Computing Stable Models for Nonmonotonic Existential Rules

Despoina Magka, Markus Krötzsch, Ian Horrocks

Department of Computer Science, University of Oxford

IJCAI, 2013

The OWLs are not what they seem

- OWL widely used for authoring biomedical ontologies

The OWLs are not what they seem

- OWL widely used for authoring biomedical ontologies

The OWLs are not what they seem

- OWL widely used for authoring biomedical ontologies

GO the Gene Ontology

NCIthesaurus

BioPortal

SNOMED INTERNATIONAL

- Not marked for its ability to model cyclic structures

THE OWLS ARE NOT WHAT THEY SEEM

- OWL widely used for authoring biomedical ontologies

GO the Gene Ontology

NCIthesaurus

BioPortal

SNOMED INTERNATIONAL

- Not marked for its ability to model cyclic structures
- Such structures abound in life science (and other) domains

THE OWLS ARE NOT WHAT THEY SEEM

- OWL widely used for authoring biomedical ontologies

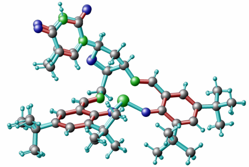
GO the Gene Ontology

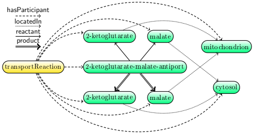
NCIthesaurus

BioPortal

SNOMED INTERNATIONAL

- Not marked for its ability to model cyclic structures
- Such structures abound in life science (and other) domains





Nonmonotonic Existential Rules

- Rules with nonmonotonic negation in the body and existentials in the head

$$B_1 \wedge \ldots \wedge B_n \wedge \ \mathbf{not}\ B_{n+1} \wedge \ldots \wedge \ \mathbf{not}\ B_m \rightarrow \exists \mathbf{y}.H_1 \wedge \ldots \wedge H_k$$

Nonmonotonic Existential Rules

- Rules with nonmonotonic negation in the body and existentials in the head

$$B_1 \wedge \ldots \wedge B_n \wedge \ \textbf{not}\ B_{n+1} \wedge \ldots \wedge \ \textbf{not}\ B_m \rightarrow \exists \mathbf{y}.H_1 \wedge \ldots \wedge H_k$$

- Interpreted under stable model semantics

NONMONOTONIC EXISTENTIAL RULES

- Rules with nonmonotonic negation in the body and existentials in the head

$$B_1 \wedge \ldots \wedge B_n \wedge \ \mathbf{not}\ B_{n+1} \wedge \ldots \wedge \ \mathbf{not}\ B_m \rightarrow \exists \mathbf{y}.H_1 \wedge \ldots \wedge H_k$$

- Interpreted under stable model semantics

- Good for representing non-tree-shaped structures

Nonmonotonic Existential Rules

- Rules with nonmonotonic negation in the body and existentials in the head

$$B_1 \land \ldots \land B_n \land \ \textbf{not}\ B_{n+1} \land \ldots \land \ \textbf{not}\ B_m \rightarrow \exists \mathbf{y}.H_1 \land \ldots \land H_k$$

- Interpreted under stable model semantics
- Good for representing non-tree-shaped structures
  - Existentials allow us to infer new structures

Nonmonotonic Existential Rules

- Rules with nonmonotonic negation in the body and existentials in the head

$$B_1 \wedge \ldots \wedge B_n \wedge \ \textbf{not}\ B_{n+1} \wedge \ldots \wedge \ \textbf{not}\ B_m \rightarrow \exists \mathbf{y}.H_1 \wedge \ldots \wedge H_k$$

- Interpreted under stable model semantics

- Good for representing non-tree-shaped structures
  - Existentials allow us to infer new structures
  - Nonmonotonicity adds extra expressivity in modelling

Nonmonotonic Existential Rules

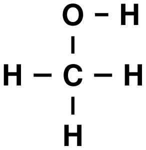
- Rules with nonmonotonic negation in the body and existentials in the head

$$B_1 \wedge \ldots \wedge B_n \wedge \ \mathbf{not}\ B_{n+1} \wedge \ldots \wedge \ \mathbf{not}\ B_m \rightarrow \exists \mathbf{y}.H_1 \wedge \ldots \wedge H_k$$

- Interpreted under stable model semantics

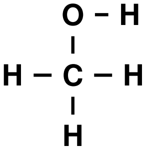
- Good for representing non-tree-shaped structures
  - Existentials allow us to infer new structures
  - Nonmonotonicity adds extra expressivity in modelling
  - Stable model semantics supported by many tools: DLV, clasp, . . .

Classification of Structured Objects I



Methanol molecule

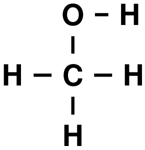
Classification of Structured Objects I



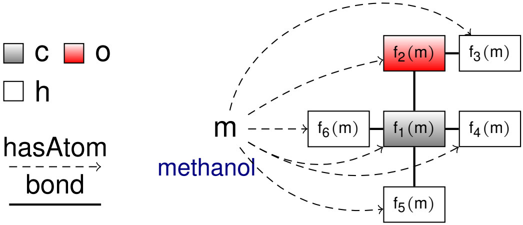
Methanol molecule

$$\mathsf{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i .\ \wedge_{i=1}^{6} \mathsf{hasAtom}(x, y_i) \wedge c(y_1) \wedge o(y_2) \wedge \wedge_{i=3}^{6} h(y_i) \wedge \wedge_{i=2}^{5} \mathsf{bond}(y_1, y_i) \wedge \mathsf{bond}(y_2, y_6)$$

Classification of Structured Objects I

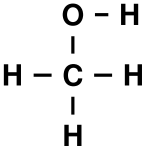


Methanol molecule

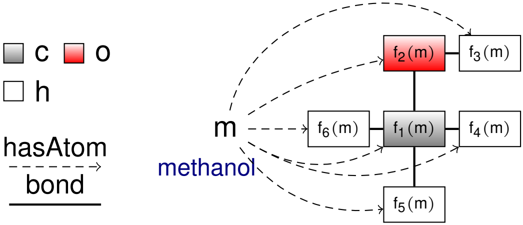


$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i .\ \wedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge c(y_1) \wedge o(y_2) \wedge \wedge_{i=3}^{6} h(y_i) \wedge \wedge_{i=2}^{5} \text{bond}(y_1, y_i) \wedge \text{bond}(y_2, y_6)$$

Classification of Structured Objects I



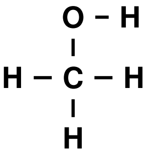
Methanol molecule



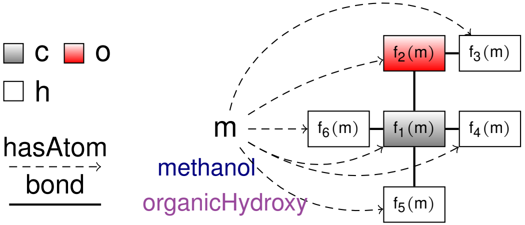
$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \bigwedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge c(y_1) \wedge o(y_2) \wedge \bigwedge_{i=3}^{6} h(y_i) \wedge \bigwedge_{i=2}^{5} \text{bond}(y_1, y_i) \wedge \text{bond}(y_2, y_6)$$

$$\bigwedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge c(z_1) \wedge o(z_2) \wedge h(z_3) \wedge \text{bond}(z_1, z_2) \wedge \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$$

Classification of Structured Objects I



Methanol molecule

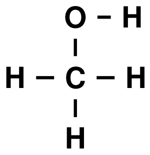


$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \wedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge c(y_1) \wedge o(y_2) \wedge$$
$$\wedge_{i=3}^{6} h(y_i) \wedge \wedge_{i=2}^{5} \text{bond}(y_1, y_i) \wedge \text{bond}(y_2, y_6)$$

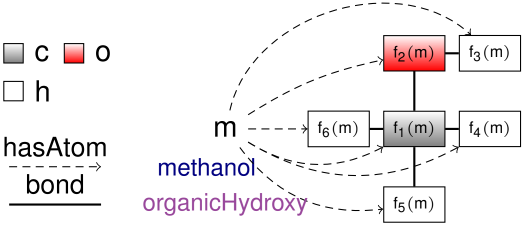
$$\wedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge c(z_1) \wedge o(z_2) \wedge$$
$$h(z_3) \wedge \text{bond}(z_1, z_2) \wedge \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$$

$\text{methanol} \sqsubseteq \text{organicHydroxy}$ ✓

CLASSIFICATION OF STRUCTURED OBJECTS I



Methanol molecule



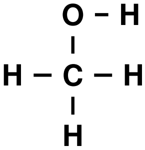
$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \bigwedge_{i=1}^{6} \text{hasAtom}(x, y_i) \land c(y_1) \land o(y_2) \land \\ \bigwedge_{i=3}^{6} h(y_i) \land \bigwedge_{i=2}^{5} \text{bond}(y_1, y_i) \land \text{bond}(y_2, y_6)$$

$$\bigwedge_{i=1}^{3} \text{hasAtom}(x, z_i) \land c(z_1) \land o(z_2) \land \\ h(z_3) \land \text{bond}(z_1, z_2) \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$$

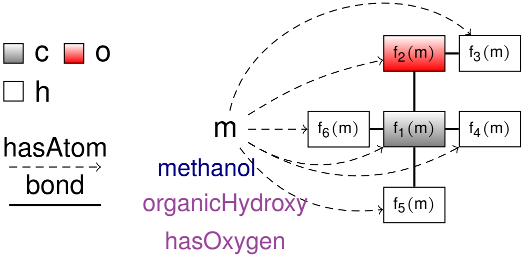
$$\text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x)$$

$\text{methanol} \sqsubseteq \text{organicHydroxy}$ ✓

Classification of Structured Objects I



Methanol molecule



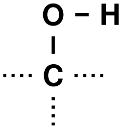
$$\mathrm{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \bigwedge_{i=1}^{6} \mathrm{hasAtom}(x,y_i) \wedge c(y_1) \wedge o(y_2) \wedge \bigwedge_{i=3}^{6} h(y_i) \wedge \bigwedge_{i=2}^{5} \mathrm{bond}(y_1,y_i) \wedge \mathrm{bond}(y_2,y_6)$$

$$\bigwedge_{i=1}^{3} \mathrm{hasAtom}(x,z_i) \wedge c(z_1) \wedge o(z_2) \wedge h(z_3) \wedge \mathrm{bond}(z_1,z_2) \wedge \mathrm{bond}(z_2,z_3) \rightarrow \mathrm{organicHydroxy}(x)$$

$$\mathrm{hasAtom}(x,z) \wedge o(z) \rightarrow \mathrm{hasOxygen}(x)$$

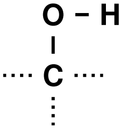
$\mathrm{methanol} \sqsubseteq \mathrm{organicHydroxy}$ ✓ $\mathrm{methanol} \sqsubseteq \mathrm{hasOxygen}$ ✓

Classification of Structured Objects II



Organic hydroxy group

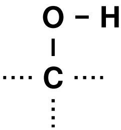
Classification of Structured Objects II



Organic hydroxy group

$$\mathsf{organicHydroxy}(x) \rightarrow \exists_{\mathsf{i}=1}^{3} \mathsf{y_i}.\ \wedge_{\mathsf{i}=1}^{3} \mathsf{hasAtom}(\mathsf{x}, \mathsf{y_i}) \wedge \mathsf{c}(\mathsf{y_1}) \\ \wedge \mathsf{o}(\mathsf{y_2}) \wedge \mathsf{h}(\mathsf{y_3}) \wedge \mathsf{bond}(\mathsf{y_1}, \mathsf{y_2}) \\ \wedge \mathsf{bond}(\mathsf{y_2}, \mathsf{y_3})$$

Classification of Structured Objects II

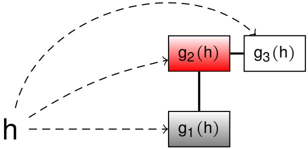


Organic hydroxy group

▨ c 🟥 o

□ h

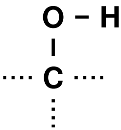
$\xrightarrow[\text{bond}]{\text{hasAtom}}$



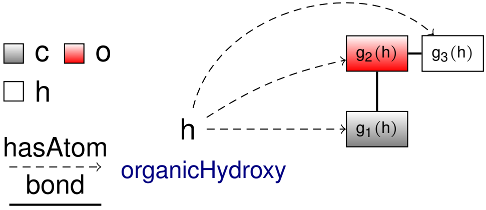
organicHydroxy

$$
\begin{aligned}
\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i .\ \wedge_{i=1}^{3}\ &\text{hasAtom}(x, y_i) \wedge c(y_1) \\
&\wedge o(y_2) \wedge h(y_3) \wedge \text{bond}(y_1, y_2) \\
&\wedge \text{bond}(y_2, y_3)
\end{aligned}
$$

Classification of Structured Objects II



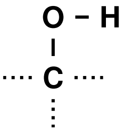
Organic hydroxy group



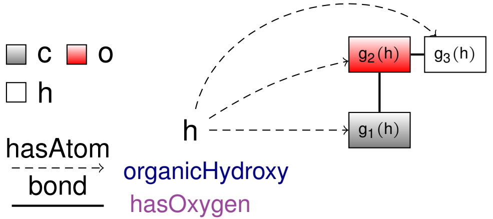
$$\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i .\ \wedge_{i=1}^{3} \text{hasAtom}(x, y_i) \wedge c(y_1) \wedge o(y_2) \wedge h(y_3) \wedge \text{bond}(y_1, y_2) \wedge \text{bond}(y_2, y_3)$$

$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

Classification of Structured Objects II



Organic hydroxy group



$$\text{organicHydroxy}(x) \rightarrow \exists^{3}_{i=1} y_i .\ \wedge^{3}_{i=1} \text{hasAtom}(x, y_i) \wedge c(y_1) \wedge o(y_2) \wedge h(y_3) \wedge \text{bond}(y_1, y_2) \wedge \text{bond}(y_2, y_3)$$

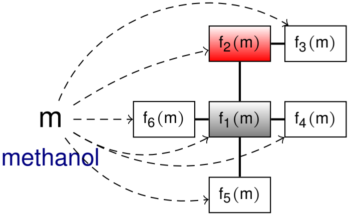
$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

$$\text{organicHydroxy} \sqsubseteq \text{hasOxygen} \checkmark$$

Incorrect Modelling

$$
\begin{aligned}
\mathsf{methanol}(x) &\rightarrow \exists_{i=1}^{6} y_i .\ \wedge_{i=1}^{6} \mathsf{hasAtom}(x, y_i) \wedge \ldots \\
&\qquad\qquad \wedge\, \mathsf{bond}(y_2, y_6) \\
\wedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \ldots \wedge & \\
\mathsf{bond}(z_2, z_3) &\rightarrow \mathsf{organicHydroxy}(x) \\
\mathsf{organicHydroxy}(x) &\rightarrow \exists_{i=1}^{3} y_i .\ \wedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \wedge \ldots \\
&\qquad\qquad \wedge\, \mathsf{bond}(y_2, y_3) \\
\mathsf{hasAtom}(x, z) \wedge o(z) &\rightarrow \mathsf{hasOxygen}(x)
\end{aligned}
$$

Incorrect Modelling



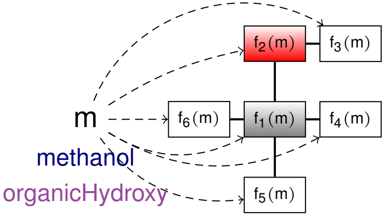
$$\mathrm{methanol}(x) \rightarrow \exists^{6}_{i=1} y_i . \bigwedge^{6}_{i=1} \mathrm{hasAtom}(x, y_i) \wedge \ldots \\ \wedge\, \mathrm{bond}(y_2, y_6)$$

$$\bigwedge^{3}_{i=1} \mathrm{hasAtom}(x, z_i) \wedge \ldots \wedge \\ \mathrm{bond}(z_2, z_3) \rightarrow \mathrm{organicHydroxy}(x)$$

$$\mathrm{organicHydroxy}(x) \rightarrow \exists^{3}_{i=1} y_i . \bigwedge^{3}_{i=1} \mathrm{hasAtom}(x, y_i) \wedge \ldots \\ \wedge\, \mathrm{bond}(y_2, y_3)$$

$$\mathrm{hasAtom}(x, z) \wedge o(z) \rightarrow \mathrm{hasOxygen}(x)$$

Incorrect Modelling



$$\text{methanol} \sqsubseteq \text{organicHydroxy} \checkmark$$

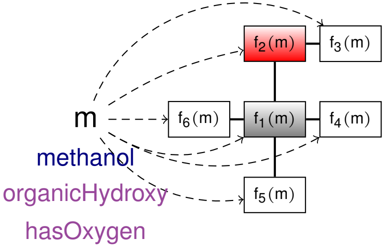
$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i . \bigwedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_6)$$

$$\bigwedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge \ldots \wedge \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$$

$$\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i . \bigwedge_{i=1}^{3} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_3)$$

$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

Incorrect Modelling



methanol $\sqsubseteq$ hasOxygen ✓

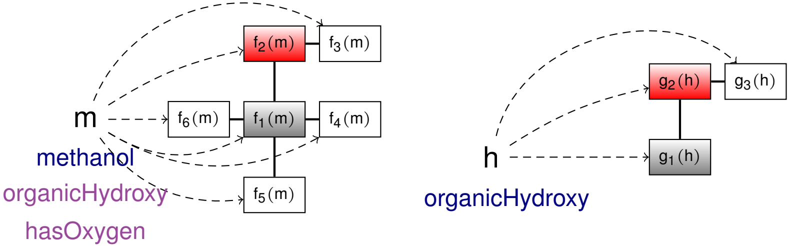
$$\mathrm{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \wedge_{i=1}^{6} \mathrm{hasAtom}(x, y_i) \wedge \ldots \wedge \mathrm{bond}(y_2, y_6)$$

$$\wedge_{i=1}^{3} \mathrm{hasAtom}(x, z_i) \wedge \ldots \wedge \mathrm{bond}(z_2, z_3) \rightarrow \mathrm{organicHydroxy}(x)$$

$$\mathrm{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i.\ \wedge_{i=1}^{3} \mathrm{hasAtom}(x, y_i) \wedge \ldots \wedge \mathrm{bond}(y_2, y_3)$$

$$\mathrm{hasAtom}(x, z) \wedge o(z) \rightarrow \mathrm{hasOxygen}(x)$$

Incorrect Modelling



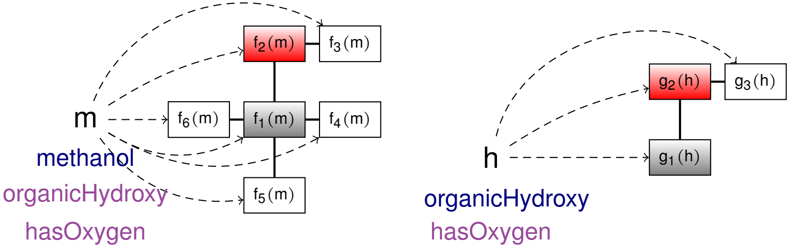
$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i . \wedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_6)$$

$$\wedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge \ldots \wedge \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$$

$$\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i . \wedge_{i=1}^{3} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_3)$$

$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

Incorrect Modelling



m

methanol

organicHydroxy

hasOxygen

h

organicHydroxy

hasOxygen

organicHydroxy $\sqsubseteq$ hasOxygen ✓

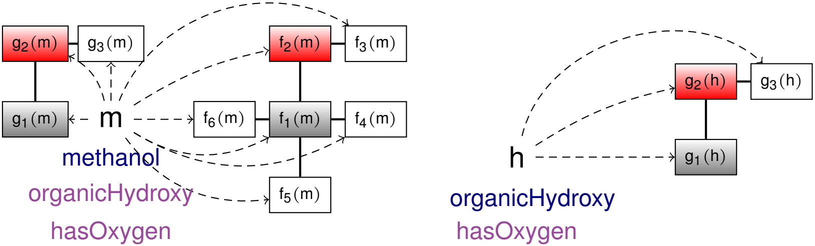
$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i . \bigwedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_6)$$

$$\bigwedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge \ldots \wedge \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$$

$$\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i . \bigwedge_{i=1}^{3} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_3)$$

$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

Incorrect Modelling



$$\text{methanol} \sqsubseteq \text{hasOneCarbon}$$ ✗

$$\text{methanol}(x) \rightarrow \exists^{6}_{i=1} y_i .\ \bigwedge^{6}_{i=1} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_6)$$

$$\bigwedge^{3}_{i=1} \text{hasAtom}(x, z_i) \wedge \ldots \wedge \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$$

$$\text{organicHydroxy}(x) \rightarrow \exists^{3}_{i=1} y_i .\ \bigwedge^{3}_{i=1} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_3)$$

$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

Repair With Auxiliary Predicates

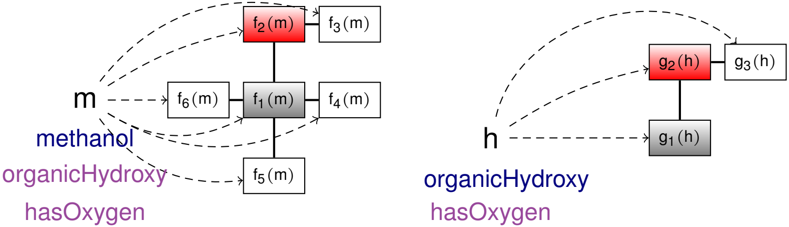
$$\mathsf{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \wedge_{i=1}^{6} \mathsf{hasAtom}(x, y_i) \wedge \ldots \wedge \mathsf{bond}(y_2, y_6)$$

$$\wedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \ldots \wedge \mathsf{bond}(z_2, z_3) \wedge \textbf{ not } g_h(z_1) \wedge \textbf{ not } g_h(z_2) \wedge \textbf{ not } g_h(z_3) \rightarrow \mathsf{organicHydroxy}(x) \wedge r_h(x)$$

$$\mathsf{organicHydroxy}(x) \wedge \textbf{ not } r_h(x) \rightarrow \exists_{i=1}^{3} y_i.\ \wedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \wedge \ldots \wedge \mathsf{bond}(y_2, y_3) \wedge \wedge_{i=1}^{3} g_h(y_i)$$

$$\mathsf{hasAtom}(x, z) \wedge o(z) \rightarrow \mathsf{hasOxygen}(x)$$

Repair With Auxiliary Predicates



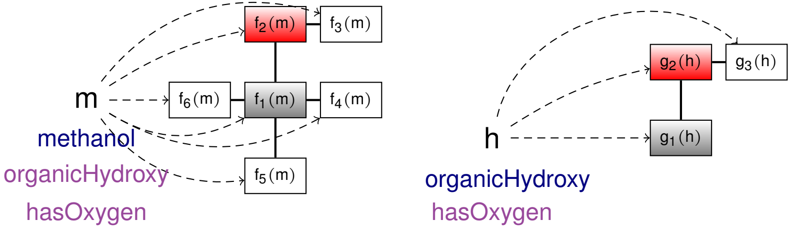
$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \wedge_{i=1}^{6}\ \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_6)$$

$$\wedge_{i=1}^{3}\ \text{hasAtom}(x, z_i) \wedge \ldots \wedge \text{bond}(z_2, z_3) \wedge\ \textbf{not}\ g_h(z_1) \wedge\ \textbf{not}\ g_h(z_2) \wedge\ \textbf{not}\ g_h(z_3) \rightarrow \text{organicHydroxy}(x) \wedge r_h(x)$$

$$\text{organicHydroxy}(x) \wedge\ \textbf{not}\ r_h(x) \rightarrow \exists_{i=1}^{3} y_i.\ \wedge_{i=1}^{3}\ \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_3) \wedge \wedge_{i=1}^{3} g_h(y_i)$$

$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

Repair With Auxiliary Predicates



methanol
organicHydroxy
hasOxygen

organicHydroxy
hasOxygen

methanol $\sqsubseteq$ hasOneCarbon ✓

$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \wedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_6)$$

$$\wedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge \ldots \wedge \text{bond}(z_2, z_3) \wedge \textbf{ not } g_h(z_1) \wedge \textbf{ not } g_h(z_2) \wedge \textbf{ not } g_h(z_3) \rightarrow \text{organicHydroxy}(x) \wedge r_h(x)$$

$$\text{organicHydroxy}(x) \wedge \textbf{ not } r_h(x) \rightarrow \exists_{i=1}^{3} y_i.\ \wedge_{i=1}^{3} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_3) \wedge \wedge_{i=1}^{3} g_h(y_i)$$

$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

WHAT'S THE PROBLEM?

- Reasoning is undecidable
  (even fact entailment, even without **not**)

WHAT'S THE PROBLEM?

- Reasoning is undecidable
  (even fact entailment, even without **not**)
  - many known conditions for regaining decidability
  - acyclicity conditions ensure finite models: (super)-weak acyclicity, joint acyclicity, aGRD, MSA, MFA, . . .

What's the Problem?

- Reasoning is undecidable
  (even fact entailment, even without **not**)
  - many known conditions for regaining decidability
  - acyclicity conditions ensure finite models: (super)-weak acyclicity, joint acyclicity, aGRD, MSA, MFA, . . .

- Reasoning is hard (even for finite models)

WHAT'S THE PROBLEM?

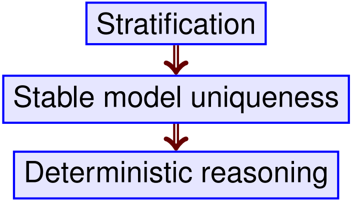
- Reasoning is undecidable
  (even fact entailment, even without **not**)
  - many known conditions for regaining decidability
  - acyclicity conditions ensure finite models: (super)-weak acyclicity, joint acyclicity, aGRD, MSA, MFA, . . .

- Reasoning is hard (even for finite models)
  - stable models lead to non-determinism
  - stratification conditions ensure determinism

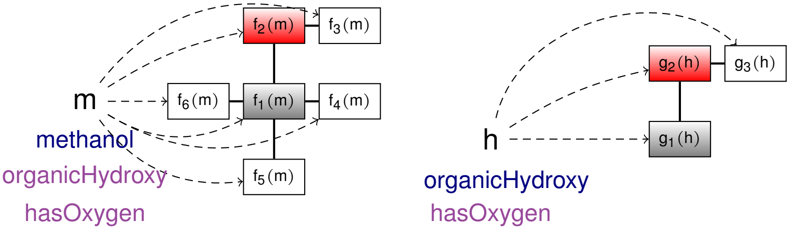
WHAT'S THE PROBLEM?

- Reasoning is undecidable
  (even fact entailment, even without **not**)
  - many known conditions for regaining decidability
  - acyclicity conditions ensure finite models: (super)-weak acyclicity, joint acyclicity, aGRD, MSA, MFA, ...

- Reasoning is hard (even for finite models)
  - stable models lead to non-determinism
  - stratification conditions ensure determinism



What's our Problem?



Repaired program not stratified

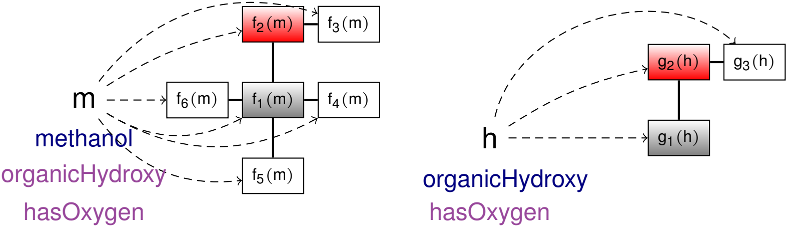
$$\begin{aligned} \mathrm{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \wedge_{i=1}^{6} \mathrm{hasAtom}(x, y_i) \wedge \ldots \\ \wedge\, \mathrm{bond}(y_2, y_6) \end{aligned}$$

$$\begin{aligned} &\wedge_{i=1}^{3} \mathrm{hasAtom}(x, z_i) \wedge \ldots \wedge \\ &\mathrm{bond}(z_2, z_3) \wedge\ \mathbf{not}\ g_h(z_1) \\ &\wedge\ \mathbf{not}\ g_h(z_2) \wedge\ \mathbf{not}\ g_h(z_3) \rightarrow \mathrm{organicHydroxy}(x) \wedge r_h(x) \end{aligned}$$

$$\begin{aligned} \mathrm{organicHydroxy}(x) \wedge\ \mathbf{not}\ r_h(x) \rightarrow \exists_{i=1}^{3} y_i.\ \wedge_{i=1}^{3} \mathrm{hasAtom}(x, y_i) \wedge \ldots \\ \wedge\, \mathrm{bond}(y_2, y_3) \wedge \wedge_{i=1}^{3} g_h(y_i) \end{aligned}$$

$$\mathrm{hasAtom}(x, z) \wedge o(z) \rightarrow \mathrm{hasOxygen}(x)$$

What's our Problem?



Repaired program not stratified

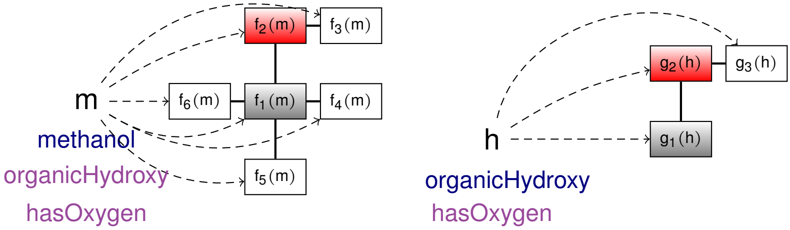
$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i.\ \wedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_6)$$

$$\wedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge \ldots \wedge \text{bond}(z_2, z_3) \wedge \boxed{\textbf{not}\ g_h(z_1)} \wedge \textbf{not}\ g_h(z_2) \wedge \textbf{not}\ g_h(z_3) \rightarrow \text{organicHydroxy}(x) \wedge r_h(x)$$

$$\text{organicHydroxy}(x) \wedge \textbf{not}\ r_h(x) \rightarrow \exists_{i=1}^{3} y_i.\ \wedge_{i=1}^{3} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_3) \wedge \wedge_{i=1}^{3} \boxed{g_h(y_i)}$$

$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

What's our Problem?



Repaired program not stratified

$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i .\ \wedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_6)$$

$$\wedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge \ldots \wedge \text{bond}(z_2, z_3) \wedge \textbf{not } g_h(z_1) \wedge \textbf{not } g_h(z_2) \wedge \textbf{not } g_h(z_3) \rightarrow \text{organicHydroxy}(x) \wedge \boxed{r_h(x)}$$

$$\text{organicHydroxy}(x) \wedge \boxed{\textbf{not } r_h(x)} \rightarrow \exists_{i=1}^{3} y_i .\ \wedge_{i=1}^{3} \text{hasAtom}(x, y_i) \wedge \ldots \wedge \text{bond}(y_2, y_3) \wedge \wedge_{i=1}^{3} g_h(y_i)$$

$$\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)$$

RESULTS OVERVIEW

1. R-acyclicity and R-stratification conditions
   - R-stratification ensures stable model uniqueness
   - Both coNP-complete to check

Results Overview

1. R-acyclicity and R-stratification conditions
   - R-stratification ensures stable model uniqueness
   - Both coNP-complete to check
2. Complexity of reasoning

Fact entailment	Program comp.	Data comp.
R-acyclic	coN2ExpTime-complete	coNP-complete
R-acyclic+R-stratified	2ExpTime-complete	PTime-complete

RESULTS OVERVIEW

1. R-acyclicity and R-stratification conditions
   - R-stratification ensures stable model uniqueness
   - Both coNP-complete to check
2. Complexity of reasoning

Fact entailment	Program comp.	Data comp.
R-acyclic	coN2ExpTime-complete	coNP-complete
R-acyclic+R-stratified	2ExpTime-complete	PTime-complete

3. Generalise R-acyclicity and R-stratification with constraints
   $\rightsquigarrow$ new conditions Π_2^P-complete to check

RESULTS OVERVIEW

1. R-acyclicity and R-stratification conditions
   - R-stratification ensures stable model uniqueness
   - Both coNP-complete to check
2. Complexity of reasoning

Fact entailment	Program comp.	Data comp.
R-acyclic	coN2ExpTime-complete	coNP-complete
R-acyclic+R-stratified	2ExpTime-complete	PTime-complete

3. Generalise R-acyclicity and R-stratification with constraints
   $\rightsquigarrow$ new conditions Π_2^P-complete to check
4. Experiments over ChEBI with DLV
   - Performance gains in DLV using R-stratification
   - Missing subsumptions from ChEBI ontology

Positive reliances

- Rule r_2 positively relies on r_1 (written $r_1 \xrightarrow{+} r_2$): there is a situation when r_1 can trigger r_2 to derive something new

Positive reliances

- Rule r_2 positively relies on r_1 (written $r_1 \xrightarrow{+} r_2$): there is a situation when r_1 can trigger r_2 to derive something new

Example

$$\begin{aligned} r_1 : \quad & \wedge_{i=1}^{3} \mathsf{hasAtom}(\mathsf{x}, \mathsf{z_i}) \wedge \\ & \mathsf{c}(\mathsf{z_1}) \wedge \mathsf{o}(\mathsf{z_2}) \wedge \mathsf{h}(\mathsf{z_3}) \wedge \\ & \mathsf{bond}(\mathsf{z_1}, \mathsf{z_2}) \wedge \mathsf{bond}(\mathsf{z_2}, \mathsf{z_3}) \rightarrow \mathsf{organicHydroxy}(\mathsf{x}) \end{aligned}$$

$$\begin{aligned} r_2 : \quad \mathsf{organicHydroxy}(\mathsf{x}) \rightarrow & \exists_{i=1}^{3} \mathsf{y_i}. \wedge_{i=1}^{3} \mathsf{hasAtom}(\mathsf{x}, \mathsf{y_i}) \wedge \\ & \mathsf{c}(\mathsf{y_1}) \wedge \mathsf{o}(\mathsf{y_2}) \wedge \mathsf{h}(\mathsf{y_3}) \wedge \\ & \mathsf{bond}(\mathsf{y_1}, \mathsf{y_2}) \wedge \mathsf{bond}(\mathsf{y_2}, \mathsf{y_3}) \end{aligned}$$

POSITIVE RELIANCES

- Rule r_2 positively relies on r_1 (written $r_1 \xrightarrow{+} r_2$): there is a situation when r_1 can trigger r_2 to derive something new

EXAMPLE

$$
\begin{aligned}
r_1 :\quad & \wedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \\
& c(z_1) \wedge o(z_2) \wedge h(z_3) \wedge \\
& \mathsf{bond}(z_1, z_2) \wedge \mathsf{bond}(z_2, z_3) \rightarrow \boxed{\mathsf{organicHydroxy}(x)} \\
r_2 :\quad & \boxed{\mathsf{organicHydroxy}(x)} \rightarrow \exists_{i=1}^{3} y_i . \wedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \wedge \\
& c(y_1) \wedge o(y_2) \wedge h(y_3) \wedge \\
& \mathsf{bond}(y_1, y_2) \wedge \mathsf{bond}(y_2, y_3)
\end{aligned}
$$

$$r_1 \xrightarrow{+} r_2$$

POSITIVE RELIANCES

- Rule r_2 positively relies on r_1 (written $r_1 \xrightarrow{+} r_2$): there is a situation when r_1 can trigger r_2 to derive something new

EXAMPLE

$$
\begin{aligned}
r_1 : \quad & \wedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \\
& c(z_1) \wedge o(z_2) \wedge h(z_3) \wedge \\
& \mathsf{bond}(z_1, z_2) \wedge \mathsf{bond}(z_2, z_3) \to \mathsf{organicHydroxy}(x)
\end{aligned}
$$

$$
\begin{aligned}
r_2 : \quad \mathsf{organicHydroxy}(x) \to & \exists_{i=1}^{3} y_i . \wedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \wedge \\
& c(y_1) \wedge o(y_2) \wedge h(y_3) \wedge \\
& \mathsf{bond}(y_1, y_2) \wedge \mathsf{bond}(y_2, y_3)
\end{aligned}
$$

$r_1 \xrightarrow{+} r_2$ but $r_2 \not\xrightarrow{+} r_1$

Positive reliances

- Rule r_2 positively relies on r_1 (written $r_1 \xrightarrow{+} r_2$): there is a situation when r_1 can trigger r_2 to derive something new

Example

$$r_1 : \quad \wedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \mathsf{c}(z_1) \wedge \mathsf{o}(z_2) \wedge \mathsf{h}(z_3) \wedge \mathsf{bond}(z_1, z_2) \wedge \mathsf{bond}(z_2, z_3) \to \mathsf{organicHydroxy}(x)$$

$$r_2 : \quad \mathsf{organicHydroxy}(x) \to \exists_{i=1}^{3} y_i . \wedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \wedge \mathsf{c}(y_1) \wedge \mathsf{o}(y_2) \wedge \mathsf{h}(y_3) \wedge \mathsf{bond}(y_1, y_2) \wedge \mathsf{bond}(y_2, y_3)$$

$$r_1 \xrightarrow{+} r_2 \qquad \text{but} \qquad r_2 \stackrel{+}{\not\to} r_1$$

- NP-complete to check
  (but only w.r.t. the size of the rules)

Positive reliances

- Rule r_2 positively relies on r_1 (written $r_1 \xrightarrow{+} r_2$): there is a situation when r_1 can trigger r_2 to derive something new

Example

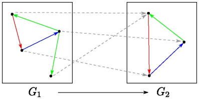
$$
\begin{aligned}
r_1 : \quad & \textstyle\bigwedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \\
& \mathsf{c}(z_1) \wedge \mathsf{o}(z_2) \wedge \mathsf{h}(z_3) \wedge \\
& \mathsf{bond}(z_1, z_2) \wedge \mathsf{bond}(z_2, z_3) \rightarrow \mathsf{organicHydroxy}(x)
\end{aligned}
$$

$$
\begin{aligned}
r_2 : \quad \mathsf{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i . & \textstyle\bigwedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \wedge \\
& \mathsf{c}(y_1) \wedge \mathsf{o}(y_2) \wedge \mathsf{h}(y_3) \wedge \\
& \mathsf{bond}(y_1, y_2) \wedge \mathsf{bond}(y_2, y_3)
\end{aligned}
$$

$r_1 \xrightarrow{+} r_2$ but $r_2 \overset{+}{\not\rightarrow} r_1$

- NP-complete to check

(but only w.r.t. the size of the rules)



R-ACYCLICITY

- A program is R-acyclic: there is no cycle of positive reliances that involves a rule with an existential
    - Checking R-acyclicity is coNP-complete
    - Similar to $\prec$-stratification [Deutsch et al., PODS, 2008]; extension of aGRD [Baget et al., RR, 2011]

R-ACYCLICITY

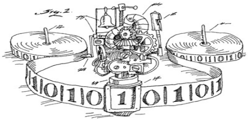
- A program is R-acyclic: there is no cycle of positive reliances that involves a rule with an existential
  - Checking R-acyclicity is coNP-complete
  - Similar to $\prec$-stratification [Deutsch et al., PODS, 2008]; extension of aGRD [Baget et al., RR, 2011]
- Fact entailment for R-acyclic programs
  - Stable models bounded in size (double exp), but many models possible

R-ACYCLICITY

- A program is R-acyclic: there is no cycle of positive reliances that involves a rule with an existential
  - Checking R-acyclicity is coNP-complete
  - Similar to $\prec$-stratification [Deutsch et al., PODS, 2008]; extension of aGRD [Baget et al., RR, 2011]
- Fact entailment for R-acyclic programs
  - Stable models bounded in size (double exp), but many models possible
  - coN2ExpTime-complete w.r.t. program complexity

R-ACYCLICITY

- A program is R-acyclic: there is no cycle of positive reliances that involves a rule with an existential
  - Checking R-acyclicity is coNP-complete
  - Similar to $\prec$-stratification [Deutsch et al., PODS, 2008]; extension of aGRD [Baget et al., RR, 2011]
- Fact entailment for R-acyclic programs
  - Stable models bounded in size (double exp), but many models possible
  - coN2ExpTime-complete w.r.t. program complexity



R-ACYCLICITY

- A program is R-acyclic: there is no cycle of positive reliances that involves a rule with an existential
  - Checking R-acyclicity is coNP-complete
  - Similar to $\prec$-stratification [Deutsch et al., PODS, 2008]; extension of aGRD [Baget et al., RR, 2011]
- Fact entailment for R-acyclic programs
  - Stable models bounded in size (double exp), but many models possible
  - coN2ExpTime-complete w.r.t. program complexity
  - coNP-complete w.r.t. data complexity

NEGATIVE RELIANCES

- Rule r_2 negatively relies on r_1 (written $r_1 \xrightarrow{-} r_2$): there is a situation when r_1 can inhibit the application of r_2

NEGATIVE RELIANCES

- Rule r_2 negatively relies on r_1 (written $r_1 \xrightarrow{-} r_2$): there is a situation when r_1 can inhibit the application of r_2

EXAMPLE

$$\begin{aligned} r_1 :\quad & \wedge_{i=1}^{3} \mathrm{hasAtom}(x, z_i) \wedge \mathrm{c}(z_1) \wedge \\ & \mathrm{o}(z_2) \wedge \mathrm{h}(z_3) \wedge \mathrm{bond}(z_1, z_2) \wedge \\ & \mathrm{bond}(z_2, z_3) \wedge \ \mathbf{not}\ \mathrm{g_h}(z_1) \wedge \\ & \mathbf{not}\ \mathrm{g_h}(z_2) \wedge \ \mathbf{not}\ \mathrm{g_h}(z_3) \rightarrow \mathrm{organicHydroxy}(x) \wedge \mathrm{r_h}(x) \end{aligned}$$

$$\begin{aligned} r_2 :\quad & \mathrm{organicHydroxy}(x) \wedge \ \mathbf{not}\ \mathrm{r_h}(x) \rightarrow \exists_{i=1}^{3} y_i . \wedge_{i=1}^{3} \mathrm{hasAtom}(x, y_i) \\ & \wedge \mathrm{c}(y_1) \wedge \mathrm{o}(y_2) \wedge \mathrm{h}(y_3) \wedge \\ & \mathrm{bond}(y_1, y_2) \wedge \mathrm{bond}(y_2, y_3) \\ & \wedge \mathrm{g_h}(y_1) \wedge \mathrm{g_h}(y_2) \wedge \mathrm{g_h}(y_3) \end{aligned}$$

Negative reliances

- Rule r_2 negatively relies on r_1 (written $r_1 \overset{-}{\rightarrow} r_2$): there is a situation when r_1 can inhibit the application of r_2

Example

$$
\begin{aligned}
r_1 :\quad & \wedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \mathsf{c}(z_1) \wedge \\
& \mathsf{o}(z_2) \wedge \mathsf{h}(z_3) \wedge \mathsf{bond}(z_1, z_2) \wedge \\
& \mathsf{bond}(z_2, z_3) \wedge \ \mathbf{not}\ \mathsf{g_h}(z_1) \wedge \\
& \mathbf{not}\ \mathsf{g_h}(z_2) \wedge \ \mathbf{not}\ \mathsf{g_h}(z_3) \rightarrow \mathsf{organicHydroxy}(x) \wedge \boxed{\mathsf{r_h}(x)}
\end{aligned}
$$

$$
\begin{aligned}
r_2 :\quad & \mathsf{organicHydroxy}(x) \wedge \boxed{\mathbf{not}\ \mathsf{r_h}(x)} \rightarrow \exists_{i=1}^{3} y_i .\ \wedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \\
& \wedge \mathsf{c}(y_1) \wedge \mathsf{o}(y_2) \wedge \mathsf{h}(y_3) \wedge \\
& \mathsf{bond}(y_1, y_2) \wedge \mathsf{bond}(y_2, y_3) \\
& \wedge \mathsf{g_h}(y_1) \wedge \mathsf{g_h}(y_2) \wedge \mathsf{g_h}(y_3)
\end{aligned}
$$

$$r_1 \overset{-}{\rightarrow} r_2$$

Negative reliances

- Rule r_2 negatively relies on r_1 (written $r_1 \xrightarrow{-} r_2$): there is a situation when r_1 can inhibit the application of r_2

Example

$$
\begin{aligned}
r_1 :\quad & \textstyle\bigwedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \mathsf{c}(z_1) \wedge \\
& \mathsf{o}(z_2) \wedge \mathsf{h}(z_3) \wedge \mathsf{bond}(z_1, z_2) \wedge \\
& \mathsf{bond}(z_2, z_3) \wedge \ \textbf{not}\ \mathsf{g_h}(z_1) \wedge \\
& \textbf{not}\ \mathsf{g_h}(z_2) \wedge \ \textbf{not}\ \mathsf{g_h}(z_3) \rightarrow \mathsf{organicHydroxy}(x) \wedge \mathsf{r_h}(x)
\end{aligned}
$$

$$
\begin{aligned}
r_2 :\quad & \mathsf{organicHydroxy}(x) \wedge \ \textbf{not}\ \mathsf{r_h}(x) \rightarrow \exists_{i=1}^{3} y_i .\ \textstyle\bigwedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \\
& \wedge \mathsf{c}(y_1) \wedge \mathsf{o}(y_2) \wedge \mathsf{h}(y_3) \wedge \\
& \mathsf{bond}(y_1, y_2) \wedge \mathsf{bond}(y_2, y_3) \\
& \wedge \mathsf{g_h}(y_1) \wedge \mathsf{g_h}(y_2) \wedge \mathsf{g_h}(y_3)
\end{aligned}
$$

$r_1 \xrightarrow{-} r_2$ but $r_2 \overset{-}{\not\rightarrow} r_1$

Negative reliances

- Rule r_2 negatively relies on r_1 (written $r_1 \overset{-}{\rightharpoonup} r_2$): there is a situation when r_1 can inhibit the application of r_2

Example

$$
\begin{aligned}
r_1 : \quad & \wedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \mathsf{c}(z_1) \wedge \\
& \mathsf{o}(z_2) \wedge \mathsf{h}(z_3) \wedge \mathsf{bond}(z_1, z_2) \wedge \\
& \mathsf{bond}(z_2, z_3) \wedge \ \mathbf{not}\ \mathsf{g_h}(z_1) \wedge \\
& \mathbf{not}\ \mathsf{g_h}(z_2) \wedge \ \mathbf{not}\ \mathsf{g_h}(z_3) \rightarrow \mathsf{organicHydroxy}(x) \wedge \mathsf{r_h}(x)
\end{aligned}
$$

$$
\begin{aligned}
r_2 : \quad \mathsf{organicHydroxy}(x) \wedge \ \mathbf{not}\ \mathsf{r_h}(x) \rightarrow\ & \exists_{i=1}^{3} y_i .\ \wedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \\
& \wedge \mathsf{c}(y_1) \wedge \mathsf{o}(y_2) \wedge \mathsf{h}(y_3) \wedge \\
& \mathsf{bond}(y_1, y_2) \wedge \mathsf{bond}(y_2, y_3) \\
& \wedge \mathsf{g_h}(y_1) \wedge \mathsf{g_h}(y_2) \wedge \mathsf{g_h}(y_3)
\end{aligned}
$$

$r_1 \overset{-}{\rightharpoonup} r_2$ but $r_2 \not\overset{-}{\rightharpoonup} r_1$

- Polynomial time to check

R-STRATIFICATION

- A program P is R-stratified if there is a partition $P_1, \ldots, P_n$ of P such that for P_i, P_j and rules $r_1 \in P_i$ and $r_2 \in P_j$, we have:

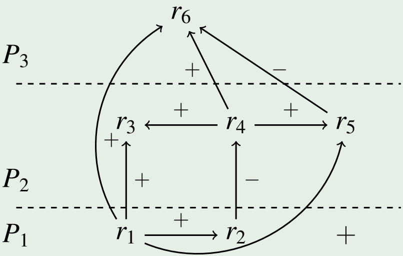
$$\text{if } r_1 \xrightarrow{+} r_2 \text{ then } i \leq j \qquad \text{and} \qquad \text{if } r_1 \xrightarrow{-} r_2 \text{ then } i < j.$$

R-STRATIFICATION

- A program P is R-stratified if there is a partition $P_1, \ldots, P_n$ of P such that for P_i, P_j and rules $r_1 \in P_i$ and $r_2 \in P_j$, we have:

  if $r_1 \xrightarrow{+} r_2$ then $i \leq j$ and if $r_1 \xrightarrow{-} r_2$ then $i < j$.

EXAMPLE

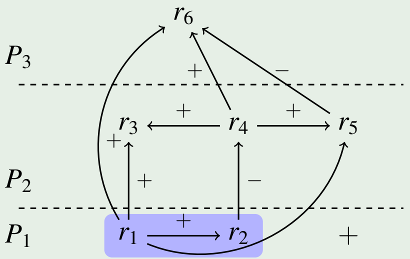


R-STRATIFICATION

- A program P is R-stratified if there is a partition $P_1, \ldots, P_n$ of P such that for P_i, P_j and rules $r_1 \in P_i$ and $r_2 \in P_j$, we have:

$$\text{if } r_1 \xrightarrow{+} r_2 \text{ then } i \leq j \qquad \text{and} \qquad \text{if } r_1 \xrightarrow{-} r_2 \text{ then } i < j.$$

EXAMPLE



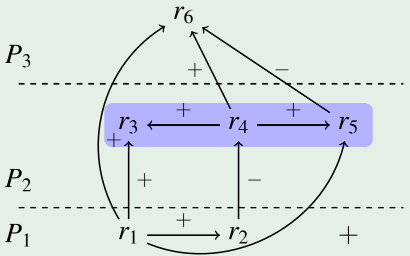
$+$ $\quad S^1_{\vec{\mathrm{P}}} = T_{P_1}(F)$

R-STRATIFICATION

- A program P is R-stratified if there is a partition $P_1, \ldots, P_n$ of P such that for P_i, P_j and rules $r_1 \in P_i$ and $r_2 \in P_j$, we have:

$$\text{if } r_1 \xrightarrow{+} r_2 \text{ then } i \leq j \qquad \text{and} \qquad \text{if } r_1 \xrightarrow{-} r_2 \text{ then } i < j.$$

EXAMPLE



$$S^2_{\vec{\mathsf{P}}} = T_{P_2}(S^1_{\vec{\mathsf{P}}})$$

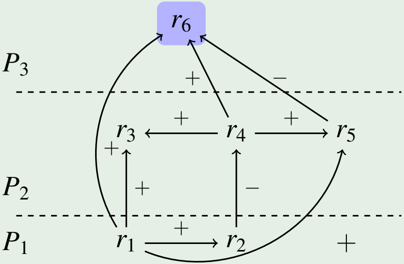
$$S^1_{\vec{\mathsf{P}}} = T_{P_1}(F)$$

R-STRATIFICATION

- A program P is R-stratified if there is a partition $P_1, \ldots, P_n$ of P such that for P_i, P_j and rules $r_1 \in P_i$ and $r_2 \in P_j$, we have:

$$\text{if } r_1 \xrightarrow{+} r_2 \text{ then } i \leq j \qquad \text{and} \qquad \text{if } r_1 \xrightarrow{-} r_2 \text{ then } i < j.$$

EXAMPLE



$$S^3_{\vec{\mathsf{P}}} = T_{P_3}(S^2_{\vec{\mathsf{P}}})$$

$$S^2_{\vec{\mathsf{P}}} = T_{P_2}(S^1_{\vec{\mathsf{P}}})$$

$$S^1_{\vec{\mathsf{P}}} = T_{P_1}(F)$$

R-STRATIFICATION

- A program P is R-stratified if there is a partition $P_1, \ldots, P_n$ of P such that for P_i, P_j and rules $r_1 \in P_i$ and $r_2 \in P_j$, we have:

  if $r_1 \xrightarrow{+} r_2$ then $i \leq j$ and if $r_1 \xrightarrow{-} r_2$ then $i < j$.

  - Strictly extends 'classical' stratification
    $\rightsquigarrow$ ensures stable model uniqueness
  - coNP-complete to check

R-STRATIFICATION

- A program P is R-stratified if there is a partition $P_1, \ldots, P_n$ of P such that for P_i, P_j and rules $r_1 \in P_i$ and $r_2 \in P_j$, we have:

  if $r_1 \xrightarrow{+} r_2$ then $i \leq j$ and if $r_1 \xrightarrow{-} r_2$ then $i < j$.

  - Strictly extends ‘classical’ stratification
    $\rightsquigarrow$ ensures stable model uniqueness
  - coNP-complete to check

- Fact entailment for R-acyclic, R-stratified programs
  - Stable models bounded in size (double exp),
    and at most one stable model

R-STRATIFICATION

- A program P is R-stratified if there is a partition $P_1, \ldots, P_n$ of P such that for P_i, P_j and rules $r_1 \in P_i$ and $r_2 \in P_j$, we have:

  if $r_1 \xrightarrow{+} r_2$ then $i \leq j$ and if $r_1 \xrightarrow{-} r_2$ then $i < j$.

  - Strictly extends ‘classical’ stratification
    $\rightsquigarrow$ ensures stable model uniqueness
  - coNP-complete to check

- Fact entailment for R-acyclic, R-stratified programs
  - Stable models bounded in size (double exp), and at most one stable model
  - 2ExpTime-complete w.r.t. program complexity

R-STRATIFICATION

- A program P is R-stratified if there is a partition $P_1, \ldots, P_n$ of P such that for P_i, P_j and rules $r_1 \in P_i$ and $r_2 \in P_j$, we have:

  if $r_1 \xrightarrow{+} r_2$ then $i \leq j$ and if $r_1 \xrightarrow{-} r_2$ then $i < j$.

  - Strictly extends 'classical' stratification
    $\rightsquigarrow$ ensures stable model uniqueness
  - coNP-complete to check

- Fact entailment for R-acyclic, R-stratified programs
  - Stable models bounded in size (double exp), and at most one stable model
  - 2ExpTime-complete w.r.t. program complexity
  - PTime-complete w.r.t. data complexity

Reliances under Constraints

- Restrict input sets of facts to relax R-acyclicity and R-stratification using constraints

RELIANCES UNDER CONSTRAINTS

- Restrict input sets of facts to relax R-acyclicity and R-stratification using constraints

EXAMPLE

$$
\begin{aligned}
r_1 :& \quad \mathsf{mol}(\mathsf{x}) \wedge \mathsf{hasAtom}(\mathsf{x}, \mathsf{z}) \wedge \mathsf{c}(\mathsf{z}) \rightarrow \mathsf{organic}(\mathsf{x}) \\
r_2 :& \quad \mathsf{mol}(\mathsf{x}) \wedge \ \mathbf{not}\ \mathsf{organic}(\mathsf{x}) \rightarrow \mathsf{inorganic}(\mathsf{x}) \\
r_3 :& \quad \mathsf{inorganic}(\mathsf{x}) \rightarrow \mathsf{mol}(\mathsf{x}) \wedge \mathsf{geoOrigin}(\mathsf{x})
\end{aligned}
$$

Reliances under Constraints

- Restrict input sets of facts to relax R-acyclicity and R-stratification using constraints

Example

$$r_1 : \quad \mathsf{mol}(x) \land \mathsf{hasAtom}(x, z) \land \mathsf{c}(z) \rightarrow \mathsf{organic}(x)$$

$$r_2 : \quad \mathsf{mol}(x) \land \ \textbf{not}\ \mathsf{organic}(x) \rightarrow \mathsf{inorganic}(x)$$

$$r_3 : \quad \mathsf{inorganic}(x) \rightarrow \mathsf{mol}(x) \land \mathsf{geoOrigin}(x)$$

$$r_1 \xrightarrow{-} r_2 \xrightarrow{+} r_3 \xrightarrow{+} r_1$$

Reliances under Constraints

- Restrict input sets of facts to relax R-acyclicity and R-stratification using constraints

Example

$$r_1: \quad \mathsf{mol}(\mathsf{x}) \wedge \mathsf{hasAtom}(\mathsf{x},\mathsf{z}) \wedge \mathsf{c}(\mathsf{z}) \rightarrow \mathsf{organic}(\mathsf{x})$$

$$r_2: \quad \mathsf{mol}(\mathsf{x}) \wedge \textbf{not}\ \mathsf{organic}(\mathsf{x}) \rightarrow \mathsf{inorganic}(\mathsf{x})$$

$$r_3: \quad \mathsf{inorganic}(\mathsf{x}) \rightarrow \mathsf{mol}(\mathsf{x}) \wedge \mathsf{geoOrigin}(\mathsf{x})$$

$$\mathsf{C} = \{\mathsf{inorganic}(\mathsf{x}) \wedge \mathsf{hasAtom}(\mathsf{x},\mathsf{z}) \wedge \mathsf{c}(\mathsf{z}) \rightarrow \bot\}$$

$$r_1 \xrightarrow{-} r_2 \xrightarrow{+} r_3 \xrightarrow{+} r_1$$

Reliances under Constraints

- Restrict input sets of facts to relax R-acyclicity and R-stratification using constraints

Example

$$r_1: \quad \mathsf{mol}(\mathsf{x}) \land \mathsf{hasAtom}(\mathsf{x},\mathsf{z}) \land \mathsf{c}(\mathsf{z}) \to \mathsf{organic}(\mathsf{x})$$

$$r_2: \quad \mathsf{mol}(\mathsf{x}) \land \ \textbf{not}\ \mathsf{organic}(\mathsf{x}) \to \mathsf{inorganic}(\mathsf{x})$$

$$r_3: \quad \mathsf{inorganic}(\mathsf{x}) \to \mathsf{mol}(\mathsf{x}) \land \mathsf{geoOrigin}(\mathsf{x})$$

$$\mathsf{C} = \{\mathsf{inorganic}(\mathsf{x}) \land \mathsf{hasAtom}(\mathsf{x},\mathsf{z}) \land \mathsf{c}(\mathsf{z}) \to \bot\}$$

$$r_1 \xrightarrow{-} r_2 \xrightarrow{+} r_3 \xrightarrow{+} r_1 \qquad \text{but} \qquad r_3 \not\xrightarrow{+}_{\mathsf{C}} r_1$$

Reliances under Constraints

- Restrict input sets of facts to relax R-acyclicity and R-stratification using constraints

Example

$$r_1 : \quad \mathsf{mol}(x) \land \mathsf{hasAtom}(x, z) \land \mathsf{c}(z) \to \mathsf{organic}(x)$$

$$r_2 : \quad \mathsf{mol}(x) \land \ \mathbf{not}\ \mathsf{organic}(x) \to \mathsf{inorganic}(x)$$

$$r_3 : \quad \mathsf{inorganic}(x) \to \mathsf{mol}(x) \land \mathsf{geoOrigin}(x)$$

$$\mathsf{C} = \{\mathsf{inorganic}(x) \land \mathsf{hasAtom}(x, z) \land \mathsf{c}(z) \to \bot\}$$

$$r_1 \xrightarrow{-} r_2 \xrightarrow{+} r_3 \xrightarrow{+} r_1 \qquad \text{but} \qquad r_3 \not\xrightarrow{+}_{\mathsf{C}} r_1$$

- Slightly more complex to check:

Positive reliance	Negative reliance	R-acyclicity/R-stratification
Σ_2^P-complete	in Δ_2^P	Π_2^P-complete

$\rightsquigarrow$ Σ_2^P-hardness follows from satisfiability of a QBF $\exists\vec{p}.\forall\vec{q}.\varphi$

Experimental setup

- Chemical Entities of Biological Interest

  ChEBI
  The database and ontology of Chemical Entities of Biological Interest

  - Reference terminology adopted for chemical annotation by major bio-ontologies
  - ~20,000 molecule and ~8,000 chemical class descriptions
  - ChEBI taxonomy manually curated

EXPERIMENTAL SETUP

- Chemical Entities of Biological Interest

  ChEBI
  The database and ontology of Chemical Entities of Biological Interest

  - Reference terminology adopted for chemical annotation by major bio-ontologies
  - ~20,000 molecule and ~8,000 chemical class descriptions
  - ChEBI taxonomy manually curated
- Our knowledge base consisted of rules derived from ChEBI that represented

Experimental setup

- Chemical Entities of Biological Interest

  ChEBI
  The database and ontology of Chemical Entities of Biological Interest

  - Reference terminology adopted for chemical annotation by major bio-ontologies
  - ~20,000 molecule and ~8,000 chemical class descriptions
  - ChEBI taxonomy manually curated
- Our knowledge base consisted of rules derived from ChEBI that represented
  - 500 molecules

Example

$$\mathrm{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i . \wedge_{i=1}^{6} \mathrm{hasAtom}(x, y_i) \wedge \ldots \wedge \mathrm{bond}(y_2, y_6)$$

Experimental setup

- Chemical Entities of Biological Interest

ChEBI
The database and ontology of Chemical Entities of Biological Interest

  - Reference terminology adopted for chemical annotation by major bio-ontologies
  - ~20,000 molecule and ~8,000 chemical class descriptions
  - ChEBI taxonomy manually curated

- Our knowledge base consisted of rules derived from ChEBI that represented
  - 500 molecules
  - 30 molecular part descriptions

Example

$$
\begin{aligned}
&\wedge_{i=1}^{3} \mathsf{hasAtom}(x, z_i) \wedge \ldots \wedge \\
&\mathsf{bond}(z_2, z_3) \wedge \ \mathbf{not}\ g_h(z_1) \\
&\wedge\ \mathbf{not}\ g_h(z_2) \wedge\ \mathbf{not}\ g_h(z_3) \rightarrow \mathsf{organicHydroxy}(x) \wedge r_h(x)
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{organicHydroxy}(x) \wedge\ \mathbf{not}\ r_h(x) \rightarrow \exists_{i=1}^{3} y_i.\ \wedge_{i=1}^{3} \mathsf{hasAtom}(x, y_i) \wedge \ldots \\
\wedge\ \mathsf{bond}(y_2, y_3) \wedge \wedge_{i=1}^{3} g_h(y_i)
\end{aligned}
$$

Experimental setup

- Chemical Entities of Biological Interest

  ChEBI
  The database and ontology of Chemical Entities of Biological Interest

  - Reference terminology adopted for chemical annotation by major bio-ontologies
  - ~20,000 molecule and ~8,000 chemical class descriptions
  - ChEBI taxonomy manually curated
- Our knowledge base consisted of rules derived from ChEBI that represented
  - 500 molecules
  - 30 molecular part descriptions
  - 50 chemical class descriptions

Example

$$\mathsf{hasAtom}(\mathsf{x},\mathsf{z}) \wedge \mathsf{o}(\mathsf{z}) \rightarrow \mathsf{hasOxygen}(\mathsf{x})$$

Experimental setup

- Chemical Entities of Biological Interest

  ChEBI
  The database and ontology of Chemical Entities of Biological Interest

  - Reference terminology adopted for chemical annotation by major bio-ontologies
  - ~20,000 molecule and ~8,000 chemical class descriptions
  - ChEBI taxonomy manually curated

- Our knowledge base consisted of rules derived from ChEBI that represented
  - 500 molecules
  - 30 molecular part descriptions
  - 50 chemical class descriptions

  ⇝ 78,957 rules in total (R-stratified and R-acyclic)

Experimental setup

- Chemical Entities of Biological Interest

  ChEBI
  The database and ontology of Chemical Entities of Biological Interest

  - Reference terminology adopted for chemical annotation by major bio-ontologies
  - ~20,000 molecule and ~8,000 chemical class descriptions
  - ChEBI taxonomy manually curated

- Our knowledge base consisted of rules derived from ChEBI that represented
  - 500 molecules
  - 30 molecular part descriptions
  - 50 chemical class descriptions

⇝ 78,957 rules in total (R-stratified and R-acyclic)

DLV

- Used DLV for stable model computation

EMPIRICAL RESULTS

- First attempt to compute the stable model of the overall program P failed (no result after 600 secs)

EMPIRICAL RESULTS

- First attempt to compute the stable model of the overall program P failed (no result after 600 secs)
- Second attempt exploited partition of the program into two rule sets according to R-stratification

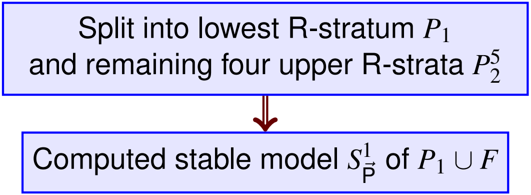
EMPIRICAL RESULTS

- First attempt to compute the stable model of the overall program P failed (no result after 600 secs)
- Second attempt exploited partition of the program into two rule sets according to R-stratification

Split into lowest R-stratum P_1
and remaining four upper R-strata P_2^5

EMPIRICAL RESULTS

- First attempt to compute the stable model of the overall program P failed (no result after 600 secs)
- Second attempt exploited partition of the program into two rule sets according to R-stratification

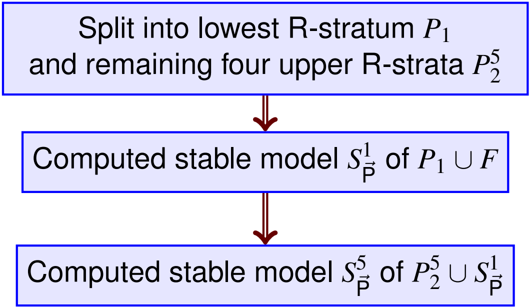


Split into lowest R-stratum P_1
and remaining four upper R-strata P_2^5

Computed stable model $S^1_{\vec{\mathsf{P}}}$ of $P_1 \cup F$

Empirical results

- First attempt to compute the stable model of the overall program P failed (no result after 600 secs)
- Second attempt exploited partition of the program into two rule sets according to R-stratification



Split into lowest R-stratum P_1
and remaining four upper R-strata P_2^5

Computed stable model $S^1_{\vec{\mathrm{P}}}$ of $P_1 \cup F$

Computed stable model $S^5_{\vec{\mathrm{P}}}$ of $P_2^5 \cup S^1_{\vec{\mathrm{P}}}$

EMPIRICAL RESULTS

- First attempt to compute the stable model of the overall program P failed (no result after 600 secs)
- Second attempt exploited partition of the program into two rule sets according to R-stratification
- Computed 8,639 subclass relations in 13.5 secs

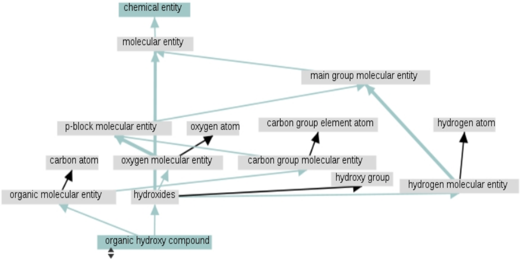
Empirical results

- First attempt to compute the stable model of the overall program P failed (no result after 600 secs)
- Second attempt exploited partition of the program into two rule sets according to R-stratification
- Computed 8,639 subclass relations in 13.5 secs
- Revealed missing subsumptions from the ChEBI ontology

EMPIRICAL RESULTS

- First attempt to compute the stable model of the overall program P failed (no result after 600 secs)
- Second attempt exploited partition of the program into two rule sets according to R-stratification
- Computed 8,639 subclass relations in 13.5 secs
- Revealed missing subsumptions from the ChEBI ontology

E.g. organicHydroxy $\sqsubseteq$ organoOxygenCompound ✓



✗

Conclusions

- R-acyclicity and R-stratification conditions (coNP-complete to check)

Conclusions

- R-acyclicity and R-stratification conditions (coNP-complete to check)

Fact entailment	Program comp.	Data comp.
R-acyclic	coN2ExpTime-complete	coNP-complete
R-acyclic+R-stratified	2ExpTime-complete	PTime-complete

CONCLUSIONS

- R-acyclicity and R-stratification conditions (coNP-complete to check)

Fact entailment	Program comp.	Data comp.
R-acyclic	coN2ExpTime-complete	coNP-complete
R-acyclic+R-stratified	2ExpTime-complete	PTime-complete

- Generalise with constraints (Π_2^P-complete to check)

CONCLUSIONS

- R-acyclicity and R-stratification conditions (coNP-complete to check)

Fact entailment	Program comp.	Data comp.
R-acyclic	coN2ExpTime-complete	coNP-complete
R-acyclic+R-stratified	2ExpTime-complete	PTime-complete

- Generalise with constraints (Π_2^P-complete to check)
- Performance gains in DLV & new subsumptions in ChEBI

Conclusions

- R-acyclicity and R-stratification conditions (coNP-complete to check)

Fact entailment	Program comp.	Data comp.
R-acyclic	coN2ExpTime-complete	coNP-complete
R-acyclic+R-stratified	2ExpTime-complete	PTime-complete

- Generalise with constraints (Π_2^P-complete to check)
- Performance gains in DLV & new subsumptions in ChEBI
- Future directions:
  - More general notions of 'rule' + equality in rule heads [LPNMR'13]

Conclusions

- R-acyclicity and R-stratification conditions (coNP-complete to check)

Fact entailment	Program comp.	Data comp.
R-acyclic	coN2ExpTime-complete	coNP-complete
R-acyclic+R-stratified	2ExpTime-complete	PTime-complete

- Generalise with constraints (Π_2^P-complete to check)
- Performance gains in DLV & new subsumptions in ChEBI

- Future directions:
  - More general notions of ‘rule’ + equality in rule heads [LPNMR’13]
  - Compare performance with other ASP solvers [chemical classification problem, ASPCOMP’13]

Conclusions

- R-acyclicity and R-stratification conditions (coNP-complete to check)

Fact entailment	Program comp.	Data comp.
R-acyclic	coN2ExpTime-complete	coNP-complete
R-acyclic+R-stratified	2ExpTime-complete	PTime-complete

- Generalise with constraints (Π_2^P-complete to check)
- Performance gains in DLV & new subsumptions in ChEBI
- Future directions:
  - More general notions of 'rule' + equality in rule heads [LPNMR'13]
  - Compare performance with other ASP solvers [chemical classification problem, ASPCOMP'13]

Thank you! Questions?!?