1 Motivation

2 MFA and MSA

3 Querying Acyclic DL Ontologies

4 Experimental Results
ONTLOGICAL QUERY ANSWERING

- Key reasoning task for DL and rule-based applications

Query $Q$ s.t. $\mathcal{T} \cup \mathcal{A} \models Q$
Ontological Query Answering

- Key reasoning task for DL and rule-based applications
- Answering CQs over DLs $\leadsto$ high computational complexity
Ontological Query Answering

- Key reasoning task for DL and rule-based applications
- Answering CQs over DLs $\leadsto$ high computational complexity
- Materialisation-based paradigm: chase ABox using TBox and evaluate $Q$ in the computed ABox
**EXISTENTIAL RULES**

- Positive, function-free, FOL implications with existentially quantified variables in the head

1. Schema constraints in databases
2. Data transformation rules in data exchange
3. Foundation for Datalog $\pm$ family of KR languages
4. Ubiquitous in Description Logics

Chase termination is undecidable for existential rules

CQ answering is undecidable for existential rules
EXISTENTIAL RULES

- Positive, function-free, FOL implications with existentially quantified variables in the head

**Example**

A(x) → ∃y. R(x, y) ∧ B(y)  
DL-equivalent  ⇝  A ⊑ ∃R.B
Existential Rules

- Positive, function-free, FOL implications with existentially quantified variables in the head

**Example**

\[ A(x) \rightarrow \exists y. R(x, y) \land B(y) \quad \text{DL-equivalent} \quad \leadsto \quad A \sqsubseteq \exists R.B \]

- Existential rules fundamental for several KR formalisms:
**Existential Rules**

- Positive, function-free, FOL implications with existentially quantified variables in the head

**Example**

\[ A(x) \rightarrow \exists y. R(x, y) \land B(y) \quad \text{DL-equivalent} \quad \rightsquigarrow \quad A \sqsubseteq \exists R.B \]

- Existential rules fundamental for several KR formalisms:
  1. Schema constraints in databases
  2. Data transformation rules in data exchange
  3. Foundation for Datalog± family of KR languages
  4. Ubiquitous in Description Logics
Existential Rules

- Positive, function-free, FOL implications with existentially quantified variables in the head

**Example**

\[ A(x) \rightarrow \exists y. R(x, y) \land B(y) \]

DL-equivalent \( \iff \)

\[ A \sqsubseteq \exists R.B \]

- Existential rules fundamental for several KR formalisms:
  1. Schema constraints in databases
  2. Data transformation rules in data exchange
  3. Foundation for Datalog\(\pm\) family of KR languages
  4. Ubiquitous in Description Logics

- Chase termination is undecidable for existential rules
Existential Rules

- Positive, function-free, FOL implications with existentially quantified variables in the head

**Example**

\[ A(x) \rightarrow \exists y. R(x, y) \land B(y) \quad \text{DL-equivalent} \quad \leadsto \quad A \sqsubseteq \exists R.B \]

- Existential rules fundamental for several KR formalisms:
  1. Schema constraints in **databases**
  2. Data transformation rules in **data exchange**
  3. Foundation for **Datalog±** family of KR languages
  4. Ubiquitous in **Description Logics**

- Chase termination is **undecidable** for existential rules
- CQ answering is **undecidable** for existential rules
Tackling Undecidability

1 Identify groups of rules for which query answering is decidable
TACKLING UNDECIDABILITY

Identify groups of rules for which query answering is decidable
- Guarded rules, sticky rules, bounded treewidth sets

Acyclic set of rules
- Guarantees chase termination
- Yields finite materialisation
- No restriction on the shape of rules (unlike guarded rules)

Materialised ABoxes can be stored as databases

Acyclicity conditions might be too restrictive
Tackling Undecidability

1. Identify groups of rules for which query answering is decidable
   - Guarded rules, sticky rules, bounded treewidth sets

Tackling Undecidability

1. Identify groups of rules for which query answering is decidable
   - Guarded rules, sticky rules, bounded treewidth sets


   Acyclic set of rules
TACKLING UNDECIDABILITY

1. Identify groups of rules for which query answering is decidable
   - Guarded rules, sticky rules, bounded treewidth sets


Acyclic set of rules

Guarantees chase termination

- No restriction on the shape of rules (unlike guarded rules)
- Materialised ABoxes can be stored as databases

- Only sets of rules with models of bounded size
- Acyclicity conditions might be too restrictive
**Tackling Undecidability**

1. Identify groups of rules for which query answering is decidable
   - Guarded rules, sticky rules, bounded treewidth sets

2. **Acyclicity** conditions: weak acyclicity [Kolaitis et al., ICDT, 2002], super-weak acyclicity [Marnette, PODS, 2009], joint acyclicity [Krötzsch and Rudolph, IJCAI, 2011],...

   - Acyclic set of rules
     - Guarantees chase termination
     - Yields finite materialisation

---

Plus
- No restriction on the shape of rules (unlike guarded rules)

Minus
- Only sets of rules with models of bounded size
- Acyclicity conditions might be too restrictive
Tackling Undecidability

1. Identify groups of rules for which query answering is decidable
   - Guarded rules, sticky rules, bounded treewidth sets


Acyclic set of rules

- Guarantees chase termination
- Yields finite materialisation

Plus

- No restriction on the shape of rules (unlike guarded rules)
Tackling Undecidability

1. Identify groups of rules for which query answering is decidable
   - Guarded rules, sticky rules, bounded treewidth sets


Acyclic set of rules

Guarantees chase termination

Yields finite materialisation

Plus

I. No restriction on the shape of rules (unlike guarded rules)

II. Materialised ABoxes can be stored as databases
Tackling Undecidability

1. Identify groups of rules for which query answering is decidable
   - Guarded rules, sticky rules, bounded treewidth sets

2. Acyclicity conditions: weak acyclicity [Kolaitis et al., ICDT, 2002], super-weak acyclicity [Marnette, PODS, 2009], joint acyclicity [Krötzsch and Rudolph, IJCAI, 2011], ...

   **Acyclic set of rules**
   
   Guarantees chase termination
   
   Yields finite materialisation

**Plus**

1. No restriction on the shape of rules (unlike guarded rules)
2. Materialised ABoxes can be stored as databases

**Minus**

1. Only sets of rules with models of bounded size
**Tackling Undecidability**

1. Identify groups of rules for which query answering is decidable
   - Guarded rules, sticky rules, bounded treewidth sets

2. **Acyclicity** conditions: weak acyclicity [Kolaitis et al., ICDT, 2002], super-weak acyclicity [Marnette, PODS, 2009], joint acyclicity [Krötzsch and Rudolph, IJCAI, 2011], ...

---

**Acyclic set of rules**

- Guarantees chase termination
- Yields finite materialisation

---

**Plus**

1. No restriction on the shape of rules (unlike guarded rules)
2. Materialised ABoxes can be stored as databases

**Minus**

1. Only sets of rules with models of bounded size
2. Acyclicity conditions might be too restrictive
Materialisation-based Reasoning

- Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-SHOIQ [Ortiz, Rudolph and Simkus, 2011]

Approaches taken:

1. Saturate only non-existential rules (OWL 2 RL)
2. Apply existential rules in a restricted way

Suggestion: materialise ABoxes only over acyclic TBoxes

Always complete
Provably terminating
Materialisation-based Reasoning

- Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-\textit{SHOIQ} [Ortiz, Rudolph and Simkus, 2011]

- For Horn ontologies, consequences can be precomputed, stored and used for query evaluation, e.g. by the RDF repositories Sesame, Jena, OWLIM, DLE-Jena, ...
Materialisation-based Reasoning

- Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-\textit{SHOIQ} [Ortiz, Rudolph and Simkus, 2011]

- For Horn ontologies, consequences can be precomputed, stored and used for query evaluation, e.g. by the RDF repositories Sesame, Jena, OWLIM, DLE-Jena, ...
**Materialisation-based Reasoning**

- Answering CQs over expressive DLs is **expensive**, e.g. EXPTIME-complete for Horn-\textit{SHOIQ} [Ortiz, Rudolph and Simkus, 2011]

- For Horn ontologies, consequences can be **precomputed**, **stored** and **used for query evaluation**, e.g. by the RDF repositories Sesame, Jena, OWLIM, DLE-Jena,…

  **Risk of non-termination**

- Approaches taken:
  
  1. Saturate **only non-existential** rules (OWL 2 RL)
Materialisation-based Reasoning

- Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-\textit{SHOIQ} [Ortiz, Rudolph and Simkus, 2011]

- For Horn ontologies, consequences can be precomputed, stored and used for query evaluation, e.g. by the RDF repositories Sesame, Jena, OWLIM, DLE-Jena,…

Risk of non-termination

- Approaches taken:
  1. Saturate only non-existential rules (OWL 2 RL): can miss answers ✗
**Materialisation-based Reasoning**

- Answering CQs over expressive DLs is **expensive**, e.g. EXPTIME-complete for Horn-\textit{SHOIQ} [Ortiz, Rudolph and Simkus, 2011]

- For Horn ontologies, consequences can be **precomputed**, **stored** and **used for query evaluation**, e.g. by the RDF repositories Sesame, Jena, OWLIM, DLE-Jena, . . .

  **Risk of non-termination**

- Approaches taken:
  1. Saturate only non-existential rules (OWL 2 RL): can miss answers \(\times\)
  2. Apply existential rules **in a restricted way**
Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-$\mathbf{SHOIQ}$ [Ortiz, Rudolph and Simkus, 2011]

For Horn ontologies, consequences can be precomputed, stored and used for query evaluation, e.g. by the RDF repositories Sesame, Jena, OWLIM, DLE-Jena,…

Risk of non-termination

Approaches taken:

1. Saturate only non-existential rules (OWL 2 RL): can miss answers $\times$
2. Apply existential rules in a restricted way: can still miss answers and/or not terminate $\times$
Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-\textit{SHOIQ} [Ortiz, Rudolph and Simkus, 2011].

For Horn ontologies, consequences can be precomputed, stored and used for query evaluation, e.g. by the RDF repositories Sesame, Jena, OWLIM, DLE-Jena,…

Risk of non-termination

Approaches taken:
1. Saturate only non-existential rules (OWL 2 RL): can miss answers  
2. Apply existential rules in a restricted way: can still miss answers and/or not terminate

Suggestion: materialise ABoxes only over acyclic TBoxes
- Always complete  
- Provably terminating
RESULTS OVERVIEW

1. More general acyclicity conditions: MSA and MFA
RESULTS OVERVIEW

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

3. DL query answering under acyclicity conditions
   - Horn-SHIQ in WA: $T \cup A| = F$ is ExpTime-hard
   - Horn-SHIQ in MFA: $T \cup A| = Q$ is PSpace-complete

4. Experimental evaluation on DL ontologies
   - 83% ontologies found acyclic (78% JA)
   - Materialised ABoxes not too large
     - $\Rightarrow \times 5$ bigger on average for ontologies with depth < 5 (= most ontologies)

5. Materialisation-based reasoning beyond OWL 2 RL might be practically feasible
RESULTS OVERVIEW

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-SHIQ</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>
Results Overview

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-SHIQ</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>
3. DL query answering under acyclicity conditions

Materialisation-based reasoning beyond OWL 2 RL might be practically feasible.
RESULTS OVERVIEW

1. More general acyclicitiy conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-$SHIQ$</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicitiy conditions
   - Horn-$SRI\ T$ in WA: $\mathcal{T} \cup \mathcal{A} \models F$ is ExpTime-hard
RESULTS OVERVIEW

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-$SHIQ$</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicity conditions
   - Horn-$SR$I $\mathcal{T}$ in WA: $\mathcal{T} \cup \mathcal{A} \models F$ is ExpTime-hard
   - Horn-$SHIQ$ $\mathcal{T}$ in MFA: $\mathcal{T} \cup \mathcal{A} \models Q$ is PSpace-complete
# Results Overview

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-\textit{SHIQ}</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicity conditions
   - Horn-\textit{SRI} $\mathcal{T}$ in WA: $\mathcal{T} \cup \mathcal{A} \models F$ is ExpTime-hard
   - Horn-\textit{SHIQ} $\mathcal{T}$ in MFA: $\mathcal{T} \cup \mathcal{A} \models Q$ is PSpace-complete

4. Experimental evaluation on DL ontologies
RESULTS OVERVIEW

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-$SHIQ$</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicity conditions
   - Horn-$SRI$ $\mathcal{T}$ in $WA$: $\mathcal{T} \cup \mathcal{A} \models F$ is ExpTime-hard
   - Horn-$SHIQ$ $\mathcal{T}$ in $MFA$: $\mathcal{T} \cup \mathcal{A} \models Q$ is PSpace-complete

4. Experimental evaluation on DL ontologies
   - 83% ontologies found acyclic (78% JA)

Materialisation-based reasoning beyond OWL 2 RL might be practically feasible.
RESULTS OVERVIEW

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-\textit{SHIQ}</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicity conditions
   - Horn-\textit{SRI} $\mathcal{T}$ in WA: $\mathcal{T} \cup A \models F$ is ExpTime-hard
   - Horn-\textit{SHIQ} $\mathcal{T}$ in MFA: $\mathcal{T} \cup A \models Q$ is PSpace-complete

4. Experimental evaluation on DL ontologies
   - 83% ontologies found acyclic (78% JA)
   - materialised ABoxes not too large
RESULTS OVERVIEW

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-\textit{SHIQ}</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicity conditions
   - Horn-\textit{SRI} \( T \) in WA: \( T \cup A \models F \) is ExpTime-hard
   - Horn-\textit{SHIQ} \( T \) in MFA: \( T \cup A \models Q \) is PSpace-complete

4. Experimental evaluation on DL ontologies
   - 83\% ontologies found acyclic (78\% JA)
   - materialised ABoxes not too large \( \sim \times 5 \) bigger on average for ontologies with depth < 5 (= most ontologies)
RESULTS OVERVIEW

1. More general acyclicity conditions: MSA and MFA

2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-$SHIQ$</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicity conditions
   - Horn-$SRI$ $\mathcal{T}$ in WA: $\mathcal{T} \cup \mathcal{A} \models F$ is ExpTime-hard
   - Horn-$SHIQ$ $\mathcal{T}$ in MFA: $\mathcal{T} \cup \mathcal{A} \models Q$ is PSpace-complete

4. Experimental evaluation on DL ontologies
   - 83% ontologies found acyclic (78% JA)
   - materialised ABoxes not too large

Materialisation-based reasoning beyond OWL 2 RL might be practically feasible
1 Motivation

2 MFA and MSA

3 Querying Acyclic DL Ontologies

4 Experimental Results
## Existing Acyclicity Conditions

**Example**

$r_1 : A(u) \rightarrow \exists y_1. R(u, y_1) \land B(y_1)$

$r_2 : B(v) \rightarrow \exists y_2. R(v, y_2) \land C(y_2)$

$r_3 : R(w, z) \land B(z) \rightarrow A(w)$
EXISTINGACYCLICITYCONDITIONS

**Example**

\[r_1 : A(u) \rightarrow \exists y_1. R(u, y_1) \land B(y_1)\]

\[r_2 : B(v) \rightarrow \exists y_2. R(v, y_2) \land C(y_2)\]

\[r_3 : R(w, z) \land B(z) \rightarrow A(w)\]

\[A \equiv \exists R. B\]

\[B \subseteq \exists R. C\]
**Existing Acyclicity Conditions**

**Example**

\[ r_1 : A(u) \rightarrow \exists y_1. R(u, y_1) \land B(y_1) \]

\[ r_2 : B(v) \rightarrow \exists y_2. R(v, y_2) \land C(y_2) \]

\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]
**Existing Acyclicity Conditions**

**Example**

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow \exists y_2 . R(v, y_2) \land C(y_2) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]
**Existing Acyclicity Conditions**

**Example**

\[
\begin{align*}
\ r_1 : & A(u) \rightarrow R(u,f(u)) \land B(f(u)) \\
\ r_2 : & B(v) \rightarrow \exists y_2. R(v,y_2) \land C(y_2) \\
\ r_3 : & R(w,z) \land B(z) \rightarrow A(w)
\end{align*}
\]
**Existing Acyclicity Conditions**

**Example**

\[
\begin{align*}
    r_1 : A(u) & \rightarrow R(u, f(u)) \land B(f(u)) \\
    r_2 : B(v) & \rightarrow R(v, g(v)) \land C(g(v)) \\
    r_3 : R(w, z) \land B(z) & \rightarrow A(w)
\end{align*}
\]
**Existing Acyclicity Conditions**

**Example**

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]
Existing Acyclicity Conditions

Example

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]
**Existing Acyclicity Conditions**

**Example**

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]

\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]

\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]
**Existing Acyclicity Conditions**

**Example**

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]

- **Joint acyclicity**
  - Tracks value generation and propagation to detect cyclic creation of terms
  - Polynomial time to check
Existing Acyclicity Conditions

**Example**

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]

\[
\text{Move}(f(u)) = \{ R|_2, B|_1 \}
\]

- **Joint acyclicity**
  1. Tracks *value generation and propagation* to detect cyclic creation of terms
  2. Polynomial time to check
**Existing Acyclicity Conditions**

**Example**

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]

\[ \text{Move}(f(u)) = \{ R|_2, B|_1, R|_1 \} \]

- **Joint acyclicity**
  
  1. Tracks *value generation and propagation* to detect cyclic creation of terms
  2. Polynomial time to check
Existing Acyclicity Conditions

**Example**

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]

\[
\text{Move}(f(u)) = \{ R|_2, B|_1, R|_1, A|_1 \}
\]

- **Joint acyclicity**
  1. Tracks **value generation and propagation** to detect cyclic creation of terms
  2. Polynomial time to check
**Existing Acyclicity Conditions**

**Example**

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]

\[ \text{Pos}_B(u) = \{A|_1\} \quad \text{Move}(f(u)) = \{R|_2, B|_1, R|_1, A|_1\} \]

- **Joint acyclicity**
  1. Tracks *value generation and propagation* to detect cyclic creation of terms
  2. Polynomial time to check
Existing Acyclicity Conditions

**Example**

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]

\[ \text{Pos}_B(u) = \{A|_1\} \subseteq \text{Move}(f(u)) = \{R|_2, B|_1, R|_1, A|_1\} \]

- **Joint acyclicity**
  - Tracks *value generation and propagation* to detect cyclic creation of terms
  - Polynomial time to check
**Existing Acyclicity Conditions**

### Example

\[ r_1 : A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ r_2 : B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ r_3 : R(w, z) \land B(z) \rightarrow A(w) \]

\[ \text{Pos}_{B(u)} = \{A \mid_1\} \subseteq \text{Move}(f(u)) = \{R \mid_2, B \mid_1, R \mid_1, A \mid_1\} \]

- Joint acyclicity
  1. Tracks value generation and propagation to detect cyclic creation of terms
  2. Polynomial time to check
**Existing Acyclicity Conditions**

**Example**

\[ \begin{align*} r_1 : A(u) & \rightarrow R(u, f(u)) \land B(f(u)) \\ r_2 : B(v) & \rightarrow R(v, g(v)) \land C(g(v)) \\ r_3 : R(w, z) \land B(z) & \rightarrow A(w) \end{align*} \]

\[ \text{Pos}_B(u) = \{A\}_{1} \subseteq \text{Move}(f(u)) = \{R\}_{2}, B\}_{1}, R\}_{1}, A\}_{1} \]

- Joint acyclicity
  1. Tracks **value generation and propagation** to detect cyclic creation of terms
  2. Polynomial time to check
- May **overestimate** rule applicability
Model-faithful Acyclicity

- Track rule applications more ‘faithfully’
Model-faithful Acyclicity

- Track rule applications more ‘faithfully’

Example:

\[
A(u) \rightarrow R(u, f(u)) \land B(f(u))
\]

\[
B(v) \rightarrow R(v, g(v)) \land C(g(v))
\]

\[
R(w, z) \land B(z) \rightarrow A(w)
\]
MODEL-FAITHFUL ACYCLICITY

- Track rule applications more ‘faithfully’

**Example**

\[
\begin{align*}
A(u) & \rightarrow R(u, f(u)) \land B(f(u)) \land S(u, f(u)) \land F_f(f(u)) \\
B(v) & \rightarrow R(v, g(v)) \land C(g(v)) \land S(v, g(v)) \land F_g(g(v)) \\
R(w, z) \land B(z) & \rightarrow A(w)
\end{align*}
\]
MODEL-FAITHFUL ACYCLICITY

- Track rule applications more ‘faithfully’

**Example**

\[
A(u) \rightarrow R(u, f(u)) \land B(f(u)) \land S(u, f(u)) \land F_f(f(u))
\]

\[
B(v) \rightarrow R(v, g(v)) \land C(g(v)) \land S(v, g(v)) \land F_g(g(v))
\]

\[
R(w, z) \land B(z) \rightarrow A(w)
\]

\[
S(x, y) \rightarrow D(x, y)
\]

\[
D(x, y) \land S(y, z) \rightarrow D(x, z)
\]
Model-faithful Acyclicity

- Track rule applications more ‘faithfully’

**Example**

\[
\begin{align*}
A(u) & \rightarrow R(u,f(u)) \land B(f(u)) \land S(u,f(u)) \land F_f(f(u)) \\
B(v) & \rightarrow R(v,g(v)) \land C(g(v)) \land S(v,g(v)) \land F_g(g(v)) \\
R(w,z) \land B(z) & \rightarrow A(w) \\
S(x,y) & \rightarrow D(x,y) \\
D(x,y) \land S(y,z) & \rightarrow D(x,z) \\
F_f(x) \land D(x,y) \land F_f(y) & \rightarrow \text{Cycle} \\
F_g(x) \land D(x,y) \land F_g(y) & \rightarrow \text{Cycle}
\end{align*}
\]
**Model-faithful Acyclicity**

- Track rule applications more ‘faithfully’

**Example**

\[
A(u) \rightarrow R(u,f(u)) \land B(f(u)) \land S(u,f(u)) \land F_f(f(u))
\]

\[
B(v) \rightarrow R(v,g(v)) \land C(g(v)) \land S(v,g(v)) \land F_g(g(v))
\]

\[
R(w,z) \land B(z) \rightarrow A(w)
\]

\[
S(x,y) \rightarrow D(x,y)
\]

\[
D(x,y) \land S(y,z) \rightarrow D(x,z)
\]

\[
F_f(x) \land D(x,y) \land F_f(y) \rightarrow \text{Cycle}
\]

\[
F_g(x) \land D(x,y) \land F_g(y) \rightarrow \text{Cycle}
\]

- For \( \Sigma \) a set of rules, \( \Sigma \) is MFA if \( I_\Sigma^* \cup \text{MFA}(\Sigma) \not\models \text{Cycle} \)
Model-faithful Acyclicity

- Track rule applications more ‘faithfully’

**Example**

\[
\begin{align*}
A(u) & \rightarrow R(u, f(u)) \land B(f(u)) \land S(u, f(u)) \land F_f(f(u)) \\
B(v) & \rightarrow R(v, g(v)) \land C(g(v)) \land S(v, g(v)) \land F_g(g(v)) \\
R(w, z) \land B(z) & \rightarrow A(w) \\
S(x, y) & \rightarrow D(x, y) \\
D(x, y) \land S(y, z) & \rightarrow D(x, z) \\
F_f(x) \land D(x, y) \land F_f(y) & \rightarrow \text{Cycle} \\
F_g(x) \land D(x, y) \land F_g(y) & \rightarrow \text{Cycle}
\end{align*}
\]

- For $\Sigma$ a set of rules, $\Sigma$ is MFA if \( I_\Sigma^* \cup MFA(\Sigma) \not\models \text{Cycle} \)
**Model-faithful Acyclicity**

- Track rule applications more ‘faithfully’

**Example**

\[
\begin{align*}
A(u) & \rightarrow R(u,f(u)) \land B(f(u)) \land S(u,f(u)) \land F_f(f(u)) \\
B(v) & \rightarrow R(v,g(v)) \land C(g(v)) \land S(v,g(v)) \land F_g(g(v)) \\
R(w,z) \land B(z) & \rightarrow A(w) \\
S(x,y) & \rightarrow D(x,y) \\
D(x,y) \land S(y,z) & \rightarrow D(x,z) \\
F_f(x) \land D(x,y) \land F_f(y) & \rightarrow Cycle \\
F_g(x) \land D(x,y) \land F_g(y) & \rightarrow Cycle
\end{align*}
\]

- For \( \sum \) a set of rules, \( \sum \) is MFA if \( I^*_\sum \cup MFA(\sum) \not\models Cycle \)
MODEL-FAITHFUL ACYCLICITY

- Track rule applications more ‘faithfully’

**Example**

\[ A(u) \rightarrow R(u, f(u)) \land B(f(u)) \land S(u, f(u)) \land F_f(f(u)) \]
\[ B(v) \rightarrow R(v, g(v)) \land C(g(v)) \land S(v, g(v)) \land F_g(g(v)) \]
\[ R(w, z) \land B(z) \rightarrow A(w) \]
\[ S(x, y) \rightarrow D(x, y) \]
\[ D(x, y) \land S(y, z) \rightarrow D(x, z) \]
\[ F_f(x) \land D(x, y) \land F_f(y) \rightarrow \text{Cycle} \]
\[ F_g(x) \land D(x, y) \land F_g(y) \rightarrow \text{Cycle} \]

- For \( \Sigma \) a set of rules, \( \Sigma \) is MFA if \( I_\Sigma^* \cup MFA(\Sigma) \not\models \text{Cycle} \)
**Model-faithful Acyclicity**

- Track rule applications more ‘faithfully’

**Example**

\[
A(u) \rightarrow R(u, f(u)) \land B(f(u)) \land S(u, f(u)) \land F_f(f(u))
\]

\[
B(v) \rightarrow R(v, g(v)) \land C(g(v)) \land S(v, g(v)) \land F_g(g(v))
\]

\[
R(w, z) \land B(z) \rightarrow A(w)
\]

\[
S(x, y) \rightarrow D(x, y)
\]

\[
D(x, y) \land S(y, z) \rightarrow D(x, z)
\]

\[
F_f(x) \land D(x, y) \land F_f(y) \rightarrow \text{Cycle}
\]

\[
F_g(x) \land D(x, y) \land F_g(y) \rightarrow \text{Cycle}
\]

- For $\Sigma$ a set of rules, $\Sigma$ is MFA if $I_\Sigma^* \cup \text{MFA}(\Sigma) \not\models \text{Cycle}$
Model-faithful Acyclicity

- Track rule applications more ‘faithfully’

Example

\[
A(u) \rightarrow R(u, f(u)) \land B(f(u)) \land S(u, f(u)) \land F_f(f(u)) \\
B(v) \rightarrow R(v, g(v)) \land C(g(v)) \land S(v, g(v)) \land F_g(g(v)) \\
R(w, z) \land B(z) \rightarrow A(w)
\]

\[
S(x, y) \rightarrow D(x, y) \\
D(x, y) \land S(y, z) \rightarrow D(x, z) \\
F_f(x) \land D(x, y) \land F_f(y) \rightarrow \text{Cycle} \\
F_g(x) \land D(x, y) \land F_g(y) \rightarrow \text{Cycle}
\]

- For \( \Sigma \) a set of rules, \( \Sigma \) is MFA if \( I_\Sigma^* \cup MFA(\Sigma) \not\models \text{Cycle} \)
- Set of rules that correspond to DL subsumptions \( \{A \equiv \exists R.B, B \subseteq \exists R.C\} \) is MFA
Cost of Checking MFA

- Testing model-faithful acyclicity for a set of rules $\Sigma$
Cost of Checking MFA

- Testing model-faithful acyclicity for a set of rules $\Sigma$

1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ (no restriction)

$\leadsto$ \text{2EXPTIME-complete (tree with branching factor $|\vec{x}|$ and height the total number of function symbols)}
Cost of Checking MFA

Testing model-faithful acyclicity for a set of rules $\Sigma$

1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ (no restriction)
   \[ \sim 2\text{EXPTIME}-\text{complete} \] (tree with branching factor $|\vec{x}|$ and height the total number of function symbols)

2. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ with predicates of bounded arity
   \[ \sim 2\text{EXPTIME}-\text{complete} \]
Cost of Checking MFA

Testing model-faithful acyclicity for a set of rules $\Sigma$

1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ (no restriction)
   $\leadsto$ $2\text{EXPTIME}$-complete (tree with branching factor $|\vec{x}|$ and height the total number of function symbols)

2. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ with predicates of bounded arity
   $\leadsto$ $2\text{EXPTIME}$-complete

3. Rules from Horn-$\mathbf{SRI}$
   $\leadsto$ $\text{EXPTIME}$-hard
Cost of Checking MFA

Testing model-faithful acyclicity for a set of rules $\Sigma$

1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ (no restriction)

   $\leadsto$ 2EXPTIME-complete (tree with branching factor $|\vec{x}|$ and height the total number of function symbols)

2. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ with predicates of bounded arity

   $\leadsto$ 2EXPTIME-complete

3. Rules from Horn-$SRI$

   $\leadsto$ EXPTIME-hard

4. Rules from Horn-$SHIQ$

   $\leadsto$ PSPACE-complete
Cost of Checking MFA

- Testing model-faithful acyclicity for a set of rules $\Sigma$
  1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ (no restriction)
     $\leadsto$ 2EXPTIME-complete (tree with branching factor $|\vec{x}|$ and height the total number of function symbols)
  2. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ with predicates of bounded arity
     $\leadsto$ 2EXPTIME-complete
  3. Rules from Horn-$SRI$
     $\leadsto$ EXPTIME-hard
  4. Rules from Horn-$SHIQ$
     $\leadsto$ PSPACE-complete

- Existing acyclicity conditions can be checked in PTIME
Cost of Checking MFA

- Testing model-faithful acyclicity for a set of rules $\Sigma$
  
  1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ (no restriction)
     
     $\leadsto$ 2EXPTIME-complete (tree with branching factor $|\vec{x}|$ and height the total number of function symbols)

  2. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ with predicates of bounded arity
     
     $\leadsto$ 2EXPTIME-complete

  3. Rules from Horn-SRI
     
     $\leadsto$ EXPTIME-hard

  4. Rules from Horn-SHIQ
     
     $\leadsto$ PSPACE-complete

- Existing acyclicity conditions can be checked in PTIME
- Isn’t computational complexity too high?
MODEL-SUMMARISING ACYCLICITY

- Track rule applications just ‘faithfully’ enough
Track rule applications just ‘faithfully’ enough

**Example**

\[
\begin{align*}
A(u) &\rightarrow R(u, f(u)) \land B(f(u)) \\
B(v) &\rightarrow R(v, g(v)) \land C(g(v)) \\
R(w, z) \land B(z) &\rightarrow A(w)
\end{align*}
\]
Model-summarising Acyclicity

- Track rule applications just ‘faithfully’ enough

**Example**

\[ A(u) \rightarrow R(u, f(u)) \land B(f(u)) \]
\[ B(v) \rightarrow R(v, g(v)) \land C(g(v)) \]
\[ R(w, z) \land B(z) \rightarrow A(w) \]
**Model-summarising Acyclicity**

- Track rule applications just ‘faithfully’ enough

**Example**

\[
\begin{align*}
A(u) & \rightarrow R(u, c_1) \land B(c_1) \\
B(v) & \rightarrow R(v, c_2) \land C(c_2) \\
R(w, z) \land B(z) & \rightarrow A(w)
\end{align*}
\]
Model-summarising Acyclicity

- Track rule applications just ‘faithfully’ enough

**Example**

\[ A(u) \rightarrow R(u, c_1) \land B(c_1) \land S(u, c_1) \land F_{c_1}(c_1) \]
\[ B(v) \rightarrow R(v, c_2) \land C(c_2) \land S(v, c_2) \land F_{c_2}(c_2) \]
\[ R(w, z) \land B(z) \rightarrow A(w) \]
MODEL-SUMMARISING ACYCLICITY

- Track rule applications just ‘faithfully’ enough

**EXAMPLE**

\[
\begin{align*}
A(u) & \rightarrow R(u, c_1) \land B(c_1) \land S(u, c_1) \land F_{c_1}(c_1) \\
B(v) & \rightarrow R(v, c_2) \land C(c_2) \land S(v, c_2) \land F_{c_2}(c_2) \\
R(w, z) \land B(z) & \rightarrow A(w) \\
S(x, y) & \rightarrow D(x, y) \\
D(x, y) \land S(y, z) & \rightarrow D(x, z) \\
F_{c_1}(x) \land D(x, y) \land F_{c_1}(y) & \rightarrow \text{Cycle} \\
F_{c_2}(x) \land D(x, y) \land F_{c_2}(y) & \rightarrow \text{Cycle}
\end{align*}
\]
Track rule applications just ‘faithfully’ enough

**Example**

\[ A(u) \rightarrow R(u, c_1) \land B(c_1) \land S(u, c_1) \land F_{c_1}(c_1) \]
\[ B(v) \rightarrow R(v, c_2) \land C(c_2) \land S(v, c_2) \land F_{c_2}(c_2) \]
\[ R(w, z) \land B(z) \rightarrow A(w) \]
\[ S(x, y) \rightarrow D(x, y) \]
\[ D(x, y) \land S(y, z) \rightarrow D(x, z) \]
\[ F_{c_1}(x) \land D(x, y) \land F_{c_1}(y) \rightarrow \text{Cycle} \]
\[ F_{c_2}(x) \land D(x, y) \land F_{c_2}(y) \rightarrow \text{Cycle} \]

For \( \Sigma \) a set of rules, \( \Sigma \) is MSA if \( I_{\Sigma}^* \cup \text{MSA}(\Sigma) \not\models \text{Cycle} \)
Model-summarising Acyclicity

- Track rule applications just ‘faithfully’ enough

**Example**

\[ A(u) \rightarrow R(u, c_1) \land B(c_1) \land S(u, c_1) \land F_{c_1}(c_1) \]
\[ B(v) \rightarrow R(v, c_2) \land C(c_2) \land S(v, c_2) \land F_{c_2}(c_2) \]
\[ R(w, z) \land B(z) \rightarrow A(w) \]
\[ S(x, y) \rightarrow D(x, y) \]
\[ D(x, y) \land S(y, z) \rightarrow D(x, z) \]
\[ F_{c_1}(x) \land D(x, y) \land F_{c_1}(y) \rightarrow \text{Cycle} \]
\[ F_{c_2}(x) \land D(x, y) \land F_{c_2}(y) \rightarrow \text{Cycle} \]

- For \( \Sigma \) a set of rules, \( \Sigma \) is MSA if \( I_{\Sigma}^* \cup MSA(\Sigma) \not\models \text{Cycle} \)
MODEL-SUMMARISING A CYClicity

- Track rule applications just ‘faithfully’ enough

**EXAMPLE**

\[
\begin{align*}
A(u) & \rightarrow R(u, c_1) \land B(c_1) \land S(u, c_1) \land F_{c_1}(c_1) \\
B(v) & \rightarrow R(v, c_2) \land C(c_2) \land S(v, c_2) \land F_{c_2}(c_2) \\
R(w, z) & \land B(z) \rightarrow A(w) \\
S(x, y) & \rightarrow D(x, y) \\
D(x, y) & \land S(y, z) \rightarrow D(x, z) \\
F_{c_1}(x) & \land D(x, y) \land F_{c_1}(y) \rightarrow \text{Cycle} \\
F_{c_2}(x) & \land D(x, y) \land F_{c_2}(y) \rightarrow \text{Cycle}
\end{align*}
\]

- For \( \Sigma \) a set of rules, \( \Sigma \) is MSA if \( I_{\Sigma}^* \cup \text{MSA}(\Sigma) \not\models \text{Cycle} \)
MODEL-SUMMARISING ACYCLICITY

- Track rule applications just ‘faithfully’ enough

**EXAMPLE**

\[
\begin{align*}
A(u) & \rightarrow R(u, c_1) \land B(c_1) \land S(u, c_1) \land F_{c_1}(c_1) \\
B(v) & \rightarrow R(v, c_2) \land C(c_2) \land S(v, c_2) \land F_{c_2}(c_2) \\
R(w, z) \land B(z) & \rightarrow A(w) \\
S(x, y) & \rightarrow D(x, y) \\
D(x, y) \land S(y, z) & \rightarrow D(x, z) \\
F_{c_1}(x) \land D(x, y) \land F_{c_1}(y) & \rightarrow \text{Cycle} \\
F_{c_2}(x) \land D(x, y) \land F_{c_2}(y) & \rightarrow \text{Cycle}
\end{align*}
\]

- For \( \Sigma \) a set of rules, \( \Sigma \) is MSA if \( I^{*}_\Sigma \cup \text{MSA}(\Sigma) \not\models \text{Cycle} \)
**Model-summarising Acyclicity**

- Track rule applications just ‘faithfully’ enough

**Example**

\[
\begin{align*}
A(u) &\rightarrow R(u, c_1) \land B(c_1) \land S(u, c_1) \land F_{c_1}(c_1) \\
B(v) &\rightarrow R(v, c_2) \land C(c_2) \land S(v, c_2) \land F_{c_2}(c_2) \\
R(w, z) \land B(z) &\rightarrow A(w)
\end{align*}
\]

\[
\begin{align*}
S(x, y) &\rightarrow D(x, y) \\
D(x, y) \land S(y, z) &\rightarrow D(x, z) \\
F_{c_1}(x) \land D(x, y) \land F_{c_1}(y) &\rightarrow \text{Cycle} \\
F_{c_2}(x) \land D(x, y) \land F_{c_2}(y) &\rightarrow \text{Cycle}
\end{align*}
\]

- For \(\Sigma\) a set of rules, \(\Sigma\) is MSA if \(I_\Sigma^* \cup MSA(\Sigma) \not\models \text{Cycle}\)
- Set of rules that correspond to DL subsumptions \(\{A \equiv \exists R.B, B \sqsubseteq \exists R.C\}\) is still MSA
Cost of Checking MSA

- Testing model-faithful acyclicity for a set of rules $\Sigma$
Testing model-faithful acyclicity for a set of rules $\Sigma$

1. Rules of the form $\varphi(\bar{x}, \bar{z}) \rightarrow \exists \bar{y}. \psi(\bar{x}, \bar{y})$ (no restriction)
   $\leadsto$ EXPTIME-complete
Cost of Checking MSA

- Testing model-faithful acyclicity for a set of rules $\Sigma$
  
  1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ (no restriction)
     $\leadsto$ EXPTIME-complete
  
  2. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ with predicates of bounded arity
     $\leadsto$ coNP-complete
Cost of Checking MSA

- Testing model-faithful acyclicity for a set of rules $\Sigma$
  1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ (no restriction)
     $\rightsquigarrow$ EXPTIME-complete
  2. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ with predicates of bounded arity
     $\rightsquigarrow$ coNP-complete
  3. Rules from Horn-$SHIQ$
     $\rightsquigarrow$ PTIME-complete
Cost of Checking MSA

- Testing model-faithful acyclicity for a set of rules $\Sigma$
  1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ (no restriction)
     $\leadsto$ EXPTIME-complete
  2. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}. \psi(\vec{x}, \vec{y})$ with predicates of bounded arity
     $\leadsto$ coNP-complete
  3. Rules from Horn-$SHIQ$
     $\leadsto$ PTIME-complete

- Horn-$SHIQ$ TBoxes can be checked in PTIME for MSA before potential materialisation-based query answering
Acyclicity Conditions (Partial) Taxonomy

Our contributions:
1. MSA strictly subsumes SWA
2. MFA strictly subsumes MSA

Example:
\[
A(x) \rightarrow \exists y. R(x, y) \land B(y) \\
B(x) \rightarrow \exists y. S(x, y) \land T(y, x) \\
A(z) \land S(z, x) \rightarrow C(x) \\
C(z) \land T(z, x) \rightarrow A(x)
\]

MFA but not MSA

MSA and MFA coincide in experimental evaluation of DL ontologies
Acyclicity Conditions (Partial) Taxonomy

- Our contributions:

<table>
<thead>
<tr>
<th>JA</th>
<th>$\not\subseteq$</th>
<th>SWA</th>
<th>MSA</th>
<th>MFA</th>
</tr>
</thead>
</table>

$B(x) \rightarrow \exists y. R(x, y) \land B(y)$

$A(z) \land S(z, x) \rightarrow C(x)$

$C(z) \land T(z, x) \rightarrow A(x)$

MSA and MFA coincide in experimental evaluation of DL ontologies

MSA strictly subsumes SWA

MFA strictly subsumes MSA

$JA \subseteq SWA \subseteq MSA \subseteq MFA$
Acyclicity Conditions (Partial) Taxonomy

- Our contributions:
  1. MSA strictly subsumes SWA

\[
\begin{array}{cccc}
\text{JA} & \subset & \text{SWA} & \subset \\
& & \text{MSA} & \text{MFA}
\end{array}
\]
# Acyclicity Conditions (Partial) Taxonomy

- **Our contributions:**
  1. MSA strictly subsumes SWA
  2. MFA strictly subsumes MSA

<table>
<thead>
<tr>
<th>JA</th>
<th>SWA</th>
<th>MSA</th>
<th>MFA</th>
</tr>
</thead>
</table>

## Example

\[
\begin{align*}
A(x) & \rightarrow \exists y. R(x, y) \land B(y) \\
B(x) & \rightarrow \exists y. S(x, y) \land T(y, x) \\
A(z) \land S(z, x) & \rightarrow C(x) \\
C(z) \land T(z, x) & \rightarrow A(x)
\end{align*}
\]

- MFA but not MSA
Our contributions:

1. MSA strictly subsumes SWA
2. MFA strictly subsumes MSA

\[
\text{JA} \subsetneq \text{SWA} \subsetneq \text{MSA} \subsetneq \text{MFA}
\]

**Example**

\[
\begin{align*}
A(x) & \rightarrow \exists y. R(x, y) \land B(y) \\
B(x) & \rightarrow \exists y. S(x, y) \land T(y, x) \\
A(z) \land S(z, x) & \rightarrow C(x) \\
C(z) \land T(z, x) & \rightarrow A(x)
\end{align*}
\]

MFA but not MSA

- MSA and MFA coincide in experimental evaluation of DL ontologies
1 Motivation
2 MFA and MSA
3 Querying Acyclic DL Ontologies
4 Experimental Results
### Translating DLs into Rules

- Axioms of normalised Horn-\textit{SRIQ} ontologies can be converted to (existential) rules

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \sqsubseteq \exists R.B$</td>
<td>$A(x) \rightarrow \exists y. R(x, y) \land B(y)$</td>
</tr>
<tr>
<td>$A \sqsubseteq \leq 1 R.B$</td>
<td>$A(z) \land R(z, x_1) \land B(x_1) \land R(z, x_2) \land B(x_2) \rightarrow x_1 \approx x_2$</td>
</tr>
<tr>
<td>$A \cap B \sqsubseteq C$</td>
<td>$A(x) \land B(x) \rightarrow C(x)$</td>
</tr>
<tr>
<td>$A \sqsubseteq \forall R.B$</td>
<td>$A(z) \land R(z, x) \rightarrow B(x)$</td>
</tr>
<tr>
<td>$R \sqsubseteq S$</td>
<td>$R(x_1, x_2) \rightarrow S(x_1, x_2)$</td>
</tr>
<tr>
<td>$R \circ S \sqsubseteq T$</td>
<td>$R(x_1, z) \land S(z, x_2) \rightarrow T(x_1, x_2)$</td>
</tr>
</tbody>
</table>
Translating DLs into rules

- Axioms of normalised Horn-\textit{SRIQ} ontologies can be converted to (existential) rules

<table>
<thead>
<tr>
<th>Condition</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \sqsubseteq \exists R.B$</td>
<td>$A(x) \rightarrow \exists y. R(x, y) \land B(y)$</td>
</tr>
<tr>
<td>$A \sqsubseteq \leq 1 R.B$</td>
<td>$A(z) \land R(z, x_1) \land B(x_1) \land R(z, x_2) \land B(x_2) \rightarrow x_1 \approx x_2$</td>
</tr>
<tr>
<td>$A \cap B \sqsubseteq C$</td>
<td>$A(x) \land B(x) \rightarrow C(x)$</td>
</tr>
<tr>
<td>$A \sqsubseteq \forall R.B$</td>
<td>$A(z) \land R(z, x) \rightarrow B(x)$</td>
</tr>
<tr>
<td>$R \sqsubseteq S$</td>
<td>$R(x_1, x_2) \rightarrow S(x_1, x_2)$</td>
</tr>
<tr>
<td>$R \circ S \sqsubseteq T$</td>
<td>$R(x_1, z) \land S(z, x_2) \rightarrow T(x_1, x_2)$</td>
</tr>
</tbody>
</table>

- Equality is handled with a modification of the singularisation [Marnette, PODS, 2009] technique
Answering conjunctive queries for the DL Horn-$SHIQ$ is EXPTIME-complete [Eiter et al., 2008]
Answering conjunctive queries for the DL Horn-$SHIQ$ is \textsc{EXPTIME}-complete [Eiter et al., 2008]

Does acyclicity affect complexity for DL Query Answering?
**DL Query Answering under Acyclicity**

- Answering conjunctive queries for the DL Horn-\textit{SHIQ} is EXPTIME-complete [Eiter et al., 2008]

- Does acyclicity affect complexity for DL Query Answering?

1. Horn-\textit{SHIQ} TBox $\mathcal{T}$ and ABox $\mathcal{A}$
   - $\mathcal{T}$ is MFA
   - $\mathcal{Q}$ Boolean conjunctive query

   $\models$ Deciding $\mathcal{T} \cup \mathcal{A} \models \mathcal{Q}$ is PSPACE-complete
DL Query Answering under Acyclicity

- Answering conjunctive queries for the DL Horn-\textit{SHIQ} is EXPTIME-complete [Eiter et al., 2008]

- Does acyclicity affect complexity for DL Query Answering?

1. Horn-\textit{SHIQ} TBox \( \mathcal{T} \) and ABox \( \mathcal{A} \)
   \( \mathcal{T} \) is MFA
   \( Q \) Boolean conjunctive query
   \[ \models \] Deciding \( \mathcal{T} \cup \mathcal{A} \models Q \) is PSPACE-complete

2. Horn-\textit{SRI} TBox \( \mathcal{T} \) and ABox \( \mathcal{A} \)
   \( \mathcal{T} \) is weakly acyclic
   \( F \) set of facts
   \[ \models \] Deciding \( \mathcal{T} \cup \mathcal{A} \models F \) is EXPTIME-hard
ACYCLICITY TESTS

- Checked 149 DL ontologies for WA, JA, MSA, MFA
ACYCLICITY TESTS

- Checked 149 DL ontologies for WA, JA, MSA, MFA

<table>
<thead>
<tr>
<th>Existential rules</th>
<th>Total</th>
<th>MSA</th>
<th>JA</th>
<th>WA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 100</td>
<td>70</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>100–1K</td>
<td>33</td>
<td>30</td>
<td>30</td>
<td>23</td>
</tr>
<tr>
<td>1K–5K</td>
<td>20</td>
<td>14</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>5K–12K</td>
<td>14</td>
<td>11</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>12K–160K</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>All sizes</td>
<td>149</td>
<td>124</td>
<td>117</td>
<td>108</td>
</tr>
</tbody>
</table>
ACYCLICITY TESTS

- Checked 149 DL ontologies for WA, JA, MSA, MFA

<table>
<thead>
<tr>
<th>Existential rules</th>
<th>Total</th>
<th>MSA</th>
<th>JA</th>
<th>WA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 100</td>
<td>70</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>100–1K</td>
<td>33</td>
<td>30</td>
<td>30</td>
<td>23</td>
</tr>
<tr>
<td>1K–5K</td>
<td>20</td>
<td>14</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>5K–12K</td>
<td>14</td>
<td>11</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>12K–160K</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>All sizes</td>
<td>149</td>
<td>124</td>
<td>117</td>
<td>108</td>
</tr>
</tbody>
</table>

- MSA and MFA coincide w.r.t. the test ontologies
Acyclicity Tests

- Checked 149 DL ontologies for WA, JA, MSA, MFA

<table>
<thead>
<tr>
<th>Existential rules</th>
<th>Total</th>
<th>MSA</th>
<th>JA</th>
<th>WA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 100</td>
<td>70</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>100–1K</td>
<td>33</td>
<td>30</td>
<td>30</td>
<td>23</td>
</tr>
<tr>
<td>1K–5K</td>
<td>20</td>
<td>14</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>5K–12K</td>
<td>14</td>
<td>11</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>12K–160K</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>All sizes</td>
<td>149</td>
<td>124</td>
<td>117</td>
<td>108</td>
</tr>
</tbody>
</table>

- MSA and MFA coincide w.r.t. the test ontologies
- 83% were found MSA
Acyliclity Tests

- Checked 149 DL ontologies for WA, JA, MSA, MFA

<table>
<thead>
<tr>
<th>Existential rules</th>
<th>Total</th>
<th>MSA</th>
<th>JA</th>
<th>WA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 100</td>
<td>70</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>100–1K</td>
<td>33</td>
<td>30</td>
<td>30</td>
<td>23</td>
</tr>
<tr>
<td>1K–5K</td>
<td>20</td>
<td>14</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>5K–12K</td>
<td>14</td>
<td>11</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>12K–160K</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>All sizes</td>
<td>149</td>
<td>124</td>
<td>117</td>
<td>108</td>
</tr>
</tbody>
</table>

- MSA and MFA coincide w.r.t. the test ontologies
- 83% were found MSA
- 7 large and expressive OBO ontologies MSA but not JA (only two of them were $\mathcal{ELH}^r$ and DL-Lite)
Materialisation Tests

- Computed materialisation of *acyclic* TBoxes

<table>
<thead>
<tr>
<th>Depth</th>
<th>generated size</th>
<th>materialisation size</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 5</td>
<td>82</td>
<td>27</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>37</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>281</td>
</tr>
</tbody>
</table>

- Depth = length of function symbol nesting
- generated size = # facts generated by existential rules
- materialisation size = # facts in materialisation

For ontologies with small depths materialisation seems practically feasible.
# Materialisation Tests

- Computed materialisation of **acyclic TBoxes**

<table>
<thead>
<tr>
<th>Depth</th>
<th>#</th>
<th>generated size</th>
<th>materialisation size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>max</td>
<td>avg</td>
</tr>
<tr>
<td>&lt; 5</td>
<td>82</td>
<td>27</td>
<td>2</td>
</tr>
<tr>
<td>5–9</td>
<td>13</td>
<td>37</td>
<td>11</td>
</tr>
<tr>
<td>10–80</td>
<td>14</td>
<td>281</td>
<td>51</td>
</tr>
</tbody>
</table>

**Depth** = length of function symbol nesting

**generated size** = \( \frac{\text{# facts generated by existential rules}}{\text{# facts in initial ABox}} \)

**materialisation size** = \( \frac{\text{# facts in materialisation}}{\text{# facts in initial ABox}} \)

For ontologies with small depths materialisation seems practically feasible.
**Materialisation Tests**

- Computed materialisation of *acyclic* TBoxes

<table>
<thead>
<tr>
<th>Depth</th>
<th>#</th>
<th>generated size</th>
<th>materialisation size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>max</td>
<td>avg</td>
</tr>
<tr>
<td>&lt; 5</td>
<td>82</td>
<td>27</td>
<td>2</td>
</tr>
<tr>
<td>5–9</td>
<td>13</td>
<td>37</td>
<td>11</td>
</tr>
<tr>
<td>10–80</td>
<td>14</td>
<td>281</td>
<td>51</td>
</tr>
</tbody>
</table>

- **Depth** = length of function symbol nesting
- **generated size** = \( \frac{\text{# facts generated by existential rules}}{\text{# facts in initial ABox}} \)
- **materialisation size** = \( \frac{\text{# facts in materialisation}}{\text{# facts in initial ABox}} \)

- For ontologies with *small depths* materialisation seems practically feasible
Materialisation Tests

- Computed materialisation of acyclic TBoxes

<table>
<thead>
<tr>
<th>Depth</th>
<th>#</th>
<th>generated size</th>
<th>materialisation size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>max</td>
<td>avg</td>
</tr>
<tr>
<td>&lt; 5</td>
<td>82</td>
<td>27</td>
<td>2</td>
</tr>
<tr>
<td>5–9</td>
<td>13</td>
<td>37</td>
<td>11</td>
</tr>
<tr>
<td>10–80</td>
<td>14</td>
<td>281</td>
<td>51</td>
</tr>
</tbody>
</table>

Depth = length of function symbol nesting

generated size = \# facts generated by existential rules / \# facts in initial ABox

materialisation size = \# facts in materialisation / \# facts in initial ABox

For ontologies with small depths materialisation seems practically feasible
SUMMARY OF THE RESULTS

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-\textit{SHIQ}</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicity conditions
   - Horn-\textit{SRI} \( \mathcal{T} \) in WA: \( \mathcal{T} \cup \mathcal{A} \models F \) is ExpTime-hard
   - Horn-\textit{SHIQ} \( \mathcal{T} \) in MFA: \( \mathcal{T} \cup \mathcal{A} \models Q \) is PSpace-complete

4. Experimental evaluation on DL ontologies
   - 83% ontologies found acyclic (78% JA)
   - materialised ABoxes not too large \( \leadsto \times 5 \) bigger on average for ontologies with depth < 5 (= most ontologies)
SUMMARY OF THE RESULTS

1. More general acyclicity conditions: MSA and MFA
2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-$SHIQ$</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicity conditions
   - Horn-$SRI$ $\mathcal{T}$ in WA: $\mathcal{T} \cup \mathcal{A} \models F$ is ExpTime-hard
   - Horn-$SHIQ$ $\mathcal{T}$ in MFA: $\mathcal{T} \cup \mathcal{A} \models Q$ is PSpace-complete

4. Experimental evaluation on DL ontologies
   - 83% ontologies found acyclic (78% JA)
   - materialised ABoxes not too large

Materialisation-based reasoning beyond OWL 2 RL might be practically feasible
SUMMARY OF THE RESULTS

1. More general acyclicity conditions: MSA and MFA

2. Complexity analysis for checking MSA and MFA

<table>
<thead>
<tr>
<th></th>
<th>Horn-SHIQ</th>
<th>bounded arity</th>
<th>no restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSA</td>
<td>PTime-complete</td>
<td>coNP-complete</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>MFA</td>
<td>PSpace-complete</td>
<td>2ExpTime-complete</td>
<td>2ExpTime-complete</td>
</tr>
</tbody>
</table>

3. DL query answering under acyclicity conditions
   - Horn-SRI $\mathcal{T}$ in WA: $\mathcal{T} \cup \mathcal{A} \models F$ is ExpTime-hard
   - Horn-SHIQ $\mathcal{T}$ in MFA: $\mathcal{T} \cup \mathcal{A} \models Q$ is PSpace-complete

4. Experimental evaluation on DL ontologies
   - 83% ontologies found acyclic (78% JA)
   - materialised ABoxes not too large

Materialisation-based reasoning beyond OWL 2 RL might be practically feasible

Thank you! Questions?!!?