Ontology Debugging
—— Debugging Semantic Defects

Zlatan Dragisic and Qiang Liu
Ontologies and Ontology Engineering, VT 2011
Linköpings universitet, Sweden

Outline
1. Background
   1. Explanations for the Semantic Defects
      1. Unsatisfiable Concepts & Incoherent Ontologies
      1. Inconsistent Ontologies
   1. Find the Explanations for the Unsatisfiable Concepts
      1. Glass-box approach
      1. Black-box approach
   1. Repairing the Unsatisfiable Concepts
      1. Diagnosis
      1. Pinpointing
      1. Hitting sets with axiom ranking
   1. Summary

Ontologies based on DL
1. TBox — terminological axioms
   1. Concepts: C ⊑ D, C ≡ D
   1. Roles: hasMember
1. ABox — assertion component
   1. Individuals: C(a), D(b), R(a, b)
1. Knowledge base: $\mathcal{K} = (T, A)$

OWL DL
1. OWL ontology maps to DL knowledge base $\mathcal{K} = (T, A)$
1. OWL DL — based on SHOIN(D)
   1. $\text{OWL } \text{DL} = \text{based on } \text{SHOIN}(D)$
   1. $\text{ALC} + \text{transitive roles, } \to S$
   1. role hierarchies, $\to H$
   1. nominals, $\to O$
   1. inverse roles, $\to I$
   1. number restrictions, $\to N$
   1. data type properties, data values or data types $\to N(D)$

Ontology defects
Three main categories of defects:
1. Syntactic defects
1. Semantic defects
1. Modeling defects

Semantic defects
1. Contradictions in TBox
   1. Unsatisfiable concepts & incoherent ontologies
1. Contradictions in ABox
   1. Inconsistent ontologies
A concept is unsatisfiable iff $A^I = \emptyset$ for all models $I$ of the ontology.

Unsatisfiable concepts cannot have any individuals.

Ontology is incoherent if has an unsatisfiable concept.

An ontology is inconsistent if it has no models.

Contradiction in the ABox.

Outline

1. Background
2. Explanations for the Semantic Defects
   1. Unsatisfiable Concepts & Incoherent Ontologies
   2. Inconsistent Ontologies
3. Find the Explanations for the Unsatisfiable Concepts
   1. Glass-box approach
   2. Black-box approach
4. Repairing the Unsatisfiable Concepts
   1. Diagnosis
   2. Pinpointing
   3. Hitting sets with axiom ranking
5. Summary

Example: an Incoherent Ontology

Consider the following TBox $\mathcal{T}$, where $A$, $B$, and $C$ are primitive and $A_1, \ldots, A_n$ are defined concept names:

\begin{align*}
\forall x : A_1 \sqcap A_2 \sqcap \neg A_3 \sqsubseteq A_4 \\
\forall x : A_1 \sqcap A_2 \sqcap B \sqsubseteq C \\
\forall x : A_4 \sqcap \neg B \sqsubseteq \neg C \\
\forall x : A_3 \sqsubseteq \neg A_1 \\
\end{align*}

The ontology is incoherent! The set of unsatisfiable concepts are: \{A_1, A_2, A_3, A_4\}.

What are the root causes of these defects?

Explain the Semantic Defects

We need to identify the sets of axioms which are necessary for causing the logic contradictions.

For example, for the unsatisfiable concept "A", there are two (and only two) sets of axioms rendering $A = \bot$:

\begin{align*}
\forall x : A_1 \sqcap A_2 \sqcap \neg A_3 \sqsubseteq A_4 \\
\forall x : A_1 \sqcap A_2 \sqcap B \sqsubseteq C \\
\forall x : A_4 \sqcap \neg B \sqsubseteq \neg C \\
\forall x : A_3 \sqsubseteq \neg A_1 \\
\end{align*}

Minimal Unsatisfiability Preserving Sub-TBoxes (MUPS)

Definition 1: Let $A$ be a concept which is unsatisfiable in a TBox $\mathcal{T}$. A set $S' \subseteq S$ is a minimal unsatisfiability-preserving sub-TBox (MUPS) of $T$ if:

1. $A$ is unsatisfiable in $\mathcal{T}$, and
2. $A$ is satisfiable in every sub-TBox $T' \subseteq T$.

We will abbreviate the set of MUPS of $\mathcal{T}$ and $A$ by $\text{mups}(\mathcal{T}, A)$.

The MUPS of an unsatisfiable concept implies the solutions for repairing.

Remove at least one axiom from each axiom set in the MUPS.
Example

For all the unsatisfiable concepts in the ontology, we have

\[ \text{Example 1.} \]

Possible ways of repairing all the unsatisfiable concepts in the ontology:

\[ \{ a_1, a_2, a_3, \overline{a}_1 \} \]

How to represent all these possibilities?

---

Outline

1. Background
2. Explanations for the Semantic Defects
   - Unsatisfiable Concepts & Incoherent Ontologies
   - Inconsistent Ontologies
3. Find the Explanations for the Unsatisfiable Concepts
   - Glass-box approach
   - Black-box approach
4. Repairing the Unsatisfiable Concepts
   - Diagnosis
   - Pinpointing
   - Hitting sets with axiom ranking
5. Summary

---

Minimal Incoherence Preserving Sub-TBox (MIPS)

Definition 3. Let \( T \) be an inconsistent TBox. A TBox \( T' \subseteq T \) is a minimal incoherence preserving sub-TBox (MIPS) of \( T \) if

- \( T' \) is inconsistent,
- every sub-TBox \( T'' \subseteq T' \) is inconsistent.

We will abbreviate the set of MIPS of \( T \) by \( \text{mips}(T) \). For \( T' \) we get three MIPS:

\[ \text{mips}(T') = \{ \{a_1, a_2\}, \{a_3, a_4, a_5\}, \{a_6, a_7\} \} \]

A possible repairing is \( \{a_1, a_2\} \cup \{a_3, a_4, a_5\} \cup \{a_6, a_7\} \), where

- \( a_1 \in \{a_1, a_2\} \)
- \( a_3 \in \{a_3, a_4, a_5\} \)
- \( a_6 \in \{a_6, a_7\} \)

---

Example: an Inconsistent Ontology

\[ \text{Example 2.} \]

\[ \Gamma = \{ a_1 : A_1 \lor \neg A_2, a_2 : A_3 \lor A_4, a_3 : A_5 \lor A_6, a_4 : A_7 \lor A_8, a_5 : A_9 \lor A_{10}, a_6 : A_{11} \lor A_{12}, a_7 : A_{13} \lor A_{14}, a_8 : A_{15} \lor A_{16} \} \]

\[ \text{The ontology is inconsistent!} \]
Glass-box Approach

1. By modifying the internals of a DL reasoner for extract and reveal the cause for unsatisfiability.

   ![DL Reasoner]

2. It gives explanation to why the error occurs.
   - e.g. find out the axioms in the ontology responsible for the logic contradictions.

Glass-box Approach

- Phase 1: Clash detection procedure
  - Use tableau algorithm by applying transformation rules to individuals in the ontology until no more rules are applicable or an individual has a clash.
  - For example, an individual belongs to a concept and its complement.

- Phase 2: Tableaux trace procedure
  - Trace back to find the source axioms supporting the clash and determine the minimal sets of support (i.e. MUPS).

Glass-box Approach

- Advantage
  - By tightly integrating the debugging with the reasoning procedure, precise results can be obtained.

- Disadvantage
  - The reasoner needs to maintain extra data structures to track the source and its dependencies, which introduces additional memory and computation consumption.

Black-box Approach

- By using DL reasoner as an oracle for a certain set of questions (standard inferences such as satisfiability, subsumption, etc.)

- Give explanation to the dependencies between unsatisfiable concepts.
  - i.e. distinguish root from derived unsatisfiable concepts.

- Dependency detection algorithm
  - For each unsatisfiable concept in the ontology, the algorithm returns all its parent dependency unsatisfiable concepts along with the corresponding axioms that link this concept to the parent.

  e.g. Unsatisfiable concept A is derived if:
  1. \( A = (B_1 \cap C_1 \cap \ldots \cap D_1) \) and one of \( B, C, \ldots, D \) is unsatisfiable
  2. \( A = (B \cup C_1 \cup \ldots \cup D_1) \) and all \( B, C, \ldots, D \) are unsatisfiable
  3. \( \text{domain}(p) = B, \text{and } B \text{ is unsatisfiable} \)
  4. \( A = \exists p, B \) and \( B \) is unsatisfiable

- Advantage
  - Reasoner independence, i.e., you do not need a specialized, explanation generating reasoner.
  - Avoid the performance penalty (memory consumption) of glass box techniques.

- Disadvantage
  - Since it does not follow the standard reasoning procedure, the results are sound but not complete.
Outline
1 Background
1 Explanations for the Semantic Defects
   1 Unsatisfiable Concepts & Incoherent Ontologies
   1 Inconsistent Ontologies
2 Find the Explanations for the Unsatisfiable Concepts
   1 Glass-box approach
   1 Black-box approach
2 Repairing the Unsatisfiable Concepts
   1 Diagnosis
   1 Pinpointing
   1 Hitting sets with axiom ranking
2 Summary

Diagnosis
1 Generic method – applicable to a number of domains
1 System represented as the set (SD, COMP)
  1 SD – system description
  1 COMP – set of system components
1 Example
  1 SD
  1 COMP – ax1, ax2, ax3, ax4, ax5, ax6, ax7

Diagnosis
3 Observations made about the system – set OBS
   1 Example: A1, A3, A6, A7 – are unsatisfiable

Definition 2.4: A diagnosis for (ax, components, obs) is a minimal set atC of components such that:

\[ \forall a \in atC, \neg \neg (ax(a) \lor (\forall b \in atC \backslash \{a\} \neg ax(b))) \]

is consistent.

Diagnoses for our example:

\{ax1, ax4\}, \{ax1, ax5, ax7\}, \{ax2, ax4\},…

Naïve approach:
Generate subsets \( \Delta \) of COMP and check for consistency of:

\[ SD \cup \bigcup_{\Delta \in \text{COMP}} \{\neg \neg ax(r) \mid r \in \text{COMPONENTS} \backslash \Delta\} \]

1 Problem: complexity

Conflict sets and hitting sets

Definition 4.1: A conflict set for (SD, components, obs) is a set \( \{r_1, \ldots, r_c\} \subseteq \text{COMPONENTS} \) such that:

\[ \forall a \in \text{COMPONENTS}, \neg \neg (ax(a) \lor (\forall b \in \{r_1, \ldots, r_c\} \neg ax(b))) \]

Conflict sets for our example:

\{ax1, ax2\}, \{ax3, ax4, ax5\}, \{ax4, ax7\}

Definition 4.3: Suppose \( C \) is a collection of sets. A hitting set for \( C \) is a set \( H \subseteq \bigcup_{S \in C} S \) such that \( H \cap S \neq \emptyset \) for each \( S \in C \). A hitting set for \( C \) is minimal if no proper subset of it is a hitting set for \( C \).

Hitting set example

Conflict sets for our example:

\{ax1, ax2\}, \{ax3, ax4, ax5\}, \{ax4, ax7\}
How to find conflict sets?
Compute all refutations of $\text{core}(\text{ont}) \cup \text{Com}()$ Record the $\text{AB}$ instances entering the refutation
If $\{\text{AB}_1, \ldots, \text{AB}_n\}$ is set of $\text{AB}$ instances used in the refutation then $\{\text{AB}_1, \ldots, \text{AB}_n\}$ is a conflict set.

Pinpointing
1. Find unsatisfiable concepts
2. Search for MUPS for unsatisfiable concepts
3. Calculate MIPS from MUPS
4. Calculate the pinpoint from MIPS
5. Remove axioms in the pinpoint from the ontology

Pinpointing
1. Find unsatisfiable concepts
2. Search for MUPS for unsatisfiable concepts
3. Calculate MIPS from MUPS
4. Calculate the pinpoint from MIPS
5. Remove axioms in the pinpoint from the ontology

Hitting set with axiom ranking
1. Hitting sets generated from MUPS
2. Ranking criteria introduced into the algorithm
3. Optimality based on minimal path rank
Axiom Ranking strategies

Ranking based on:
1. Impact analysis
2. User test cases
3. Provenance information regarding change
4. Syntactic relevance

Further improvements
1. Focus on root unsatisfiable concepts

Hitting set with axiom ranking

\[ C = \{\{2, 3, 7\}, \{1, 6\}, \{4, 5, 7\}, \{1, 2, 3\}\} \]

Outline

1. Background
2. Explanations for the Semantic Defects
   - Unsatisfiable Concepts & Incoherent Ontologies
   - Inconsistent Ontologies
3. Find the Explanations for the Unsatisfiable Concepts
   - Glass-box approach
   - Black-box approach
4. Repairing the Unsatisfiable Concepts
   - Diagnosis
   - Pinpointing
   - Hitting sets with axiom ranking
5. Summary

Summary

1. In this presentation, we introduced the existing research work on ontology debugging, focusing on one kind of ontological defects — semantic defects.
2. We reviewed different kinds of semantic defects and explanations for these semantic defects.
3. We reviewed the ontology debugging techniques for:
   - Finding explanations for the unsatisfiable concepts
   - Repairing the unsatisfiable concepts

References