Large-scale Optimization Methods for Logical Reasoning: A Novel Perspective

Maryam Daryalal

MIP Workshop
June 2024
Outline

- Introduction
- Problem Description
- Solution Methodology
- Results & Conclusion
Introduction
Introduction

Problem Description

Solution Methodology

Results & Conclusion

Description Logic

- Description Logics ($\mathcal{DL}$) are a family of formal knowledge representation languages.
- Used to represent the knowledge of an application domain in a structured and formal way.
- Provides a mechanism for encoding semantics of a domain and reasoning about it.
Description Logic

- Description Logics (\(\mathcal{DL}\)) are a family of formal knowledge representation languages.
- Used to represent the knowledge of an application domain in a structured and formal way.
- Provides a mechanism for encoding semantics of a domain and reasoning about it.

Checking satisfiability; classification; answering queries
Description Logic

- Description Logics (DL) are a family of formal knowledge representation languages.
- Used to represent the knowledge of an application domain in a structured and formal way.
- Provides a mechanism for encoding semantics of a domain and reasoning about it.
  - Checking satisfiability; classification; answering queries
- It serves as a foundation for implementing ontologies and semantic web technologies
Description Logic

- Description Logics ($\mathcal{DL}$) are a family of formal knowledge representation languages.
- Used to represent the knowledge of an application domain in a structured and formal way.
- Provides a mechanism for encoding semantics of a domain and reasoning about it.
  - Checking satisfiability; classification; answering queries
- It serves as a foundation for implementing ontologies and semantic web technologies
  - Used to define a formal representation of a domain using concepts and relationships.
Description Logic

- Description Logics ($\mathcal{DL}$) are a family of formal knowledge representation languages.
- Used to represent the knowledge of an application domain in a structured and formal way.
- Provides a mechanism for encoding semantics of a domain and reasoning about it.
  - Checking satisfiability; classification; answering queries
- It serves as a foundation for implementing ontologies and semantic web technologies
  - Used to define a formal representation of a domain using concepts and relationships.
  - An extension of the traditional Web, enables computers to understand web data, using ontologies.
The Semantic Web & Description Logic

- OWL (Web Ontology Language): A language for defining and instantiating Web ontologies.
  - OWL uses $\mathcal{DL}$ to provide semantics for complex ontologies.
  - OWL $\mathcal{DL}$ is compatible with existing Web standards, e.g., HTTP, XML, RDF, RDFS
The Semantic Web & Description Logic

» OWL (Web Ontology Language): A language for defining and instantiating Web ontologies.
  
  - OWL uses DL to provide semantics for complex ontologies.
  - OWL DL is compatible with existing Web standards, e.g., HTTP, XML, RDF, RDFS
  
  → DL provides a theoretical basis for semantic reasoning on the Web.
The Semantic Web & Description Logic

- OWL (Web Ontology Language): A language for defining and instantiating Web ontologies.
  - OWL uses $\mathcal{DL}$ to provide semantics for complex ontologies.
  - OWL $\mathcal{DL}$ is compatible with existing Web standards, e.g., HTTP, XML, RDF, RDFS

→ $\mathcal{DL}$ provides a theoretical basis for semantic reasoning on the Web.

Semantic Web: Data is not just structured but also meaningful and machine-understandable.
The Semantic Web & Description Logic

OWL (Web Ontology Language): A language for defining and instantiating Web ontologies.

- OWL uses $\mathcal{DL}$ to provide semantics for complex ontologies.
- OWL $\mathcal{DL}$ is compatible with existing Web standards, e.g., HTTP, XML, RDF, RDFS

$\rightarrow$ $\mathcal{DL}$ provides a theoretical basis for semantic reasoning on the Web.

Semantic Web: Data is not just structured but also meaningful and machine-understandable.

Why not the existing Web data models? XML? RDF?

- XML: syntax ✓, semantics ✗
- RDF: syntax ✓, (basic) semantics ✓, reasoning ✗
- OWL $\mathcal{DL}$: syntax ✓, (rich) semantics ✓, reasoning ✓
A Simple Ontology: Description

Modeling a University domain including entities like Professors, Students, and Courses.

- **Concepts:** Professor, Student, Full-time Student, Part-time Student, Course
- **Roles:** teaches(Professor, Course), enrolled(Student, Course)
- **Axioms:**
  - Every Full-time Student is a Student. Every Part-time Student is a Student.
  - A Student is either Full-time Student, or Part-time Student. They cannot be both.
  - Every Full-time Student is enrolled in at least 3 Courses.
  - Every Part-time Student is enrolled in at most 2 Courses.
  - For every Course there exists some Professor teaching it.
A Simple Ontology: \( \mathcal{DL} \) Syntax

Modeling a University domain including entities like Professors, Students, and Courses.

- **Concepts**: Professor, Student, Full-time Student, Part-time Student, Course
- **Roles**: teaches(Professor, Course), enrolled(Student, Course)
- **Axioms**:
  - Full-time Student ⊆ Student, Part-time Student ⊆ Student
  - Student ⊆ Full-time Student ⊖ Part-time Student, Full-time Student ⊖ Part-time Student ⊆ ⊥
  - Full-time Student ⊇≥ 3 enrolled.Course
  - Part-time Student ⊇≤ 2 enrolled.Course
  - Course ⊆ ∃inv(teaches).Professor
A Simple Ontology: Knowledge Graph
A Simple Ontology: Knowledge Graph

Course ⊑ ∃ inv(teaches) Professor

Full-time Student ⊓ Part-time Student ⊑ ⊥

Full-time Student ⊔ Part-time Student ⊑ ⊥
A Simple Ontology: Reasoning

OWL $\mathcal{DL}$ can also infer new knowledge $\rightarrow$ reasoning
A Simple Ontology: Reasoning

- OWL $\mathcal{DL}$ can also infer new knowledge $\rightarrow$ reasoning
- Let’s add two new concepts to our ontology:
  - PhD-Student $\sqsubseteq$ Student
  - Seminar $\sqsubseteq \exists$inv(enrolled).PhD-Student
A Simple Ontology: Reasoning

- OWL $\mathcal{DL}$ can also infer new knowledge $\rightarrow$ reasoning

- Let’s add two new concepts to our ontology:
  - PhD-Student $\sqsubseteq$ Student
  - Seminar $\sqsubseteq \exists \text{inv(}\text{enrolled})$.PhD-Student

- We haven’t explicitly told the reasoner that a Seminar is a Course. It will infer this.
  - RDF cannot represent the semantics of our ontology. It lacks the vocabulary for disjointedness, cardinality, etc.
  - RDF cannot infer new knowledge.
A Simple Ontology: Reasoning

- Added concepts and axiom

Diagram of the class hierarchy and properties:

- owl:Thing
  - Course
  - Professor
  - Seminar
  - Student
    - PhD Student
    - Full-time Student
    - Part-time Student

Annotations for Seminar:

- Annotation
- Usage

Description:

- Equivalent To
- SubClass Of
  - inverse (enrolled) some PhD Student

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Annotation:

- Seminar — http://www.semanticweb.org/marymdaryalal/ontologies/2024/4/University#Seminar
A Simple Ontology: Reasoning

▶ Inferences made by the reasoner

Class hierarchy: Seminar

Annotation properties | Datatypes | Individuals
--- | --- | ---
Classes | Object properties | Data properties

Annotations: Seminar

Description: Seminar

Equivalent To

SubClass Of

Inverse (enrolled) some PhD_Student

Course

General class axioms

SubClass Of (Anonymous Ancestor)

Inverse (teaches) some Professor
A Simple Ontology: Reasoning

▶ Explanations provided by the reasoner
Takeaway: Key Features of $DL$

- **Expressiveness**: It balances expressivity with computational tractability.
- **Decidability**: The reasoning problems can be solved algorithmically.
- **Conciseness**: Provides compact and human-readable form.
- **Formal Semantics**: Uses formal logic-based semantics to avoid ambiguity.
Takeaway: Key Features of $\mathcal{DL}$

- **Expressiveness**
  It balances expressivity with computational tractability.

- **Decidability**

- **Conciseness**

- **Formal Semantics**
  - Provides compact and human-readable form.
  - Uses formal logic-based semantics to avoid ambiguity.
  - The reasoning problems can be solved algorithmically.
Takeaway: Key Features of $\mathcal{DL}$

- **Expressiveness**: It balances expressivity with computational tractability.
- **Decidability**: The reasoning problems can be solved algorithmically.
- **Conciseness**: Provides compact and human-readable form.
- **Formal Semantics**: Uses formal logic-based semantics to avoid ambiguity.

Description Logic

M. Daryalal
Large-scale Optimization for Logical Reasoning
Takeaway: Key Features of $DL$

- **Expressiveness**
  It balances expressivity with computational tractability

- **Decidability**
  The reasoning problems can be solved algorithmically.

- **Conciseness**
  Provides compact and human-readable form.

- **Formal Semantics**
  Uses formal logic-based semantics to avoid ambiguity.

- **Description Logic**
Takeaway: Key Features of DL

- **Expressiveness**: It balances expressivity with computational tractability.
- **Decidability**: The reasoning problems can be solved algorithmically.
- **Conciseness**: Provides compact and human-readable form.
- **Formal Semantics**: Uses formal logic-based semantics to avoid ambiguity.

Description Logic
Constructors & Axioms in $\mathcal{DL}$ ALCQ (I)

An interpretation: $\mathcal{I} = (\Delta^\mathcal{I}, .^\mathcal{I})$, with $\Delta^\mathcal{I}$ a non-empty domain set and $.^\mathcal{I}$ a mapping.

- **Thing**  
  $\top \equiv \top^\mathcal{I} = \Delta^\mathcal{I}$

- **Nothing**  
  $\bot \equiv \bot^\mathcal{I} = \emptyset$

- **Concept (class)**  
  $A \equiv A^\mathcal{I} \subseteq \Delta^\mathcal{I}$

- **Concept assertion**  
  $a : C \equiv a^\mathcal{I} \in C^\mathcal{I}$

- **Negation**  
  $\neg C \equiv \Delta^\mathcal{I} \setminus C^\mathcal{I}$

- **Conjunction**  
  $C \sqcap D \equiv C^\mathcal{I} \cap D^\mathcal{I}$

- **Disjunction**  
  $C \sqcup D \equiv C^\mathcal{I} \cup D^\mathcal{I}$

- **Subsumption**  
  $C \sqsubseteq D \equiv C^\mathcal{I} \subseteq D^\mathcal{I}$
Constructors & Axioms in $\mathcal{DL}$ ALCQ (II)

- An interpretation: $\mathcal{I} = (\Delta^\mathcal{I}, .^\mathcal{I})$, with $\Delta^\mathcal{I}$ a non-empty domain set and $.^\mathcal{I}$ a mapping.

Role (relationship)  
\[ R \equiv R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I} \]

Role assertion  
\[ (a, b) : R \equiv (a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I} \]

Universal restriction  
\[ \forall R.C \equiv \{ x|\forall y : (x, y) \in R^\mathcal{I} \Rightarrow y \in C^\mathcal{I} \} \]

At-least qualified cardinality restriction  
\[ \geq nR.C \equiv \{ x|\#R^\mathcal{I}(x, C) \geq n \} \]

At-most qualified cardinality restriction  
\[ \leq mR.C \equiv \{ x|\#R^\mathcal{I}(x, C) \leq m \} \]

Role hierarchy  
\[ R \subseteq S \equiv R^\mathcal{I} \subseteq S^\mathcal{I} \]

Transitive role  
\[ R \in \mathcal{N}_{RT} \equiv R^\mathcal{I} = (R^\mathcal{I})^+ \]
Problem Description
The Satisfiability Problem in $\mathcal{DL}$ Ontologies

The SAT Problem

Given an ontology $\mathcal{O}$ written in a Description Logic $\mathcal{L}$, and a concept $C$, is there a model $\mathcal{I}$ of $\mathcal{O}$ where $C^\mathcal{I} \neq \emptyset$?

- Does there exist an interpretation that satisfies all axioms in $\mathcal{O}$ and where $C$ is non-empty?
- Ontology axioms constrain possible $\mathcal{I}$s, potentially making a concept unsatisfiable.
The Satisfiability Problem in $\mathcal{DL}$ Ontologies

The SAT Problem

Given an ontology $\mathcal{O}$ written in a Description Logic $\mathcal{L}$, and a concept $C$, is there a model $\mathcal{I}$ of $\mathcal{O}$ where $C^\mathcal{I} \neq \emptyset$?

- Does there exist an interpretation that satisfies all axioms in $\mathcal{O}$ and where $C$ is non-empty?
- Ontology axioms constrain possible $\mathcal{I}$s, potentially making a concept unsatisfiable.

- **Example:**
  - Full-time Student $\sqsubseteq$ Student
  - Professor $\sqsubseteq \neg$ Student (Professors are not Students)
  - Is Professor $\sqcap$ Full-time Student satisfiable? No $\rightarrow$ Professor $\sqcap$ Full-time Student $\sqsubseteq \bot$
The Challenge of Qualified Cardinality Restrictions (QCRs)

- QCRs are expressive, but computationally challenging for reasoning algorithms.
- Reasoning with QCRs:
  - Tableaux algorithms: Introduce or merge individuals to satisfy cardinality constraints.
    - Example: For Person ⊑≥ 2hasChild.Student:
      - Start with individual ‘a: Person’, ‘a: (≥ 2hasChild.Student)’
      - Introduce ‘b1, b2: Student’ such that hasChild(a, b1), hasChild(a, b2)
    - Challenges: Non-determinism, exponential complexity.
The Challenge of Qualified Cardinality Restrictions (QCRs)

- QCRs are expressive, but computationally challenging for reasoning algorithms.

- Reasoning with QCRs:
  - Tableaux algorithms: Introduce or merge individuals to satisfy cardinality constraints.
    - Example: For $\text{Person} ⊑\geq 2\text{hasChild.Student}$:
      - Start with individual ‘$a: \text{Person}$’, ‘$a: (\geq 2\text{hasChild.Student})$’
      - Introduce ‘$b1, b2 : \text{Student}$’ such that $\text{hasChild}(a, b1), \text{hasChild}(a, b2)$
    - Challenges: Non-determinism, exponential complexity.
  - Reduction to other problems → Feasibility problem: Given a set of constraints $\mathcal{T}$, does there exist a solution $x$ that satisfies all constraints in $\mathcal{T}$?
The Challenge of Qualified Cardinality Restrictions (QCRs)

- QCRs are expressive, but computationally challenging for reasoning algorithms.

- Reasoning with QCRs:
  - Tableaux algorithms: Introduce or merge individuals to satisfy cardinality constraints.
    - Example: For Person $\sqsupseteq 2\text{hasChild}.$Student:
      - Start with individual ‘a: Person’, ‘a: ($\geq 2\text{hasChild}.$Student)’
      - Introduce ‘b1, b2 : Student’ such that hasChild(a, b1), hasChild(a, b2)
    - Challenges: Non-determinism, exponential complexity.

- Reduction to other problems $\rightarrow$ Feasibility problem: Given a set of constraints $\mathcal{T}$, does there exist a solution $x$ that satisfies all constraints in $\mathcal{T}$?
QCRs as Linear Inequalities: The Idea

▶ Example: $S = \{ \geq 3R, \leq 2T, \geq 1S, \leq 1S \}$

▶ Atomic Decomposition of $S$:

$$p_1 = \{R\}, \quad p_2 = \{T\}, \quad p_4 = \{S\},$$

$$p_3 = \{R, T\}, \quad p_5 = \{R, S\}, \quad p_6 = \{S, T\},$$

$$p_7 = \{R, S, T\}$$
QCRs as Linear Inequalities: The Idea

- Example: \( S = \{ \geq 3R, \leq 2T, \geq 1S, \leq 1S \} \)

- Atomic Decomposition of \( S \):

1. Define int variable \( v_{iS}, v_{iT}, v_{iR} \) for each partition:

\[
p_1 \rightarrow v_{001}, \ p_2 \rightarrow v_{010}, \ p_3 \rightarrow v_{011}, \ p_4 \rightarrow v_{100},
\]
\[
p_5 \rightarrow v_{101}, \ p_6 \rightarrow v_{110}, \ p_7 \rightarrow v_{111}
\]

\[
p_1 = \{ R \}, \ p_2 = \{ T \}, \ p_4 = \{ S \},
\]
\[
p_3 = \{ R, T \}, \ p_5 = \{ R, S \}, \ p_6 = \{ S, T \},
\]
\[
p_7 = \{ R, S, T \}
\]
QCRs as Linear Inequalities: The Idea

➤ Example: $S = \{ \geq 3R, \leq 2T, \geq 1S, \leq 1S \}$

➤ Atomic Decomposition of $S$:

1. Define int variable $v_{i_S}, v_{i_T}, v_{i_R}$ for each partition:

   $p_1 \rightarrow v_{001}$, $p_2 \rightarrow v_{010}$, $p_3 \rightarrow v_{011}$, $p_4 \rightarrow v_{100}$,
   $p_5 \rightarrow v_{101}$, $p_6 \rightarrow v_{110}$, $p_7 \rightarrow v_{111}$

2. Write $S$ as:

   $v_{001} + v_{011} + v_{101} + v_{111} \geq 3$
   $v_{010} + v_{011} + v_{110} + v_{111} \leq 2$
   $v_{100} + v_{101} + v_{110} + v_{111} \leq 1$
   $v_{100} + v_{101} + v_{110} + v_{111} \geq 1$
QCRs as Linear Inequalities: Compact Model

- Back to our own world!

- Let $\mathcal{R}$ be the set of all roles, and $\bar{\delta}_{R}$ and $\delta_{R}$ be the right-hand side of at-most and at-least restrictions on a role $R$.

$$
\begin{align*}
\min & \quad \sum_{R \in \mathcal{R}} \sum_{i_{R} \in \{0,1\}} v_{i_{1},\ldots,i_{|\mathcal{R}|}} \\
\text{s.t.} & \quad \delta_{R} \leq \sum_{j \in \mathcal{R}, j \neq R} \sum_{i_{j} \in \{0,1\}} v_{i_{1},\ldots,i_{|\mathcal{R}|}} \leq \bar{\delta}_{R}, \quad R \in \mathcal{R} \\
& \quad v_{i_{1},\ldots,i_{|\mathcal{R}|}} \in \mathbb{Z}^{+}, \quad R \in \mathcal{R}, i_{R} \in \{0,1\}
\end{align*}
$$

How many partitions do we have?

There’s also other axioms and concepts we haven’t considered yet...
QCRs as Linear Inequalities: Compact Model

- Back to our own world!

- Let $\mathcal{R}$ be the set of all roles, and $\overline{\delta}_R$ and $\delta_R$ be the right-hand side of at-most and at-least restrictions on a role $R$.

$$\min \sum_{R \in \mathcal{R}} \sum_{i_R \in \{0,1\}} v_{i_1, \ldots, i_{|\mathcal{R}|}}$$

s.t.  
$$\delta_R \leq \sum_{j \in \mathcal{R}, j \neq R} \sum_{i_j \in \{0,1\}} v_{i_1, \ldots, i_{|\mathcal{R}|}} \leq \overline{\delta}_R \quad R \in \mathcal{R}$$

$$v_{i_1, \ldots, i_{|\mathcal{R}|}} \in \mathbb{Z}^+ \quad R \in \mathcal{R}, i_R \in \{0,1\}$$

- How many partitions do we have?
QCRs as Linear Inequalities: Compact Model

- Back to our own world!
- Let $\mathcal{R}$ be the set of all roles, and $\bar{\delta}_R$ and $\delta_R$ be the right-hand side of at-most and at-least restrictions on a role $R$.

\[
\begin{align*}
\min & \quad \sum_{R \in \mathcal{R}} \sum_{i_R \in \{0, 1\}} v_{i_1, \ldots, i_{|\mathcal{R}|}} \\
\text{s.t.} & \quad \delta_R \leq \sum_{j \in \mathcal{R}, j \neq R} \sum_{i_j \in \{0, 1\}} v_{i_1, \ldots, i_{|\mathcal{R}|}} \leq \bar{\delta}_R \quad R \in \mathcal{R} \\
& \quad v_{i_1, \ldots, i_{|\mathcal{R}|}} \in \mathbb{Z}^+ \\
& \quad R \in \mathcal{R}, i_R \in \{0, 1\}
\end{align*}
\]

- How many partitions do we have?
- There’s also other axioms and concepts we haven’t considered yet...
Solution Methodology
Extended Formulation for QCRs (I)

- Define a mapping $\alpha(.)$ that assigns a newly defined sub-role $R' \subseteq R$ to each QCR:

$$\alpha(\bowtie nR.C) = R'.$$
Extended Formulation for QCRs (I)

- Define a mapping $\alpha(.)$ that assigns a newly defined sub-role $R' \subseteq R$ to each QCR:
  \[
  \alpha(\bowtie nR.C) = R'.
  \]

- Define $S_Q = \{\alpha(\bowtie nR.C) | \bowtie nR.C \in S\} \cup \{C | \bowtie nR.C \in S\} \cup \{\top, \bot\}$. 

Extended Formulation for QCRs (I)

- Define a mapping $\alpha(.)$ that assigns a newly defined sub-role $R' \sqsubseteq R$ to each QCR:
  \[ \alpha(\bowtie nR.C) = R'. \]

- Define $S_Q = \{ \alpha(\bowtie nR.C) | \bowtie nR.C \in S \} \cup \{ C | \bowtie nR.C \in S \} \cup \{ \top, \bot \}.$

- Define $\mathcal{P}_{S_Q}$ as the power set of $S_Q$ excluding the empty set, and any subset without a role.
Extended Formulation for QCRs (I)

- Define a mapping $\alpha(.)$ that assigns a newly defined sub-role $R' \subseteq R$ to each QCR:

$$\alpha(\nabla nR.C) = R'.$$

- Define $S_Q = \{\alpha(\nabla nR.C) \mid \nabla nR.C \in S\} \cup \{C \mid \nabla nR.C \in S\} \cup \{\top, \bot\}$.

- Define $\mathcal{P}_{S_Q}$ as the power set of $S_Q$ excluding the empty set, and any subset without a role.

- A partition configuration: Represents a partition $p$ in $\mathcal{P}_{S_Q}$. It is a set of binary parameters $a^R_p$, $R' \in S_Q$:

$$a^R_p = \begin{cases} 1 & \text{if role } R' \in p \\ 0 & \text{otherwise.} \end{cases}$$
Extended Formulation for QCRs (I)

- Define a mapping $\alpha(.)$ that assigns a newly defined sub-role $R' \subseteq R$ to each QCR:
  $$\alpha(\bowtie nR.C) = R'.$$

- Define $S_Q = \{\alpha(\bowtie nR.C) \mid \bowtie nR.C \in S\} \cup \{C \mid \bowtie nR.C \in S\} \cup \{\top, \bot\}$.

- Define $P_{S_Q}$ as the power set of $S_Q$ excluding the empty set, and any subset without a role.

- A partition configuration: Represents a partition $p$ in $P_{S_Q}$. It is a set of binary parameters $a_{R'}^p$, $R' \in S_Q$:
  $$a_{R'}^p = \begin{cases} 1 & \text{if role } R' \in p \\ 0 & \text{otherwise.} \end{cases}$$

- $\text{cost}_p$: Cost of partition $p$, defined as the number of concepts in $p$
  - We want only explicitly entailed concepts
Extended Formulation for QCRs (II)

\[ x_p \in \mathbb{Z}^+ : \text{The number of individuals belonging to partition } p \text{ in the optimal solution.} \]

\[ \rightarrow \quad \text{EF}(\mathcal{P}_{SQ}) = \min \sum_{p \in \mathcal{P}_{SQ}} \text{cost}_p x_p \]

\[ \text{s.t.} \quad \sum_{p \in \mathcal{P}_{SQ}} a^R_p x_p \geq \delta_{R'}, \quad R' \in \{\alpha(\geq nR.C) | \geq nR.C \in S\} \]

\[ \sum_{p \in \mathcal{P}_{SQ}} a^R_p x_p \leq \bar{\delta}_{R'}, \quad R' \in \{\alpha(\leq nR.C) | \leq nR.C \in S\} \]

\[ x_p \in \mathbb{Z}^+, \quad p \in \mathcal{P}_{SQ}. \]

- A branch-and-price framework can implicitly enumerate the exponentially many partitions.
- We will take care of all other axioms inside the implicit enumeration.
Branch-and-Price for QCRs

Let $P' \subseteq P_{SQ}$. Then $EF^{LP}(P')$ is the LP relaxation of $EF$ over $P'$.

Partition Generation:
- Let $\pi$ and $\omega$ be the dual vectors associated with $\geq$ and $\leq$ constraints in $EF^{LP}(P')$, respectively.
- Let $a^{R'} \in \{0, 1\}$ be a decision variable equal to 1 iff role $R'$ is in the generated partition.
- Let $b_C \in \{0, 1\}$ be a decision variable equal to 1 iff concept $C$ is in the generated partition.
Branch-and-Price for QCRs

- Let $\mathcal{P}' \subseteq \mathcal{P}_{SQ}$. Then $\text{EF}^{LP}(\mathcal{P}')$ is the LP relaxation of $\text{EF}$ over $\mathcal{P}'$.

- **Partition Generation**:
  - Let $\pi$ and $\omega$ be the dual vectors associated with $\geq$ and $\leq$ constraints in $\text{EF}^{LP}(\mathcal{P}')$, respectively.
  - Let $a^{R'} \in \{0, 1\}$ be a decision variable equal to 1 iff role $R'$ is in the generated partition.
  - Let $b_C \in \{0, 1\}$ be a decision variable equal to 1 iff concept $C$ is in the generated partition.

\[ \rightarrow \text{PP} = \min \text{ Reduced-cost}(\hat{\pi}, \hat{\omega}) \]

\[ \text{s.t. } \; a^{R'} \otimes b_C, \quad R' \in \{\alpha(\bowtie nR.C) \mid \bowtie nR.C \in \mathcal{S}\}, C = R'.\text{Qualifier} \]

\[ b_C \leq \beta, \quad C \in \{C \mid \bowtie nR.C \in \mathcal{S}\} \]

\[ b_\perp = 0 \]

All other axioms

\[ b_C, a^{R'} \in \{0, 1\}, \quad R' \in \{\alpha(\bowtie nR.C) \mid \bowtie nR.C \in \mathcal{S}\}, C \in \{C \mid \bowtie nR.C \in \mathcal{S}\} \]
Mapping of Axioms (I)

- **Basic axioms:**
  - For every subsumption $A \sqsubseteq B$, add the following to PP:
    \[ b_A \leq b_B. \]
  - For every binary subsumption $A \sqcap B \sqsubseteq C$, add the following to PP:
    \[ b_A + b_B - 1 \leq b_C. \]
  - For every disjointness $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq \bot$, $n \geq 2$, add the following to PP:
    \[ \sum_{i=1}^{n} b_{A_i} - n + 1 \leq b_\bot. \]
  - For modelling the negation between $C$ and $\neg C$, add the following to PP:
    \[ b_C + b_{\neg C} = 1. \]
Mapping of Axioms (II)

- We can show that all the other axioms in $\mathcal{DL}$ ALCQ can be converted to basic axioms by introducing new concepts and basic axioms.

Example:

\[
\begin{align*}
- & A \sqsubseteq A_1 \sqcap \cdots \sqcap A_n \equiv A \sqsubseteq A_i, \quad i = 1, \ldots, n \\
- & A \sqsubseteq B \sqcup C \equiv \neg B \sqcap \neg C \sqsubseteq \neg A \\
- & \geq nR \cdot C \sqsubseteq A \equiv \neg A \sqsubseteq (n - 1)R \cdot C \\
- & \leq nR \cdot C \sqsubseteq A \equiv \neg A \sqsupseteq (n + 1)R \cdot C \\
- & A \sqsubseteq \exists R \cdot B \equiv A \sqsupseteq 1R \cdot B \\
- & \geq 1R \cdot B \sqsubseteq A \equiv \neg A \sqsubseteq 0R \cdot B \\
- & \cdots
\end{align*}
\]
Integrality

- Branching rule can be defined on binary variables $a^R'$. 
- However, in all of our instances so far (real and synthetic ontologies), optimal solution returned by the column generation method have been integral!

Conjecture

The polyhedron of $\text{EF}^{\text{LP}}(\mathcal{P}')$ is integral.

- We haven’t been able to prove this using the usual sufficient conditions. So, to be continued...
Results & Conclusion
Preliminary Experiments

- We compared our ILP-based reasoner with major OWL reasoners: FaCT++, HermiT, Konclude, and Racer.

- Benchmark ontologies:

<table>
<thead>
<tr>
<th>Ontology Name</th>
<th>#Axioms</th>
<th>#Concepts</th>
<th>#Roles</th>
<th>#QCRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>canadian-parliament-factions-1</td>
<td>48</td>
<td>21</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>canadian-parliament-factions-2</td>
<td>56</td>
<td>24</td>
<td>7</td>
<td>25</td>
</tr>
<tr>
<td>canadian-parliament-factions-3</td>
<td>64</td>
<td>27</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>canadian-parliament-factions-4</td>
<td>72</td>
<td>30</td>
<td>9</td>
<td>35</td>
</tr>
<tr>
<td>canadian-parliament-factions-5</td>
<td>81</td>
<td>34</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>C-SAT-exp-ELQ</td>
<td>26</td>
<td>10</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>C-UnSAT-exp-ELQ</td>
<td>26</td>
<td>10</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>genomic-cds rules-ELQ-fragment-1</td>
<td>716</td>
<td>358</td>
<td>1</td>
<td>357</td>
</tr>
<tr>
<td>genomic-cds rules-ELQ-fragment-2</td>
<td>718</td>
<td>359</td>
<td>1</td>
<td>357</td>
</tr>
</tbody>
</table>
Preliminary Observations

- The only reasoners that can classify all variants of the simplest of the first benchmark ontology within the given time limit of 1000s are our ILP-based reasoner and Racer.

- Second benchmark:

```
<table>
<thead>
<tr>
<th>n</th>
<th>ILP</th>
<th>Fac</th>
<th>Her</th>
<th>Kon</th>
<th>Rac</th>
<th>ILP</th>
<th>Fac</th>
<th>Her</th>
<th>Kon</th>
<th>Rac</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>0.6</td>
<td>TO</td>
<td>TO</td>
<td>TO</td>
<td>0.01</td>
<td>0.63</td>
<td>TO</td>
<td>TO</td>
<td>TO</td>
<td>0.01</td>
</tr>
<tr>
<td>20</td>
<td>0.62</td>
<td>TO</td>
<td>TO</td>
<td>TO</td>
<td>0.01</td>
<td>0.80</td>
<td>TO</td>
<td>TO</td>
<td>TO</td>
<td>0.01</td>
</tr>
<tr>
<td>10</td>
<td>0.63</td>
<td>TO</td>
<td>TO</td>
<td>TO</td>
<td>0.01</td>
<td>0.99</td>
<td>TO</td>
<td>TO</td>
<td>TO</td>
<td>0.01</td>
</tr>
<tr>
<td>5</td>
<td>0.72</td>
<td>6.3</td>
<td>4.4</td>
<td>0.91</td>
<td>0.01</td>
<td>0.74</td>
<td>TO</td>
<td>TO</td>
<td>784</td>
<td>0.01</td>
</tr>
<tr>
<td>3</td>
<td>0.62</td>
<td>0.17</td>
<td>0.18</td>
<td>0.33</td>
<td>0.01</td>
<td>0.75</td>
<td>0.25</td>
<td>1.15</td>
<td>1.18</td>
<td>0.01</td>
</tr>
</tbody>
</table>
```

- Ontologies genomic-cds rules contain many concepts using QCRs of the form $a = 2$ has $A_i$, with no interaction between $(A_i)$: all reasoners except Racer performed well.
Next...

- This is an ongoing work! So far, we’ve only focused on mappings and proof of concept.
- The normalization of non-basic axioms is taking longer than expected, however according to the $\mathcal{DL}$ literature should be possible in polynomial time. To be investigated.
- Conjecture, if proved, can simplify the implementation and presentation to non-OR communities.
Large-scale optimization methods for logical reasoning: A novel perspective

Maryam Daryalal
HEC Montréal, Canada

61 views 3 weeks ago Séminaires/Seminars
Séminaire du GERAD
Large-scale optimization methods for logical reasoning: A novel perspective...more
Thanks for listening!