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

- Not marked for its ability to model cyclic structures
THE OWLs ARE NOT WHAT THEY SEEM

- OWL widely used for authoring biomedical ontologies

- 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
- 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

\[ B_1 \land \ldots \land B_n \land \text{not } B_{n+1} \land \ldots \land \text{not } B_m \rightarrow \exists y. H_1 \land \ldots \land H_k \]
**Nonmonotonic Existential Rules**

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

\[
B_1 \land \ldots \land B_n \land \text{not } B_{n+1} \land \ldots \land \text{not } B_m \rightarrow \exists y. H_1 \land \ldots \land H_k
\]

- Interpreted under **stable model semantics**
Nonmonotonic Existential Rules

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

  \[ B_1 \land \ldots \land B_n \land \text{not } B_{n+1} \land \ldots \land \text{not } B_m \rightarrow \exists y. H_1 \land \ldots \land H_k \]

- 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

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

- 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

\[ B_1 \land \ldots \land B_n \land \text{not } B_{n+1} \land \ldots \land \text{not } B_m \rightarrow \exists y. H_1 \land \ldots \land H_k \]

- Interpreted under **stable model semantics**

- Good for representing **non-tree-shaped** structures

  - Existentials allow us to infer new structures
  - Nonmonotonicity adds **extra expressivity** in modelling
Nonmonotonic Existential Rules

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

\[ B_1 \land \ldots \land B_n \land \text{not } B_{n+1} \land \ldots \land \text{not } B_m \rightarrow \exists y. H_1 \land \ldots \land H_k \]

- Interpreted under stable model semantics

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

\[
\begin{align*}
O & - H \\
\mid & \\
H & - C - H \\
\mid & \\
H & \\
\end{align*}
\]

Methanol molecule
Classification of Structured Objects I

\[
\begin{align*}
\text{O} & \rightarrow \text{H} \\
\text{H} & \rightarrow \text{C} \rightarrow \text{H} \\
\text{H} & \rightarrow \text{H}
\end{align*}
\]

Methanol molecule

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i. \wedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge \text{c}(y_1) \wedge \text{o}(y_2) \wedge \wedge_{i=3}^{6} \text{h}(y_i) \wedge \wedge_{i=2}^{5} \text{bond}(y_1, y_i) \wedge \text{bond}(y_2, y_6)
\]
**Classification of Structured Objects I**

Methanol molecule

\[
methanol(x) \rightarrow \exists_{i=1}^{6} y_i. \land_{i=1}^{6} hasAtom(x, y_i) \land c(y_1) \land o(y_2) \land \land_{i=3}^{6} h(y_i) \land \land_{i=2}^{5} bond(y_1, y_i) \land bond(y_2, y_6)
\]
**Classification of Structured Objects I**

Methanol molecule

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \wedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge c(y_1) \wedge o(y_2) \wedge \wedge_{i=3}^{6} h(y_i) \wedge \wedge_{i=2}^{5} \text{bond}(y_1, y_i) \wedge \text{bond}(y_2, y_6) \wedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge c(z_1) \wedge o(z_2) \wedge h(z_3) \wedge \text{bond}(z_1, z_2) \wedge \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)
\]
**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

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

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

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

\[
\text{methanol} \sqsubseteq \text{organicHydroxy}
\]
Classification of Structured Objects I

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i. \wedge_{i=1}^{6} \text{hasAtom}(x, y_i) \wedge c(y_1) \wedge o(y_2) \wedge \\
\wedge_{i=3}^{6} h(y_i) \wedge \wedge_{i=2}^{5} \text{bond}(y_1, y_i) \wedge \text{bond}(y_2, y_6) \\
\wedge_{i=1}^{3} \text{hasAtom}(x, z_i) \wedge c(z_1) \wedge o(z_2) \wedge \\
h(z_3) \wedge \text{bond}(z_1, z_2) \wedge \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x) \\
\text{hasAtom}(x, z) \wedge o(z) \rightarrow \text{hasOxygen}(x)
\]

\text{methanol} \sqsubseteq \text{organicHydroxy} \checkmark \quad \text{methanol} \sqsubseteq \text{hasOxygen} \checkmark
Organic hydroxy group
Classification of Structured Objects II

O – H

· · · C · · ·

Organic hydroxy group

$$\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_{i} \cdot \land_{i=1}^{3} \text{hasAtom}(x, y_{i}) \land c(y_{1})$$

$$\land o(y_{2}) \land h(y_{3}) \land \text{bond}(y_{1}, y_{2})$$

$$\land \text{bond}(y_{2}, y_{3})$$
Classification of Structured Objects II

Organic hydroxy group

\[
\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i. \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land c(y_1) \\
\land o(y_2) \land h(y_3) \land \text{bond}(y_1, y_2) \\
\land \text{bond}(y_2, y_3)
\]
Classification of Structured Objects II

**Organic hydroxy group**

\[ \text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i. \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land c(y_1) \land o(y_2) \land h(y_3) \land \text{bond}(y_1, y_2) \land \text{bond}(y_2, y_3) \]

\[ \text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x) \]
Classification of Structured Objects II

\[
\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^3 y_i \land_{i=1}^3 \text{hasAtom}(x, y_i) \land c(y_1) \\
\land o(y_2) \land h(y_3) \land \text{bond}(y_1, y_2) \\
\land \text{bond}(y_2, y_3)
\]

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

\[
\text{organicHydroxy} \sqsubseteq \text{hasOxygen} \checkmark
\]
**Incorrect Modelling**

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \cdot \land_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_6)
\]

\[
\land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)
\]

\[
\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \cdot \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_3)
\]

\[
\text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x)
\]
**Incorrect Modelling**

\[ \text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \cdot \land_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_6) \]

\[ \land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x) \]

\[ \text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \cdot \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_3) \]

\[ \text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x) \]
**Incorrect Modelling**

methanol \sqsubseteq \text{organicHydroxy} \checkmark

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \cdot \bigwedge_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_6) \\
\bigwedge_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)
\]

\[
\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \cdot \bigwedge_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_3)
\]

\[
\text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x)
\]
**Incorrect Modelling**

![Diagram showing incorrect modelling of methanol](image)

**methanol** $\subseteq$ **hasOxygen** ✔

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \cdot \bigwedge_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \land \\
\bigwedge_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \\
\text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)
\]

\[
\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \cdot \bigwedge_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \land \\
\text{bond}(y_2, y_3)
\]

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

---

**methanol** $\subseteq$ **hasOxygen** ✔
**Incorrect Modelling**

\[ \text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \land \exists_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_6) \]

\[ \land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x) \]

\[ \text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \land \exists_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_3) \]

\[ \text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x) \]
**Incorrect Modelling**

\[ \text{methanol} \rightarrow \exists_{i=1}^{6} y_i \land \exists_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_6) \]

\[ \land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x) \]

\[ \text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \land \exists_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_3) \]

\[ \text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x) \]
**Incorrect Modelling**

\[
\begin{align*}
&g_2(m) \quad g_3(m) \\
&f_2(m) \quad f_3(m) \\
&m \quad f_6(m) \\
&f_1(m) \quad f_4(m) \\
&f_5(m) \\
&g_2(h) \quad g_3(h) \\
&g_1(h)
\end{align*}
\]

**methanol**

**organicHydroxy**

**hasOxygen**

**methanol** $\sqsubseteq$ **hasOneCarbon** $\times$

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \land \land_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \\
\land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \\
\text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)
\]

\[
\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \land \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \\
\land \text{bond}(y_2, y_3)
\]

\[
\text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x)
\]
Repair With Auxiliary Predicates

\[
\begin{align*}
\text{methanol}(x) & \rightarrow \exists_{i=1}^{6} y_i. \land_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \\
\land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \\
\text{bond}(z_2, z_3) \land \text{not } g_h(z_1) \\
\land \text{not } g_h(z_2) \land \text{not } g_h(z_3) & \rightarrow \text{organicHydroxy}(x) \land r_h(x) \\
\text{organicHydroxy}(x) \land \text{not } r_h(x) & \rightarrow \exists_{i=1}^{3} y_i. \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \\
\land \text{bond}(y_2, y_3) \land \land_{i=1}^{3} g_h(y_i) \\
\text{hasAtom}(x, z) \land o(z) & \rightarrow \text{hasOxygen}(x)
\end{align*}
\]
**REPAIR WITH AUXILIARY PREDICATES**

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \land \exists_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \\
\land \exists_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \\
\text{bond}(z_2, z_3) \land \text{not } g_h(z_1) \\
\land \text{not } g_h(z_2) \land \text{not } g_h(z_3) \rightarrow \text{organicHydroxy}(x) \land r_h(x) \\
\text{organicHydroxy}(x) \land \text{not } r_h(x) \rightarrow \exists_{i=1}^{3} y_i \land \exists_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \\
\land \text{bond}(y_2, y_3) \land \land_{i=1}^{3} g_h(y_i) \\
\text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x)
\]
REPAIR WITH AUXILIARY PREDICATES

methanol \subseteq \text{hasOneCarbon} \checkmark

\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i. \land_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \\
\land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \\
\text{bond}(z_2, z_3) \land \neg \text{gh}(z_1) \\
\land \neg \text{gh}(z_2) \land \neg \text{gh}(z_3) \rightarrow \text{organicHydroxy}(x) \land \text{rh}(x)

\text{organicHydroxy}(x) \land \neg \text{rh}(x) \rightarrow \exists_{i=1}^{3} y_i. \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \\
\land \text{bond}(y_2, y_3) \land \land_{i=1}^{3} \text{gh}(y_i)

\text{hasAtom}(x, z) \land \text{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

---

![Diagram](image-url)
WHAT'S OUR PROBLEM?

methanol
hasOxygen

Repaired program not stratified

methanol(x) → ∃_{i=1}^{6} y_i. \land_{i=1}^{6} hasAtom(x, y_i) \land ... \land hasAtom(x, z_i) \land ... \land bond(z_2, z_3) \land not g_h(z_1) \land not g_h(z_2) \land not g_h(z_3) → organicHydroxy(x) \land r_h(x)

organicHydroxy(x) \land not r_h(x) → ∃_{i=1}^{3} y_i. \land_{i=1}^{3} hasAtom(x, y_i) \land ... \land bond(y_2, y_3) \land \land_{i=1}^{3} g_h(y_i)

hasAtom(x, z) \land o(z) → hasOxygen(x)
**What’s Our Problem?**

methanol \( (m) \)  
organicHydroxy  
hasOxygen  

Repaired program not stratified

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \land \exists_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \\
\land \text{bond}(y_2, y_6) \\
\land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \\
\text{bond}(z_2, z_3) \land \text{not } g_h(z_1) \land \text{not } g_h(z_2) \land \text{not } g_h(z_3) \rightarrow \text{organicHydroxy}(x) \land r_h(x)
\]

\[
\text{organicHydroxy}(x) \land \text{not } r_h(x) \rightarrow \exists_{i=1}^{3} y_i \land \exists_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \\
\land \text{bond}(y_2, y_3) \land \land_{i=1}^{3} g_h(y_i)
\]

\[
\text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x)
\]
WHAT’S OUR PROBLEM?

Repaired program not stratified

\[
\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \land \land_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \land \land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \land \text{bond}(y_2, y_6) \\
\land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \land \text{bond}(z_2, z_3) \land \text{not } g_h(z_1) \\
\land \text{not } g_h(z_2) \land \text{not } g_h(z_3) \rightarrow \text{organicHydroxy}(x) \land r_h(x) \\
\text{organicHydroxy}(x) \land \text{not } r_h(x) \rightarrow \exists_{i=1}^{3} y_i \land \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \land \land \text{bond}(y_2, y_3) \land \land_{i=1}^{3} g_h(y_i) \\
\text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x)
\]
RESULTS OVERVIEW

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

Experiments over ChEBI with DLV
- Performance gains in DLV using R-stratification
- Missing subsumptions from ChEBI ontology
RESULTS OVERVIEW

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

2. Complexity of reasoning

<table>
<thead>
<tr>
<th>Fact entailment</th>
<th>Program comp.</th>
<th>Data comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-acyclic</td>
<td>coN2ExpTime-complete</td>
<td>coNP-complete</td>
</tr>
<tr>
<td>R-acyclic+R-stratified</td>
<td>2ExpTime-complete</td>
<td>PTime-complete</td>
</tr>
</tbody>
</table>

Experiments over ChEBI with DLV

Performance gains in DLV using R-stratification

Missing subsumptions from ChEBI ontology
RESULTS OVERVIEW

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

2. **Complexity of reasoning**

<table>
<thead>
<tr>
<th>Fact entailment</th>
<th>Program comp.</th>
<th>Data comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-acyclic</td>
<td>coN2ExpTime-complete</td>
<td>coNP-complete</td>
</tr>
<tr>
<td>R-acyclic+R-stratified</td>
<td>2ExpTime-complete</td>
<td>PTime-complete</td>
</tr>
</tbody>
</table>

3. **Generalise R-acyclicity and R-stratification with constraints**
   - New conditions **$\Pi_2^P$-complete** to check

Experiments over ChEBI with DLV
- Performance gains in DLV using R-stratification
- Missing subsumptions from ChEBI ontology
RESULTS OVERVIEW

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

2. **Complexity of reasoning**

<table>
<thead>
<tr>
<th>Fact entailment</th>
<th>Program comp.</th>
<th>Data comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-acyclic</td>
<td>coN2ExpTime-complete</td>
<td>coNP-complete</td>
</tr>
<tr>
<td>R-acyclic+R-stratified</td>
<td>2ExpTime-complete</td>
<td>PTime-complete</td>
</tr>
</tbody>
</table>

3. Generalise R-acyclicity and R-stratification with constraints
   - new conditions $\Pi_2^P$-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 $r_1 \Rightarrow r_2$): 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 \( r_1 \leadsto r_2 \)): there is a situation when \( r_1 \) can trigger \( r_2 \) to derive something new.

**Example**

\[
\begin{align*}
  r_1 : & \quad \land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \\
  & \quad c(z_1) \land o(z_2) \land h(z_3) \land \\
  & \quad \text{bond}(z_1, z_2) \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x) \\
  r_2 : & \quad \text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \cdot \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land \\
  & \quad c(y_1) \land o(y_2) \land h(y_3) \land \\
  & \quad \text{bond}(y_1, y_2) \land \text{bond}(y_2, y_3)
\end{align*}
\]
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**

$r_1 : \quad \land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \text{c}(z_1) \land \text{o}(z_2) \land \text{h}(z_3) \land \text{bond}(z_1, z_2) \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$

$r_2 : \quad \text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \cdot \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land \text{c}(y_1) \land \text{o}(y_2) \land \text{h}(y_3) \land \text{bond}(y_1, y_2) \land \text{bond}(y_2, y_3)$

$r_1 \xrightarrow{+} r_2$
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**

$r_1 : \wedge_{i=1}^{3} \text{hasAtom}(x, z_i) \land c(z_1) \land o(z_2) \land h(z_3) \land \text{bond}(z_1, z_2) \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$

$r_2 : \text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \land \exists_{i=1}^{3} \text{hasAtom}(x, y_i) \land c(y_1) \land o(y_2) \land h(y_3) \land \text{bond}(y_1, y_2) \land \text{bond}(y_2, y_3)$

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

**Example**

$r_1 : \begin{align*}
\wedge_{i=1}^{3} & \text{hasAtom}(x, z_i) \land \\
& \text{c}(z_1) \land \text{o}(z_2) \land \text{h}(z_3) \land \\
& \text{bond}(z_1, z_2) \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)
\end{align*}$

$r_2 : \begin{align*}
\text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i \cdot \wedge_{i=1}^{3} & \text{hasAtom}(x, y_i) \land \\
& \text{c}(y_1) \land \text{o}(y_2) \land \text{h}(y_3) \land \\
& \text{bond}(y_1, y_2) \land \text{bond}(y_2, y_3)
\end{align*}$

$r_1 \xrightarrow{\pm} r_2$ but $r_2 \nRightarrow r_1$

**NP-complete to check**

(but only w.r.t. the size of the rules)
Rule $r_2$ positively relies on $r_1$ (written $r_1 \rightarrow r_2$): there is a situation when $r_1$ can trigger $r_2$ to derive something new.

**EXAMPLE**

$$r_1 : \quad \land_{i=1}^{3} \text{hasAtom}(x, z_i) \land$$
$$c(z_1) \land o(z_2) \land h(z_3) \land$$
$$\text{bond}(z_1, z_2) \land \text{bond}(z_2, z_3) \rightarrow \text{organicHydroxy}(x)$$

$$r_2 : \quad \text{organicHydroxy}(x) \rightarrow \exists_{i=1}^{3} y_i . \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land$$
$$c(y_1) \land o(y_2) \land h(y_3) \land$$
$$\text{bond}(y_1, y_2) \land \text{bond}(y_2, y_3)$$

$$r_1 \rightarrow r_2 \quad \text{but} \quad r_2 \nrightarrow r_1$$

**NP-complete to check**

(but only w.r.t. the size of the rules)
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 ≺-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
A program is **R-acyclic**: there is no cycle of positive reliances that involves a rule with an existential

- Checking R-acyclicity is \( \text{coNP} \)-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

- \( \text{coN2ExpTime} \)-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 $r_1 \rightarrow r_2$): 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 $r_1 \rightarrow r_2$): there is a situation when $r_1$ can inhibit the application of $r_2$

**Example**

\[
\begin{align*}
    r_1 & : & \land_{i=1}^3 \text{hasAtom}(x, z_i) \land \text{c}(z_1) \land \\
        & & \land \text{o}(z_2) \land \text{h}(z_3) \land \text{bond}(z_1, z_2) \land \\
        & & \land \text{bond}(z_2, z_3) \land \text{not} \, \text{g}_h(z_1) \land \\
        & & \text{not} \, \text{g}_h(z_2) \land \text{not} \, \text{g}_h(z_3) \rightarrow \text{organicHydroxy}(x) \land r_h(x)
\end{align*}
\]

\[
\begin{align*}
    r_2 & : & \text{organicHydroxy}(x) \land \text{not} \, r_h(x) \rightarrow \exists_{i=1}^3 y_i. \land_{i=1}^3 \text{hasAtom}(x, y_i) \\
        & & \land \text{c}(y_1) \land \text{o}(y_2) \land \text{h}(y_3) \land \text{bond}(y_1, y_2) \land \text{bond}(y_2, y_3) \\
        & & \land \text{g}_h(y_1) \land \text{g}_h(y_2) \land \text{g}_h(y_3)
\end{align*}
\]
NEGATIVE RELIANCES

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

**Example**

\[ r_1 : \bigwedge_{i=1}^3 \text{hasAtom}(x, z_i) \wedge c(z_1) \wedge o(z_2) \wedge h(z_3) \wedge \text{bond}(z_1, z_2) \wedge \text{bond}(z_2, z_3) \land \neg g_h(z_1) \land \neg g_h(z_2) \land \neg g_h(z_3) \rightarrow \text{organicHydroxy}(x) \wedge \neg r_h(x) \]

\[ r_2 : \text{organicHydroxy}(x) \land \neg r_h(x) \rightarrow \exists_{i=1}^3 y_i. \bigwedge_{i=1}^3 \text{hasAtom}(x, y_i) \wedge c(y_1) \wedge o(y_2) \wedge h(y_3) \wedge \text{bond}(y_1, y_2) \wedge \text{bond}(y_2, y_3) \wedge g_h(y_1) \land g_h(y_2) \land g_h(y_3) \]

\[ r_1 \rightarrow r_2 \]
**Negative Reliances**

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

**Example**

\[
\begin{align*}
r_1 : & \quad \wedge_{i=1}^{3} \text{hasAtom}(x, z_i) \land c(z_1) \land o(z_2) \land h(z_3) \land \text{bond}(z_1, z_2) \land \text{bond}(z_2, z_3) \land \neg \text{g}_h(z_1) \land \neg \text{g}_h(z_2) \land \neg \text{g}_h(z_3) \rightarrow \text{organicHydroxy}(x) \land r_h(x) \\
r_2 : & \quad \text{organicHydroxy}(x) \land \neg r_h(x) \rightarrow \exists_{i=1}^{3} y_i. \wedge_{i=1}^{3} \text{hasAtom}(x, y_i) \land c(y_1) \land o(y_2) \land h(y_3) \land \text{bond}(y_1, y_2) \land \text{bond}(y_2, y_3) \land \text{g}_h(y_1) \land \text{g}_h(y_2) \land \text{g}_h(y_3)
\end{align*}
\]

$r_1 \rightarrow r_2$ but $r_2 \not\rightarrow r_1$
NEGATIVE RELIANCES

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

**Example**

$r_1 : \quad \land_{i=1}^3 \text{hasAtom}(x, z_i) \land c(z_1) \land \\
on(z_2) \land h(z_3) \land \text{bond}(z_1, z_2) \land \\
\text{bond}(z_2, z_3) \land \text{not } g_h(z_1) \land \\
\text{not } g_h(z_2) \land \text{not } g_h(z_3) \rightarrow \text{organicHydroxy}(x) \land r_h(x)$

$r_2 : \quad \text{organicHydroxy}(x) \land \text{not } r_h(x) \rightarrow \exists_{i=1}^3 y_i. \land_{i=1}^3 \text{hasAtom}(x, y_i) \land \\
c(y_1) \land o(y_2) \land h(y_3) \land \\
\text{bond}(y_1, y_2) \land \text{bond}(y_2, y_3) \land \\
g_h(y_1) \land g_h(y_2) \land g_h(y_3)$

$r_1 \rightarrow r_2 \quad \text{but} \quad r_2 \nrightarrow r_1$

Polynomial time to check
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 $r_1 \rightleftharpoons r_2$ then $i \leq j$  
  and  
- if $r_1 \leftarrow r_2$ 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 $r_1 \xrightarrow{\pm} r_2$ then $i \leq j$ and if $r_1 \xrightarrow{\rightarrow} r_2$ then $i < j$.

**Example**

![Diagram](image-url)
**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**

\[ S_1^1 = T_{P_1}(F) \]
**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 $i < j$.

**Example**

\[ S_P^2 = T_{P_2}(S_P^1) \quad S_P^1 = T_{P_1}(F) \]
**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 $i < j$.

**Example**

```
S^3_P = T_{P_3}(S^2_P)
S^2_P = T_{P_2}(S^1_P)
S^1_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 $r_1 \leftarrow \leftarrow r_2$ then $i \leq j$ and if $r_1 \leftarrow \rightarrow r_2$ then $i < j$.

- Strictly extends ‘classical’ stratification
- $\leftarrow \rightarrow$ ensures stable model uniqueness
- coNP-complete to check
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 $r_1 \xrightarrow{\pm} r_2$ then $i \leq j$ and if $r_1 \xrightarrow{\leftarrow} r_2$ then $i < j$.

- Strictly extends ‘classical’ stratification
- $\xrightarrow{\leftarrow}$ 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 $r_1 \leftrightarrow r_2$ then $i \leq j$ and if $r_1 \rightarrow r_2$ then $i < j$.

- Strictly extends ‘classical’ stratification
- $\rightarrow$ 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
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 $r_1 \leftarrow\rightarrow r_2$ then $i \leq j$ and if $r_1 \leftarrow\rightarrow r_2$ then $i < j$.

- Strictly extends ‘classical’ stratification
- $\leftarrow\rightarrow$ 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

Example:

\[ r_1: \text{mol}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \text{organic}(x) \]

\[ r_2: \text{mol}(x) \land \text{not organic}(x) \rightarrow \text{inorganic}(x) \]

\[ r_3: \text{inorganic}(x) \rightarrow \text{mol}(x) \land \text{geoOrigin}(x) \]

\[ C = \{ \text{inorganic}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \bot \} \]

\[ r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_1 \]

But \[ r_3 \not\rightarrow r_1 \rightarrow r_2 \]

Slightly more complex to check:
- Positive reliance
- Negative reliance
- R-acyclicity/R-stratification

\[ \Sigma \text{P}^2 \text{-complete} \]

\[ \Pi \text{P}^2 \text{-complete} \]

\[ \Sigma \text{P}^2 \text{-hardness follows from satisfiability of a QBF} \]

\[ \exists \text{p}. \forall \text{q}. \phi \]
Reliances under Constraints

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

**Example**

\[ r_1 : \text{mol}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \text{organic}(x) \]
\[ r_2 : \text{mol}(x) \land \textbf{not} \text{organic}(x) \rightarrow \text{inorganic}(x) \]
\[ r_3 : \text{inorganic}(x) \rightarrow \text{mol}(x) \land \text{geoOrigin}(x) \]
Restrict input sets of facts to relax R-acyclicity and R-stratification using constraints

**Example**

\[ r_1 : \text{mol}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \text{organic}(x) \]
\[ r_2 : \text{mol}(x) \land \text{not} \text{ organic}(x) \rightarrow \text{inorganic}(x) \]
\[ r_3 : \text{inorganic}(x) \rightarrow \text{mol}(x) \land \text{geoOrigin}(x) \]

\[ r_1 \xrightarrow{\text{ref}} r_2 \xrightarrow{\text{add}} r_3 \xrightarrow{\text{ref}} r_1 \]
RELIANCES UNDER CONSTRAINTS

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

**Example**

\[
\begin{align*}
  r_1 : & \quad \text{mol}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \text{organic}(x) \\
  r_2 : & \quad \text{mol}(x) \land \textbf{not} \text{ organic}(x) \rightarrow \text{inorganic}(x) \\
  r_3 : & \quad \text{inorganic}(x) \rightarrow \text{mol}(x) \land \text{geoOrigin}(x) \\
  C = & \{ \text{inorganic}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \bot \} \\
  r_1 & \rightarrow r_2 \xrightarrow{+} r_3 \xrightarrow{+} r_1
\end{align*}
\]
Restrict input sets of facts to relax R-acyclicity and R-stratification using constraints

**Example**

\[
\begin{align*}
 r_1 : & \quad \text{mol}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \text{organic}(x) \\
 r_2 : & \quad \text{mol}(x) \land \neg \text{organic}(x) \rightarrow \text{inorganic}(x) \\
 r_3 : & \quad \text{inorganic}(x) \rightarrow \text{mol}(x) \land \text{geoOrigin}(x)
\end{align*}
\]

\[
C = \{ \text{inorganic}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \bot \}
\]

\[
\begin{align*}
 r_1 & \rightarrow r_2 \rightarrow r_3 \rightarrow r_1 \\
 & \text{but} \quad r_3 \nRightarrow_C r_1
\end{align*}
\]
**Reliances under Constraints**

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

**Example**

- \( r_1 : \) \( \text{mol}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \text{organic}(x) \)
- \( r_2 : \) \( \text{mol}(x) \land \textbf{not} \text{organic}(x) \rightarrow \text{inorganic}(x) \)
- \( r_3 : \) \( \text{inorganic}(x) \rightarrow \text{mol}(x) \land \text{geoOrigin}(x) \)

\[ C = \{ \text{inorganic}(x) \land \text{hasAtom}(x, z) \land c(z) \rightarrow \bot \} \]

\[ r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow r_1 \quad \text{but} \quad r_3 \nrightarrow_C r_1 \]

- Slightly more complex to check:

<table>
<thead>
<tr>
<th>Positive reliance</th>
<th>Negative reliance</th>
<th>R-acyclicity/R-stratification</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Sigma_2^P )-complete</td>
<td>in ( \Delta_2^P )</td>
<td>( \Pi_2^P )-complete</td>
</tr>
</tbody>
</table>

\( \leadsto \Sigma_2^P \)-hardness follows from satisfiability of a QBF \( \exists \vec{p}. \forall \vec{q}. \varphi \)
**Experimental Setup**

- **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

[ChEBI](http://www.ebi.ac.uk/chebi) - The database and ontology of Chemical Entities of Biological Interest
EXPERIMENTAL SETUP

- **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**
  - 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**

$$\text{methanol}(x) \rightarrow \exists_{i=1}^{6} y_i \land \exists_{i=1}^{6} \text{hasAtom}(x, y_i) \land \ldots \land \text{bond}(y_2, y_6)$$
EXPERIMENTAL SETUP

- **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**

\[
\begin{align*}
\land_{i=1}^{3} \text{hasAtom}(x, z_i) \land \ldots \land \\
\text{bond}(z_2, z_3) \land \neg \text{gh}(z_1) \\
\land \neg \text{gh}(z_2) \land \neg \text{gh}(z_3) \rightarrow \text{organicHydroxy}(x) \land r_h(x)
\end{align*}
\]

\[
\text{organicHydroxy}(x) \land \neg r_h(x) \rightarrow \exists_{i=1}^{3} y_i. \land_{i=1}^{3} \text{hasAtom}(x, y_i) \land \ldots \\
\land \text{bond}(y_2, y_3) \land \land_{i=1}^{3} \text{gh}(y_i)
\]
EXPERIMENTAL SETUP

- **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

\[ \text{hasAtom}(x, z) \land o(z) \rightarrow \text{hasOxygen}(x) \]
EXPERIMENTAL SETUP

■ 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**
  - 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
  \(\sim 78,957\) rules in total (R-stratified and R-acyclic)

- 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 $S^1_P$ 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 $S_{P_1}^1$ of $P_1 \cup F$

Computed stable model $S_{P_2}^5$ of $P_2^5 \cup S_{P}^1$
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
  \((\text{coNP}-\text{complete to check})\)
Conclusions

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

<table>
<thead>
<tr>
<th>Fact entailment</th>
<th>Program comp.</th>
<th>Data comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-acyclic</td>
<td>coN2ExpTime-complete</td>
<td>coNP-complete</td>
</tr>
<tr>
<td>R-acyclic+R-stratified</td>
<td>2ExpTime-complete</td>
<td>PTime-complete</td>
</tr>
</tbody>
</table>
Conclusions

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

<table>
<thead>
<tr>
<th>Fact entailment</th>
<th>Program comp.</th>
<th>Data comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-acyclic</td>
<td>coN2ExpTime-complete</td>
<td>coNP-complete</td>
</tr>
<tr>
<td>R-acyclic+R-stratified</td>
<td>2ExpTime-complete</td>
<td>PTime-complete</td>
</tr>
</tbody>
</table>

- Generalise with constraints ($\Pi_2^P$-complete to check)
CONCLUSIONS

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

<table>
<thead>
<tr>
<th>Fact entailment</th>
<th>Program comp.</th>
<th>Data comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-acyclic</td>
<td>coN2ExpTime-complete</td>
<td>coNP-complete</td>
</tr>
<tr>
<td>R-acyclic+R-stratified</td>
<td>2ExpTime-complete</td>
<td>PTime-complete</td>
</tr>
</tbody>
</table>

- Generalise with constraints (Π₂^P-complete to check)
- Performance gains in DLV & new subsumptions in ChEBI
Conclusions

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

<table>
<thead>
<tr>
<th>Fact entailment</th>
<th>Program comp.</th>
<th>Data comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-acyclic</td>
<td>coN2ExpTime-complete</td>
<td>coNP-complete</td>
</tr>
<tr>
<td>R-acyclic+R-stratified</td>
<td>2ExpTime-complete</td>
<td>PTime-complete</td>
</tr>
</tbody>
</table>

- Generalise with constraints ($\Pi_2^P$-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)

<table>
<thead>
<tr>
<th>Fact entailment</th>
<th>Program comp.</th>
<th>Data comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-acyclic</td>
<td>coN2ExpTime-complete</td>
<td>coNP-complete</td>
</tr>
<tr>
<td>R-acyclic+R-stratified</td>
<td>2ExpTime-complete</td>
<td>PTime-complete</td>
</tr>
</tbody>
</table>

- Generalise with constraints (Π₂^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)

<table>
<thead>
<tr>
<th>Fact entailment</th>
<th>Program comp.</th>
<th>Data comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-acyclic</td>
<td>coN2ExpTime-complete</td>
<td>coNP-complete</td>
</tr>
<tr>
<td>R-acyclic+R-stratified</td>
<td>2ExpTime-complete</td>
<td>PTime-complete</td>
</tr>
</tbody>
</table>

- Generalise with constraints ($\Pi_2^P$-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?!?