Modelling Structured Domains Using Description Graphs and Logic Programming

Despoina Magka, Boris Motik and Ian Horrocks

Department of Computer Science, University of Oxford

May 29, 2012

OUTLINE

1 MOTIVATION

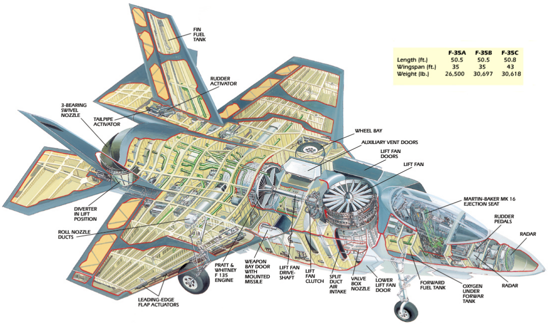
2 DGLPS, IMPLEMENTATION AND OVERVIEW

Modelling Structured Domains with OWL

- OWL used for the representation of complex structures:

MODELLING STRUCTURED DOMAINS WITH OWL

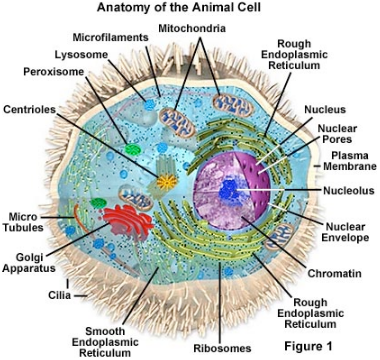
- OWL used for the representation of complex structures:
  - Aerospace



	F-35A	F-35B	F-35C
Length (ft.)	50.5	50.5	50.8
Wingspan (ft.)	35	35	43
Weight (lb.)	26,500	30,697	30,618

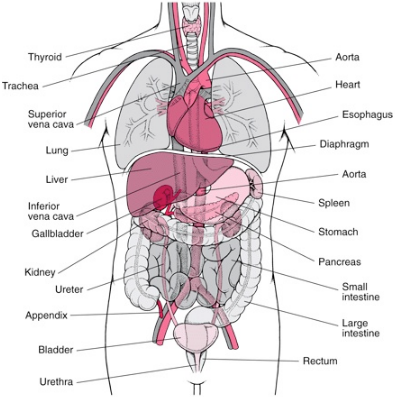
Modelling Structured Domains with OWL

- OWL used for the representation of complex structures:
  - Aerospace
  - Cellular biology



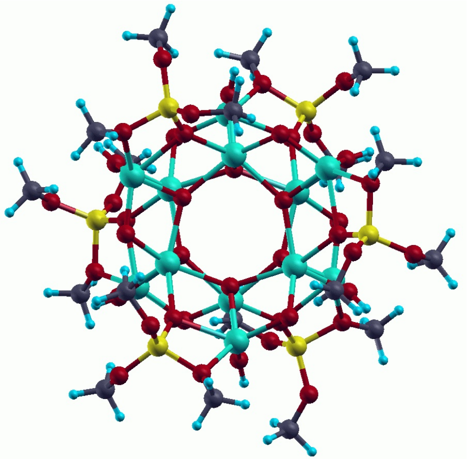
Modelling Structured Domains with OWL

- OWL used for the representation of complex structures:
  - Aerospace
  - Cellular biology
  - Human anatomy



Modelling Structured Domains with OWL

- OWL used for the representation of complex structures:
  - Aerospace
  - Cellular biology
  - Human anatomy
  - Molecules



The ChEBI Ontology

- OWL ontology Chemical Entities of Biological Interest

THE CHEBI ONTOLOGY

- OWL ontology Chemical Entities of Biological Interest
  - Freely accessible dictionary of ‘small’ molecular entities

The ChEBI Ontology

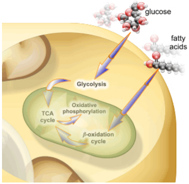
- OWL ontology Chemical Entities of Biological Interest
  - Freely accessible dictionary of ‘small’ molecular entities
  - High quality annotation and taxonomy of chemicals

THE CHEBI ONTOLOGY

- OWL ontology Chemical Entities of Biological Interest
  - Freely accessible dictionary of ‘small’ molecular entities
  - High quality annotation and taxonomy of chemicals
  - Interoperability between researchers

The ChEBI Ontology

- OWL ontology Chemical Entities of Biological Interest
  - Freely accessible dictionary of ‘small’ molecular entities
  - High quality annotation and taxonomy of chemicals
  - Interoperability between researchers
  - Drug discovery and elucidation of metabolic pathways



AUTOMATE CHEMICAL CLASSIFICATION

- ChEBI is manually incremented

AUTOMATE CHEMICAL CLASSIFICATION

- ChEBI is manually incremented

- Currently contains approx. 28,000 fully annotated entities

Automate Chemical Classification

- ChEBI is manually incremented
- Currently contains approx. 28,000 fully annotated entities
- Grows at a rate of ~1,500 entities per curator per year

AUTOMATE CHEMICAL CLASSIFICATION

- ChEBI is manually incremented
- Currently contains approx. 28,000 fully annotated entities
- Grows at a rate of ~1,500 entities per curator per year
- Biologically interesting entities possibly > 1,000,000

Automate Chemical Classification

- ChEBI is manually incremented
- Currently contains approx. 28,000 fully annotated entities
- Grows at a rate of ~1,500 entities per curator per year
- Biologically interesting entities possibly > 1,000,000
- Each new molecule is subsumed by several chemical classes

AUTOMATE CHEMICAL CLASSIFICATION

- ChEBI is manually incremented
- Currently contains approx. 28,000 fully annotated entities
- Grows at a rate of ~1,500 entities per curator per year
- Biologically interesting entities possibly > 1,000,000
- Each new molecule is subsumed by several chemical classes
    - Is dinitrogen inorganic?

AUTOMATE CHEMICAL CLASSIFICATION

- ChEBI is manually incremented

- Currently contains approx. 28,000 fully annotated entities

- Grows at a rate of ~1,500 entities per curator per year

- Biologically interesting entities possibly > 1,000,000

- Each new molecule is subsumed by several chemical classes
    - Is dinitrogen inorganic?
    - Does cyclobutane contain a four-membered ring?

AUTOMATE CHEMICAL CLASSIFICATION

- ChEBI is manually incremented
- Currently contains approx. 28,000 fully annotated entities
- Grows at a rate of ~1,500 entities per curator per year
- Biologically interesting entities possibly > 1,000,000
- Each new molecule is subsumed by several chemical classes
  - Is dinitrogen inorganic?
  - Does cyclobutane contain a four-membered ring?
  - Is acetylene a hydrocarbon?

AUTOMATE CHEMICAL CLASSIFICATION

- ChEBI is manually incremented

- Currently contains approx. 28,000 fully annotated entities

- Grows at a rate of ~1,500 entities per curator per year

- Biologically interesting entities possibly > 1,000,000

- Each new molecule is subsumed by several chemical classes
    - Is dinitrogen inorganic?
    - Does cyclobutane contain a four-membered ring?
    - Is acetylene a hydrocarbon?
    - Does benzaldehyde contain a benzene ring?

Automate Chemical Classification

- ChEBI is manually incremented
- Currently contains approx. 28,000 fully annotated entities
- Grows at a rate of ~1,500 entities per curator per year
- Biologically interesting entities possibly > 1,000,000
- Each new molecule is subsumed by several chemical classes
  - Is dinitrogen inorganic?
  - Does cyclobutane contain a four-membered ring?
  - Is acetylene a hydrocarbon?
  - Does benzaldehyde contain a benzene ring?
- Speed up curating tasks with automated reasoning tools

AUTOMATE CHEMICAL CLASSIFICATION

- ChEBI is manually incremented
- Currently contains approx. 28,000 fully annotated entities
- Grows at a rate of ~1,500 entities per curator per year
- Biologically interesting entities possibly > 1,000,000
- Each new molecule is subsumed by several chemical classes
  - Is dinitrogen inorganic? ⇝Yes
  - Does cyclobutane contain a four-membered ring? ⇝Yes
  - Is acetylene a hydrocarbon? ⇝Yes
  - Does benzaldehyde contain a benzene ring? ⇝Yes
- Speed up curating tasks with automated reasoning tools

(MIS)REPRESENTING RINGS WITH OWL

- Chemical compounds with rings are highly frequent

(MIS)REPRESENTING RINGS WITH OWL

- Chemical compounds with rings are highly frequent
- Fundamental inability of OWL to represent cycles

(MIS)REPRESENTING RINGS WITH OWL

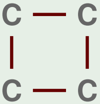
- Chemical compounds with rings are highly frequent
- Fundamental inability of OWL to represent cycles
- At least one tree-shaped model for each consistent OWL knowledge base

(MIS)REPRESENTING RINGS WITH OWL

- Chemical compounds with rings are highly frequent
- Fundamental inability of OWL to represent cycles
- At least one tree-shaped model for each consistent OWL knowledge base

EXAMPLE

$$\text{Cyclobutane} \sqsubseteq \exists(=4)\text{hasAtom}.(\text{Carbon} \sqcap \exists(=2)\text{hasBond}.\text{Carbon})$$

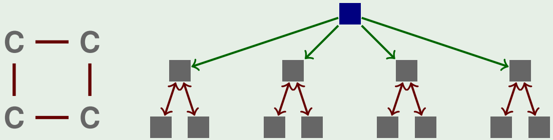


(Mis)Representing Rings with OWL

- Chemical compounds with rings are highly frequent
- Fundamental inability of OWL to represent cycles
- At least one tree-shaped model for each consistent OWL knowledge base

Example

$$\text{Cyclobutane} \sqsubseteq \exists(=4)\text{hasAtom}.(\text{Carbon} \sqcap \exists(=2)\text{hasBond}.\text{Carbon})$$

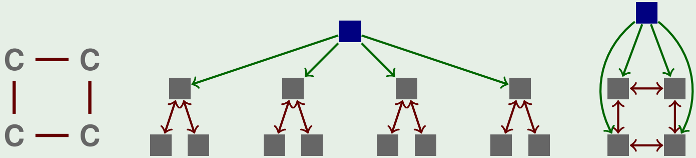


(MIS)REPRESENTING RINGS WITH OWL

- Chemical compounds with rings are highly frequent
- Fundamental inability of OWL to represent cycles
- At least one tree-shaped model for each consistent OWL knowledge base

EXAMPLE

$$\text{Cyclobutane} \sqsubseteq \exists(=4)\text{hasAtom}.(\text{Carbon} \sqcap \exists(=2)\text{hasBond}.\text{Carbon})$$

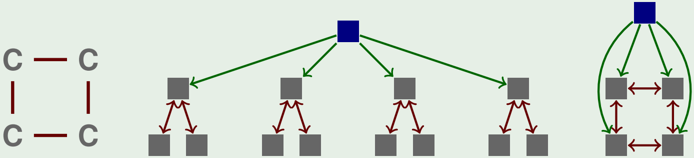


(MIS)REPRESENTING RINGS WITH OWL

- Chemical compounds with rings are highly frequent
- Fundamental inability of OWL to represent cycles
- At least one tree-shaped model for each consistent OWL knowledge base

EXAMPLE

$$\text{Cyclobutane} \sqsubseteq \exists(=4)\text{hasAtom}.(\text{Carbon} \sqcap \exists(=2)\text{hasBond}.\text{Carbon})$$



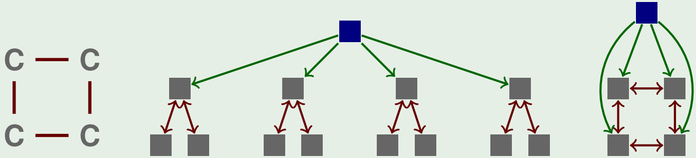
- OWL-based reasoning support

(MIS)REPRESENTING RINGS WITH OWL

- Chemical compounds with rings are highly frequent
- Fundamental inability of OWL to represent cycles
- At least one tree-shaped model for each consistent OWL knowledge base

EXAMPLE

$$\text{Cyclobutane} \sqsubseteq \exists(=4)\text{hasAtom}.(\text{Carbon} \sqcap \exists(=2)\text{hasBond}.\text{Carbon})$$



- OWL-based reasoning support
  - Does cyclobutane contain a four-membered ring? ✘
  - Does benzaldehyde contain a benzene ring? ✘

OWL Extensions

- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]

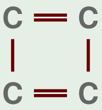
OWL Extensions

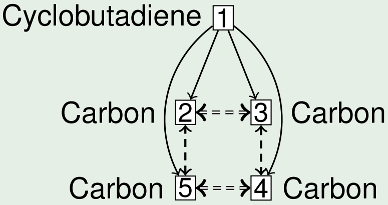
- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]
- A Description Graph represents structures by means of a directed labeled graph

OWL Extensions

- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]
- A Description Graph represents structures by means of a directed labeled graph

Example

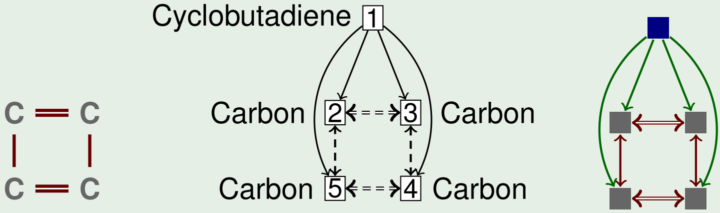




OWL Extensions

- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]
- A Description Graph represents structures by means of a directed labeled graph

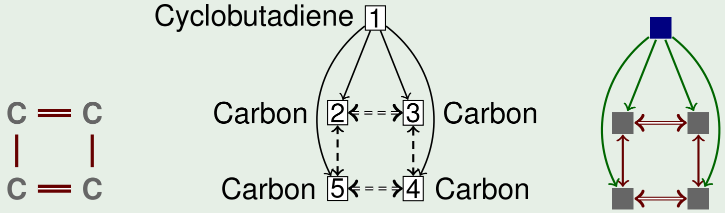
Example



OWL Extensions

- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]
- A Description Graph represents structures by means of a directed labeled graph

Example

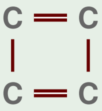


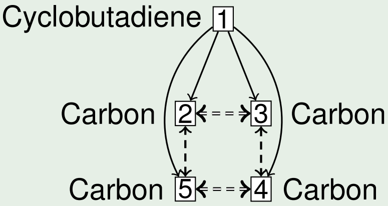
- Does cyclobutadiene have a conjugated four-membered ring?

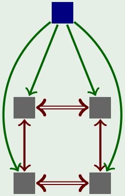
OWL EXTENSIONS

- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]
- A Description Graph represents structures by means of a directed labeled graph

EXAMPLE





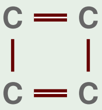


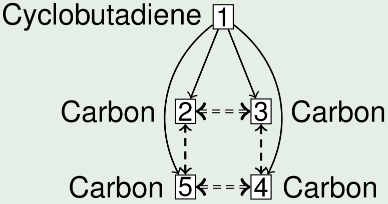
- Does cyclobutadiene have a conjugated four-membered ring? ✓

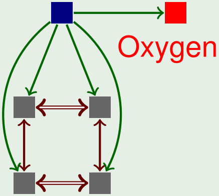
OWL Extensions

- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]
- A Description Graph represents structures by means of a directed labeled graph

Example



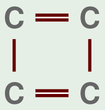


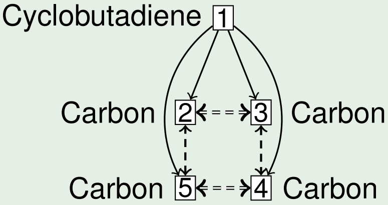


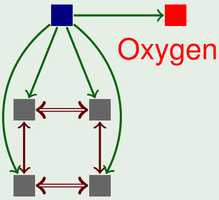
OWL Extensions

- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]
- A Description Graph represents structures by means of a directed labeled graph

Example





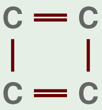


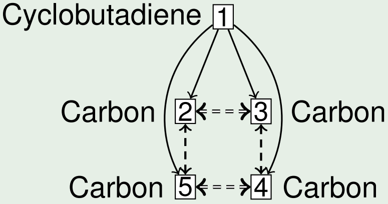
$\forall \mathit{hasAtom}.(\mathit{Carbon} \sqcup \mathit{Hydrogen}) \sqsubseteq \mathit{Hydrocarbon}$

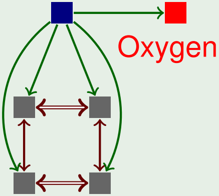
OWL Extensions

- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]
- A Description Graph represents structures by means of a directed labeled graph

Example







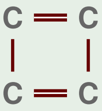
$\forall \mathrm{hasAtom}.(\mathrm{Carbon} \sqcup \mathrm{Hydrogen}) \sqsubseteq \mathrm{Hydrocarbon}$

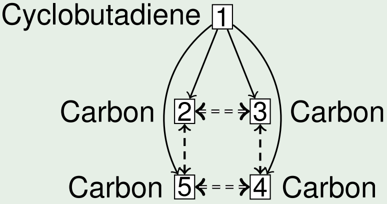
- Is cyclobutadiene a hydrocarbon?

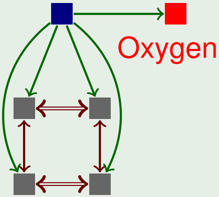
OWL Extensions

- Limitation of OWL to represent cycles (partially) remedied by extension of OWL with Description Graphs and rules [Motik et al., 2009]
- A Description Graph represents structures by means of a directed labeled graph

Example







$\forall$hasAtom.(Carbon $\sqcup$ Hydrogen) $\sqsubseteq$ Hydrocarbon

- Is cyclobutadiene a hydrocarbon? ✘

RESULTS OVERVIEW

- Key idea:

Results Overview

- Key idea:
  - Switch from first-order logic to logic programming semantics

RESULTS OVERVIEW

- Key idea:
  - Switch from first-order logic to logic programming semantics
  - Use negation-as-failure to derive non-monotonic inferences

Results Overview

- Key idea:
  - Switch from first-order logic to logic programming semantics
  - Use negation-as-failure to derive non-monotonic inferences
- Expressive decidable logic-based formalism for modelling structured entities: Description Graph Logic Programs (DGLPs)

Results Overview

- Key idea:
  - Switch from first-order logic to logic programming semantics
  - Use negation-as-failure to derive non-monotonic inferences
- Expressive decidable logic-based formalism for modelling structured entities: Description Graph Logic Programs (DGLPs)

DGLPs	all cycles	CWA
OWL+DGs+rules	some cycles	OWA
OWL	no cycles	OWA

Results Overview

- Key idea:
  - Switch from first-order logic to logic programming semantics
  - Use negation-as-failure to derive non-monotonic inferences
- Expressive decidable logic-based formalism for modelling structured entities: Description Graph Logic Programs (DGLPs)

DGLPs	all cycles	CWA
OWL+DGs+rules	some cycles	OWA
OWL	no cycles	OWA

Negation-as-failure ↔ Closed-world assumption ↔ Missing information treated as *false*

Results Overview

- Key idea:
  - Switch from first-order logic to logic programming semantics
  - Use negation-as-failure to derive non-monotonic inferences
- Expressive decidable logic-based formalism for modelling structured entities: Description Graph Logic Programs (DGLPs)

DGLPs	all cycles	CWA
OWL+DGs+rules	some cycles	OWA
OWL	no cycles	OWA

Negation-as-failure ↔ Closed-world assumption ↔ Missing information treated as *false*

Classical negation ↔ Open-world assumption ↔ Missing information treated as *not known*

Results Overview

- Key idea:
  - Switch from first-order logic to logic programming semantics
  - Use negation-as-failure to derive non-monotonic inferences
- Expressive decidable logic-based formalism for modelling structured entities: Description Graph Logic Programs (DGLPs)

DGLPs	all cycles	CWA
OWL+DGs+rules	some cycles	OWA
OWL	no cycles	OWA

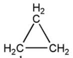
- Prototypical implementation of DGLPs with application in structure-based chemical classification

Results Overview

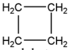
- Key idea:
  - Switch from first-order logic to logic programming semantics
  - Use negation-as-failure to derive non-monotonic inferences
- Expressive decidable logic-based formalism for modelling structured entities: Description Graph Logic Programs (DGLPs)

DGLPs	all cycles	CWA
OWL+DGs+RULES	some cycles	OWA
OWL	no cycles	OWA

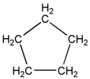
- Prototypical implementation of DGLPs with application in structure-based chemical classification



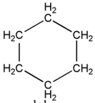
cyclopropane



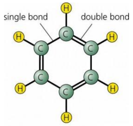
cyclobutane



cyclopentane



cyclohexane



OUTLINE

1 MOTIVATION

2 DGLPS, IMPLEMENTATION AND OVERVIEW

What is a DGLP Ontology?

- The syntactic objects of a DGLP ontology:

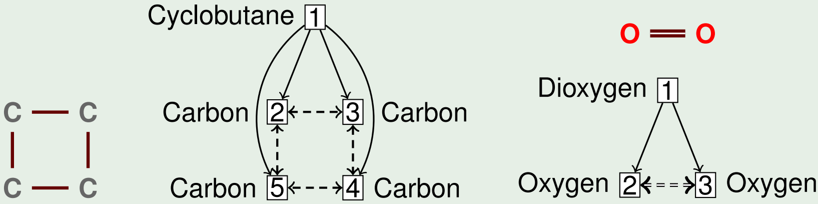
What is a DGLP Ontology?

- The syntactic objects of a DGLP ontology:
    - Description graphs

What is a DGLP Ontology?

- The syntactic objects of a DGLP ontology:
  - Description graphs

Example



WHAT IS A DGLP ONTOLOGY?

- The syntactic objects of a DGLP ontology:
    - Description graphs
    - Function-free FOL Horn rules

WHAT IS A DGLP ONTOLOGY?

- The syntactic objects of a DGLP ontology:
  - Description graphs
  - Function-free FOL Horn rules

EXAMPLE

$$\begin{aligned} \mathsf{Bond}(x,y) &\rightarrow \mathsf{Bond}(y,x) \\ \mathsf{SingleBond}(x,y) &\rightarrow \mathsf{Bond}(x,y) \end{aligned}$$

WHAT IS A DGLP ONTOLOGY?

- The syntactic objects of a DGLP ontology:
  - Description graphs
  - Function-free FOL Horn rules

EXAMPLE

$$\begin{aligned} \mathsf{Bond}(x, y) &\rightarrow \mathsf{Bond}(y, x) \\ \mathsf{SingleBond}(x, y) &\rightarrow \mathsf{Bond}(x, y) \end{aligned}$$

  - Rules with negation-as-failure

What is a DGLP Ontology?

- The syntactic objects of a DGLP ontology:
  - Description graphs
  - Function-free FOL Horn rules

Example

$$\begin{array}{lcl} \mathsf{Bond}(x, y) & \rightarrow & \mathsf{Bond}(y, x) \\ \mathsf{SingleBond}(x, y) & \rightarrow & \mathsf{Bond}(x, y) \end{array}$$

  - Rules with negation-as-failure

Example

$$\begin{array}{lcl} \mathsf{HasAtom}(x, y) \land \mathsf{Carbon}(y) & \rightarrow & \mathsf{HasCarbon}(x) \\ \mathsf{Molecule}(x) \land \textbf{not } \mathsf{HasCarbon}(x) & \rightarrow & \mathsf{Inorganic}(x) \end{array}$$

WHAT IS A DGLP ONTOLOGY?

- The syntactic objects of a DGLP ontology:
  - Description graphs
  - Function-free FOL Horn rules

EXAMPLE

$$\begin{array}{lcl} \mathsf{Bond}(x,y) & \rightarrow & \mathsf{Bond}(y,x) \\ \mathsf{SingleBond}(x,y) & \rightarrow & \mathsf{Bond}(x,y) \end{array}$$

- Rules with negation-as-failure

EXAMPLE

$$\begin{array}{lcl} \mathsf{HasAtom}(x,y) \wedge \mathsf{Carbon}(y) & \rightarrow & \mathsf{HasCarbon}(x) \\ \mathsf{Molecule}(x) \wedge \textbf{not } \mathsf{HasCarbon}(x) & \rightarrow & \mathsf{Inorganic}(x) \end{array}$$

- Facts

What is a DGLP Ontology?

- The syntactic objects of a DGLP ontology:
  - Description graphs
  - Function-free FOL Horn rules

Example

$$\begin{aligned} \mathsf{Bond}(x,y) &\rightarrow \mathsf{Bond}(y,x)\\ \mathsf{SingleBond}(x,y) &\rightarrow \mathsf{Bond}(x,y) \end{aligned}$$

-
  - Rules with negation-as-failure

Example

$$\begin{aligned} \mathsf{HasAtom}(x,y) \wedge \mathsf{Carbon}(y) &\rightarrow \mathsf{HasCarbon}(x)\\ \mathsf{Molecule}(x) \wedge \textbf{not}\ \mathsf{HasCarbon}(x) &\rightarrow \mathsf{Inorganic}(x) \end{aligned}$$

-
  - Facts

Example

$\mathsf{Cyclobutane}(c_1),\quad \mathsf{Dinitrogen}(c_2), \ldots$

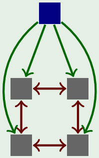
ENCODING DESCRIPTION GRAPHS

- Translate DGs into logic programs with function symbols

Encoding Description Graphs

- Translate DGs into logic programs with function symbols

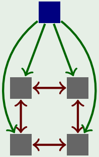
Example



Encoding Description Graphs

- Translate DGs into logic programs with function symbols

Example



$$\begin{aligned}
\mathrm{Cyclobutane}(x) &\rightarrow G_{cb}(x, f_1(x), f_2(x), f_3(x), f_4(x)) \\
G_{cb}(x, y_1, y_2, y_3, y_4) &\rightarrow \mathrm{Cyclobutane}(x) \wedge \\
&\quad \mathrm{Carbon}(y_1) \wedge \mathrm{Carbon}(y_2) \wedge \\
&\quad \mathrm{Carbon}(y_3) \wedge \mathrm{Carbon}(y_4) \wedge \\
&\quad \mathrm{HasAtom}(x, y_1) \wedge \mathrm{Bond}(y_1, y_2) \wedge \\
&\quad \mathrm{HasAtom}(x, y_2) \wedge \mathrm{Bond}(y_2, y_3) \wedge \\
&\quad \mathrm{HasAtom}(x, y_3) \wedge \mathrm{Bond}(y_3, y_4) \wedge \\
&\quad \mathrm{HasAtom}(x, y_4) \wedge \mathrm{Bond}(y_4, y_1)
\end{aligned}$$

Classifying Objects

Example

$$\mathsf{Molecule}(x) \wedge \mathsf{HasAtom}(x, y) \wedge \textbf{not } \mathsf{Carbon}(y) \wedge \textbf{not } \mathsf{Hydrogen}(y) \rightarrow \mathsf{NotHydroCarbon}(x)$$

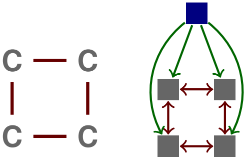
$$\mathsf{Molecule}(x) \wedge \textbf{not } \mathsf{NotHydroCarbon}(x) \rightarrow \mathsf{HydroCarbon}(x)$$

Classifying Objects

Example

$$\text{Molecule}(x) \wedge \text{HasAtom}(x, y) \wedge \textbf{not } \text{Carbon}(y) \wedge \textbf{not } \text{Hydrogen}(y) \rightarrow \text{NotHydroCarbon}(x)$$

$$\text{Molecule}(x) \wedge \textbf{not } \text{NotHydroCarbon}(x) \rightarrow \text{HydroCarbon}(x)$$

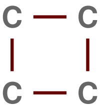


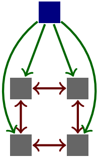
Classifying Objects

Example

$$\text{Molecule}(x) \wedge \text{HasAtom}(x, y) \wedge \textbf{not } \text{Carbon}(y) \wedge \textbf{not } \text{Hydrogen}(y) \rightarrow \text{NotHydroCarbon}(x)$$

$$\text{Molecule}(x) \wedge \textbf{not } \text{NotHydroCarbon}(x) \rightarrow \text{HydroCarbon}(x)$$





- Is cyclobutane a hydrocarbon? ✓

Classifying Objects

Example

$$\mathsf{Molecule}(\mathsf{x}) \wedge \bigwedge_{1 \leq \mathsf{i} \leq 4} \mathsf{HasAtom}(\mathsf{x}, \mathsf{y_i}) \wedge \bigwedge_{1 \leq \mathsf{i} \leq 3} \mathsf{Bond}(\mathsf{y_i}, \mathsf{y_{i+1}}) \wedge$$

$$\mathsf{Bond}(\mathsf{y_4}, \mathsf{y_1}) \bigwedge_{1 \leq i < j \leq 4} \textbf{not } \mathsf{y_i} = \mathsf{y_j}$$

$$\rightarrow \mathsf{MoleculeWith4MemberedRing}(\mathsf{x})$$

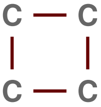
Classifying Objects

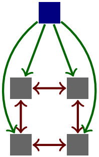
Example

$$\mathsf{Molecule}(\mathsf{x}) \wedge \bigwedge_{1\leq \mathsf{i}\leq 4} \mathsf{HasAtom}(\mathsf{x},\mathsf{y_i}) \wedge \bigwedge_{1\leq \mathsf{i}\leq 3} \mathsf{Bond}(\mathsf{y_i},\mathsf{y_{i+1}}) \wedge$$

$$\mathsf{Bond}(\mathsf{y_4},\mathsf{y_1}) \bigwedge_{1\leq i<j\leq 4} \textbf{not } \mathsf{y_i} = \mathsf{y_j}$$

$$\rightarrow \mathsf{MoleculeWith4MemberedRing}(\mathsf{x})$$





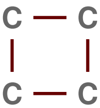
Classifying Objects

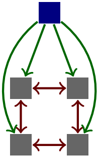
Example

$$\mathrm{Molecule}(x) \wedge \bigwedge_{1\leq i\leq 4} \mathrm{HasAtom}(x, y_i) \wedge \bigwedge_{1\leq i\leq 3} \mathrm{Bond}(y_i, y_{i+1}) \wedge$$

$$\mathrm{Bond}(y_4, y_1) \bigwedge_{1\leq i<j\leq 4} \textbf{not } y_i = y_j$$

$$\rightarrow \mathrm{MoleculeWith4MemberedRing}(x)$$





- Does cyclobutane contain a four-membered ring? ✓

UNDECIDABILITY

- Logic programs with function symbols can axiomatise infinitely large structures

UNDECIDABILITY

- Logic programs with function symbols can axiomatise infinitely large structures
- Reasoning with DGLP ontologies is trivially undecidable

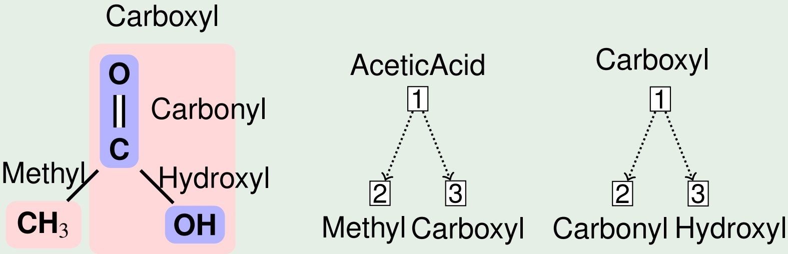
UNDECIDABILITY

- Logic programs with function symbols can axiomatise infinitely large structures
- Reasoning with DGLP ontologies is trivially undecidable
- We are only interested in finite structures

UNDECIDABILITY

- Logic programs with function symbols can axiomatise infinitely large structures
- Reasoning with DGLP ontologies is trivially undecidable
- We are only interested in finite structures

EXAMPLE



Syntactic Acyclicity Conditions

- Chase [Maier et al., 1979] termination is undecidable

Syntactic Acyclicity Conditions

- Chase [Maier et al., 1979] termination is undecidable
- Problem extensively studied in theory of databases

Syntactic Acyclicity Conditions

- Chase [Maier et al., 1979] termination is undecidable
- Problem extensively studied in theory of databases
- Various syntax-based acyclicity conditions

SYNTACTIC ACYCLICITY CONDITIONS

- Chase [Maier et al., 1979] termination is undecidable
- Problem extensively studied in theory of databases
- Various syntax-based acyclicity conditions
  - weak acyclicity [Fagin et al., ICDT, 2002]
  - super-weak acyclicity [Marnette, PODS, 2009]
  - joint acyclicity [Krötzsch and Rudolph, IJCAI, 2011]

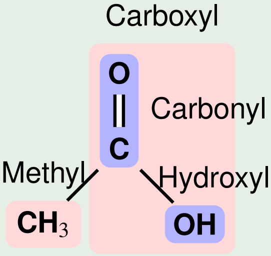
Syntactic Acyclicity Conditions

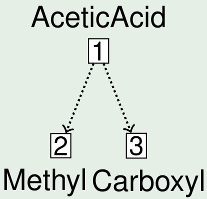
- Chase [Maier et al., 1979] termination is undecidable
- Problem extensively studied in theory of databases
- Various syntax-based acyclicity conditions
  - weak acyclicity [Fagin et al., ICDT, 2002]
  - super-weak acyclicity [Marnette, PODS, 2009]
  - joint acyclicity [Krötzsch and Rudolph, IJCAI, 2011]
  - rule out naturally-arising nested structures

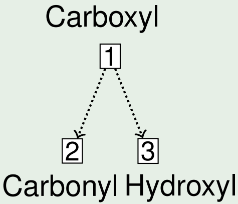
Syntactic Acyclicity Conditions

- Chase [Maier et al., 1979] termination is undecidable
- Problem extensively studied in theory of databases
- Various syntax-based acyclicity conditions
  - weak acyclicity [Fagin et al., ICDT, 2002]
  - super-weak acyclicity [Marnette, PODS, 2009]
  - joint acyclicity [Krötzsch and Rudolph, IJCAI, 2011]
  - rule out naturally-arising nested structures

Example







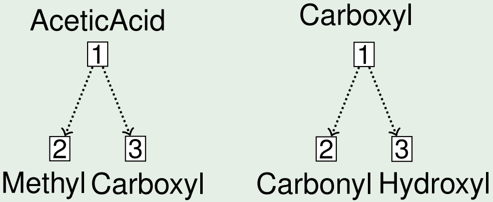
Semantic acyclicity

1. Transitive and irreflexive graph ordering which specifies which graph instances may imply the existence of other graph instances

SEMANTIC ACYCLICITY

1. Transitive and irreflexive graph ordering which specifies which graph instances may imply the existence of other graph instances

EXAMPLE



AceticAcid $\prec$ Carboxyl

SEMANTIC ACYCLICITY

1. Transitive and irreflexive graph ordering which specifies which graph instances may imply the existence of other graph instances
2. Extend the logic program with rules that detect violation of the graph ordering

SEMANTIC ACYCLICITY

1. Transitive and irreflexive graph ordering which specifies which graph instances may imply the existence of other graph instances
2. Extend the logic program with rules that detect violation of the graph ordering
3. Repetitive construction of graph instances during reasoning triggers derivation of Cycle

Semantic acyclicity

1. Transitive and irreflexive graph ordering which specifies which graph instances may imply the existence of other graph instances
2. Extend the logic program with rules that detect violation of the graph ordering
3. Repetitive construction of graph instances during reasoning triggers derivation of Cycle

- A DGLP ontology $\mathcal{O}$ is semantically acyclic if $\mathcal{O} \not\models$ Cycle

SEMANTIC ACYCLICITY

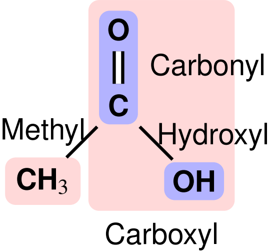
1. Transitive and irreflexive graph ordering which specifies which graph instances may imply the existence of other graph instances
2. Extend the logic program with rules that detect violation of the graph ordering
3. Repetitive construction of graph instances during reasoning triggers derivation of Cycle

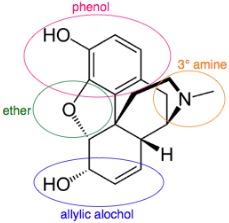
- A DGLP ontology $\mathcal{O}$ is semantically acyclic if $\mathcal{O} \not\models \text{Cycle}$
- DGLP ontology with acetic acid is semantically acyclic ✓

Semantic acyclicity

1. Transitive and irreflexive graph ordering which specifies which graph instances may imply the existence of other graph instances
2. Extend the logic program with rules that detect violation of the graph ordering
3. Repetitive construction of graph instances during reasoning triggers derivation of Cycle

- A DGLP ontology $\mathcal{O}$ is semantically acyclic if $\mathcal{O} \not\models$ Cycle
- DGLP ontology with acetic acid is semantically acyclic ✓





Technical results

1. Termination guarantee for semantically acyclic ontologies

TECHNICAL RESULTS

1. Termination guarantee for semantically acyclic ontologies
2. Decidability of semantic acyclicity for negation-free DGLP ontologies

Technical results

1. Termination guarantee for semantically acyclic ontologies
2. Decidability of semantic acyclicity for negation-free DGLP ontologies
3. Decidability of semantic acyclicity for DGLP ontologies with stratified negation

TECHNICAL RESULTS

1. Termination guarantee for semantically acyclic ontologies
2. Decidability of semantic acyclicity for negation-free DGLP ontologies
3. Decidability of semantic acyclicity for DGLP ontologies with stratified negation

- Semantically acyclic DGLP ontologies with stratified negation capture a wide range of chemical classes:

TECHNICAL RESULTS

1. Termination guarantee for semantically acyclic ontologies
2. Decidability of semantic acyclicity for negation-free DGLP ontologies
3. Decidability of semantic acyclicity for DGLP ontologies with stratified negation

- Semantically acyclic DGLP ontologies with stratified negation capture a wide range of chemical classes:
    - Is dinitrogen inorganic?
    - Does cyclobutane contain a four-membered ring?
    - Is acetylene a hydrocarbon?
    - Does benzaldehyde contain a benzene ring?

TECHNICAL RESULTS

1. Termination guarantee for semantically acyclic ontologies
2. Decidability of semantic acyclicity for negation-free DGLP ontologies
3. Decidability of semantic acyclicity for DGLP ontologies with stratified negation

- Semantically acyclic DGLP ontologies with stratified negation capture a wide range of chemical classes:
  - Is dinitrogen inorganic? ✓
  - Does cyclobutane contain a four-membered ring? ✓
  - Is acetylene a hydrocarbon? ✓
  - Does benzaldehyde contain a benzene ring? ✓

EMPIRICAL EVALUATION

- Data extracted from ChEBI in Molfile format

EMPIRICAL EVALUATION

- Data extracted from ChEBI in Molfile format
- XSB logic programming engine

Empirical Evaluation

- Data extracted from ChEBI in Molfile format
- XSB logic programming engine
- Chemical classes:

Empirical Evaluation

- Data extracted from ChEBI in Molfile format
- XSB logic programming engine
- Chemical classes:
    - Hydrocarbons
    - Inorganic molecules
    - Molecules with exactly two carbons
    - Molecules with a four-membered ring
    - Molecules with a benzene

Empirical Evaluation

- Data extracted from ChEBI in Molfile format
- XSB logic programming engine
- Chemical classes:
  - Hydrocarbons
  - Inorganic molecules
  - Molecules with exactly two carbons
  - Molecules with a four-membered ring
  - Molecules with a benzene
- Preliminary evaluation ranging from 10 to 70 molecules

Empirical Evaluation

- Data extracted from ChEBI in Molfile format
- XSB logic programming engine
- Chemical classes:
  - Hydrocarbons
  - Inorganic molecules
  - Molecules with exactly two carbons
  - Molecules with a four-membered ring
  - Molecules with a benzene
- Preliminary evaluation ranging from 10 to 70 molecules
- Results:

Empirical Evaluation

- Data extracted from ChEBI in Molfile format
- XSB logic programming engine
- Chemical classes:
  - Hydrocarbons
  - Inorganic molecules
  - Molecules with exactly two carbons
  - Molecules with a four-membered ring
  - Molecules with a benzene
- Preliminary evaluation ranging from 10 to 70 molecules
- Results:
  - All DGLP ontologies were found acyclic

Empirical Evaluation

- Data extracted from ChEBI in Molfile format
- XSB logic programming engine
- Chemical classes:
  - Hydrocarbons
  - Inorganic molecules
  - Molecules with exactly two carbons
  - Molecules with a four-membered ring
  - Molecules with a benzene
- Preliminary evaluation ranging from 10 to 70 molecules
- Results:
  - All DGLP ontologies were found acyclic
  - Molecules classified as expected

EMPIRICAL EVALUATION

- Data extracted from ChEBI in Molfile format
- XSB logic programming engine
- Chemical classes:
  - Hydrocarbons
  - Inorganic molecules
  - Molecules with exactly two carbons
  - Molecules with a four-membered ring
  - Molecules with a benzene
- Preliminary evaluation ranging from 10 to 70 molecules
- Results:
  - All DGLP ontologies were found acyclic
  - Molecules classified as expected
  - Suite of subsumption tests for largest ontology performed in few minutes

Overview and Future Directions

1. Expressive and decidable formalism for representation of structured objects

Overview and Future Directions

1. Expressive and decidable formalism for representation of structured objects
2. Novel acyclicity condition for logic programs with restricted use of function symbols

Overview and Future Directions

1. Expressive and decidable formalism for representation of structured objects
2. Novel acyclicity condition for logic programs with restricted use of function symbols
3. Prototype for the structure-based classification of complex objects

Overview and Future Directions

1. Expressive and decidable formalism for representation of structured objects
2. Novel acyclicity condition for logic programs with restricted use of function symbols
3. Prototype for the structure-based classification of complex objects

- Future directions:
  - Generalise acyclicity condition for datalog rules with existentials in the head

Overview and Future Directions

1. Expressive and decidable formalism for representation of structured objects
2. Novel acyclicity condition for logic programs with restricted use of function symbols
3. Prototype for the structure-based classification of complex objects

- Future directions:
  - Generalise acyclicity condition for datalog rules with existentials in the head
  - Relax stratifiability criteria for negation

OVERVIEW AND FUTURE DIRECTIONS

1. Expressive and decidable formalism for representation of structured objects
2. Novel acyclicity condition for logic programs with restricted use of function symbols
3. Prototype for the structure-based classification of complex objects

- Future directions:
  - Generalise acyclicity condition for datalog rules with existentials in the head
  - Relax stratifiability criteria for negation
  - User-friendly surface syntax

Overview and Future Directions

1. Expressive and decidable formalism for representation of structured objects
2. Novel acyclicity condition for logic programs with restricted use of function symbols
3. Prototype for the structure-based classification of complex objects

- Future directions:
  - Generalise acyclicity condition for datalog rules with existentials in the head
  - Relax stratifiability criteria for negation
  - User-friendly surface syntax
  - Fully-fledged classification system for graph-shaped objects

Overview and Future Directions

1. Expressive and decidable formalism for representation of structured objects
2. Novel acyclicity condition for logic programs with restricted use of function symbols
3. Prototype for the structure-based classification of complex objects

- Future directions:
  - Generalise acyclicity condition for datalog rules with existentials in the head
  - Relax stratifiability criteria for negation
  - User-friendly surface syntax
  - Fully-fledged classification system for graph-shaped objects
- Thank you for listening. Questions?