Island Reasoning for ALCHI Ontologies

Ralf Möller, Sebastian Wandelt

Hamburg University of Technology, Germany

November 2, 2008
What is the Problem?

- Real world ontologies will have a growing amount of assertional information
  - concept assertions
  - role assertions
- Today’s DL reasoner have to load the whole ontology into main memory to solve decision problems.
  - Exception: DLs with limited expressivity, which can be “accessed” through SQL (e.g. DL-Lite)
- Idea for more expressive languages: Identify parts of ontologies, which are relevant for reasoning
We extract a small subset of assertions (called *island*), which is worst-case relevant for checking, whether $\mathcal{O} \models a : C$.
Focus

- We consider \textit{ALCHI}, i.e.
  - Basic \textit{ALC} concept-constructors ($\sqcup$, $\sqcap$, $\neg$, $\forall$, $\exists$)
  - + Role hierarchy
  - + Inverse roles

- Decision problems we take into account are
  - Instance Checking: $\mathcal{O} \vDash ?a : C$
  - Role Checking: $\mathcal{O} \vDash ?R(a, b)$

- Furthermore: only atomic ABoxes
Related Work

- **Summarization approaches**
  - *Fokue06*: Merge ABox individuals (merging preserves completeness w.r.t. instance checking $\Rightarrow$ preselection, UNA?)

- **Partitioning approaches**
  - *Heflin07*: Precompute all possible dependencies among ABox individuals (even for very simple ontologies the partitions become almost equivalent to the source ABox)

$\Rightarrow$ Both approaches

- have a time-consuming precomputation, not suitable for updates, streams
- Both approaches do not consider locality
Guiding Example - TBox and RBox

- Snapshot taken from Lehigh University Benchmark (LUBM)

\[
\mathcal{T}_{EX} = \left\{ \right. \\
\text{Person} \sqcap \exists \text{headOf} . \text{Department} \sqsubseteq \text{Chair}, \text{Professor} \sqsubseteq \text{Faculty}, \\
\text{Book} \sqsubseteq \text{Publication}, \text{GraduateStudent} \sqsubseteq \text{Student}, \\
\text{Student} \equiv \text{Person} \sqcap \exists \text{takesCourse} . \text{Course}, \\
\top \sqsubseteq \forall \text{teacherOf} . \text{Course}, \exists \text{teacherOf} . \top \sqsubseteq \text{Faculty}, \text{Faculty} \sqsubseteq \text{Person}, \\
\top \sqsubseteq \forall \text{publicationAuthor}^- . (\text{Book} \sqcup \text{ConferencePaper}) \\
\left. \right\}
\]

\[
\mathcal{R}_{EX} = \{ \text{headOf} \sqsubseteq \text{worksFor}, \text{worksFor} \sqsubseteq \text{memberOf}, \text{memberOf} \equiv \text{member}^- \}
\]
Guiding Example - ABox
Steps for Island Computation

- Informally: For $\text{ALCHI}$, only $\forall$-constraints make “use” of role assertions and related individuals
- Solution:
  1. Transform $\mathcal{T}$ into some kind of normal form, to read off all possible $\forall$-constraints ($\text{Shallow Normal Form}$)
  2. Extract $\forall$-constraints and take into account the role hierarchy ($\forall$-info structure)
  3. Identify role assertions, which can be used to propagate new information ($O$-separability)
  4. Island extraction for a given individual
Shallow Normal Form

- Convert concept inclusions from the TBox into disjunctions
- Apply transformation rules (e.g. De Morgan, NNF, $\forall R.C \sqcap D \equiv \forall R.C \sqcap \forall R.D$, ...) to propagate $\sqcap$ to the outer parts
- Break up conjunctions
- Example:
  - Input:
    $$\mathcal{I} = \{ Person \sqcap \exists headOf . Department \sqsubseteq Chair, \neg Faculty \sqsubseteq \neg Professor \}$$
  - Output:
    $$\text{Shallow}(\mathcal{I}) = \{ \neg Person \sqcup \forall headOf . \neg Department \sqcup Chair, Faculty \sqcup \neg Professor \}$$
∀-info structure

A **∀-info structure** for TBox $\mathcal{T}$ is a function $f^\forall_\mathcal{T} : N_R \rightarrow \mathcal{P}(N_C \cup \{-A|A \in N_C\} \cup \{\bot\}) \cup \{\ast\}$, s.t. $N_C$ ($N_R$) is a set of atomic concepts (roles) used in $\mathcal{T}$. The value of $f^\forall_\mathcal{T}(R)$ is:

- $\emptyset$, if we know that there is no ∀-constraint for $R$ in $\mathcal{T}$
- a subset $S$ of $N_C \cup \{-A|A \in N_C\} \cup \{\bot\}$, s.t. there is no other concept but those in $S$, which occur ∀-bound (i.e. they are a subconcept of a ∀-constraint) on $R$ in $\mathcal{T}$
- $\ast$, if there are arbitrary complex ∀-constraints on role $R$ in $\mathcal{T}$, but we don’t give additional information on the structure of these constraints.
∀-info Structure - Example

\[
Shallow(T) = \{\neg Person \sqcup \forall headOf. \neg Department \sqcup Chair, \neg Student \sqcup \exists takesCourse. Course, \bot \sqcup \\
\forall publicationAuthor^\neg. (Book \sqcup ConferencePaper)\}
\]

\[
f_\forall^O(R) = \begin{cases} 
\{\neg Department\} & \text{if } R = headOf \\
* & \text{if } R = publicationAuthor^\neg \\
\emptyset & \text{else}
\end{cases}
\]
Given an ontology $\mathcal{O} = \langle T, R, A \rangle$, a role assertion $R(a, b)$ is called $\mathcal{O}$-separable, if, informally speaking, semantics are preserved when splitting up $R(a, b)$. $R(a, b)$ is called $\mathcal{O}$-inseparable otherwise.
\( \mathcal{O} \)-separability continued

- Idea: a role assertion \( R(a, b) \) is \( \mathcal{O} \)-separable, if no or only known/obvious information can be propagated over \( R(a, b) \)

- Example:
  - \( \mathcal{T} = \{ ..., \text{ResearchDepartment} \sqsubseteq \text{Department}, ... \} \)
  - \( \mathcal{A} = \{ ..., \text{headOf}(p2, d1), \text{ResearchDepartment}(d1), ... \} \)
  - \( f_{\mathcal{O}}(\text{headOf}) = \{ \text{Department} \} \)
  - \( \Rightarrow \) The assertion \( \text{headOf}(p2, d1) \) is \( \mathcal{O} \)-separable.
Island Extraction

1. Given a named individual $a$, perform graph search (e.g. BFS) on the ABox, starting from $a$, such that one follows only $\mathcal{O}$-inseparable assertions in the ABox. All visited assertions, denoted $island_{\mathcal{O}}(a)$, are worst-case relevant for reasoning.

2. Example results:
Naive Extension to $SHIQ$

- Transitive roles: Naive approach is to assume that each transitive role (+ their sub-roles) cause $O$-inseparability, i.e. arbitrary complex concepts can be propagated
- Qualified number restrictions: Naive approach is to assume that each involved role (+ their sub-roles) cause $O$-inseparability, i.e. individuals can be worst-case merged
- $\Rightarrow$ These roles can be identified by simple analysis of TBox and RBox
How to improve instance checking?

If we want to check $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle \models a : C$, then

1. Compute $\text{island}_\mathcal{O}(a)$
2. Check whether $\langle \mathcal{T}, \mathcal{R}, \text{island}_\mathcal{O}(a) \rangle \models a : C$
How to improve role checking?

If we want to check $\langle T, R, A \rangle \models R(a, b)$, then

1. Compute $island_{O}(a)$ (or $island_{O}(b)$)
2. Check whether $\langle T, R, island_{O}(a/b) \rangle \models R(a, b)$
How to improve (grounded) instance retrieval?

- Island reasoning can be used for *post-filtering*
- Given a complete, but possibly unsound, reasoning algorithm for instance retrieval ($\vdash$), we can perform optimized instance retrieval for a concept $C$ as follows:
  1. Compute the set of candidates $S = \{ a \mid \mathcal{O} \vdash a : C \}$
  2. Post-filter (remove unsound answers) by island reasoning for each $a \in S$
- There exist several unsound algorithms: e.g. Summarization, OWL-Approximation with Screech
- Easy to lift approach to grounded conjunctive queries
Implementation

- Java
- Implementation is straight-forward, given OWLAPI for ontology loading/analysis
- For DL reasoning we use Racer
- Structure:
Evaluation - Precomputation of the Terminological Part

<table>
<thead>
<tr>
<th>Ontology</th>
<th>—Inclusions—</th>
<th>—Equivalences—</th>
<th>—$N_{RN}$—</th>
<th>Time for analysis (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUBM</td>
<td>75</td>
<td>6</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>Cyc</td>
<td>43541</td>
<td>2</td>
<td>4853</td>
<td>93</td>
</tr>
<tr>
<td>GODaily</td>
<td>28997</td>
<td>0</td>
<td>1</td>
<td>103</td>
</tr>
<tr>
<td>Galen</td>
<td>3388</td>
<td>699</td>
<td>413</td>
<td>55</td>
</tr>
<tr>
<td>Pizza</td>
<td>57</td>
<td>2</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Boemie</td>
<td>10000+</td>
<td>0</td>
<td>50</td>
<td>130</td>
</tr>
</tbody>
</table>
Evaluation - Island Extraction for LUBM

<table>
<thead>
<tr>
<th>Individual</th>
<th>Island size</th>
<th>Time for island computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dep0.Uni0/GraduateStudent128</td>
<td>9</td>
<td>0 ms</td>
</tr>
<tr>
<td>Dep0.Uni0/Publication2</td>
<td>4</td>
<td>0 ms</td>
</tr>
<tr>
<td>Dep0.Uni0/FullProfessor7</td>
<td>93</td>
<td>0 ms</td>
</tr>
<tr>
<td>Dep0.Uni0/Course4</td>
<td>37</td>
<td>0 ms</td>
</tr>
</tbody>
</table>

- The average island size for all individuals in LUBM-X is 29 individuals
- First experiments: compute around 10k islands/s
- Similar results for other ontologies (e.g. BOEMIE multimedia annotation)
Conclusions and Future Work

▶ Conclusions
  ▶ Additional approach to perform more efficient ABox reasoning in the average case
  ▶ For our investigated ontologies the results are encouraging
  ▶ Most useful in combination with other techniques (ideally for post-filtering purposes)

▶ Future Work
  ▶ Non-naive extension for $\mathcal{SHIQ}$
  ▶ What about TBox updates?
  ▶ More benchmarks with “real-world” ontologies