Approximation for the Semantic Web

The KnowledgeWeb point of view

Holger Wache

Semantic Web Systems in General

Ontology

Input -> DL-Reasoning -> Output
Problems tackled in KWEB

- **Scalability (Performance)**
  - DL-Reasoning in EXPTIME (OWL DL in NEXPTIME)

Input ➔ DL-Reasoning ➔ Output

- **Robustness**
  - For example during query answering
Approximation Approaches

- Language Weakening
- Knowledge Compilation
- Approximate Deduction

Input → Ontology → Output

Approximate Deduction through Simplification

- Simplify query
- Simple query ⇒ fast query answering
- Simple query ⇒ approximated answers
- Continuously complete query
- Anytime behavior
How to simplify?

First Idea:
Omit some parts (e.g. $\Phi$, $\Psi$)

Query = ... $\times \Psi$ $\times$ ... $\times$ ( ...$\times$ $\Psi$ $\times$ ...)

How to simplify? (II)

Second Idea:
Rewrite some parts (e.g. $\Phi$, $\Psi$)

Query = ... $\times \Phi$ $\times$ ... $\times$ ( ...$\times$ $\Phi$ $\times$ ...)

$\phi \leftrightarrow \psi$
Cadoli-Schaerf-Approximation for DLs

\[ C^{\top}_{i} : \exists R.C \leftrightarrow \top \]
\[ C^{\bot}_{i} : \exists R.C \leftrightarrow \bot \]

- Replacing some sub terms in concept expressions
- Well-founded theory with (theoretically) nice results

Cadoli-Schaerf-Approximation: Example

Depth of subconcept \( D \):
number of universal quantifiers that have \( D \) in its scope.

\[
(\exists \text{friend.tall}) \sqcap \forall \text{friend}.((\forall \text{friend.doctor}) \sqcap \exists \text{friend}.\neg \text{doctor})
\]

Depth: 0

Depth: 2

Depth: 1

\[ s^{\top}_{0} \quad \top \sqcap \forall \text{friend}.((\forall \text{friend.doctor}) \sqcap \top) \]

\[ s^{\top}_{1} \quad (\exists \text{friend.tall}) \sqcap \forall \text{friend}.((\forall \text{friend.doctor}) \sqcap \top) \]

\[ s^{\top}_{2} \quad (\exists \text{friend.tall}) \sqcap \forall \text{friend}.((\forall \text{friend.doctor}) \sqcap \exists \text{friend}.\neg \text{doctor}). \]
Application: Classification

- Central process
  Classify Term Q
- Contained in
  - Generating the subsumption hierarchy
  - Instance Retrieval

Mixed Results: Classifying in TAMBIS

- Application: Classification of Concepts
  ⇒ sequence of subsumption test: \( C \sqsubseteq D \)

<table>
<thead>
<tr>
<th></th>
<th>normal</th>
<th>( C_l )</th>
<th>( C_l^t )</th>
<th>( C_l^t &amp; C_l^f )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>Tambis (16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C_1</td>
<td>157</td>
<td>8</td>
<td>157</td>
</tr>
<tr>
<td></td>
<td>C_2</td>
<td>32</td>
<td>161</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>C_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C_4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>24</td>
<td>279</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>X</td>
<td>Y</td>
<td>X</td>
</tr>
</tbody>
</table>

\( (C \sqsupseteq D)_{i}^T \Rightarrow C \sqsupseteq D \)  
\( (C \sqsubseteq D)_{i}^T \Rightarrow C \sqsubseteq D \)

\( (C \sqcap \neg D)_{i}^T \) is satisfiable \( (C \sqcap \neg D)_{i}^T \) is unsatisfiable
\( (C \sqcap \neg D) \) is satisfiable \( (C \sqcap \neg D) \) is unsatisfiable
Further Results

<table>
<thead>
<tr>
<th></th>
<th>normal</th>
<th>( c_1^f )</th>
<th>( c_2^f )</th>
<th>( c_1^e &amp; c_2^f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dolce (10)</td>
<td>true</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>normal</td>
<td></td>
<td>10</td>
<td>113</td>
<td>10</td>
</tr>
<tr>
<td>Galen (10)</td>
<td>true</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>normal</td>
<td></td>
<td>10</td>
<td>12190</td>
<td>10</td>
</tr>
<tr>
<td>Monet (10)</td>
<td>true</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>normal</td>
<td></td>
<td>20</td>
<td>656</td>
<td>20</td>
</tr>
<tr>
<td>MadCow (10)</td>
<td>true</td>
<td>145</td>
<td>0</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>normal</td>
<td></td>
<td>60</td>
<td>152</td>
<td>61</td>
</tr>
<tr>
<td>Wine (10)</td>
<td>true</td>
<td>-</td>
<td>228</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>false</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>normal</td>
<td></td>
<td>33</td>
<td>252</td>
<td>27</td>
</tr>
</tbody>
</table>

Problem: Term Collapsing

Query = \( \forall \)

- **Term C**
  - very often conjunction of subterms
  - e.g. conjunctive queries

- **Term D**
  - Very often also conjunction of subterms

Subsumption Queries have this structure very often
Classifying in TAMBIS (IV)

<table>
<thead>
<tr>
<th>normal</th>
<th>$C^I$</th>
<th>$C^U$</th>
<th>$C^I &amp; C^U$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>Tambis (16)</td>
<td>$C^I$</td>
<td>157</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>$C^I$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>24</td>
<td>219</td>
<td>$\psi$</td>
</tr>
</tbody>
</table>

Term Collapsing: 157 = 100%  65 = 35,9%  190 = 62,1%

Lessons learned

$\phi \mapsto \psi$

- Avoid Term Collapsing
  - Replace $\psi$ with something else than $A$ or $B$

- Find better places to rewrite
  - Ontology-adapted $\phi$?
Focused Case: Instance Retrieval

- Find all instances \( a \) which belongs to a query \( Q \):
  \[ a : Q \]
- Tool \textit{InstanceStore}:
  - Try to replace DL reasoning as much as possible with (scalable) DB retrieval
  - Only applicable to role-free A-Boxes
    \[ a : Q \leftrightarrow l_a \land Q; l_a \text{ concept description of instance } a \]
- Boolean Conjunctive Queries
  - \( q_1 \land \cdots \land q_n \), where \( q_1, \ldots, q_n \) are of the form \( x : C \) or \( \langle x, y \rangle : R \)
  - Restrict to those which can be converted to a concept expression \( C \)

New Approximation Method: Heuristic Ordering of Conjuncts

- Compute a ranking value for each conjunct
  \[ \Phi(q_i) : C \rightarrow I \mathbb{R} \]
- Order the conjuncts \( q_n, \ldots, q_1 \) according to their value
- Complete approximated query with more and more expensive conjuncts
How to order conjuncts?

- According to the needed computation time for each conjunction
  - Estimate the computation time a priori
- According to the possible search space reduction
  - Prefer conjuncts which eliminate a lot of instances

How to estimate the computation costs?

\[
\begin{align*}
\Phi(A) &= 1 \\
\Phi(\neg A) &= 0 \\
\Phi(C \cap D) &= 2 + \Phi(C') + \Phi(D) \\
\Phi(C \cup D) &= \phi + 2 + \Phi(C') + \Phi(D) \\
\Phi(\exists R.C') &= 2 + \Phi(C') \\
\Phi(\forall R.C') &= n + n \cdot \Phi(C')
\end{align*}
\]

where \( \phi \) is the current value of \( \Phi(E) \)
where \( n \) is the number of existential quantifiers in \( E \)
Effects of Cadoli-Schaerf for Subsumption

\[ C \subseteq D \]

Semantics

\((C \subseteq D)^B \equiv a \perp \]

\[ C \subseteq D \iff \models (C \lor \neg D) \perp \]

21

22
Effects of CS for Subsumption: Term Collapsing

$C \sqsubseteq D$

Term collapsing

$\Delta (C \sqsubseteq D)^B$

Effects of new Approximation

$\Delta (C_a \sqsubseteq Q)$

only $Q$ changed

$\Delta (C_a \sqsubseteq Q)^\Delta$

not changed; Term collapsing avoided
### Results: Subsumption tests

<table>
<thead>
<tr>
<th>Results</th>
<th>$C^T$</th>
<th>$C^\perp$</th>
<th>$C^\Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q17</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Results: Time

<table>
<thead>
<tr>
<th>Results</th>
<th>$C^T$</th>
<th>$C^\perp$</th>
<th>$C^\Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q17</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Approximation Approaches

- Language Weakening
- Knowledge Compilation
- Approximate Deduction

Input \rightarrow Output

Approximation through Language Weakening

T-Box
- OWL-FULL
- OWL-DLP
- DLP-Reasoning

A-Box
- Role-free
- DataBase Queries

Input \rightarrow DL-Reasoning \rightarrow Output
Approximation Approaches

- Language Weakening
- Knowledge Compilation
- Approximate Deduction

Input $\rightarrow$ Output

Approximation through Knowledge Compilation

Ontology (optimized) $\rightarrow$ Reasoning (optimized) $\rightarrow$ Output

Input $\rightarrow$ Output
Standard: KAON2

Diagram:

- Query
- OWL DL TBox (no nominals)
- SWRL Rules (only DL-safe)
- OWL DL ABox

Translation to Disjunctive Datalog [ExpTime]

Disjunctive Datalog Reasoning Engine [coNP]

Answer

(Applied: KAON2) = Screech

Diagram:

- Query
- OWL DL TBox (no nominals)
- SWRL Rules (only DL-safe)
- OWL DL ABox

Translation to Disjunctive Datalog [ExpTime]

Disjunctive Datalog Reasoning Engine [coNP] [P]

Answer

Language weakening

Can be performed offline.
Screech simple example

serbian \& croatian \& European
eucitizen \& European
german \& french \& beneluxian \& eucitizen

**beneluxian** = **luxembourgien** \& **dutch** \& **belgian**

french(julien). croatian(boris). german(markus).
**belgian(saartje)**. german(rudi). german(york).

---

Screech simple example

**beneluxian** = **luxembourgien** \& **dutch** \& **belgian**

**KAON2** translates into the following four clauses:

- luxembourgien(x) \& dutch(x) \& belgian(x) \& beneluxian(x)
- beneluxian(x) \& luxembourgien(x)
- beneluxian(x) \& dutch(x)
- beneluxian(x) \& belgian(x)

**Screech split first clause:**

- luxembourgien(x) \& beneluxian(x)
dutch(x) \& beneluxian(x)
belgian(x) \& beneluxian(x)
Screech reasoning

- data complexity is $P$
- complete
- but unsound
- inference can be described in terms of standard notions from non-monotonic reasoning

Screech Performance (not optimized yet)

- Galen ontology
  - 673 axioms, 175 classes
  - randomly populated with 500 individuals

- After KAON2: 267 disjunctions in 133 rules eliminated

- Complete run:
  - queried for the extensions of all 175 Galen classes
  - resulting in 5809 classifications (Screech)
    - 5353 (i.e. 92.2%) correct
  - For 138 out of 175 classes: computed extension correct
  - Average time saved: 39.0%
Summary

- Approximation approaches start to improve performance
  - Cadoli-Schaerf Approximation seems to not to work in practical settings
  - Heuristic approximation but performance improvements (only) in restricted cases?!
  - Screech 40% speed-up with only 8% wrong answers but only in one use-case

- Open questions:
  - Try to understand (theoretically) why they work
  - Benchmarking (more use-cases)
  - What about Robustness?

Thank you for your attention!