# Datalog Fact Explanation Using Group-SAT Solver

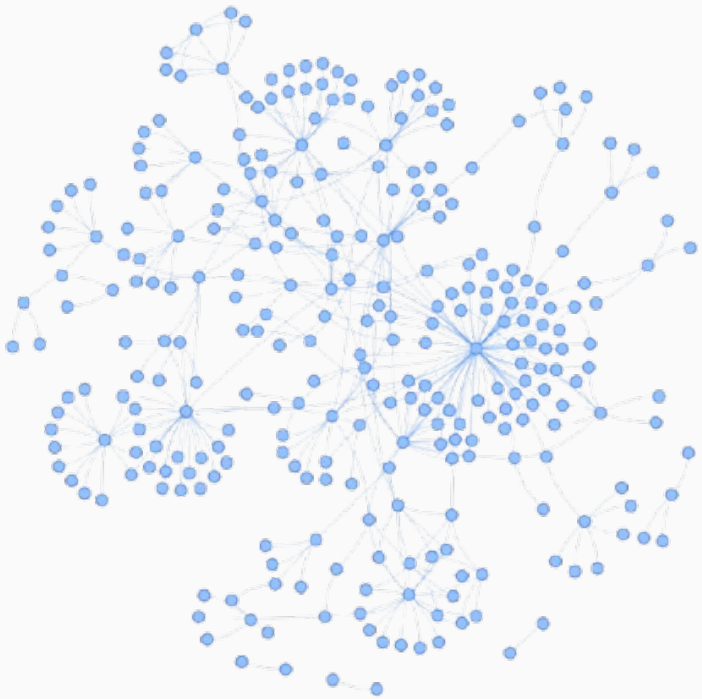
Akira Charoensit [1] David Carral [1] Pierre Bisquert [1, 2]
Lucas Rouquette [1] Federico Ulliana [1]

Sesame Seminar - 24/03/2025

[1]Inria, LIRMM, Univ Montpellier, CNRS, France

[2]IATE, Univ Montpellier, INRAE, Institut Agro, Montpellier, France

# Motivation: making sense of data





Radioactive ?

Bananas are radioactive!

- But why? ⇒ Because they contain Potassium-40

# But wait, what's an explanation for us?

- In a nutshell, "something" that justifies an entailment
  - In our context, facts and/or rules
- Why-provenance is somewhat classical
  - But other notions of explanation can be considered
- A note: we are not considering the user here
  - Nor explanation of missing entailment, etc.

# Outline

1. A bit of formalities
2. Computing explanations...
3. ...Is not easy
4. Let's be practical
5. Conclusion

# A bit of formalities

# Formal setting

- A **Datalog knowledge base**:
  - A set $\mathcal{F}$ of ground atoms
  - A set $\mathcal{R}$ of Datalog rules:
    $\forall \bar{x}.Body[\bar{x}] \rightarrow Head[\bar{z}]$ s.t. $z \subseteq x$

$$contains(banana, k40)$$
$$radioactive(k40)$$

---

$$contains(x, y) \land radioactive(y) \rightarrow radioactive(x)$$

- (Explanation) **Query**
  - A query $\varphi$ is a fact (ground atom)

$$radioactive(banana) \ ?$$

- **Entailment**
  - $\mathcal{F}, \mathcal{R} \models \varphi$

# So what's an explanation in Datalog?

- Different definitions of explanations:

**The content**

- Facts
- Rules
- Both

**The link**

- Proof trees
- Entailment

- As always, they will give different sets of explanations...

# So what's an explanation in Datalog?

- Different definitions of explanations:

**The content**

- Facts
- Rules
- Both

*Why-provenance*

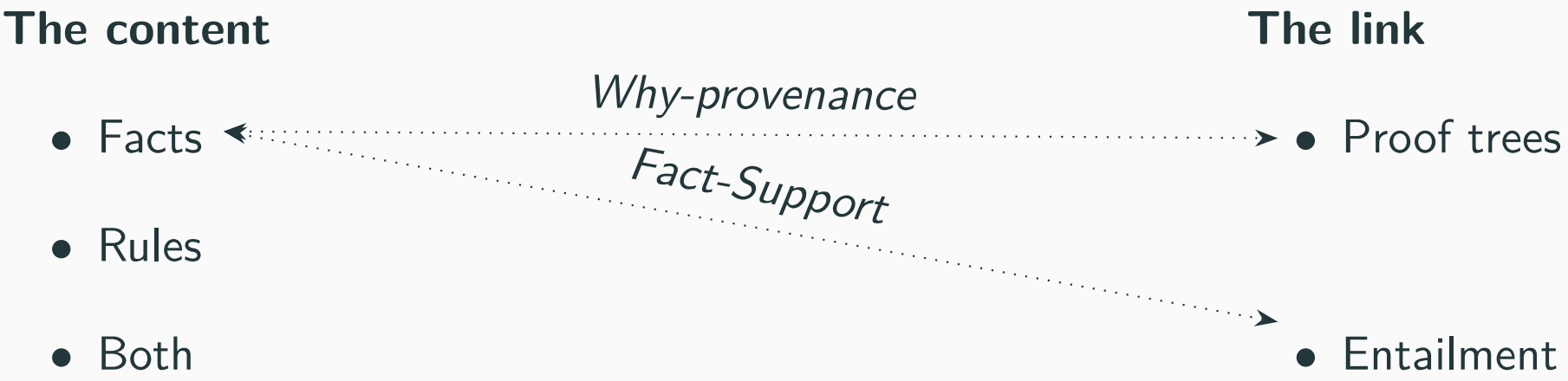
**The link**

- Proof trees
- Entailment

- As always, they will give different sets of explanations...

# So what's an explanation in Datalog?

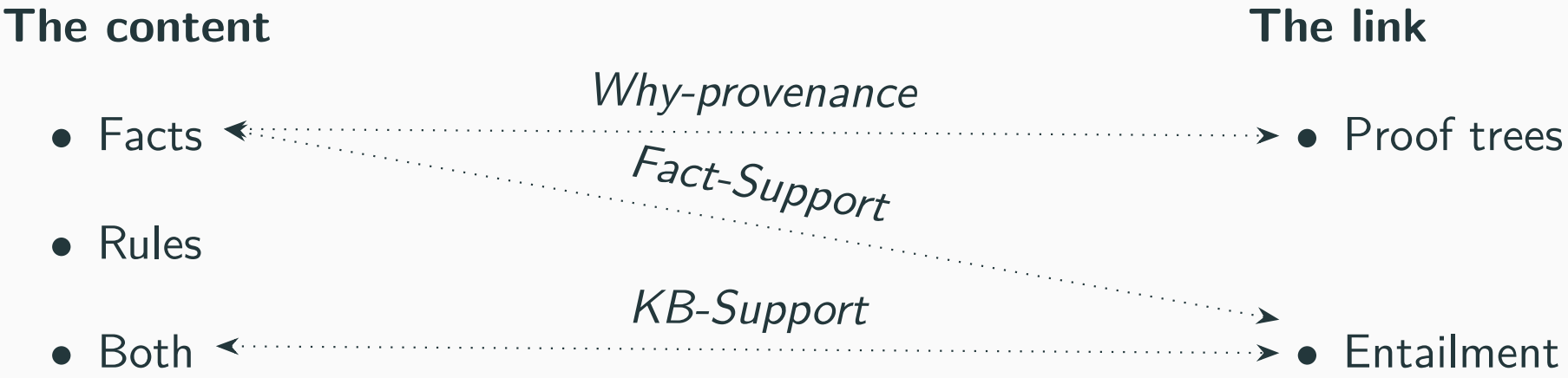
- Different definitions of explanations:




- As always, they will give different sets of explanations...

# So what's an explanation in Datalog?

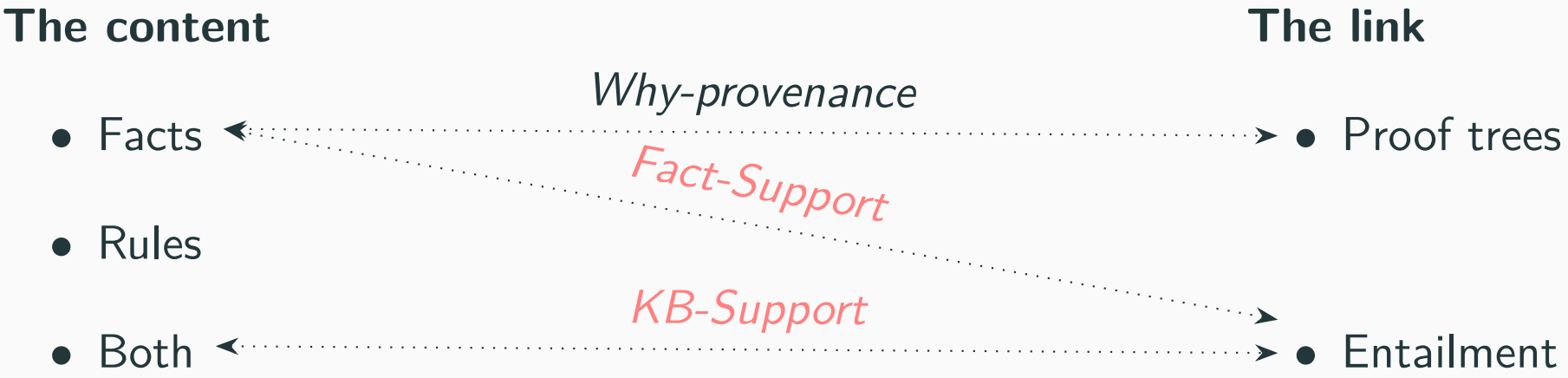
- Different definitions of explanations:




- As always, they will give different sets of explanations...

# So what's an explanation in Datalog?

- Different definitions of explanations:




- As always, they will give different sets of explanations...

# So what's an explanation in Datalog?

**Fact-Support**

- Minimal subsets $\mathcal{F}'$ of $\mathcal{F}$ s.t. $\mathcal{F}', \mathcal{R} \models \varphi$

**KB-Support**

- Minimal subsets $\mathcal{KB}'$ of $\mathcal{KB} = (\mathcal{F}, \mathcal{R})$ s.t. $\mathcal{KB}' \models \varphi$

# So what's an explanation in Datalog?

## Fact-Support

- Minimal subsets $\mathcal{F}'$ of $\mathcal{F}$ s.t. $\mathcal{F}', \mathcal{R} \models \varphi$

$boss(alice, alice)$

---

$r_1 : \forall XY.boss(X, Y) \rightarrow manager(X)$
$r_2 : \forall X.boss(X, X) \rightarrow ceo(X)$
$r_3 : \forall X.ceo(X) \rightarrow manager(X)$

## KB-Support

- Minimal subsets $\mathcal{KB}'$ of $\mathcal{KB} = (\mathcal{F}, \mathcal{R})$ s.t. $\mathcal{KB}' \models \varphi$

$manager(alice)$ ?

# So what's an explanation in Datalog?

## Fact-Support

- Minimal subsets $\mathcal{F}'$ of $\mathcal{F}$ s.t. $\mathcal{F}', \mathcal{R} \models \varphi$

$boss(alice, alice)$

---

$r_1 : \forall XY.boss(X, Y) \rightarrow manager(X)$
$r_2 : \forall X.boss(X, X) \rightarrow ceo(X)$
$r_3 : \forall X.ceo(X) \rightarrow manager(X)$

- $\{boss(alice, alice)\}$

## KB-Support

- Minimal subsets $\mathcal{KB}'$ of $\mathcal{KB} = (\mathcal{F}, \mathcal{R})$ s.t. $\mathcal{KB}' \models \varphi$

$manager(alice)$ ?

- $(\{boss(alice, alice)\}, \{r_1\})$
- $(\{boss(alice, alice)\}, \{r_2, r_3\})$

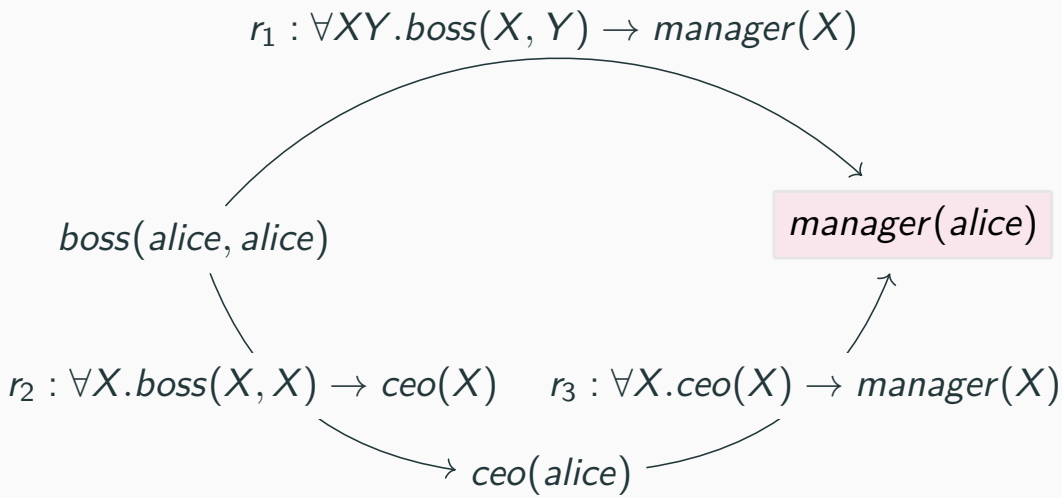
# So what's an explanation in Datalog?

## Fact-Support

- Minimal subsets $\mathcal{F}'$ of $\mathcal{F}$ s.t. $\mathcal{F}', \mathcal{R} \models \varphi$

## KB-Support

- Minimal subsets $\mathcal{KB}'$ of $\mathcal{KB} = (\mathcal{F}, \mathcal{R})$ s.t. $\mathcal{KB}' \models \varphi$




- $\{boss(alice, alice)\}$

- $(\{boss(alice, alice)\}, \{r_1\})$
- $(\{boss(alice, alice)\}, \{r_2, r_3\})$

# Computing explanation

# And how do we compute that?

- **Hint**: Minimal Inconsistent Set (MUS) and Explanations are very similar!

$$p \wedge (p \rightarrow q) \wedge (\neg q) \wedge s \wedge (q \vee \neg s) \wedge t$$

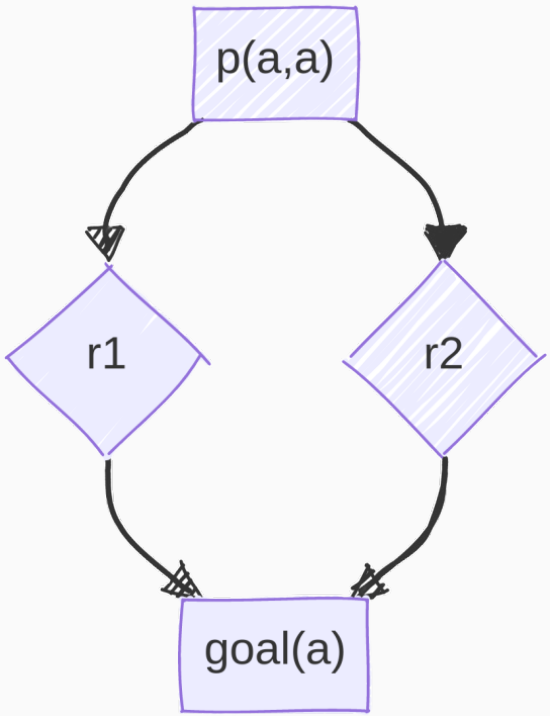
- Two explanations for $q$
  - $p \wedge (p \rightarrow q)$
  - $s \wedge (q \vee \neg s)$

# From Datalog to Propositional! Fact-Support explanation

$p(a, a)$

---

$r_1 : p(X, X) \rightarrow goal(X)$
$r_2 : p(X, Y) \rightarrow goal(X)$



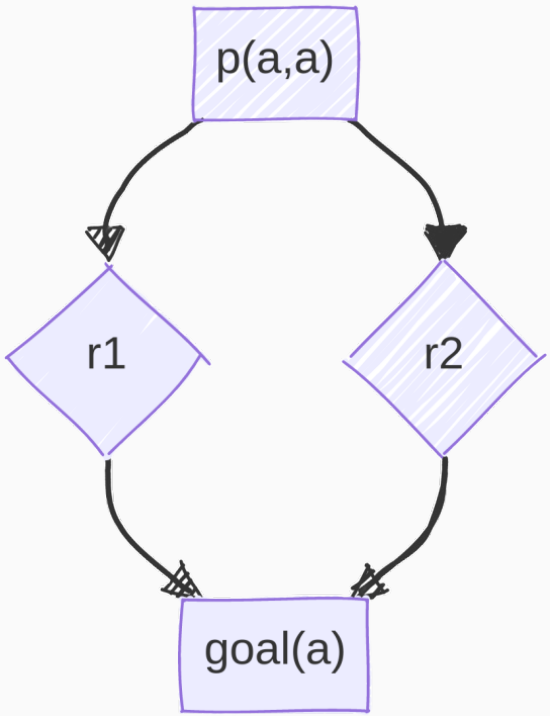
# From Datalog to Propositional! Fact-Support explanation

$p(a, a)$

$r_1 : p(X, X) \rightarrow goal(X)$
$r_2 : p(X, Y) \rightarrow goal(X)$

$\longrightarrow p(a, a) \wedge (p(a, a) \rightarrow goal(a)) \wedge (\neg goal(a))$

*Grounding*



# From Datalog to Propositional! Fact-Support explanation

$p(a, a)$

$r_1 : p(X, X) \rightarrow goal(X)$
$r_2 : p(X, Y) \rightarrow goal(X)$

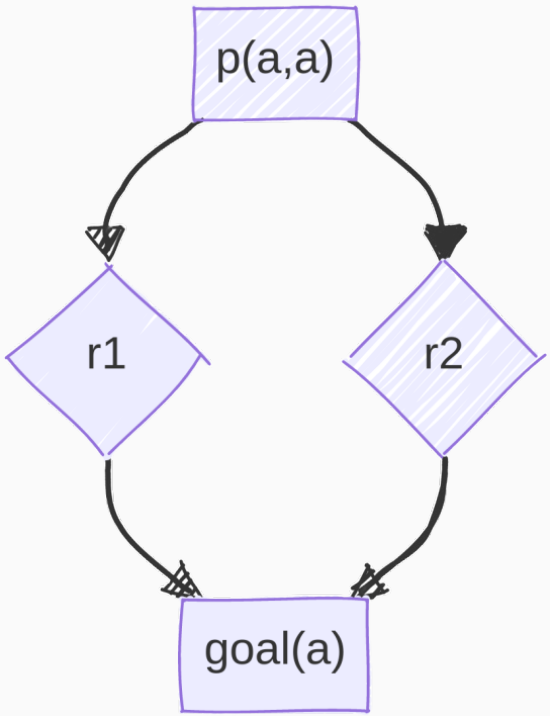
$\longrightarrow p(a, a) \wedge (p(a, a) \rightarrow goal(a)) \wedge (\neg goal(a))$

*Grounding*

*MUS solving*

1 MUS

$\{p(a, a), (p(a, a) \rightarrow goal(a)), (\neg goal(a))\}$



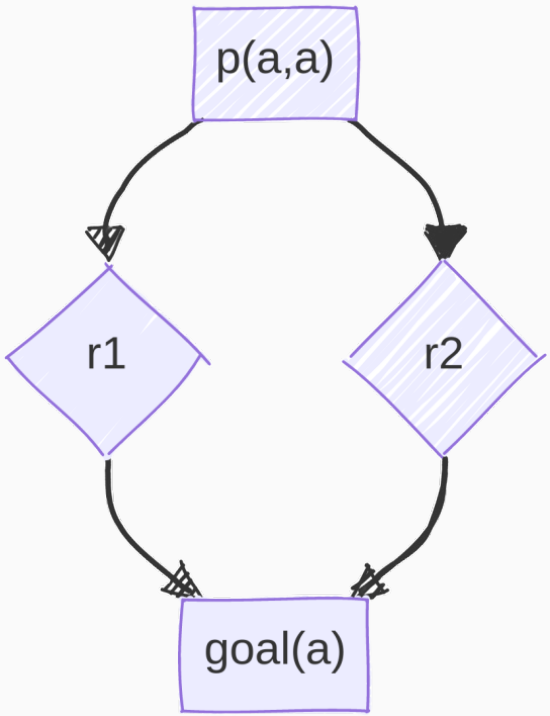
# From Datalog to Propositional! Fact-Support explanation

$p(a, a)$

$r_1 : p(X, X) \rightarrow goal(X)$
$r_2 : p(X, Y) \rightarrow goal(X)$

$\longrightarrow p(a, a) \wedge (p(a, a) \rightarrow goal(a)) \wedge (\neg goal(a))$

*Grounding*




*MUS solving*

1 MUS

$\{p(a, a), (p(a, a) \rightarrow goal(a)), (\neg goal(a))\}$

*Translation*

1 explanation

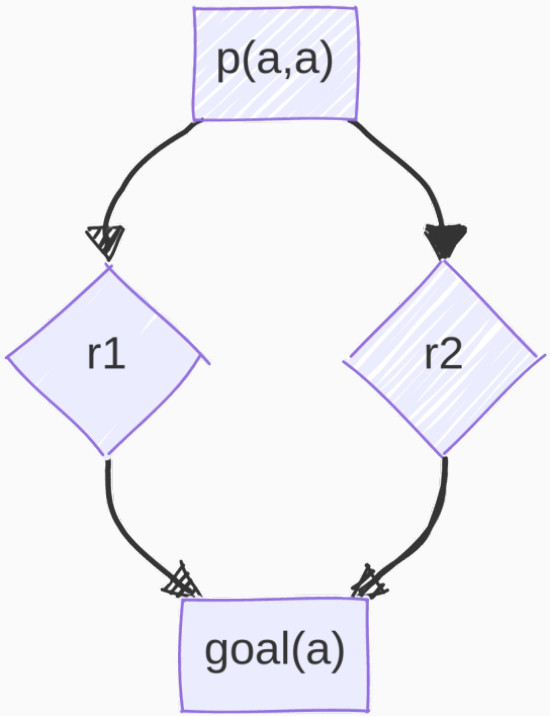
$\{p(a, a)\}$

# From Datalog to Propositional! KB-Support explanation

$p(a, a)$

---

$r_1 : p(X, X) \rightarrow goal(X)$
$r_2 : p(X, Y) \rightarrow goal(X)$



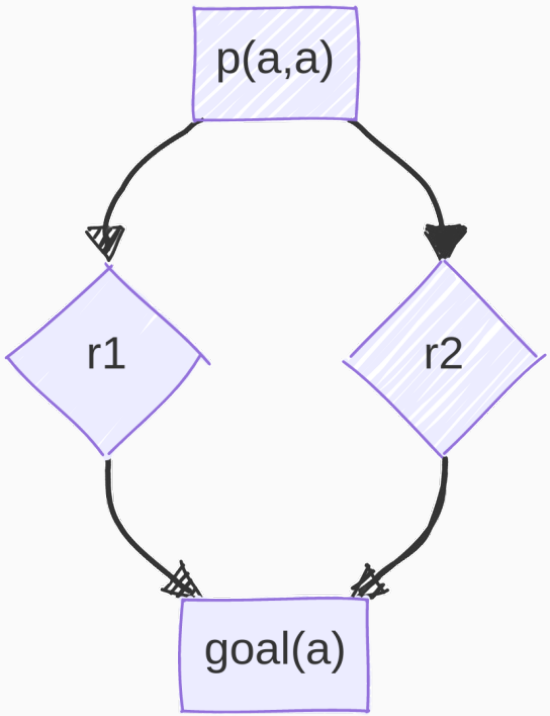
# From Datalog to Propositional! KB-Support explanation

$p(a, a)$

$r_1 : p(X, X) \rightarrow goal(X)$
$r_2 : p(X, Y) \rightarrow goal(X)$

$\longrightarrow p(a, a) \wedge (p(a, a) \rightarrow goal(a)) \wedge (\neg goal(a))$

*Grounding*



# From Datalog to Propositional! KB-Support explanation

$p(a, a)$

$r_1 : p(X, X) \rightarrow goal(X)$
$r_2 : p(X, Y) \rightarrow goal(X)$

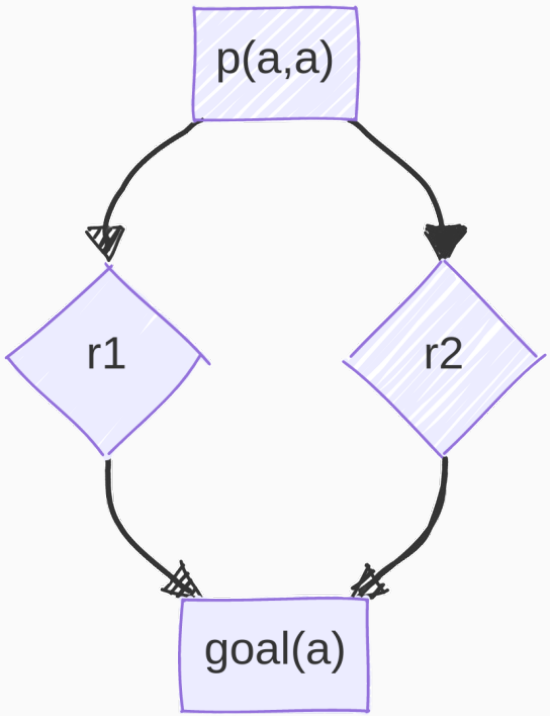
$\longrightarrow p(a, a) \wedge (p(a, a) \rightarrow goal(a)) \wedge (\neg goal(a))$

*Grounding*

*MUS solving*

1 MUS

$\{p(a, a), (p(a, a) \rightarrow goal(a)), (\neg goal(a))\}$



# From Datalog to Propositional! KB-Support explanation

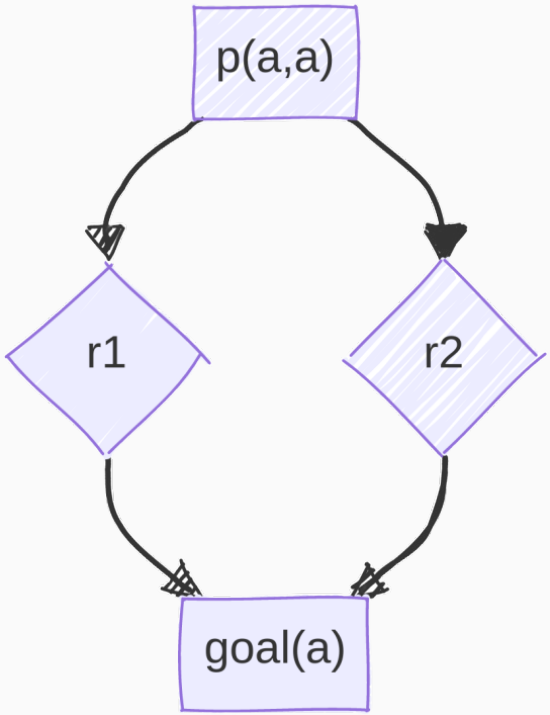
$p(a, a)$

$r_1 : p(X, X) \rightarrow goal(X)$
$r_2 : p(X, Y) \rightarrow goal(X)$

$\longrightarrow p(a, a) \wedge (p(a, a) \rightarrow goal(a)) \wedge (\neg goal(a))$

*Grounding*

*MUS solving*




1 MUS

$\{p(a, a), (p(a, a) \rightarrow goal(a)), (\neg goal(a))\}$

2 explanations

$(\{p(a, a)\}, \{r_1\})$
$(\{p(a, a)\}, \{r_2\})$

# KB-Support needs Group-MUS

- "Simple" MUS is not enough for KB-Support
  - Rules can share groundings
- Group-MUS to the rescue!
  - A group collects groundings of a given rule

$$p(a,a) \wedge (p(a,a) \rightarrow goal(a)) \wedge (p(a,a) \rightarrow goal(a)) \wedge (\neg goal(a))$$

- Two explanations for $goal(a)$
  - Group 1 , Group 2 , Group 4 $\Rightarrow$ $(\{p(a,a)\}, \{r_1\})$
  - Group 1 , Group 3 , Group 4 $\Rightarrow$ $(\{p(a,a)\}, \{r_2\})$
- KB-support explanations $\Longleftrightarrow$ Group-MUSes!

# Computing explanations is not easy

# Not easy at all!

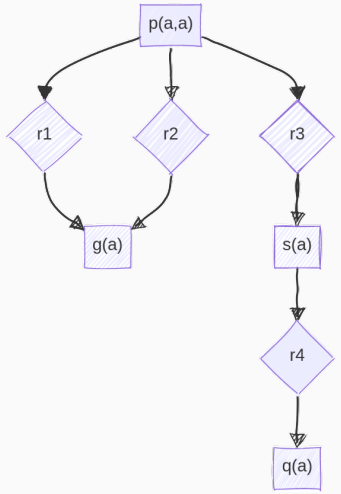
- Computing (Group-)MUSes is expensive
- And the knowledge base can be very large!
  - How can the GSAT formula be reduced?

$\Rightarrow$ Let's filter out the non-relevant part of the KB!

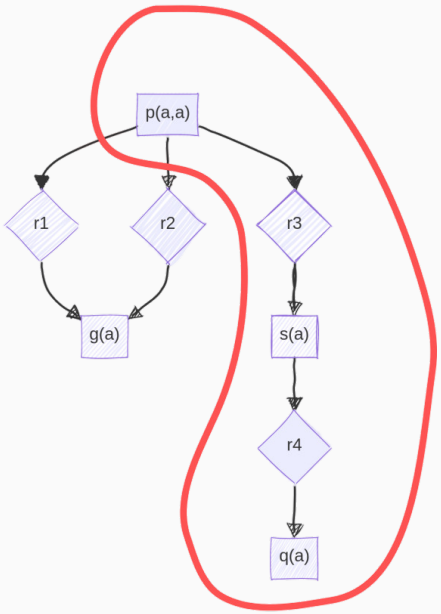
$$p \wedge (p \rightarrow q) \wedge (\neg q) \wedge s \wedge (q \vee \neg s) \wedge t$$

# Filtering: Finding the relevant part of the KB

- Two steps:
  1. Static step
     - Entailment graph building




  2. Dynamic step
     - Fact relevance tracing



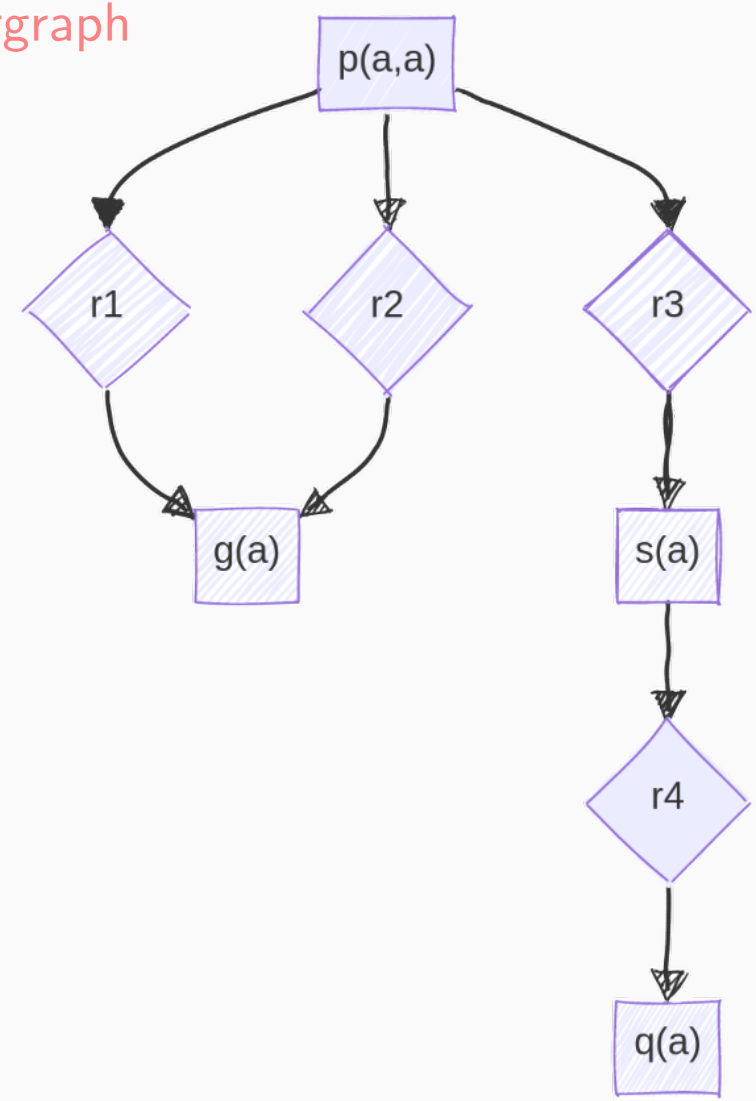
# Static step: building the atoms' lineage

- A program to build the ancestry hypergraph

$p(a, a)$

---

$r_1 : p(X, X) \to g(X)$
$r_2 : p(X, Y) \to g(X)$
$r_3 : p(X, X) \to s(X)$
$r_4 : s(X) \to q(X)$




*Rule transformation*

$$r_1' : p(X, X, f_p(X, X)) \to g(X, f_g(X)) \wedge edge(r_1, f_p(X, X), f_g(X))$$

- Computed only once!

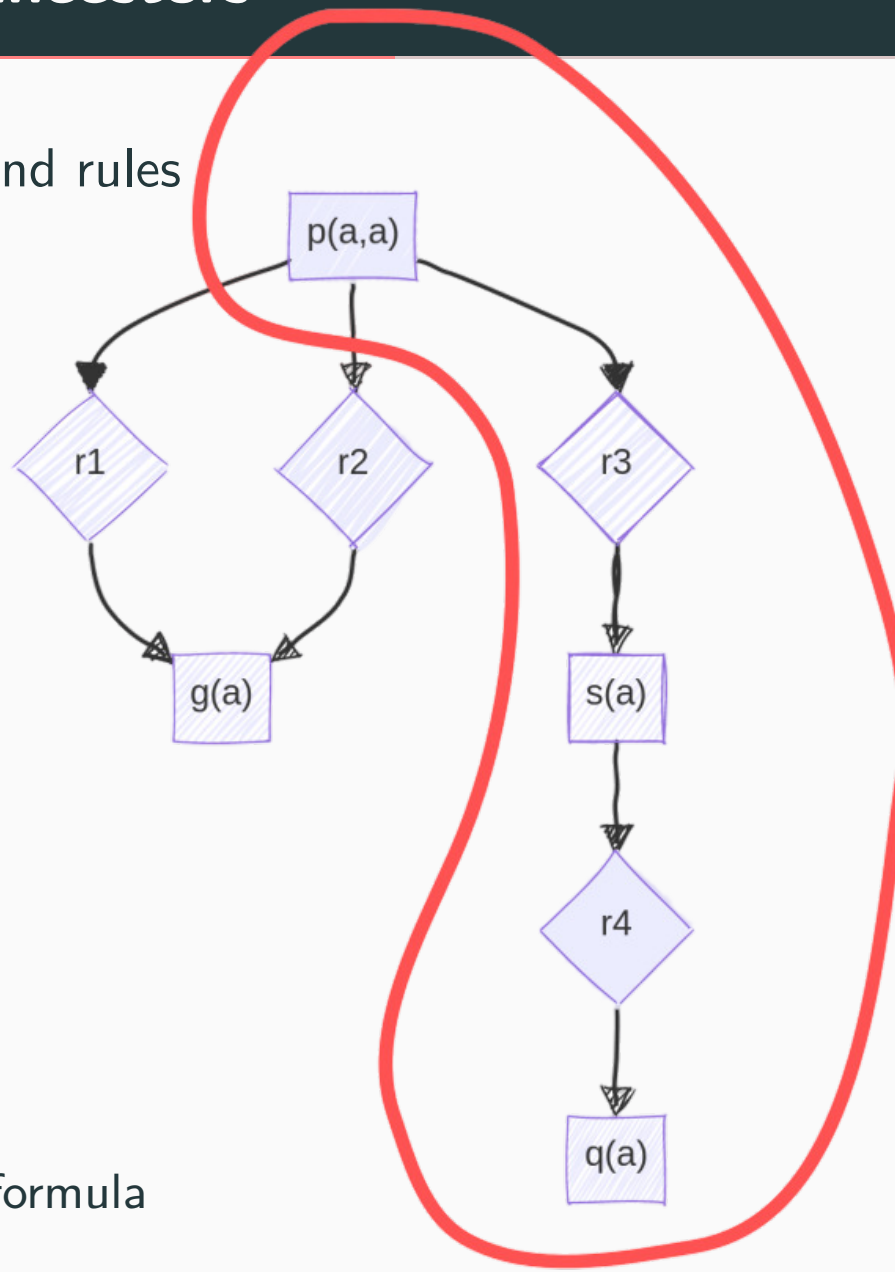
# Dynamic step: getting an atom's ancestors

- A program to extract relevant atoms and rules

$edge(r_1, f_p(a, a), f_g(a))$
$\ldots$
$rel(f_q(a))$

---

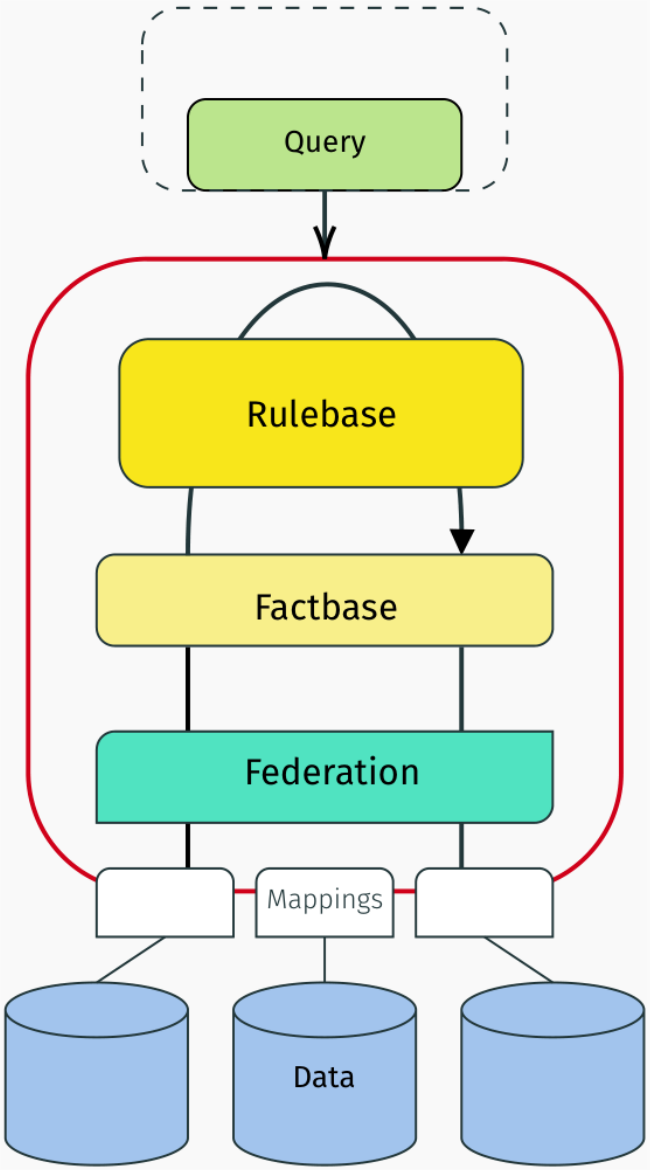
$rel(Y) \wedge edge(R, X, Y) \rightarrow rel(X) \wedge releedge(R, X, Y)$




- Computed for each query!
  - Reduces the size of the Group-SAT formula

# Let's be practical

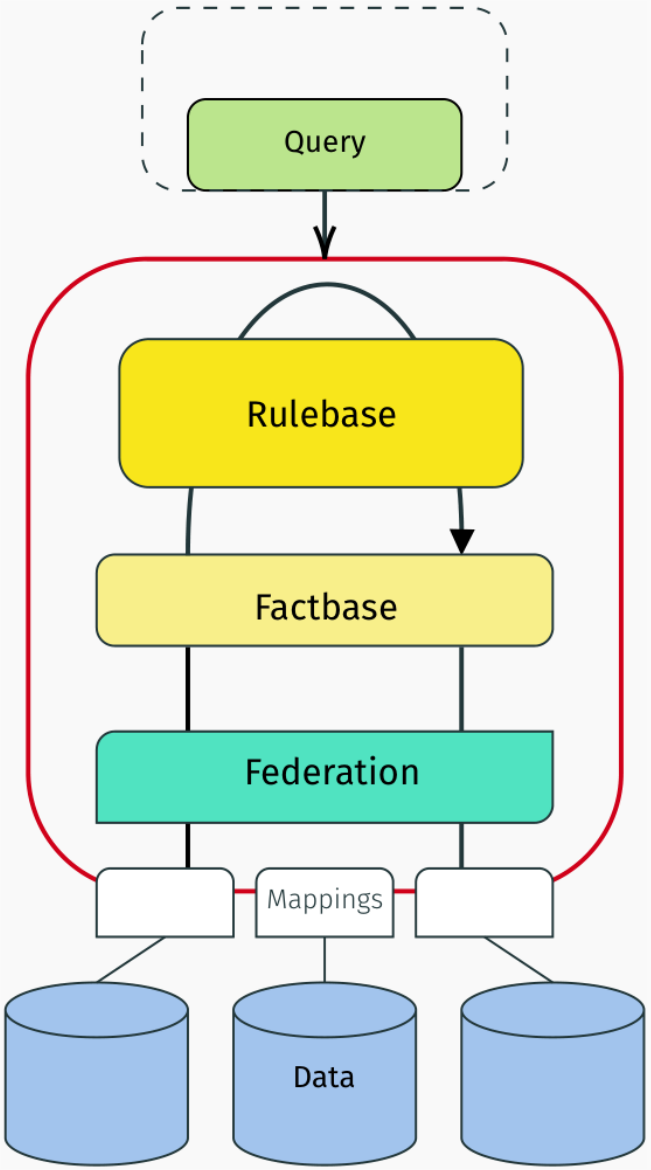
# Our playground: InteGraal




Long term **goals** for the team:

- Transfer result on (federated) OBDA
- Ground for experimental analysis
- Develop use cases and applications

# Our playground: InteGraal




- Query answering

$$\mathcal{F}, \mathcal{R} \models \mathcal{Q}$$

- Existential rules

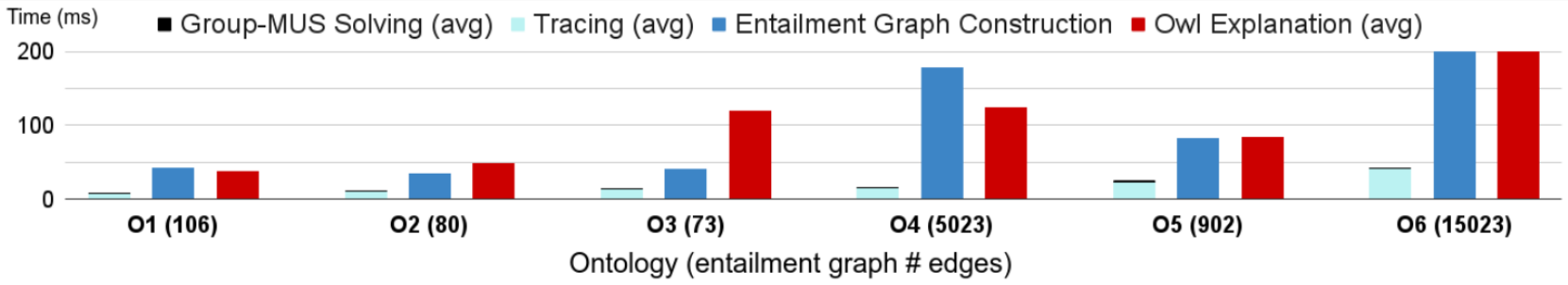
$$\forall \bar{x}\bar{y}.B[\bar{x}, \bar{y}] \rightarrow \exists z.H[\bar{x}, \bar{z}]$$

- Views

$$q_{native}(\bar{x}) \rightsquigarrow P(\bar{x})$$

# Is the approach actually worth?

- Bench:
  - 24 ontologies from MOWL corpus translated to Datalog
  - 5 facts from deepest reasoning level
  - Run against OWL Explanation (OWL API) tool




- Our approach is generally more competitive as the number of queries grows
  - And sometimes even for one query!

# Conclusion

# Related work

**Datalog**

- [Calautti et al., 2024]: SAT-solvers for computing why-provenance in Datalog.
- [Elhalawati et al., 2022]: On-demand computation of Datalog provenance using rule-based approaches.

**Description Logic**

- TBoxes (no data)
  - [Baader et al., 2007]: Axiom pinpointing for EL TBoxes.
  - [Manthey et al., 2020]: SAT-based axiom pinpointing for lightweight Description Logics.
- Provenance in EL Description Logic:
  - [Borgwardt et al., 2023]: ABox justifications via Datalog rewriting.
- Answer Set Programming:
  - [Eiter & Geibinger, 2023]: Explaining answer-set programs with abstract constraint atoms.
  - [Alviano et al., 2023]: Graph-explanations extending fact-support in ASP.

# Conclusion

- Study of KB-support explanations for Datalog
- Reduction of explanation to Group-MUS
- Filtering of relevant facts and rules exploiting Datalog programs
- Quite worth experimentally!