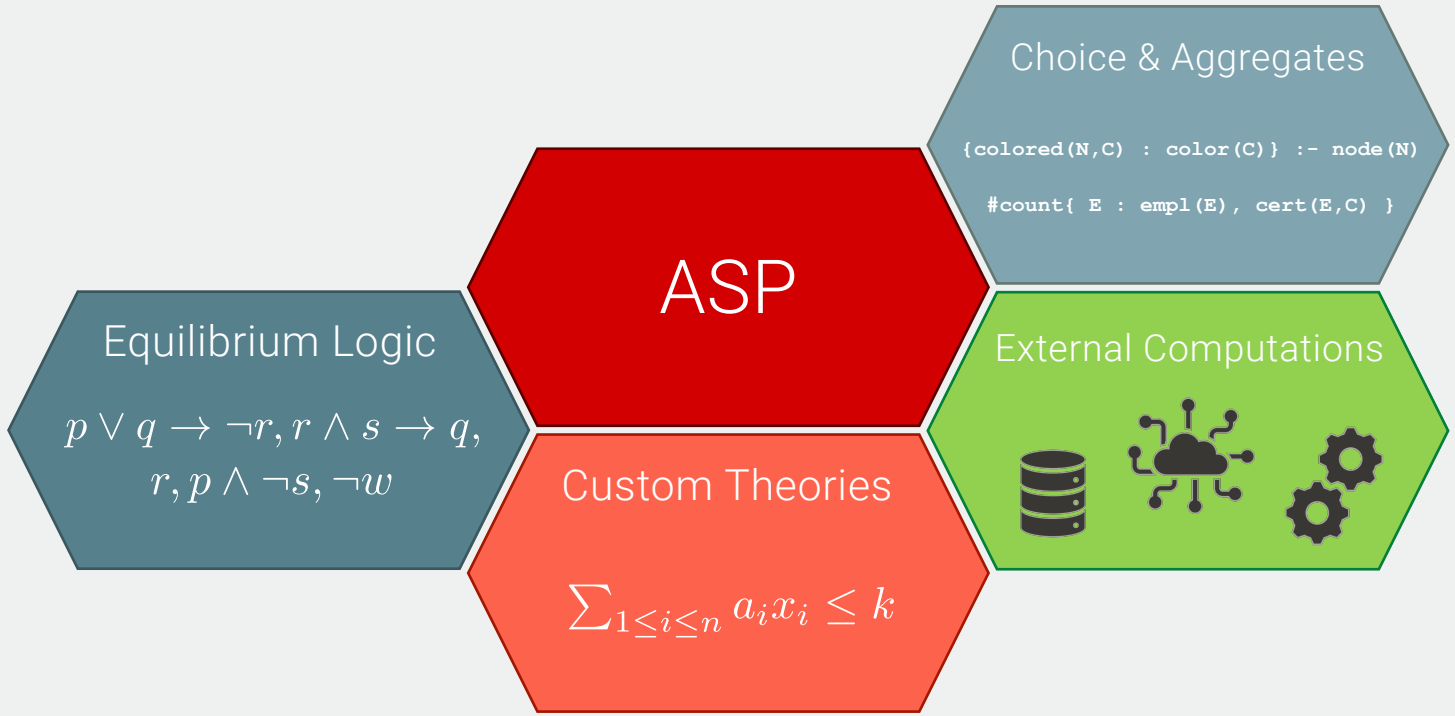
Extensions



- Choice rules & Aggregates
  - Syntactic sugar
  - but very useful
- HEX programs / clingo
- ASP modulo Theories
  - Linear Constraints (CASP)
  - Difference Logic
  - …

Answer-set Programming

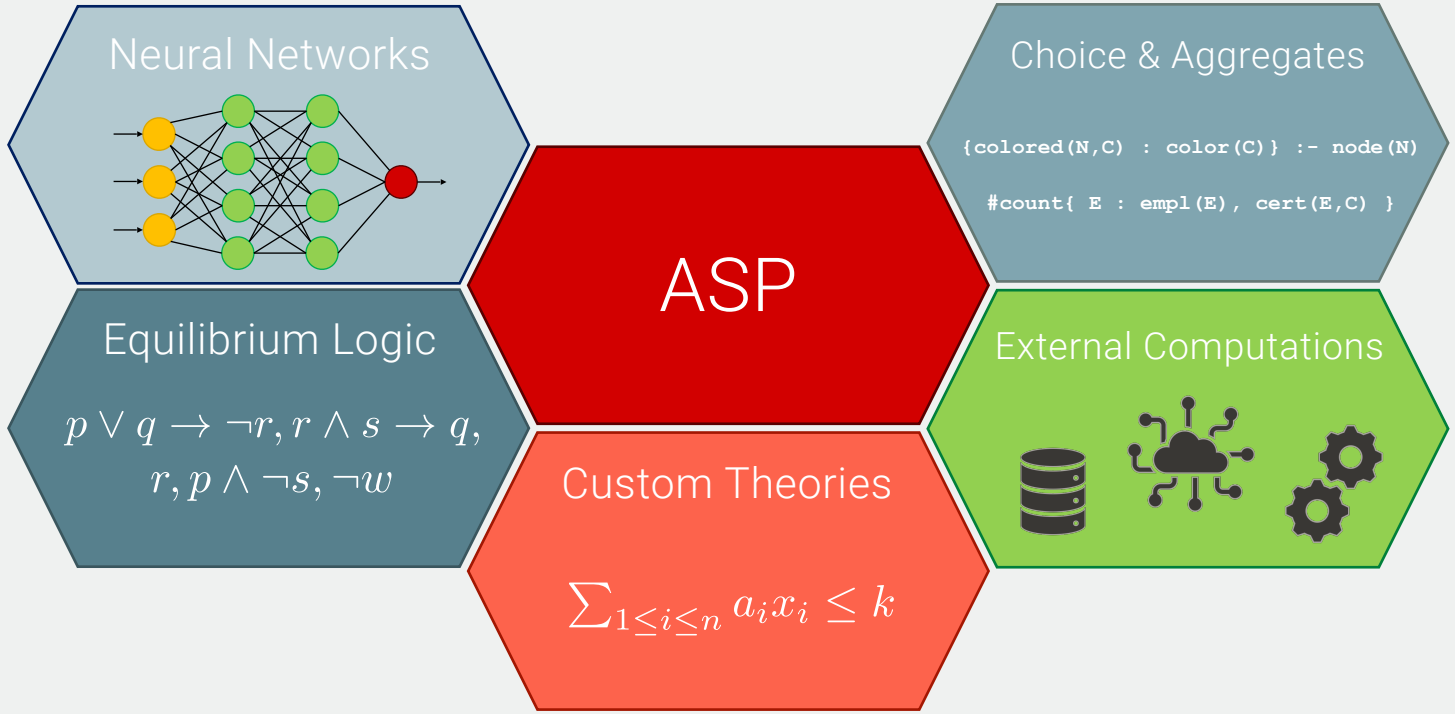
Extensions



- Choice rules & Aggregates
  - Syntactic sugar
  - but very useful
- HEX programs / clingo
- ASP modulo Theories
  - Linear Constraints (CASP)
  - Difference Logic
  - …
- Equilibrium Logic

Answer-set Programming

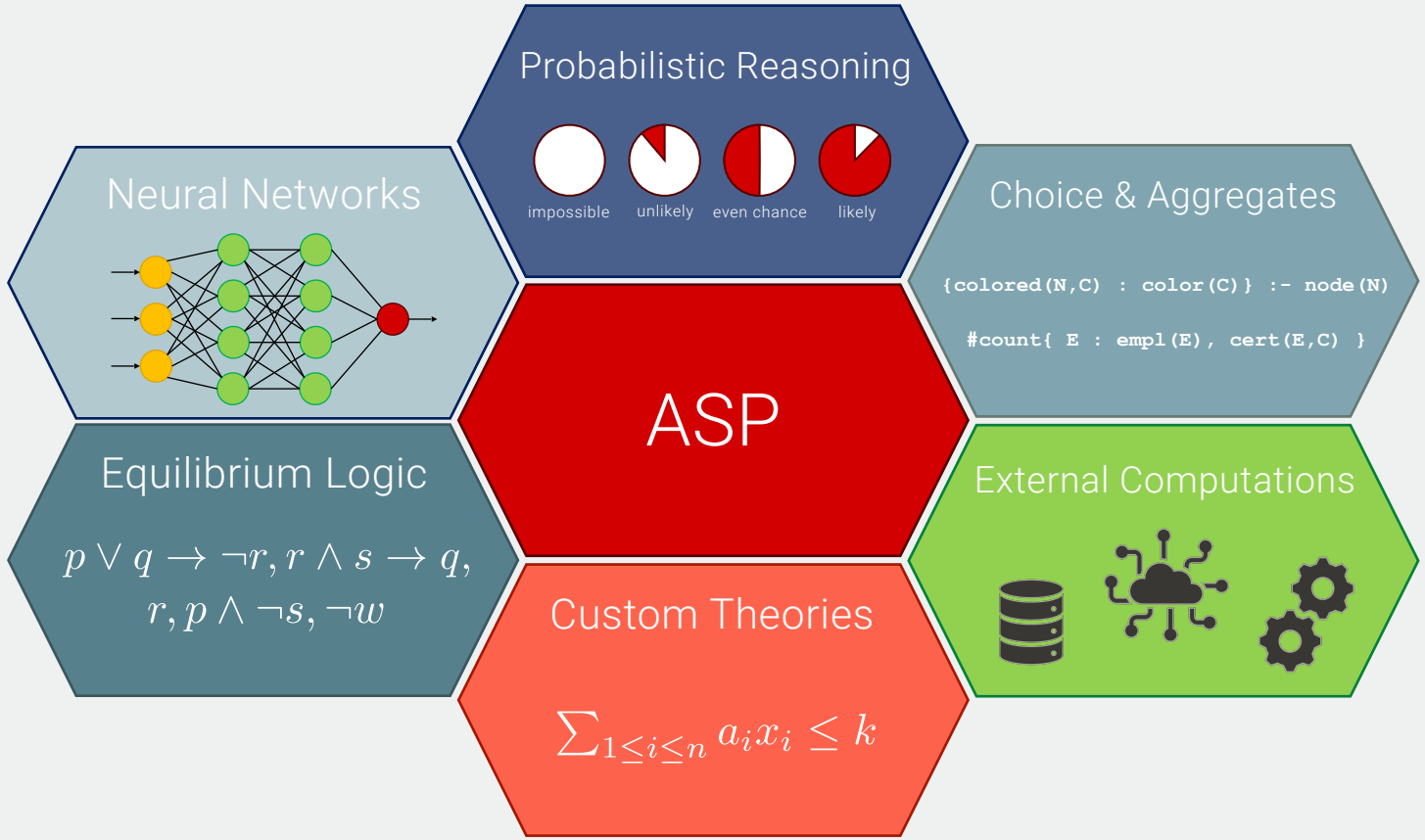
Extensions



- Choice rules & Aggregates
  - Syntactic sugar
  - but very useful
- HEX programs / clingo
- ASP modulo Theories
  - Linear Constraints (CASP)
  - Difference Logic
  - …
- Equilibrium Logic
- NeurASP

Answer-set Programming

Extensions



- Choice rules & Aggregates
  - Syntactic sugar
  - but very useful
- HEX programs / clingo
- ASP modulo Theories
  - Linear Constraints (CASP)
  - Difference Logic
  - …
- Equilibrium Logic
- NeurASP
- LPMLN

Answer-set Programming

Explanations

Answer-set Programming

Explanations

For consistent programs

Answer-set Programming

Explanations

For consistent programs

- Offline justification graphs / `xASP`

Answer-set Programming

Explanations

For consistent programs

- Offline justification graphs / `xASP`
- Causal explanations

Answer-set Programming

Explanations

For consistent programs

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses

Answer-set Programming

Explanations

For consistent programs

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`

Answer-set Programming

Explanations

For **consistent programs**

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`
- …

Answer-set Programming

Explanations

For **consistent programs**

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`
- …

Supported language features vary

Answer-set Programming

Explanations

For **consistent programs**

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`
- …

Supported language features vary

Some **common principles**

Answer-set Programming

Explanations

For **consistent programs**

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`
- …

Supported language features vary

Some **common principles**

Example:

$$warhawk(\mathsf{X}) \leftarrow republican(\mathsf{X}), not\ pacifist(\mathsf{X})$$
$$pacifist(\mathsf{X}) \leftarrow quaker(\mathsf{X}), not\ warhawk(\mathsf{X})$$
$$republic(\mathsf{nixon}) \leftarrow$$
$$quaker(\mathsf{nixon}) \leftarrow$$

$$I_1 = \{\ republican(\mathsf{nixon}), quaker(\mathsf{nixon}), warhawk(\mathsf{nixon})\ \}$$

Answer-set Programming

Explanations

For **consistent programs**

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`
- …

Supported language features vary

Some **common principles**

Example:

$$warhawk(\mathsf{X}) \leftarrow republican(\mathsf{X}), not\ pacifist(\mathsf{X})$$
$$pacifist(\mathsf{X}) \leftarrow quaker(\mathsf{X}), not\ warhawk(\mathsf{X})$$
$$republic(\mathsf{nixon}) \leftarrow$$
$$quaker(\mathsf{nixon}) \leftarrow$$

$$I_1 = \{\ republican(\mathsf{nixon}), quaker(\mathsf{nixon}), warhawk(\mathsf{nixon})\ \}$$

$$warhawk(\mathsf{nixon})$$

Answer-set Programming

Explanations

For **consistent programs**

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`
- …

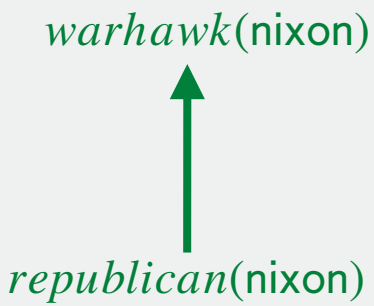
Supported language features vary

Some **common principles**

Example:

$$warhawk(\mathsf{X}) \leftarrow republican(\mathsf{X}),\ not\ pacifist(\mathsf{X})$$
$$pacifist(\mathsf{X}) \leftarrow quaker(\mathsf{X}),\ not\ warhawk(\mathsf{X})$$
$$republic(\mathsf{nixon}) \leftarrow$$
$$quaker(\mathsf{nixon}) \leftarrow$$

$$I_1 = \{\ republican(\mathsf{nixon}),\ quaker(\mathsf{nixon}),\ warhawk(\mathsf{nixon})\ \}$$



Answer-set Programming

Explanations

For **consistent programs**

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`
- …

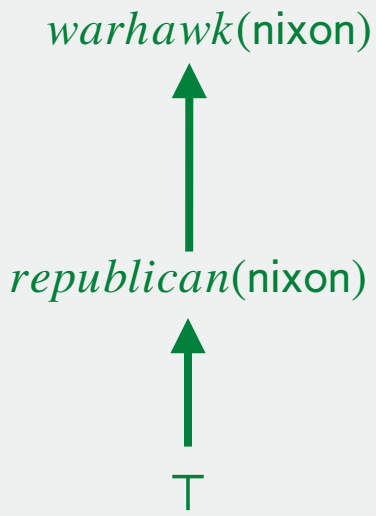
Supported language features vary

Some **common principles**

Example:

$$warhawk(\mathsf{X}) \leftarrow republican(\mathsf{X}),\ not\ pacifist(\mathsf{X})$$
$$pacifist(\mathsf{X}) \leftarrow quaker(\mathsf{X}),\ not\ warhawk(\mathsf{X})$$
$$republic(\mathsf{nixon}) \leftarrow$$
$$quaker(\mathsf{nixon}) \leftarrow$$

$$I_1 = \{\ republican(\mathsf{nixon}),\ quaker(\mathsf{nixon}),\ warhawk(\mathsf{nixon})\ \}$$



Answer-set Programming

Explanations

For **consistent programs**

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`
- …

Supported language features vary

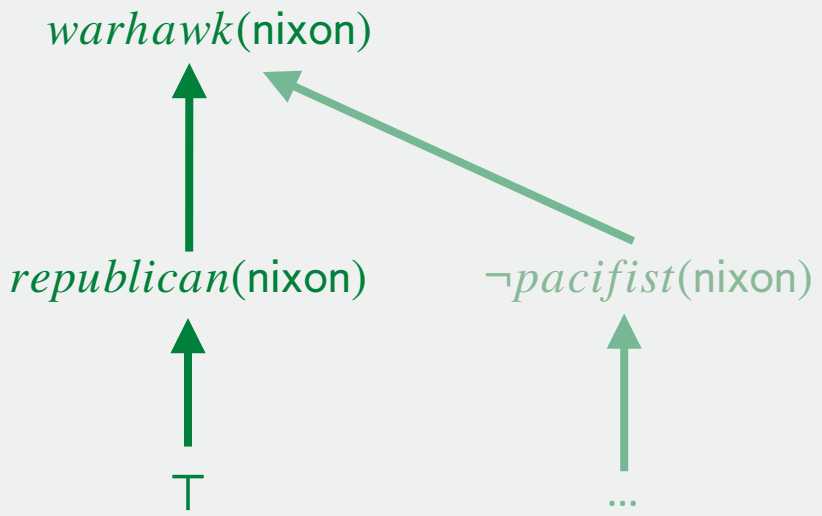
Some **common principles**

Differences in how **negation** is handled

Example:

$$warhawk(\mathrm{X}) \leftarrow republican(\mathrm{X}),\ not\ pacifist(\mathrm{X})$$
$$pacifist(\mathrm{X}) \leftarrow quaker(\mathrm{X}),\ not\ warhawk(\mathrm{X})$$
$$republic(\mathrm{nixon}) \leftarrow$$
$$quaker(\mathrm{nixon}) \leftarrow$$

$$I_1 = \{\ republican(\mathrm{nixon}),\ quaker(\mathrm{nixon}),\ warhawk(\mathrm{nixon})\ \}$$



Answer-set Programming

Explanations

For **consistent programs**

- Offline justification graphs / `xASP`
- Causal explanations
- Witnesses
- `xclingo`
- …

Supported language features vary

Some **common principles**

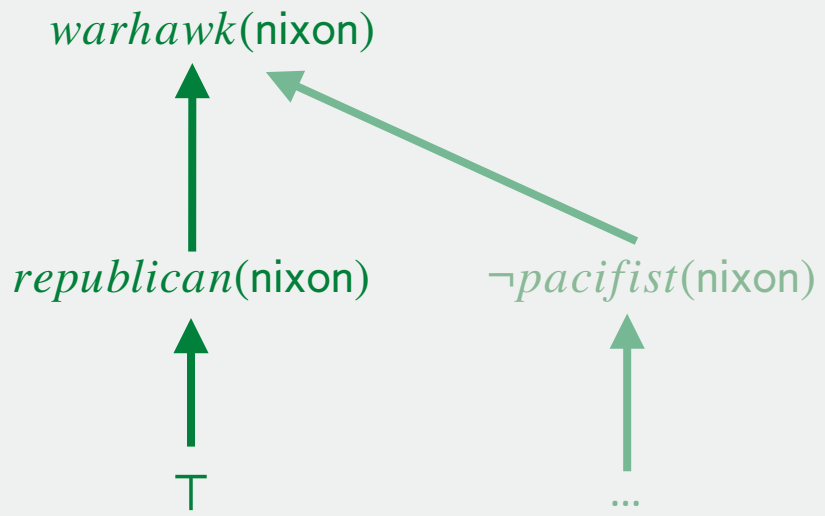
Differences in how **negation** is handled

Explanations are **not always satisfactory**

Example:

$$warhawk(\mathrm{X}) \leftarrow republican(\mathrm{X}),\ not\ pacifist(\mathrm{X})$$
$$pacifist(\mathrm{X}) \leftarrow quaker(\mathrm{X}),\ not\ warhawk(\mathrm{X})$$
$$republic(\mathrm{nixon}) \leftarrow$$
$$quaker(\mathrm{nixon}) \leftarrow$$

$$I_1 = \{\ republican(\mathrm{nixon}),\ quaker(\mathrm{nixon}),\ warhawk(\mathrm{nixon})\ \}$$



Answer-set Programming

Explanations

Answer-set Programming

Explanations

Explanations for **inconsistent programs** are mostly considered for **debugging**

Answer-set Programming

Explanations

Explanations for **inconsistent programs** are mostly considered for **debugging**

Approaches are based on:

Answer-set Programming

Explanations

Explanations for **inconsistent programs** are mostly considered for **debugging**

Approaches are based on:

- Giving reasons as to **why each interpretation is not an answer-set**

Answer-set Programming

Explanations

Explanations for **inconsistent programs** are mostly considered for **debugging**

Approaches are based on:

- Giving reasons as to **why each interpretation is not an answer-set**
- **Minimally inconsistent** sets of rules

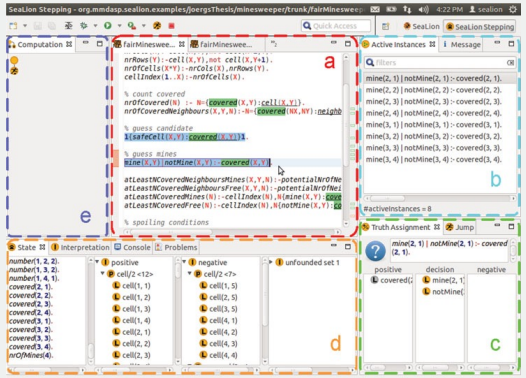
Answer-set Programming

Explanations

Explanations for **inconsistent programs** are mostly considered for **debugging**

Approaches are based on:

- Giving reasons as to **why each interpretation is not an answer-set**
- **Minimally inconsistent** sets of rules
- Interactive, **user-guided solving**



Answer-set Programming

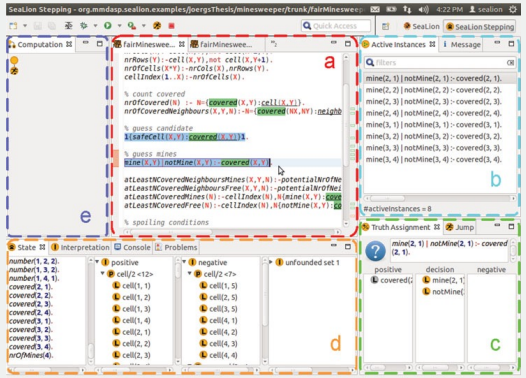
Explanations

Explanations for **inconsistent programs** are mostly considered for **debugging**

Approaches are based on:

- Giving reasons as to **why each interpretation is not an answer-set**
- **Minimally inconsistent** sets of rules
- Interactive, **user-guided solving**

Supported **language features** again vary



Answer-set Programming

Explanations

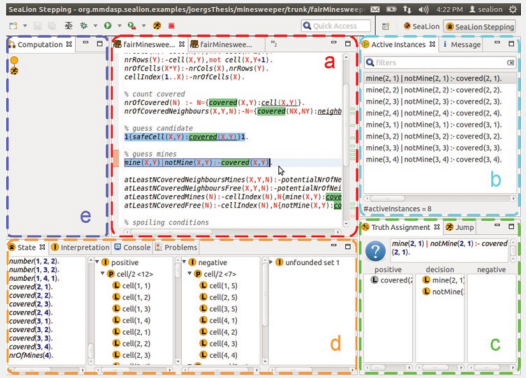
Explanations for **inconsistent programs** are mostly considered for **debugging**

Approaches are based on:

- Giving reasons as to **why each interpretation is not an answer-set**
- **Minimally inconsistent** sets of rules
- Interactive, **user-guided solving**

Supported **language features** again vary

The produced explanations are **very technical**!



Open Problems

Open Problems

- Most existing approaches lack support for language extensions

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

Example:

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

Example:

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

Example:

Why crow and not magpie?

Classifier

Crow

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

Example:

Classifier → Crow

Why crow and not magpie?

- Black beak

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

Example:

Classifier → Crow

Why crow and not magpie?

- Black beak
- Feathers

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

Example:

Why crow and not magpie?

- Black beak
- Feathers
- Size

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

Example:

Why crow and not magpie?

- ~~Black beak~~
- ~~Feathers~~
- ~~Size~~

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

Example:

Classifier → Crow

Why crow and not magpie?

- ~~Black beak~~
- ~~Feathers~~
- ~~Size~~
- Wing colour

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Open Problems

- Most existing approaches **lack support for language extensions**
- How can **variables**, **external computations**, **theories** or **neuro-symbolic extensions** be included?
- To **which detail** should those extensions be involved?
- **Nonmonotonicity** poses challenges
- Miller[1] argues that explanations should be **causal**, **interactive** and **contrastive**

Contrastive Question: Why P and not Q?

Example:

→ Classifier → Crow

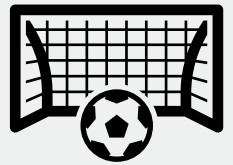
Why crow and not magpie?

- ~~Black beak~~
- ~~Feathers~~
- ~~Size~~
- Wing colour

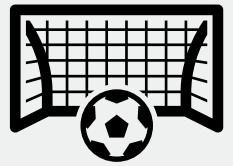
- How should we approach **contrastive explanation for ASP**?

1 Tim Miller (2019): Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence

Research Goals

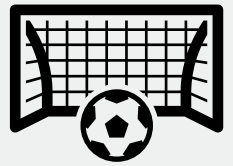


Research Goals



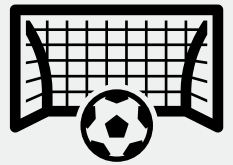
- Explaining ASP **extensions** and **advanced features**

Research Goals



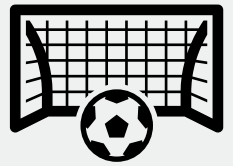
- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box

Research Goals



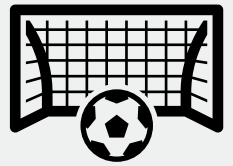
- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box
  - Contrastive explanation through counterfactual reasoning

Research Goals



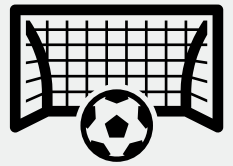
- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box
  - Contrastive explanation through counterfactual reasoning
- **Explaining** instead of debugging **inconsistency**

Research Goals



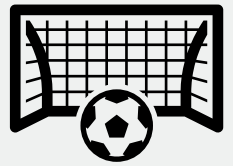
- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box
  - Contrastive explanation through counterfactual reasoning
- **Explaining** instead of debugging **inconsistency**
- Explainability in **Equilibrium Logic**

Research Goals



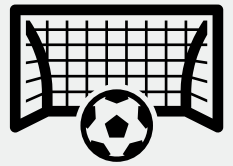
- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box
  - Contrastive explanation through counterfactual reasoning
- **Explaining** instead of debugging **inconsistency**
- Explainability in **Equilibrium Logic**
  - Proof systems

Research Goals



- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box
  - Contrastive explanation through counterfactual reasoning
- **Explaining** instead of debugging **inconsistency**
- Explainability in **Equilibrium Logic**
  - Proof systems
- Towards **practical algorithms**

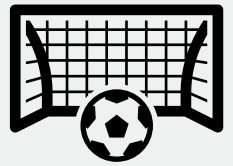
Research Goals



- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box
  - Contrastive explanation through counterfactual reasoning
- **Explaining** instead of debugging **inconsistency**
- Explainability in **Equilibrium Logic**
  - Proof systems
- Towards **practical algorithms**

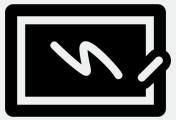
Methodology

Research Goals



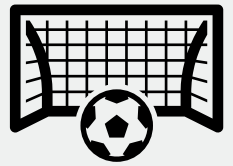
- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box
  - Contrastive explanation through counterfactual reasoning
- **Explaining** instead of debugging **inconsistency**
- Explainability in **Equilibrium Logic**
  - Proof systems
- Towards **practical algorithms**

Methodology



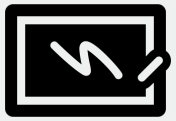
Formal concepts

Research Goals



- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box
  - Contrastive explanation through counterfactual reasoning
- **Explaining** instead of debugging **inconsistency**
- Explainability in **Equilibrium Logic**
  - Proof systems
- Towards **practical algorithms**

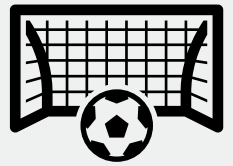
Methodology



Formal concepts

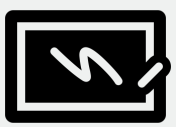
Analysis

Research Goals



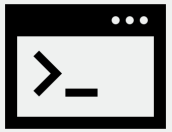
- Explaining ASP **extensions** and **advanced features**
  - White-box, grey-box, black-box
  - Contrastive explanation through counterfactual reasoning
- **Explaining** instead of debugging **inconsistency**
- Explainability in **Equilibrium Logic**
  - Proof systems
- Towards **practical algorithms**

Methodology



Formal concepts

Analysis



Prototypes

Research Status

Publications

Explaining Answer-Set Programs with Abstract Constraint Atoms

Thomas Eiter and Tobias Geibinger

32rd International Joint Conference on Artificial Intelligence (IJCAI 2023)

Contributions: Formal notions of justification for ASP with choice and aggregates, Complexity Analysis

A Logic-based Approach to Contrastive Explainability for Neurosymbolic Visual Question Answering

Thomas Eiter, Tobias Geibinger, Nelson Higuera Ruiz and Johannes Oetsch

32rd International Joint Conference on Artificial Intelligence (IJCAI 2023)

Contributions: Contrastive explanation approach for the Visual Question Answering domain

Contrastive Explanations for Answer-Set Programs

Thomas Eiter, Tobias Geibinger and Johannes Oetsch

18th Edition of the European Conference on Logics in Artificial Intelligence (JELIA 2023)

Contributions: Problem independent formulation of contrastive explanation for ASP including study of complexity

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain $C \subseteq 2^D$

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain: D — $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

```
#sum{2 : a, 1 : b, 1 : c} > 1
```

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

$$\texttt{\#sum\{2 : a, 1 : b, 1 : c\} > 1}$$

$$\updownarrow$$

$$A = \langle \{a, b, c\}, \{\{a\}, \{b, c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\} \rangle$$

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

$$I = \{a, b, c\}$$

$$\texttt{\#sum\{2 : a, 1 : b, 1 : c\}} > 1$$

$$\updownarrow$$

$$A = \langle \{a, b, c\}, \{\{a\}, \{b, c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\} \rangle$$

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

$$\texttt{\#sum\{2 : a, 1 : b, 1 : c\} > 1}$$

$$A = \langle \{a, b, c\}, \{\{a\}, \{b, c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\} \rangle$$

$I = \{a, b, c\}$ $I \models A$

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain D; $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

$$\texttt{\#sum\{2 : a, 1 : b, 1 : c\} > 1}$$

$$\Updownarrow$$

$$A = \langle \{a, b, c\}, \{\{a\}, \{b, c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\} \rangle$$

$I = \{a, b, c\}$ $\quad I \models A$

Model-based justifications:

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain D; $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

$$\texttt{\#sum\{2 : a, 1 : b, 1 : c\} > 1}$$

$$\Updownarrow$$

$$A = \langle \{a, b, c\}, \{\{a\}, \{b, c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\} \rangle$$

$I = \{a, b, c\}$ $\quad I \models A$

Model-based justifications:

$\langle \{a\}, \emptyset \rangle$

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain D; $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

$$\texttt{\#sum\{2 : a, 1 : b, 1 : c\} > 1}$$

$$\Updownarrow$$

$$A = \langle \{a, b, c\}, \{\{a\}, \{b, c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\} \rangle$$

$I = \{a, b, c\}$ $\quad I \models A$

Model-based justifications:

$\langle \{a\}, \emptyset \rangle$ $\quad \langle \{b, c\}, \emptyset \rangle$

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain: D — $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

$$\texttt{\#sum\{2 : a, 1 : b, 1 : c\} > 1}$$

$$A = \langle \{a, b, c\}, \{\{a\}, \{b, c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\} \rangle$$

$I = \{a, b, c\}$ $\quad I \models A$

Model-based justifications:

$\langle \{a\}, \emptyset \rangle$ $\quad \langle \{b, c\}, \emptyset \rangle$

minimal partial models

Research Status

Explaining Answer-Set Programs with Abstract Constraint Atoms

We introduced **justifications** for ASP programs with **Abstract Constraint Atoms**:

$$A = \langle D, C \rangle$$

Domain: D — $C \subseteq 2^D$

An interpretation I satisfies A whenever $I \cap D \in C$

Choice rules and **aggregates** are captured by abstract constraint atoms

Example:

$$\texttt{\#sum\{2 : a, 1 : b, 1 : c\} > 1}$$

$$\Updownarrow$$

$$A = \langle \{a, b, c\}, \{\{a\}, \{b, c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\} \rangle$$

$I = \{a, b, c\}$ $\quad I \models A$

Model-based justifications:

$\langle \{a\}, \emptyset \rangle$ $\quad \langle \{b, c\}, \emptyset \rangle$

minimal partial models

We also define **rule-based justifications** which take the application of rules into account

Research Status

Publications

Explaining Answer-Set Programs with Abstract Constraint Atoms

Thomas Eiter and Tobias Geibinger

32rd International Joint Conference on Artificial Intelligence (IJCAI 2023)

Contributions: Formal notions of justification for ASP with choice and aggregates, Complexity Analysis

A Logic-based Approach to Contrastive Explainability for Neurosymbolic Visual Question Answering

Thomas Eiter, Tobias Geibinger, Nelson Higuera Ruiz and Johannes Oetsch

32rd International Joint Conference on Artificial Intelligence (IJCAI 2023)

Contributions: Contrastive explanation approach for the Visual Question Answering domain

Contrastive Explanations for Answer-Set Programs

Thomas Eiter, Tobias Geibinger and Johannes Oetsch

18th Edition of the European Conference on Logics in Artificial Intelligence (JELIA 2023)

Contributions: Problem independent formulation of contrastive explanation for ASP including study of complexity

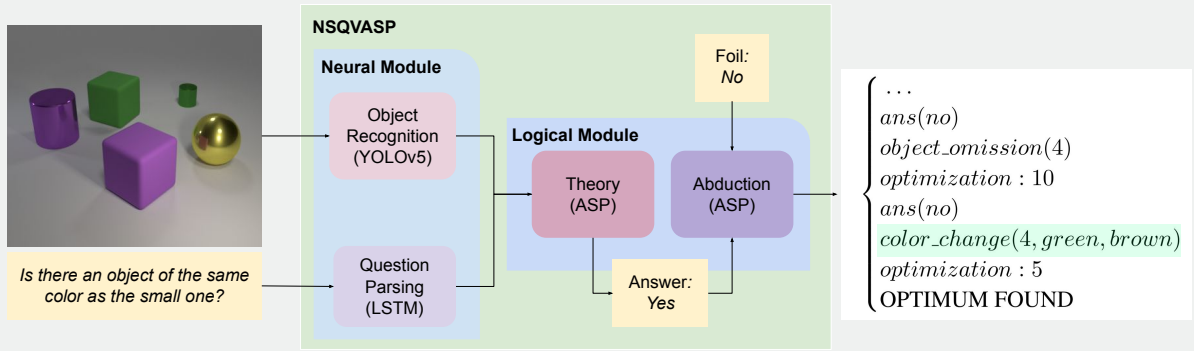
Research Status

A Logic-based Approach to Contrastive Explainability for Neurosymbolic VQA

Research Status

A Logic-based Approach to Contrastive Explainability for Neurosymbolic VQA

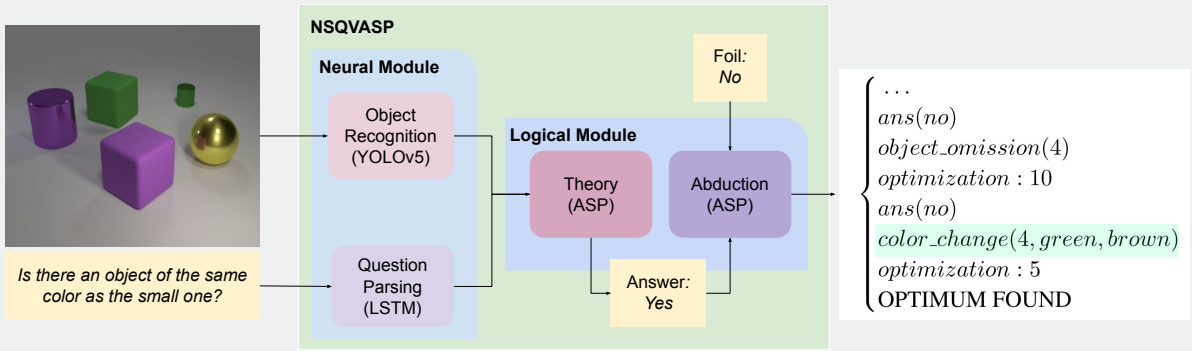
VQA Framework using ASP



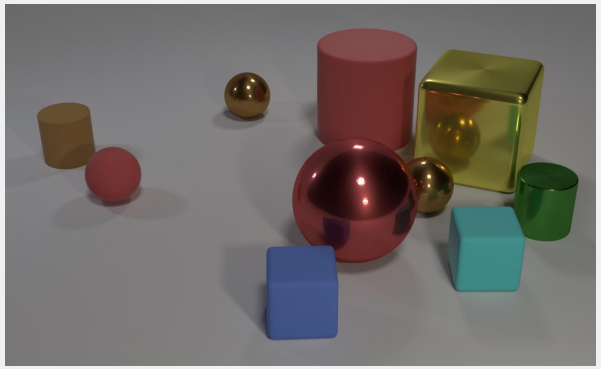
Research Status

A Logic-based Approach to Contrastive Explainability for Neurosymbolic VQA

VQA Framework using ASP



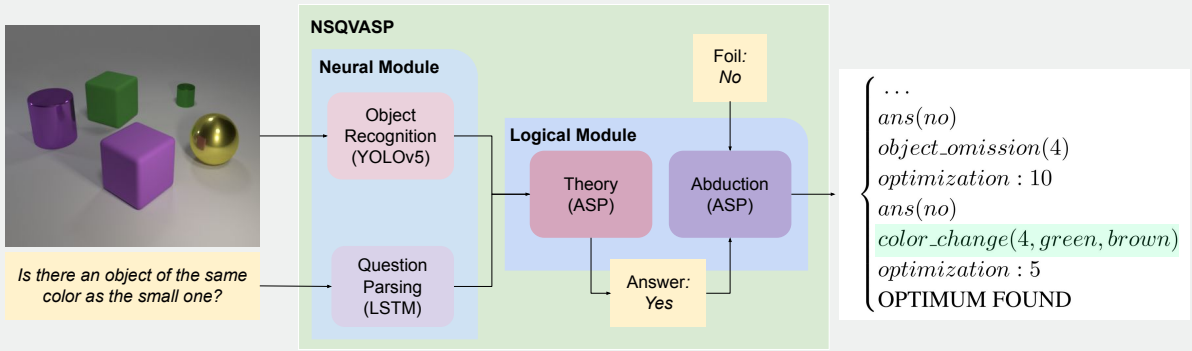
Example:



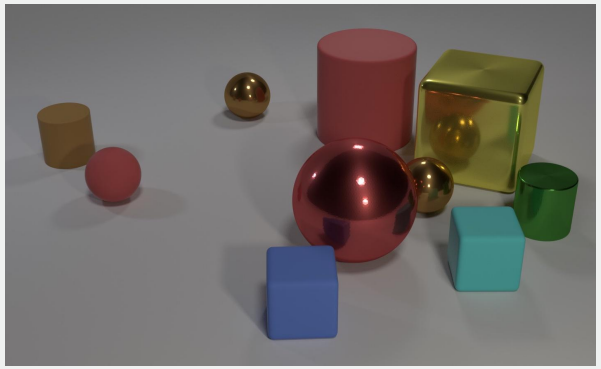
Research Status

A Logic-based Approach to Contrastive Explainability for Neurosymbolic VQA

VQA Framework using ASP



Example:

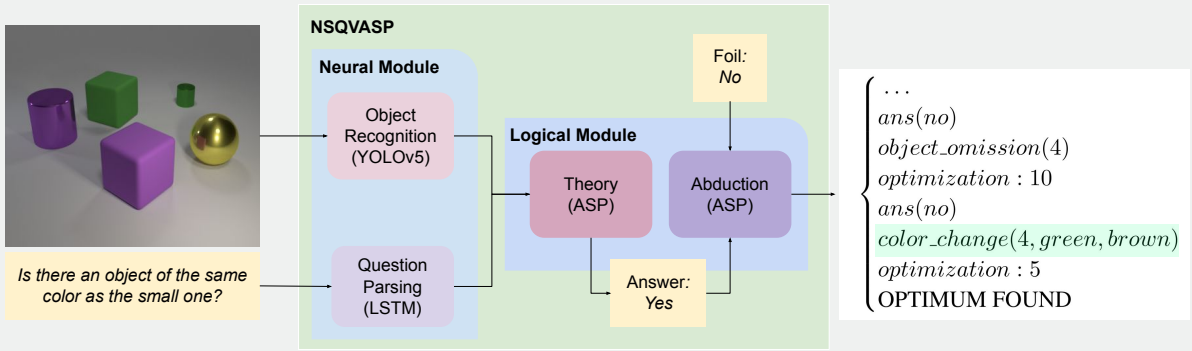


What size is the cylinder that is left of the brown metal thing that is left of the big sphere?

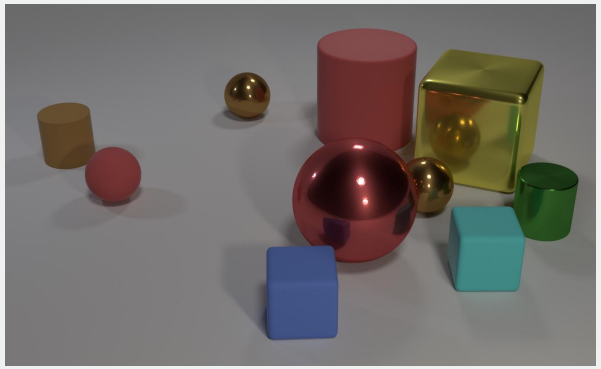
Research Status

A Logic-based Approach to Contrastive Explainability for Neurosymbolic VQA

VQA Framework using ASP



Example:



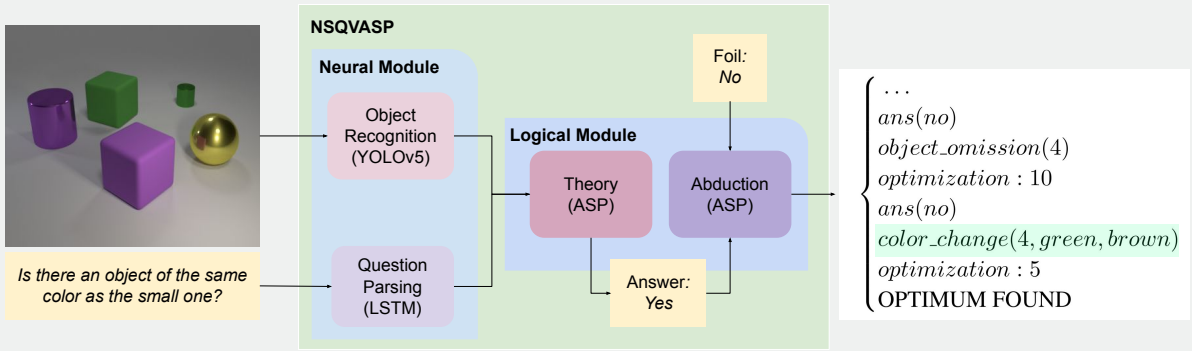
What size is the cylinder that is left of the brown metal thing that is left of the big sphere?

Answer: small

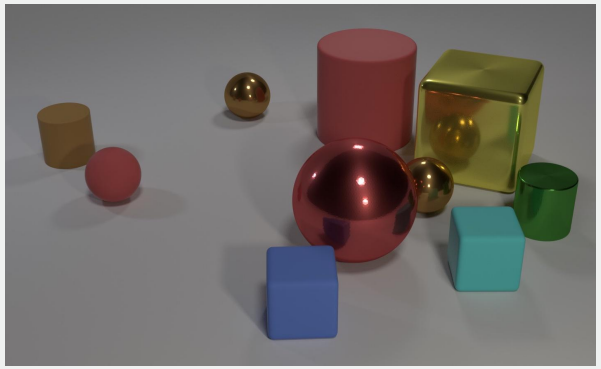
Research Status

A Logic-based Approach to Contrastive Explainability for Neurosymbolic VQA

VQA Framework using ASP



Example:



What size is the cylinder that is left of the brown metal thing that is left of the big sphere?

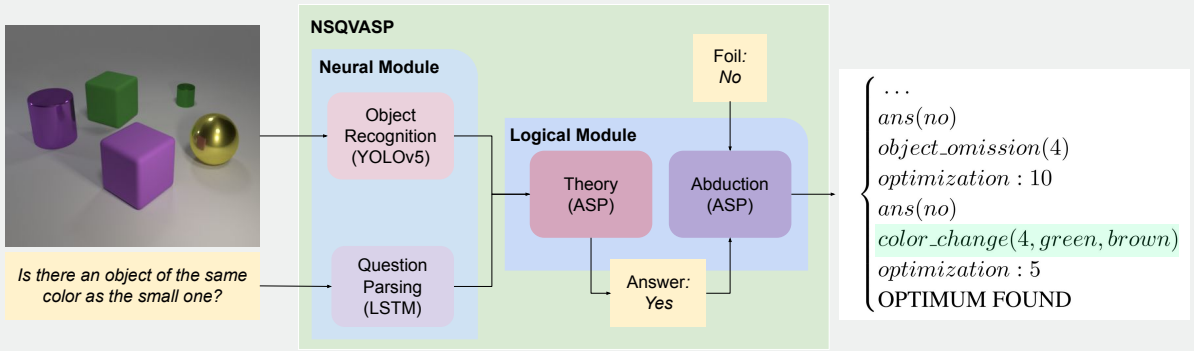
Answer: small

Why **small** and not **large**?

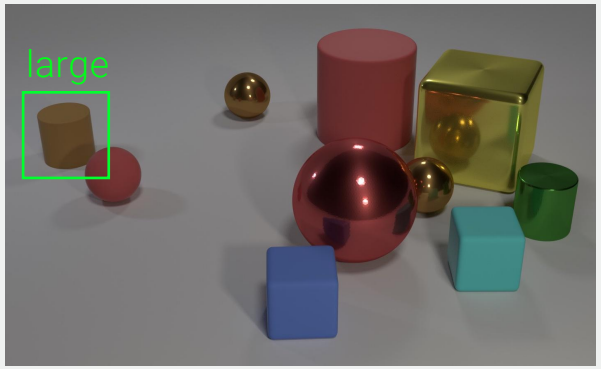
Research Status

A Logic-based Approach to Contrastive Explainability for Neurosymbolic VQA

VQA Framework using ASP



Example:

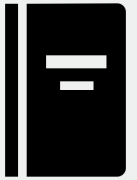


What size is the cylinder that is left of the brown metal thing that is left of the big sphere?

Answer: small

Why **small** and not **large**?

Research Status



Publications

Explaining Answer-Set Programs with Abstract Constraint Atoms

Thomas Eiter and Tobias Geibinger

32rd International Joint Conference on Artificial Intelligence (IJCAI 2023)

Contributions: Formal notions of justification for ASP with choice and aggregates, Complexity Analysis

A Logic-based Approach to Contrastive Explainability for Neurosymbolic Visual Question Answering

Thomas Eiter, Tobias Geibinger, Nelson Higuera Ruiz and Johannes Oetsch

32rd International Joint Conference on Artificial Intelligence (IJCAI 2023)

Contributions: Contrastive explanation approach for the Visual Question Answering domain

Contrastive Explanations for Answer-Set Programs

Thomas Eiter, Tobias Geibinger and Johannes Oetsch

18th Edition of the European Conference on Logics in Artificial Intelligence (JELIA 2023)

Contributions: Problem independent formulation of contrastive explanation for ASP including study of complexity

Research Status

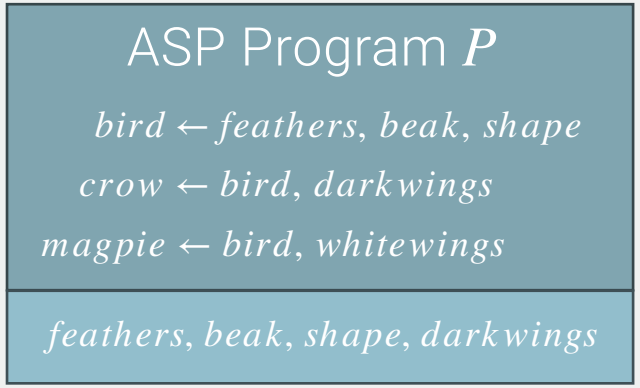
Contrastive Explanations for Answer-Set Programs

We consider the following setting:

Research Status

Contrastive Explanations for Answer-Set Programs

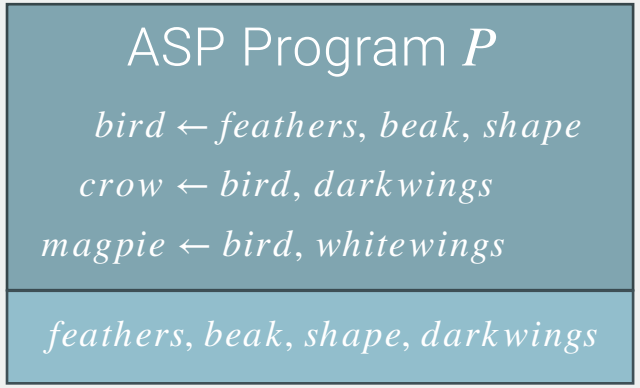
We consider the following setting:

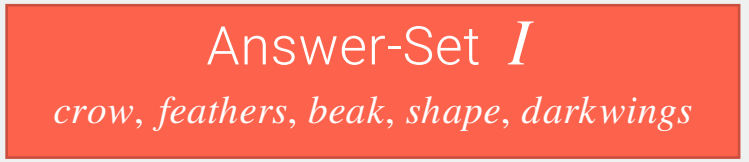


Research Status

Contrastive Explanations for Answer-Set Programs

We consider the following setting:

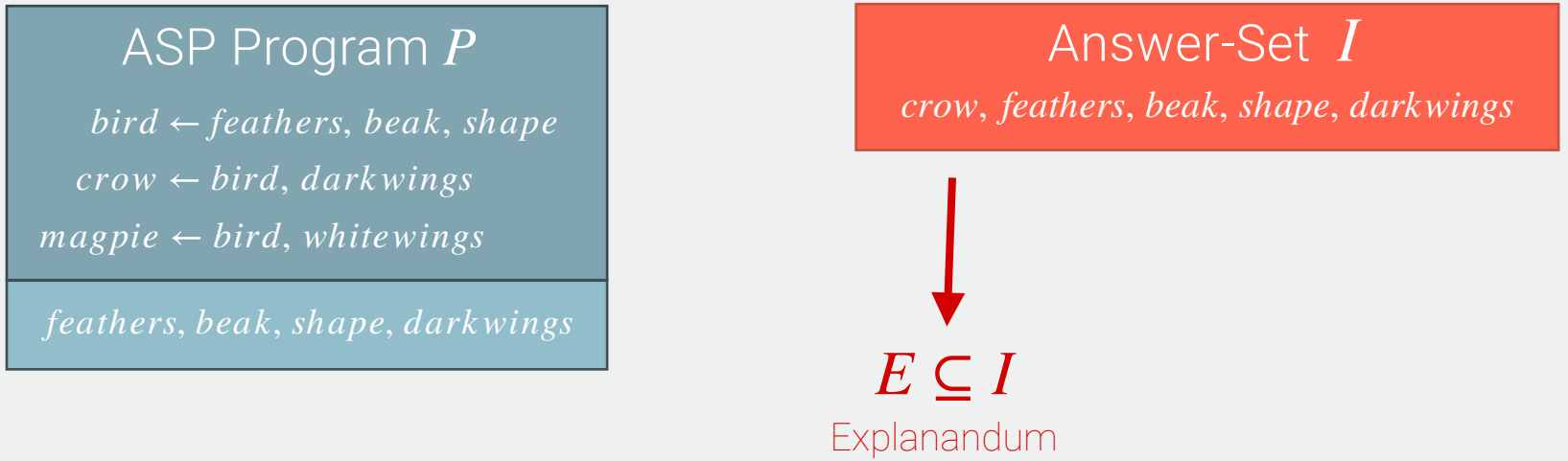




Research Status

Contrastive Explanations for Answer-Set Programs

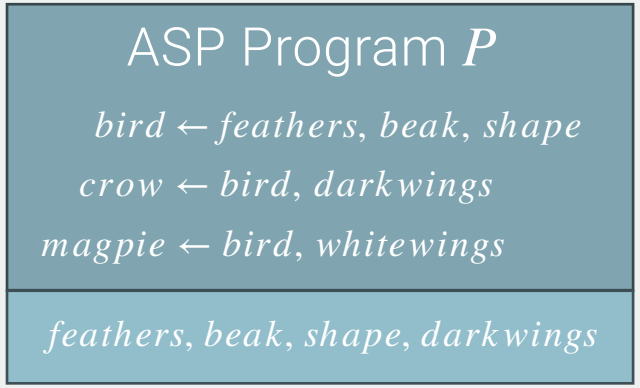
We consider the following setting:

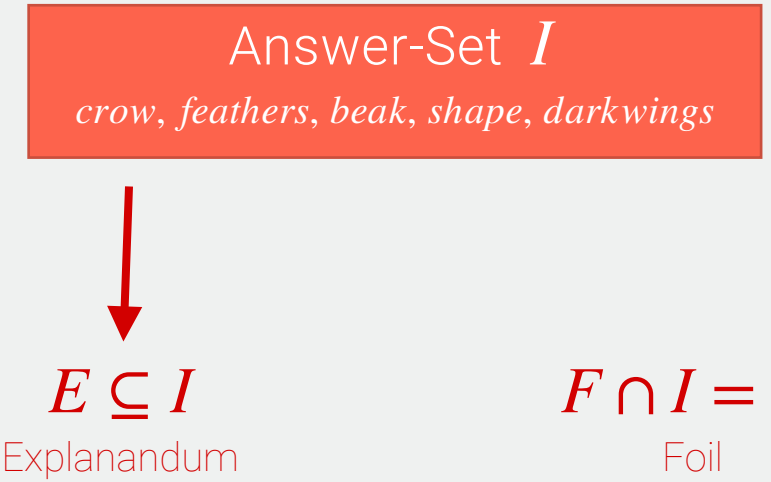


Research Status

Contrastive Explanations for Answer-Set Programs

We consider the following setting:

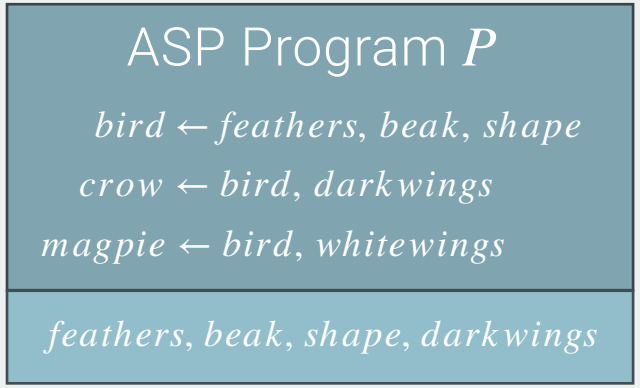


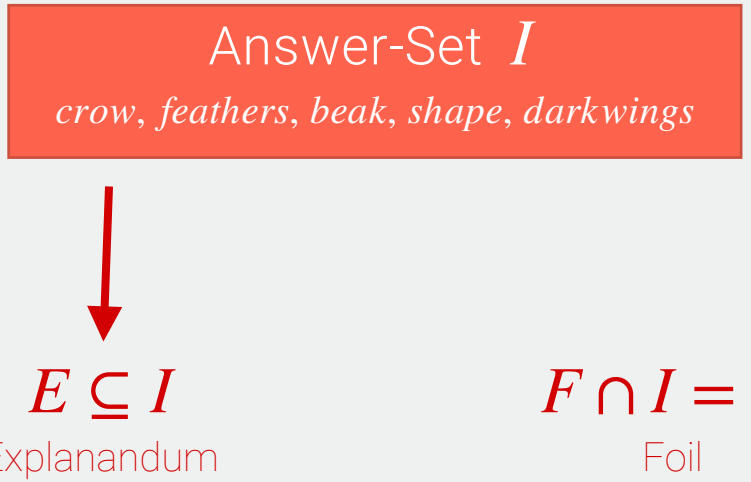


Research Status

Contrastive Explanations for Answer-Set Programs

We consider the following setting:



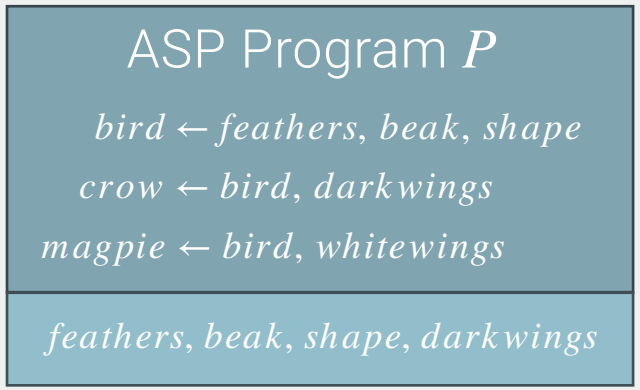


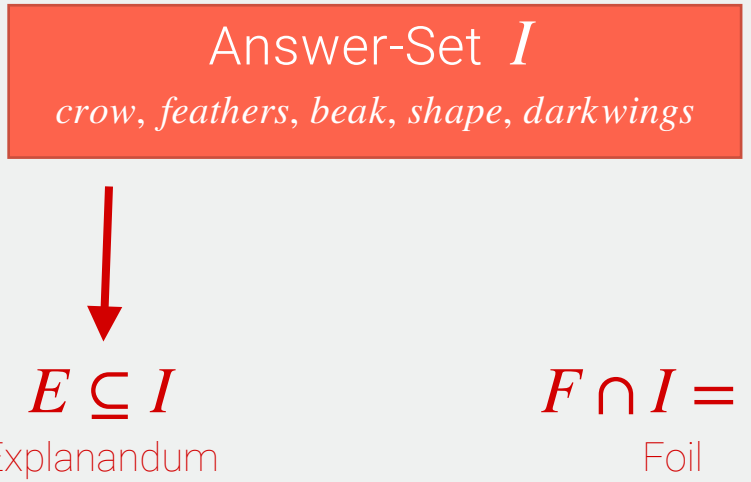
Why E rather than F?

Research Status

Contrastive Explanations for Answer-Set Programs

We consider the following setting:





Why E rather than F?

We want to find a program P' with $I' \in AS(P')$ such that $F \subseteq I'$ and $E \nsubseteq I'$

Next Steps

Next Steps

Continue investigation of **contrastive explanation**

Next Steps

Continue investigation of **contrastive explanation**

- Sharpen definitions and **theoretical foundation**
- **Encodings** and/or **algorithms**

Next Steps

Continue investigation of **contrastive explanation**

- Sharpen definitions and **theoretical foundation**
- **Encodings** and/or **algorithms**

Study **how extensions can be incorporated** in the explanations

- white-box, black-box, grey-box

Next Steps

Continue investigation of **contrastive explanation**

- Sharpen definitions and **theoretical foundation**
- **Encodings** and/or **algorithms**

Study **how extensions can be incorporated** in the explanations

- white-box, black-box, grey-box

Development of a prototype using **contrastive** and **non-contrastive explanations** interactively