Acyclicity Conditions and their Application to Query Answering in Description Logics

Bernardo Cuenca Grau, Ian Horrocks, Markus Krötzsch, Clemens Kupke, Despoina Magka, Boris Motik, Zhe Wang

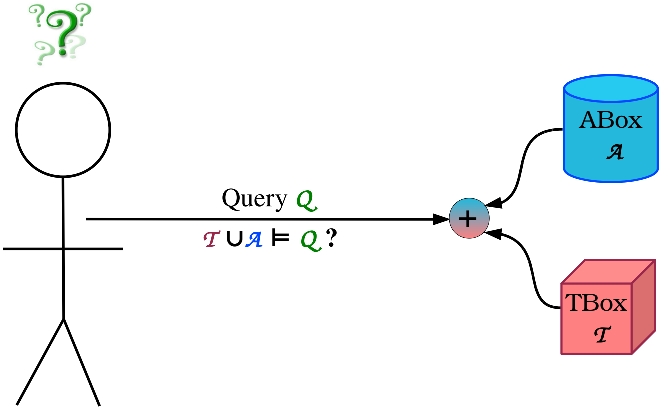
Department of Computer Science, University of Oxford

June 14, 2012

OUTLINE

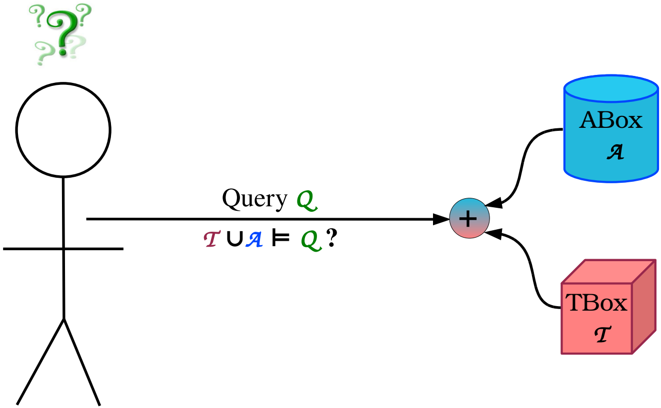
Ontological Query Answering

- Key reasoning task for DL and rule-based applications



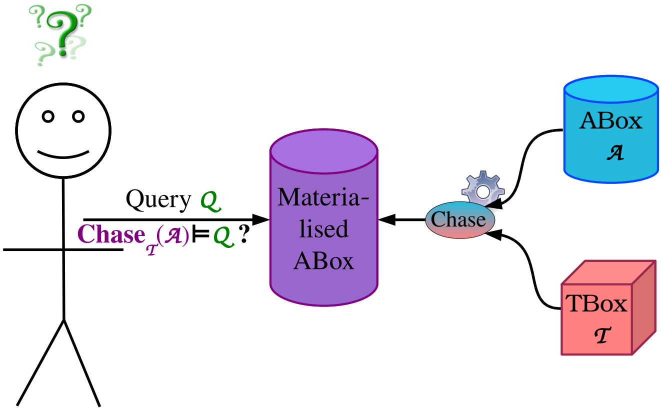
ONTOLOGICAL QUERY ANSWERING

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



ONTOLOGICAL QUERY ANSWERING

- Key reasoning task for DL and rule-based applications
- Answering CQs over DLs $\rightsquigarrow$ 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

Existential Rules

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

Example

$A(x) \to \exists y.R(x, y) \land B(y)$ DL-equivalent $\rightsquigarrow$ $A \sqsubseteq \exists R.B$

Existential Rules

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

Example

$A(x) \rightarrow \exists y.R(x,y) \land B(y)$ DL-equivalent $\rightsquigarrow$ $A \sqsubseteq \exists R.B$

- Existential rules fundamental for several KR formalisms:

Existential Rules

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

Example

$A(x) \to \exists y.R(x,y) \wedge B(y)$ DL-equivalent $\rightsquigarrow$ $A \sqsubseteq \exists R.B$

- 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

Existential Rules

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

Example

$A(x) \rightarrow \exists y.R(x, y) \wedge B(y)$ DL-equivalent $\rightsquigarrow$ $A \sqsubseteq \exists R.B$

- 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

$A(x) \rightarrow \exists y.R(x,y) \wedge B(y)$ DL-equivalent $\rightsquigarrow$ $A \sqsubseteq \exists R.B$

- 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
- CQ answering is undecidable for existential rules

TACKLING UNDECIDABILITY

1. Identify groups of rules for which query answering is decidable

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

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

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],…

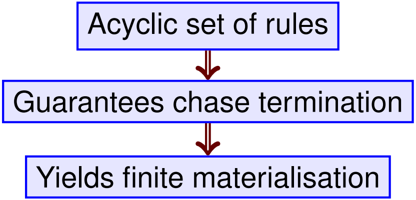
Acyclic set of rules

$\Downarrow$

Guarantees chase termination

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],...



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

⇓

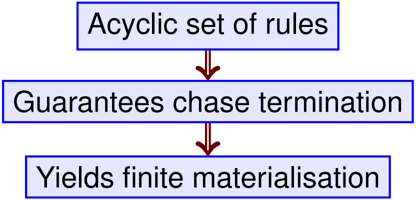
Yields finite materialisation

Plus

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

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],...



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

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

II. Materialised ABoxes can be stored as databases

Minus

I. 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

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

II. Materialised ABoxes can be stored as databases

Minus

I. Only sets of rules with models of bounded size

II. Acyclicity conditions might be too restrictive

Materialisation-based Reasoning

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

Materialisation-based Reasoning

- Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-$\mathcal{SHOIQ}$ [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-$\mathcal{SHOIQ}$ [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

Materialisation-based Reasoning

- Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-$\mathcal{SHOIQ}$ [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-$\mathcal{SHOIQ}$ [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-$\mathcal{SHOIQ}$ [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

Materialisation-based Reasoning

- Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-$\mathcal{SHOIQ}$ [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 ✘

Materialisation-based Reasoning

- Answering CQs over expressive DLs is expensive, e.g. EXPTIME-complete for Horn-$\mathcal{SHOIQ}$ [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

Results Overview

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

Results Overview

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

3. DL query answering under acyclicity conditions

Results Overview

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

3. DL query answering under acyclicity conditions
   - Horn-$\mathcal{SRIT}$ 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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

3. DL query answering under acyclicity conditions
   - Horn-$\mathcal{SRIT}$ $\mathcal{T}$ in WA: $\mathcal{T} \cup \mathcal{A} \models F$ is ExpTime-hard
   - Horn-$\mathcal{SHIQ}$ $\mathcal{T}$ in MFA: $\mathcal{T} \cup \mathcal{A} \models Q$ is PSpace-complete
4. Experimental evaluation on DL ontologies
   - 83% ontologies found acyclic (78% JA)

Results Overview

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

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

Results Overview

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

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

Outline

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \to \exists y_1.R(u, y_1) \wedge B(y_1)$$
$$r_2 : B(v) \to \exists y_2.R(v, y_2) \wedge C(y_2)$$
$$r_3 : R(w, z) \wedge B(z) \to A(w)$$

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow \exists y_1.R(u, y_1) \wedge B(y_1)$$
$$r_2 : B(v) \rightarrow \exists y_2.R(v, y_2) \wedge C(y_2)$$
$$r_3 : R(w, z) \wedge B(z) \rightarrow A(w)$$

$$A \equiv \exists R.B$$
$$B \sqsubseteq \exists R.C$$

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow \exists y_1.R(u, \underline{y_1}) \wedge B(\underline{y_1})$$

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

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

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$
$$r_2 : B(v) \rightarrow \exists y_2 . R(v, y_2) \wedge C(y_2)$$
$$r_3 : R(w, z) \wedge B(z) \rightarrow A(w)$$

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \to R(u, f(u)) \land B(f(u))$$

$$r_2 : B(v) \to \exists y_2 . R(v, \underline{y_2}) \land C(\underline{y_2})$$

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

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$

$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$

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

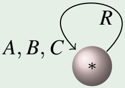
Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$

$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$

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



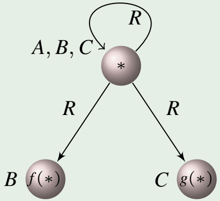
EXISTING ACYCLICITY CONDITIONS

EXAMPLE

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$

$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$

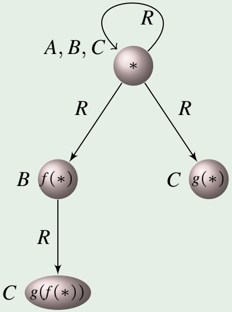
$$r_3 : R(w, z) \wedge B(z) \rightarrow A(w)$$



EXISTING ACYCLICITY CONDITIONS

EXAMPLE

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$
$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$
$$r_3 : R(w, z) \wedge B(z) \rightarrow A(w)$$



Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$
$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$
$$r_3 : R(w, z) \wedge B(z) \rightarrow A(w)$$

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

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow R(u, \underline{f(u)}) \wedge B(\underline{f(u)})$$

$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$

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

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

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

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$
$$r_2 : B(v) \rightarrow R(\underline{v}, g(v)) \wedge C(g(v))$$
$$r_3 : R(w, z) \wedge B(z) \rightarrow A(w)$$

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

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

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$

$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$

$$r_3 : R(w, z) \wedge B(z) \rightarrow A(\underline{w})$$

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

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

Existing Acyclicity Conditions

Example

$$r_1 : A(\underline{u}) \rightarrow R(u, f(u)) \wedge B(f(u))$$
$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$
$$r_3 : R(w, z) \wedge B(z) \rightarrow A(w)$$

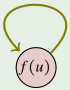
$\text{Pos}_B(u) = \{A|_1\}$ $\quad$ $\text{Move}(f(u)) = \{R|_2,\ B|_1,\ R|_1,\ A|_1\}$

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

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$
$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$
$$r_3 : R(w, z) \wedge B(z) \rightarrow A(w)$$



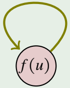
$$\mathsf{Pos}_B(u) = \{A|_1\} \subseteq \mathsf{Move}(f(u)) = \{R|_2,\ B|_1,\ R|_1,\ A|_1\}$$

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

Existing Acyclicity Conditions

Example

$$r_1 : A(u) \rightarrow R(u, f(u)) \wedge B(f(u))$$
$$r_2 : B(v) \rightarrow R(v, g(v)) \wedge C(g(v))$$
$$r_3 : R(w, z) \wedge B(z) \rightarrow A(w)$$



not in JA

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

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

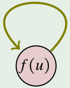
Existing Acyclicity Conditions

Example

$r_1 : A(u) \to R(u, f(u)) \wedge B(f(u))$

$r_2 : B(v) \to R(v, g(v)) \wedge C(g(v))$

$r_3 : R(w, z) \wedge B(z) \to A(w)$



not in JA

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

- 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

$A(u) \to R(u, f(u)) \wedge B(f(u))$

$B(v) \to R(v, g(v)) \wedge C(g(v))$

$R(w, z) \wedge B(z) \to A(w)$

Model-faithful Acyclicity

- Track rule applications more 'faithfully'

Example

$A(u) \to R(u, f(u)) \wedge B(f(u)) \wedge S(u, f(u)) \wedge F_f(f(u))$

$B(v) \to R(v, g(v)) \wedge C(g(v)) \wedge S(v, g(v)) \wedge F_g(g(v))$

$R(w, z) \wedge B(z) \to A(w)$

Model-faithful Acyclicity

- Track rule applications more 'faithfully'

Example

$A(u) \to R(u, f(u)) \wedge B(f(u)) \wedge S(u, f(u)) \wedge F_f(f(u))$

$B(v) \to R(v, g(v)) \wedge C(g(v)) \wedge S(v, g(v)) \wedge F_g(g(v))$

$R(w, z) \wedge B(z) \to A(w)$

$S(x, y) \to D(x, y)$

$D(x, y) \wedge S(y, z) \to D(x, z)$

Model-faithful Acyclicity

- Track rule applications more 'faithfully'

Example

$A(u) \rightarrow R(u, f(u)) \wedge B(f(u)) \wedge S(u, f(u)) \wedge F_f(f(u))$

$B(v) \rightarrow R(v, g(v)) \wedge C(g(v)) \wedge S(v, g(v)) \wedge F_g(g(v))$

$R(w, z) \wedge B(z) \rightarrow A(w)$

$S(x, y) \rightarrow D(x, y)$

$D(x, y) \wedge S(y, z) \rightarrow D(x, z)$

$F_f(x) \wedge D(x, y) \wedge F_f(y) \rightarrow Cycle$

$F_g(x) \wedge D(x, y) \wedge F_g(y) \rightarrow Cycle$

MODEL-FAITHFUL ACYCLICITY

- Track rule applications more ‘faithfully’

EXAMPLE

$A(u) \rightarrow R(u, f(u)) \wedge B(f(u)) \wedge S(u, f(u)) \wedge F_f(f(u))$

$B(v) \rightarrow R(v, g(v)) \wedge C(g(v)) \wedge S(v, g(v)) \wedge F_g(g(v))$

$R(w, z) \wedge B(z) \rightarrow A(w)$

$S(x, y) \rightarrow D(x, y)$

$D(x, y) \wedge S(y, z) \rightarrow D(x, z)$

$F_f(x) \wedge D(x, y) \wedge F_f(y) \rightarrow Cycle$

$F_g(x) \wedge D(x, y) \wedge F_g(y) \rightarrow Cycle$

- For Σ a set of rules, Σ is MFA if $I^*_\Sigma \cup MFA(\Sigma) \not\models Cycle$

MODEL-FAITHFUL ACYCLICITY

- Track rule applications more ‘faithfully’

EXAMPLE

$A(u) \to R(u, f(u)) \wedge B(f(u)) \wedge S(u, f(u)) \wedge F_f(f(u))$

$B(v) \to R(v, g(v)) \wedge C(g(v)) \wedge S(v, g(v)) \wedge F_g(g(v))$

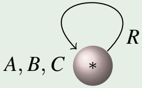
$R(w, z) \wedge B(z) \to A(w)$

$S(x, y) \to D(x, y)$

$D(x, y) \wedge S(y, z) \to D(x, z)$

$F_f(x) \wedge D(x, y) \wedge F_f(y) \to Cycle$

$F_g(x) \wedge D(x, y) \wedge F_g(y) \to Cycle$



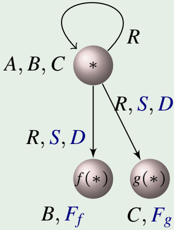
- For Σ a set of rules, Σ is MFA if $I^*_\Sigma \cup MFA(\Sigma) \not\models Cycle$

Model-faithful Acyclicity

- Track rule applications more 'faithfully'

Example

$$A(u) \rightarrow R(u, f(u)) \wedge B(f(u)) \wedge S(u, f(u)) \wedge F_f(f(u))$$
$$B(v) \rightarrow R(v, g(v)) \wedge C(g(v)) \wedge S(v, g(v)) \wedge F_g(g(v))$$
$$R(w, z) \wedge B(z) \rightarrow A(w)$$
$$S(x, y) \rightarrow D(x, y)$$
$$D(x, y) \wedge S(y, z) \rightarrow D(x, z)$$
$$F_f(x) \wedge D(x, y) \wedge F_f(y) \rightarrow Cycle$$
$$F_g(x) \wedge D(x, y) \wedge F_g(y) \rightarrow Cycle$$



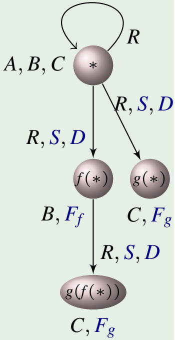
- For Σ a set of rules, Σ is MFA if $I^*_\Sigma \cup MFA(\Sigma) \not\models Cycle$

Model-faithful Acyclicity

- Track rule applications more 'faithfully'

Example

$$A(u) \to R(u,f(u)) \wedge B(f(u)) \wedge S(u,f(u)) \wedge F_f(f(u))$$
$$B(v) \to R(v,g(v)) \wedge C(g(v)) \wedge S(v,g(v)) \wedge F_g(g(v))$$
$$R(w,z) \wedge B(z) \to A(w)$$
$$S(x,y) \to D(x,y)$$
$$D(x,y) \wedge S(y,z) \to D(x,z)$$
$$F_f(x) \wedge D(x,y) \wedge F_f(y) \to Cycle$$
$$F_g(x) \wedge D(x,y) \wedge F_g(y) \to Cycle$$



- For Σ a set of rules, Σ is MFA if $I^*_\Sigma \cup MFA(\Sigma) \not\models Cycle$

Model-faithful Acyclicity

- Track rule applications more ‘faithfully’

Example

$A(u) \to R(u,f(u)) \wedge B(f(u)) \wedge S(u,f(u)) \wedge F_f(f(u))$

$B(v) \to R(v,g(v)) \wedge C(g(v)) \wedge S(v,g(v)) \wedge F_g(g(v))$

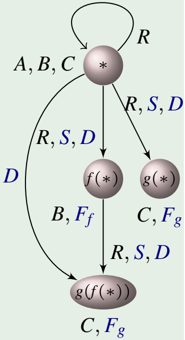
$R(w,z) \wedge B(z) \to A(w)$

$S(x,y) \to D(x,y)$

$D(x,y) \wedge S(y,z) \to D(x,z)$

$F_f(x) \wedge D(x,y) \wedge F_f(y) \to Cycle$

$F_g(x) \wedge D(x,y) \wedge F_g(y) \to Cycle$



- For Σ a set of rules, Σ is MFA if $I^*_\Sigma \cup MFA(\Sigma) \not\models Cycle$

MODEL-FAITHFUL ACYCLICITY

- Track rule applications more 'faithfully'

EXAMPLE

$A(u) \to R(u,f(u)) \land B(f(u)) \land S(u,f(u)) \land F_f(f(u))$

$B(v) \to R(v,g(v)) \land C(g(v)) \land S(v,g(v)) \land F_g(g(v))$

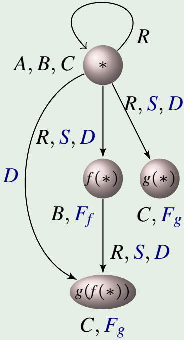
$R(w,z) \land B(z) \to A(w)$

$S(x,y) \to D(x,y)$

$D(x,y) \land S(y,z) \to D(x,z)$

$F_f(x) \land D(x,y) \land F_f(y) \to Cycle$

$F_g(x) \land D(x,y) \land F_g(y) \to Cycle$



- For Σ a set of rules, Σ is MFA if $I^*_\Sigma \cup MFA(\Sigma) \not\models Cycle$
- Set of rules that correspond to DL subsumptions $\{A \equiv \exists R.B, B \sqsubseteq \exists R.C\}$ is MFA

Cost of Checking MFA

- Testing model-faithful acyclicity for a set of rules Σ

Cost of Checking MFA

- Testing model-faithful acyclicity for a set of rules Σ

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

     $\rightsquigarrow$ 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 Σ

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

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

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

     $\rightsquigarrow$ 2EXPTIME-complete

Cost of Checking MFA

- Testing model-faithful acyclicity for a set of rules Σ

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

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

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

   $\rightsquigarrow$ 2EXPTIME-complete

3. Rules from Horn-$\mathcal{SRI}$

   $\rightsquigarrow$ EXPTIME-hard

Cost of Checking MFA

- Testing model-faithful acyclicity for a set of rules Σ

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

     $\rightsquigarrow$ 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

     $\rightsquigarrow$ 2EXPTIME-complete

  3. Rules from Horn-$\mathcal{SRI}$

     $\rightsquigarrow$ EXPTIME-hard

  4. Rules from Horn-$\mathcal{SHIQ}$

     $\rightsquigarrow$ PSPACE-complete

COST OF CHECKING MFA

- Testing model-faithful acyclicity for a set of rules Σ

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

     $\rightsquigarrow$ 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

     $\rightsquigarrow$ 2EXPTIME-complete

  3. Rules from Horn-$\mathcal{SRI}$

     $\rightsquigarrow$ EXPTIME-hard

  4. Rules from Horn-$\mathcal{SHIQ}$

     $\rightsquigarrow$ PSPACE-complete

- Existing acyclicity conditions can be checked in PTIME

Cost of Checking MFA

- Testing model-faithful acyclicity for a set of rules Σ

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

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

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

     $\rightsquigarrow$ 2EXPTIME-complete

  3. Rules from Horn-$\mathcal{SRI}$

     $\rightsquigarrow$ EXPTIME-hard

  4. Rules from Horn-$\mathcal{SHIQ}$

     $\rightsquigarrow$ 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

Model-summarising Acyclicity

- Track rule applications just 'faithfully' enough

Example

$$A(u) \to R(u, f(u)) \land B(f(u))$$

$$B(v) \to R(v, g(v)) \land C(g(v))$$

$$R(w, z) \land B(z) \to A(w)$$

Model-summarising Acyclicity

- Track rule applications just 'faithfully' enough

Example

$A(u) \rightarrow R(u, \underline{f(u)}) \wedge B(\underline{f(u)})$

$B(v) \rightarrow R(v, \underline{g(v)}) \wedge C(\underline{g(v)})$

$R(w, z) \wedge B(z) \rightarrow A(w)$

Model-summarising Acyclicity

- Track rule applications just 'faithfully' enough

Example

$A(u) \rightarrow R(u, c_1) \wedge B(c_1)$

$B(v) \rightarrow R(v, c_2) \wedge C(c_2)$

$R(w, z) \wedge B(z) \rightarrow A(w)$

MODEL-SUMMARISING ACYCLICITY

- Track rule applications just ‘faithfully’ enough

EXAMPLE

$$A(u) \to R(u, c_1) \wedge B(c_1) \wedge S(u, c_1) \wedge F_{c_1}(c_1)$$

$$B(v) \to R(v, c_2) \wedge C(c_2) \wedge S(v, c_2) \wedge F_{c_2}(c_2)$$

$$R(w, z) \wedge B(z) \to A(w)$$

MODEL-SUMMARISING ACYCLICITY

- Track rule applications just 'faithfully' enough

EXAMPLE

$$A(u) \to R(u, c_1) \wedge B(c_1) \wedge S(u, c_1) \wedge F_{c_1}(c_1)$$

$$B(v) \to R(v, c_2) \wedge C(c_2) \wedge S(v, c_2) \wedge F_{c_2}(c_2)$$

$$R(w, z) \wedge B(z) \to A(w)$$

$$S(x, y) \to D(x, y)$$

$$D(x, y) \wedge S(y, z) \to D(x, z)$$

$$F_{c_1}(x) \wedge D(x, y) \wedge F_{c_1}(y) \to \mathit{Cycle}$$

$$F_{c_2}(x) \wedge D(x, y) \wedge F_{c_2}(y) \to \mathit{Cycle}$$

Model-summarising Acyclicity

- Track rule applications just 'faithfully' enough

Example

$A(u) \rightarrow R(u, c_1) \wedge B(c_1) \wedge S(u, c_1) \wedge F_{c_1}(c_1)$

$B(v) \rightarrow R(v, c_2) \wedge C(c_2) \wedge S(v, c_2) \wedge F_{c_2}(c_2)$

$R(w, z) \wedge B(z) \rightarrow A(w)$

$S(x, y) \rightarrow D(x, y)$

$D(x, y) \wedge S(y, z) \rightarrow D(x, z)$

$F_{c_1}(x) \wedge D(x, y) \wedge F_{c_1}(y) \rightarrow Cycle$

$F_{c_2}(x) \wedge D(x, y) \wedge F_{c_2}(y) \rightarrow Cycle$

- For Σ a set of rules, Σ is MSA if $I^*_\Sigma \cup MSA(\Sigma) \not\models Cycle$

Model-summarising Acyclicity

- Track rule applications just ‘faithfully’ enough

Example

$A(u) \rightarrow R(u, c_1) \wedge B(c_1) \wedge S(u, c_1) \wedge F_{c_1}(c_1)$

$B(v) \rightarrow R(v, c_2) \wedge C(c_2) \wedge S(v, c_2) \wedge F_{c_2}(c_2)$

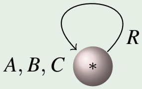
$R(w, z) \wedge B(z) \rightarrow A(w)$

$S(x, y) \rightarrow D(x, y)$

$D(x, y) \wedge S(y, z) \rightarrow D(x, z)$

$F_{c_1}(x) \wedge D(x, y) \wedge F_{c_1}(y) \rightarrow \mathit{Cycle}$

$F_{c_2}(x) \wedge D(x, y) \wedge F_{c_2}(y) \rightarrow \mathit{Cycle}$



- For Σ a set of rules, Σ is MSA if $I^*_\Sigma \cup \mathit{MSA}(\Sigma) \not\models \mathit{Cycle}$

Model-summarising Acyclicity

- Track rule applications just 'faithfully' enough

Example

$A(u) \to R(u, c_1) \wedge B(c_1) \wedge S(u, c_1) \wedge F_{c_1}(c_1)$

$B(v) \to R(v, c_2) \wedge C(c_2) \wedge S(v, c_2) \wedge F_{c_2}(c_2)$

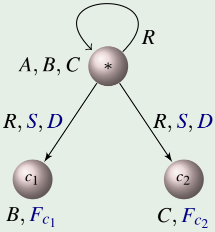
$R(w, z) \wedge B(z) \to A(w)$

$S(x, y) \to D(x, y)$

$D(x, y) \wedge S(y, z) \to D(x, z)$

$F_{c_1}(x) \wedge D(x, y) \wedge F_{c_1}(y) \to Cycle$

$F_{c_2}(x) \wedge D(x, y) \wedge F_{c_2}(y) \to Cycle$



- For Σ a set of rules, Σ is MSA if $I^*_\Sigma \cup MSA(\Sigma) \not\models Cycle$

Model-summarising Acyclicity

- Track rule applications just 'faithfully' enough

Example

$A(u) \rightarrow R(u, c_1) \wedge B(c_1) \wedge S(u, c_1) \wedge F_{c_1}(c_1)$

$B(v) \rightarrow R(v, c_2) \wedge C(c_2) \wedge S(v, c_2) \wedge F_{c_2}(c_2)$

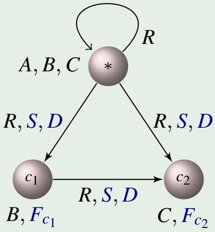
$R(w, z) \wedge B(z) \rightarrow A(w)$

$S(x, y) \rightarrow D(x, y)$

$D(x, y) \wedge S(y, z) \rightarrow D(x, z)$

$F_{c_1}(x) \wedge D(x, y) \wedge F_{c_1}(y) \rightarrow Cycle$

$F_{c_2}(x) \wedge D(x, y) \wedge F_{c_2}(y) \rightarrow Cycle$



- For Σ a set of rules, Σ is MSA if $I^*_\Sigma \cup MSA(\Sigma) \not\models Cycle$

Model-summarising Acyclicity

- Track rule applications just 'faithfully' enough

Example

$A(u) \rightarrow R(u, c_1) \wedge B(c_1) \wedge S(u, c_1) \wedge F_{c_1}(c_1)$

$B(v) \rightarrow R(v, c_2) \wedge C(c_2) \wedge S(v, c_2) \wedge F_{c_2}(c_2)$

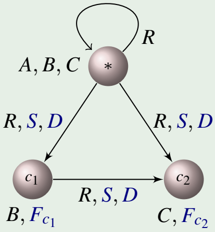
$R(w, z) \wedge B(z) \rightarrow A(w)$

$S(x, y) \rightarrow D(x, y)$

$D(x, y) \wedge S(y, z) \rightarrow D(x, z)$

$F_{c_1}(x) \wedge D(x, y) \wedge F_{c_1}(y) \rightarrow Cycle$

$F_{c_2}(x) \wedge D(x, y) \wedge F_{c_2}(y) \rightarrow Cycle$



- For Σ a set of rules, Σ is MSA if $I^*_\Sigma \cup MSA(\Sigma) \not\models Cycle$
- 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 Σ

Cost of Checking MSA

- Testing model-faithful acyclicity for a set of rules Σ
  1. Rules of the form $\varphi(\vec{x}, \vec{z}) \rightarrow \exists \vec{y}.\psi(\vec{x}, \vec{y})$ (no restriction)

     $\rightsquigarrow$ EXPTIME-complete

Cost of Checking MSA

- Testing model-faithful acyclicity for a set of rules Σ

  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

Cost of Checking MSA

- Testing model-faithful acyclicity for a set of rules Σ

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-$\mathcal{SHIQ}$

   $\leadsto$ PTIME-complete

Cost of Checking MSA

- Testing model-faithful acyclicity for a set of rules Σ

  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-$\mathcal{SHIQ}$

     $\rightsquigarrow$ PTIME-complete

- Horn-$\mathcal{SHIQ}$ TBoxes can be checked in PTIME for MSA before *potential* materialisation-based query answering

ACYCLICITY CONDITIONS (PARTIAL) TAXONOMY

JA $\subsetneq$ SWA MSA MFA

Acyclicity Conditions (Partial) Taxonomy

- Our contributions:

JA	$\subsetneq$	SWA		MSA		MFA

ACYCLICITY CONDITIONS (PARTIAL) TAXONOMY

- Our contributions:
  1. MSA strictly subsumes SWA

$$\text{JA} \subsetneq \text{SWA} \subsetneq \text{MSA} \qquad \text{MFA}$$

Acyclicity Conditions (Partial) Taxonomy

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

$$\text{JA} \subsetneq \text{SWA} \subsetneq \text{MSA} \subsetneq \text{MFA}$$

Example

$$A(x) \rightarrow \exists y.R(x,y) \wedge B(y)$$
$$B(x) \rightarrow \exists y.S(x,y) \wedge T(y,x)$$
$$A(z) \wedge S(z,x) \rightarrow C(x)$$
$$C(z) \wedge T(z,x) \rightarrow A(x)$$

MFA but not MSA

Acyclicity Conditions (Partial) Taxonomy

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

JA $\subsetneq$ SWA $\subsetneq$ MSA $\subsetneq$ MFA

Example

$$A(x) \rightarrow \exists y.R(x,y) \wedge B(y)$$
$$B(x) \rightarrow \exists y.S(x,y) \wedge T(y,x)$$
$$A(z) \wedge S(z,x) \rightarrow C(x)$$
$$C(z) \wedge T(z,x) \rightarrow A(x)$$

MFA but not MSA

- MSA and MFA coincide in experimental evaluation of DL ontologies

Outline

Translating DLs into rules

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

$A \sqsubseteq \exists R.B$	$A(x) \to \exists y.R(x,y) \land B(y)$
$A \sqsubseteq \leq 1\, R.B$	$A(z) \land R(z,x_1) \land B(x_1) \land R(z,x_2)$
	$\land B(x_2) \to x_1 \approx x_2$
$A \sqcap B \sqsubseteq C$	$A(x) \land B(x) \to C(x)$
$A \sqsubseteq \forall R.B$	$A(z) \land R(z,x) \to B(x)$
$R \sqsubseteq S$	$R(x_1,x_2) \to S(x_1,x_2)$
$R \circ S \sqsubseteq T$	$R(x_1,z) \land S(z,x_2) \to T(x_1,x_2)$

TRANSLATING DLS INTO RULES

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

$A \sqsubseteq \exists R.B$	$A(x) \rightarrow \exists y.R(x,y) \wedge B(y)$
$A \sqsubseteq \leq 1\, R.B$	$A(z) \wedge R(z,x_1) \wedge B(x_1) \wedge R(z,x_2)$
	$\wedge\ B(x_2) \rightarrow x_1 \approx x_2$
$A \sqcap B \sqsubseteq C$	$A(x) \wedge B(x) \rightarrow C(x)$
$A \sqsubseteq \forall R.B$	$A(z) \wedge R(z,x) \rightarrow B(x)$
$R \sqsubseteq S$	$R(x_1,x_2) \rightarrow S(x_1,x_2)$
$R \circ S \sqsubseteq T$	$R(x_1,z) \wedge S(z,x_2) \rightarrow T(x_1,x_2)$

- Equality is handled with a modification of the singularisation [Marnette, PODS, 2009] technique

DL Query Answering under Acyclicity

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

DL Query Answering under Acyclicity

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

- Does acyclicity affect complexity for DL Query Answering?

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

   $\rightsquigarrow$ Deciding $\mathcal{T} \cup \mathcal{A} \models Q$ is PSPACE-complete

DL Query Answering under Acyclicity

- Answering conjunctive queries for the DL Horn-$\mathcal{SHIQ}$ is EXPTIME-complete [Eiter et al., 2008]
- Does acyclicity affect complexity for DL Query Answering?

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

   $\rightsquigarrow$ Deciding $\mathcal{T} \cup \mathcal{A} \models Q$ is PSPACE-complete
2. Horn-$\mathcal{SRI}$ TBox $\mathcal{T}$ and ABox $\mathcal{A}$
   $\mathcal{T}$ is weakly acyclic
   F set of facts

   $\rightsquigarrow$ Deciding $\mathcal{T} \cup \mathcal{A} \models F$ is EXPTIME-hard

Outline

Acyclicity Tests

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

ACYCLICITY TESTS

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

Existential rules	Total	MSA	JA	WA
< 100	70	64	64	64
100–1K	33	30	30	23
1K–5K	20	14	14	12
5K–12K	14	11	6	6
12K–160K	12	5	3	3
All sizes	149	124	117	108

ACYCLICITY TESTS

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

Existential rules	Total	MSA	JA	WA
< 100	70	64	64	64
100–1K	33	30	30	23
1K–5K	20	14	14	12
5K–12K	14	11	6	6
12K–160K	12	5	3	3
All sizes	149	124	117	108

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

Acyclicity Tests

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

Existential rules	Total	MSA	JA	WA
< 100	70	64	64	64
100–1K	33	30	30	23
1K–5K	20	14	14	12
5K–12K	14	11	6	6
12K–160K	12	5	3	3
All sizes	149	124	117	108

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

ACYCLICITY TESTS

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

Existential rules	Total	MSA	JA	WA
< 100	70	64	64	64
100–1K	33	30	30	23
1K–5K	20	14	14	12
5K–12K	14	11	6	6
12K–160K	12	5	3	3
All sizes	149	124	117	108

- 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

Materialisation Tests

- Computed materialisation of acyclic TBoxes

Depth	#	generated size		materialisation size	
		max	avg	max	avg
< 5	82	27	2	35	5
5–9	13	37	11	41	13
10–80	14	281	51	283	53

Depth $=$ length of function symbol nesting

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

$$\text{materialisation size} = \frac{\text{\# facts in materialisation}}{\text{\# facts in initial ABox}}$$

Materialisation Tests

- Computed materialisation of acyclic TBoxes

Depth	#	generated size		materialisation size	
		max	avg	max	avg
< 5	82	27	2	35	5
5–9	13	37	11	41	13
10–80	14	281	51	283	53

Depth $=$ length of function symbol nesting

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

$$\text{materialisation size} = \frac{\text{\# facts in materialisation}}{\text{\# facts in initial ABox}}$$

- For ontologies with small depths materialisation seems practically feasible

Materialisation Tests

- Computed materialisation of acyclic TBoxes

Depth	#	generated size		materialisation size	
		max	avg	max	avg
< 5	82	27	2	35	5
5–9	13	37	11	41	13
10–80	14	281	51	283	53

Depth = length of function symbol nesting

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

$$\text{materialisation size} = \frac{\text{\# facts in materialisation}}{\text{\# facts in initial ABox}}$$

- 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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

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

	Horn-$\mathcal{SHIQ}$	bounded arity	no restriction
MSA	PTime-complete	coNP-complete	ExpTime-complete
MFA	PSpace-complete	2ExpTime-complete	2ExpTime-complete

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