Approximate Reasoning for the Semantic Web
Part V
Approximate Resolution for OWL

Frank van Harmelen
Pascal Hitzler
Holger Wache

ESSLLI 2006 Summer School
Malaga, Spain, August 2006

Contents Part V

• KAON2 – resolution-based reasoning with OWL

• Approximate reasoning with Screech
The KAON2 OWL Reasoner

- Completely new deduction algorithms.
- Not tableaux-based.
- Reasoning via reduction of OWL DL to (positive) disjunctive datalog.
- Goal: Efficient ABox reasoning.
- Current performance similar to state-of-the-art tableaux reasoners. Better for some tasks.
- Binaries available from http://kaon2.semanticweb.org
  - Implementation by Boris Motik.
- Treats all of OWL DL except nominals.

KAON2: Basic Ideas

- ABox reasoning (instance retrieval) is more important for practice than TBox reasoning.
- Resolution ideal for ABox reasoning (Prolog).
- Similarly good: Deductive Database techniques

  - Resolution proofs for OWL DL?
    - Naive approach does not always terminate.
    - Reason: Transformation to FOL yields existential quantifiers which are Skolemised to function symbols.
    - Termination of algorithms not guaranteed in presence of function symbols.
KAON2: How to deal with termination issue

- Finitely many usages of existential quantifiers suffice for sound and complete reasoning.
  - How many and which ones?
- First process TBox: Derive (all necessary) logical consequences using ordered resolution.
  - Finite Set!
  - Then generation of further individuals via function symbols no longer necessary!
- Existential quantifiers can then be removed!

KAON2: Inference mechanism

1. Translate TBox to function free clauses. (Exptime!)
2. Add ABox.
3. Employ standard reasoning methods for function-free clauses, e.g. magic sets. (NP-complete!)

- TBox needs to be processed only once!
- Algorithm is worst-case optimal!
- Data complexity is NP!
KAON2 Reasoner core architecture

Query \rightarrow \text{Transformation to Disjunctive Datalog [ExpTime]} \rightarrow \text{Answer}

\text{Query} \rightarrow \text{SHIQ(D) TBox (no nominals)} \rightarrow \text{SWRL Rules (only DL-safe)} \rightarrow \text{SHIQ(D) ABox}

Problem: Skolemization introduces function symbols, which cause standard Datalog algorithms to loop.

Uses standard techniques like magic sets.

Disjunctive Datalog Reasoning Engine [coNP]

KAON2 transformation algorithm

\text{Trans}(R) \Rightarrow \forall R.C \subseteq \forall R.(\forall R.C)

\text{Via FOL. Skolemization produces function symbols!}

\text{Add some inferenced clauses (basic superposition/ordered resolution). Expptime!}

Replace f(x) by new individual fx. (Finite) graph of f becomes new role.

\text{For some tasks, it suffices to deal with the TBox here!}
Simple example for transformation (ALC only)

structural transformation & clausification

<table>
<thead>
<tr>
<th>KB</th>
<th>FOL KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person ⊑ ∃ parent.Person</td>
<td>¬Person(x) ∨ parent(x,f(x))</td>
</tr>
<tr>
<td>∃ parent.(∃ parent.Person) ⊑ Grandchild</td>
<td>¬Person(x) ∨ Person(f(x))</td>
</tr>
<tr>
<td>Person(a)</td>
<td>Grandchild(x) ∨ ¬parent(x,y) ∨ Q₁(y)</td>
</tr>
<tr>
<td></td>
<td>¬Q₁(x) ∨ ¬parent(x,y) ∨ ¬Person(y)</td>
</tr>
<tr>
<td></td>
<td>Person(a)</td>
</tr>
</tbody>
</table>

Saturation

Knowledge Base Saturated!

Phase 1: Saturating TBox and RBox

Phase 2: Remove Irrelevant Rules

Translate to Datalog

Translate to Datalog
Result: Disjunctive datalog program

\[
\begin{array}{|c|}
\hline
\text{KB} \\
\text{Person} \sqsubseteq \exists \text{parent.Person} \\
\exists \text{parent.(}\exists \text{parent.Person}) \sqsubseteq \text{Grandchild} \\
\text{Person(a)} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|}
\hline
\text{DD(KB)} \\
Q_1(x), \text{Person(y)} \leftarrow \text{parent}(x,y) \\
\leftarrow \text{parent}(x,y), Q_1(y), \text{Grandchild}(x) \\
\leftarrow Q_1(x), \text{Person}(x) \\
\text{Grandchild}(x) \leftarrow \text{Person}(x) \\
\text{Person(a)} \\
\hline
\end{array}
\]

A Theorem (Hustadt, Motik, Sattler 2004)

Let KB be an $\mathcal{ALCHIQ}(D)$ knowledge base, defined over a concrete domain $D$, such that satisfiability of finite conjunctions over $D$ can be decided in deterministic exponential time. Then, the following claims hold:

1. KB is unsatisfiable if and only if DD(KB) is unsatisfiable.
2. KB $\models \alpha$ if and only if DD(KB) $\models \alpha$, where $\alpha$ is of the form $A(a)$ or $R(a, b)$, and $A$ is an atomic concept.
3. KB $\models C(a)$ for a nonatomic concept $C$ if and only if, for $Q$ a new atomic concept, DD(KB $\cup \{C \sqsubseteq Q\})$ $\models Q(a)$. 
Performance evaluation

• Different architectures difficult to compare.
  – caching mechanisms
  – preprocessing steps
  – etc.
• Generally, KAON2 seems to do better on ABox reasoning tasks, in particular if ABox is large and TBox is of medium size.
• Generally, KAON2 appears to be inferior on TBox reasoning tasks.

KAON2 additional features

• Additional features
  – an API for programmatic management of OWL-DL and SWRL and F-Logic ontologies,
  – a stand-alone server providing access to ontologies in a distributed manner,
  – an inference engine for answering queries (including support for SPARQL),
  – a DIG interface, allowing access from tools such as Protégé
  – efficient access to instances via relational databases

• Download: http://kaon2.semanticweb.org/
Part V Contents

- KAON2 – resolution-based reasoning with OWL
- Approximate reasoning with Screech

Problem Description

- Reasoning with OWL DL is hard. (Expressivity vs. scalability)
- For certain Semantic Web applications quick responses are more important than absolute accuracy of answering. e.g. scenario.
  - Answering of human queries in an open domain.
- We trade soundness for time, using approximate reasoning.
Approximate Reasoning

- do not confuse with *fuzzy* or *probabilistic* reasoning!

- speed up obtained by
  - modifying the underlying inference relation
  - in a semantically controlled and well-understood way.

- e.g. by decreasing the complexity class of a reasoning task
- e.g. by utilizing intimate knowledge of the bottlenecks in a reasoning algorithm.

Approximate reasoning with Screech for large ABoxes

1. **Query**
   - OWL DL TBox (no nominals)

2. **Translation to Disjunctive Datalog**
   - [ExpTime]

3. **Split program**

4. **Disjunctive Datalog Reasoning Engine**
   - [coNP] [P]

5. **Answer**

- C = \{a, b\}
- \{a, b\} \subseteq C

- Can be performed offline.

- Suffices for some queries e.g. instance retrieval for named classes

**OWL DL TBox**

**OWL DL ABox**

**Translation to Disjunctive Datalog**

**Query**

**Split program**

**Disjunctive Datalog Reasoning Engine**

**Answer**
Screech simple example

serbian △ croatian △ european
eucitizen △ european
german △ french △ beneluxian △ eucitizen

**beneluxian** ≡ **luxembourgian △ dutch △ belgian**

serbian(ljiljana). serbian(nenad). german(pascal).
franc(julien). croatian(boris). german(markus).
**belgian(saartje).** german(rudi). german(york).

---

Screech simple example

beneluxian ≡ Luxembourgian △ dutch △ belgian

translates into the following four clauses:

- Luxembourgian(x) △ Dutch(x) △ Belgian(x) ← Beneluxian(x)
- Beneluxian(x) ← Luxembourgian(x)
- Beneluxian(x) ← Dutch(x)
- Beneluxian(x) ← Belgian(x)

**split of first clause:**

- Luxembourgian(x) ← Beneluxian(x)
- Dutch(x) ← Beneluxian(x)
- Belgian(x) ← Beneluxian(x)

\[ \vdash Luxembourgian(saartje) \]
\[ \vdash Dutch(saartje) \]
\[ \vdash Belgian(saartje) \]
Screech reasoning

- data complexity is \( P \)
- complete
- but unsound

Screech Performance (not optimized yet)

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

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%
Screech Performance example run

<table>
<thead>
<tr>
<th>Time (DD)</th>
<th>Time (SPLIT)</th>
<th>Instances</th>
<th>Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>11036 ms</td>
<td>6489 ms</td>
<td>154/154</td>
<td>BiologicalObject</td>
</tr>
<tr>
<td>11026 ms</td>
<td>5950 ms</td>
<td>9/9</td>
<td>Specified_set</td>
</tr>
<tr>
<td>11006 ms</td>
<td>6219 ms</td>
<td>9/13</td>
<td>Multiple</td>
</tr>
<tr>
<td>11015 ms</td>
<td>5898 ms</td>
<td>16/16</td>
<td>Probe_structural_part_of_heart</td>
</tr>
<tr>
<td>11036 ms</td>
<td>7711 ms</td>
<td>4/4</td>
<td>Human_red_blood_cell_mature</td>
</tr>
<tr>
<td>11055 ms</td>
<td>5949 ms</td>
<td>24/58</td>
<td>BiologicalObject_that...</td>
</tr>
</tbody>
</table>

Table 2. Performance comparison for instance retrieval using disjunctive datalog (DD) vs. the corresponding split program (SPLIT), on the KAON2 datalog engine. *Instances* indicates the number of instances retrieved using DD versus SPLIT, e.g. class *Multiple* contained 9 individuals, while the split program allowed to retrieve 13 (i.e. the 9 correct individuals plus 4 incorrect ones). The full name of the class in the last row is *BiologicalObject_that_has_left_right_symmetry*.

Screech conclusions

- Screech reasoner based on KAON2
- complete but unsound
- 40% speed-up with only 8% wrong answers

- future work
  - optimise further
  - tackle other parts of the KAON2 algorithms
  - combine with other approximate reasoning and optimization techniques
One idea for improving Screech

- Use statistical knowledge about distribution of individuals among classes.
- E.g. there are few luxemburgians compared to dutch and belgians.

\[
\text{luxembourgian}(x) \lor \text{dutch}(x) \lor \text{belgian}(x) \leftarrow \text{beneluxian}(x)
\]

\text{split :}
\[
\begin{align*}
\text{dutch}(x) & \leftarrow \text{beneluxian}(x) \\
\text{belgian}(x) & \leftarrow \text{beneluxian}(x)
\end{align*}
\]

Wrap-up: what we did
Wrap-up: What we did

• Introduction Semantic Web
  • OWL/Description Logics
    – Semantics
    – Tableau reasoning
    – Resolution-based reasoning
  • General approximate reasoning
    – Cadoli-Schaerf
    – BCP, abstraction, knowledge compilation, Top-k,…
  • Tableaux-based approximate reasoning for OWL
    – Cadoli-Schaerf
    – Approximate query answering
  • Resolution-based approximate reasoning
    – Screech

Wrap-up: Why you should do approximate reasoning

• We need expressive reasoning for the semantic web.
• Sound and complete algorithms are too slow.

• Hence: We need approximate reasoning!

• Lots of literature on approximate reasoning exists and has not yet been applied to the semantic web!

⇒ Go do it!
Discussion

• We think that approximate reasoning will help to make semantic web a reality.

• Objections?
• Questions?
• Remarks?

End of course

• Thanks for your interest.
• Feedback is most welcome!
Additional semantic web applications

- **SmartWeb**
  See [http://www.smartweb-project.org](http://www.smartweb-project.org)
- Semantic Mediawiki

---

**SmartWeb**

Additional semantic web applications

- SmartWeb
- **Semantic Mediawiki**
  See http://wiki.ontoworld.org/

Semantic Mediawiki

- Enhancement of Mediawiki (used in Wikipedia)
- Simple knowledge representation techniques
- Added value for user

  - In particular:
    - enhancement of querying
    - better data reuse
We go to the article on the RuleML2006 conference ...


RuleML2006 (non semantic version)

RuleML2006 is the Second International Conference on Rules and Rule Markup Languages for the Semantic Web. It is held from November 9 2006 to November 10 2006 in Athens, Georgia, USA. For more information, see http://2006.ruleml.org/.

Call for Papers

Semantic Web technologies have matured to the point where they are being adopted by many organizations for applications as diverse as data integration, optimized search, and decision support. The increasing use of the technology has resulted from mainstream commercial software vendors providing solutions that support Semantic Web technologies, and W3C making RDF and OWL standard recommendations.

... and edit it


Editing RuleML2006 (non semantic version)

RuleML2006 is the Second International Conference on Rules and Rule Markup Languages for the Semantic Web. It is held from November 9 2006 to November 10 2006 in Athens, Georgia, USA. For more information, see http://2006.ruleml.org/.

== Call for Papers ==

Semantic Web technologies have matured to the point where they are being adopted by many organizations for applications as diverse as data integration, optimized search, and decision support. The increasing use of the technology has resulted from...
RuleML2006 is the Second International Conference on Rules and Rule Markup Languages for the Semantic Web. It is held from November 9 2006 to November 10 2006 in [[Athens, Georgia]], [[USA]]. For more information, see http://2006.ruleml.org/.

There is already an ordinarily link to the article of „Athens, Georgia“.

Just say what the relation between this page (RuleML2006) and „Athens, Georgia“ is.
From links...

... in [[Athens, Georgia]], [[USA]]. ...

... to typed links

... in [[located in::Athens, Georgia]], [[USA]]. ...

From values...

... It is held from November 9 2006 to November 10 2006 in...

... to attributes

... It is held from [[start date:=November 9 2006]] to [[end date:=November 10 2006]] in...
RuleML2006 is the Second International Conference on Rules and Rule Markup Languages for the Semantic Web. It is held from November 9, 2006 to November 10, 2006 in Athens, Georgia, USA. For more information, see http://2006.ruleml.org.
What the humans see, when they scroll down

RuleML2006 is the Second International Conference on Rules and Rule Markup Languages for the Semantic Web. It is held from November 9 2006 to November 10 2006 in Athens, Georgia, USA. For more information, see http://2006.ruleml.org for demonstration purposes, we have

- RuleML2006 (non semantic version)
- RuleML2006 (simple semantic version)

**Relations to other articles** — Click + to find similar articles.

RuleML2006 (simple semantic version) Has location Athens, Georgia

**Attributes of RuleML2006 (simple semantic version)** — Click + to find similar articles.

- **Start date:** 2006-11-09
- **End date:** 2006-11-10

**Editing help on relations and attributes** View as RDF
Benefits for Wikipedians: `<ask>` for your data

• Inline queries allow for questions like …
  - …movies from the 70s starring Sean Connery
  - …list of events (all conferences and workshops)

```markdown
<ask format="ul" link="all">
  [[Category:Event]]
</ask>
```

Benefits for Wikipedians: `<ask>` for your data

• Inline queries allow for questions like …
  - …movies from the 70s starring Sean Connery
  - …list of events with their deadline

```markdown
<ask format="ul" link="all">
  [[Category:Event]]
  [[paper deadline:=*]]
</ask>
```
Benefits for Wikipedians: <ask> for your data

<ask format="ul" link="all">
  [[Category:Event]]
  [[paper deadline:=>June 1 2006]]
  [[paper deadline:=<December 31 2006]]
  [[title:=*]]
  [[paper deadline:=*]]
  [[Category:Topic Semantic Web query languages]]
</ask>

<table>
<thead>
<tr>
<th>title</th>
<th>paper deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleML2006</td>
<td>Second International Conference on Rules and Rule Markup Languages for the Semantic Web</td>
</tr>
</tbody>
</table>

Applications

• Automatic tables and lists
  – E.g. Countries sorted by area, population, alphabet, …

• **Maintenance** with hand crafted checks
  – Does every country have one capital?

• Integration in applications
  – `latte = wikipedia.get(“Latte Macchiato”);`
  `print latte[“contains”]`

• Visualization and browsing

• …And many unexpected ones
Who is using Semantic MediaWiki?

- Wikicompny ~ 9,000 pages, 227 user accounts
- The Bible Wiki ~ 40,000 pages, 156 user accounts
- WWW2006 Social Wiki ~ 40 pages, 34 user accounts
- Semantic eGovernment Community Portal, ~ 30 pages
- ScubaWiki, ~ 170 pages, 33 user accounts
- NeOn Wiki ~10 pages
- this wiki = Wiki.Ontoworld.org ~400 pages, 158 user accounts
- SourceryForge ~1864 pages, 272 user accounts

Demo and more information at http://wiki.ontoworld.org/

Semantic Mediawiki

- Lightweight use of metadata (no semantics in the stronger sense)
- Already added value for the user!

- Simple: Introduce background knowledge by means of ontologies.
  - Beware of scalability problems!
  - ... and others ...