OWL and Description Logics: Cleaning up

Sebastian Wandelt

Based on slides from Ralf Möller
Roadmap

• Three lecture slots left
  ♦ Today (01/21/2013):
    ▪ Complex reasoning techniques + Upper layers in the SW Layer Cake
  ♦ 01/28/13:
    ▪ Success stories for SW applications and Future Trends (a.k.a. SW in 10 years)
  ♦ 02/04/13:
    ▪ Recap of the lecture
    ▪ focus for preparation of the exam
    ▪ Presentation of Georg/Arne (30 min)
  ♦ 02/11/13: Oral exam
Answering expressive queries

query $Q$ \rightarrow \text{DL-System} \rightarrow \text{answers to $Q$}

\text{TBox} \rightarrow \text{DL-System} \rightarrow \text{ABox}
Query answering for expressive DLs

- Grounded conjunctive queries
  - \texttt{ans(X) :- P_1(X_1), \ldots P_n(X_n)}, variables bound to named objects \texttt{i} (no UNA)
  - RacerPro, Pellet, KAON2

- Extended grounded conjunctive queries
  - \texttt{ans(X) :- Atoms}, vars bound to named objects
  - \texttt{Atoms ::= Atom, \ldots , Atom}
  - \texttt{Atom ::= P(X) | \Pi(Y):Atoms | \setminus Atom}
    - \texttt{ans(X) :- mother(X), \setminus \Pi(X) : has-child(X,Y)}
  - RacerPro (nRQL)
Query answering for expressive DLs

- (Unions of) conjunctive queries
  - \( \text{ans}(X) : P_1(X_1), \ldots P_n(X_n) \lor \ldots \lor \ldots, \) vars bound to any object
  - Variables in \( X_i \) not occurring in \( X \) are existentially quantified
  - QuOnto (for DL-Lite+UNA, see below)
  - Decidability results for SHIQ/SHOQ
  - Discouraging complexity results
  - Decidability w.r.t. SROIQ and transitive (or non-simple) roles in query is an open problem
Quonto: The picture

http://www.dis.uniroma1.it/quonto/

Taken from a presentation by Riccardo Rosati
Example

TBox:
MALE ⊑ PERSON
MALE ⊑ ¬FEMALE
∃hasFather ¬ ⊑ MALE
∃hasMother ¬ ⊑ FEMALE

FEMALE ⊑ PERSON
PERSON ⊑ ∃hasFather
PERSON ⊑ ∃hasMother

input query:
q(x) ← PERSON(x)

rewritten query:
q’(x) ← PERSON(x) ∨ FEMALE(x) ∨ MALE(x) ∨ hasFather(y,x) ∨ hasMother(y,x)

Taken from a presentation by Riccardo Rosati
Example (continued)

rewritten query:
\[ q'(x) \leftarrow \text{PERSON}(x) \lor \text{FEMALE}(x) \lor \text{MALE}(x) \lor \text{hasFather}(y,x) \lor \text{hasMother}(y,x) \]

ABox:
- \text{MALE}(\text{Bob})
- \text{MALE}(\text{Paul})
- \text{FEMALE}(\text{Ann})
- \text{hasFather}(\text{Paul},\text{Ann})
- \text{hasMother}(\text{Mary},\text{Paul})

answers to query:
\{ \text{Bob, Paul, Ann, Mary} \}
Problems

• Relies on UNA
  - Not good in case of Linked Data
• Query rewriting might cause exponential blowup
• Relies on SQL query optimizer
  - Why is that a disadvantage?
Ok…

- … so reducing the expressiveness is the key?
- Probably not in all cases
- Expressive Tboxes required for problem solving
- A lot of shouting emerged around 2005: “Current reasoners are assumed not to scale w.r.t. Abox reasoning…”
Disjunctive Datalog/KAON2

KAON2 answers queries in two processing steps:

- OWL DL ABox
- OWL DL TBox (w/o nominals)
- OWL DL query

Transformation to disjunctive datalog program

Disjunctive datalog engine

Datalog can be cached and reused for various queries

Karlsruhe & Manchester
[Motik and Sattler 06]

Diagram taken from a presentation by Markus Krötzsch
Problems

- Exponential blowups possible in transformation steps (in particular for number restrictions)
- No domains such as strings, reals, etc. supported
New ideas

- Store ABox in database/triple store
- Provide “excerpt” that fits into main memory, and provides an index to the right individuals
- Summary ABox

query \( Q \) (GCQ) → DL-System → answers to \( Q \)

ABox → Summarizer → ABox

TBox
Summary Abox

• Aggregate individuals of the same type as long as Abox remains consistent: yields summary Abox’
• Instance test on summary Abox’ gives non-instances
• Refinement step to make Abox’ more precise using justifications gives an instance test
Example

Legend:
- C – Course
- P – Person
- M – Man
- W – Woman
- H – Hobby

Original ABox

TBox:
- Functional (isTaughtBy)
- Disjoint (Man, Woman)

Summary

Taken from a presentation by A. Fokoue, IBM
Resolving inconsistencies

Legend:
- C – Course
- P - Person
- M - Man
- W – Woman
- H - Hobby

TBox:
- Functional (isTaughtBy)
- Disjoint (Man, Woman)

Original ABox

Summary

Summary is inconsistent

Taken from a presentation by A. Fokoue, IBM
Refinement & Justifications

After 1st Refinement

- $C_x' \rightarrow \{C_1, C_2\}$
- $C_y' \rightarrow \{C_3\}$
- $M'$
- $W'$
- $H'$

Summary still inconsistent!

Sample Q: PeopleWithHobby?

Solns: $P_1$, $P_2$

After 2nd Refinement – Consistent Summary

- $C_x' \rightarrow \{C_1, C_2\}$
- $C_y' \rightarrow \{C_3\}$
- $M'$
- $W'$
- $H'$
- $P_x' \rightarrow \{P_1, P_2\}$
- $P_y' \rightarrow \{P_3\}$

PeopleWithHobby = (some likes Hobby)

Taken from a presentation by A. Fokoue, IBM
SHER

- Based on Abox summaries
- Refinement strategy is based on identifying *justifications* for an inconsistency in the summary

*Taken from a presentation by A. Fokoue, IBM*
Problems with summaries

• UNA?
• Datatypes?
• Updates?
• Approach patented …
  ♦ Maybe one of the major reasons why the Semantic Web still does not scale up until today …
Idea: Try to define Abox modules

• Due to Tbox, individuals in Aboxes
  - are potentially merged
  - are “annotated” with additional concepts
    • due to value restrictions triggered by role assertions
    • due to axioms

• What if we assume the worst case?
• If something that will potentially be propagated to an individual is already known, the role corresponding assertion can be “removed”
Island reasoning

• Extract subset of axioms which are sufficient for reasoning about a given individual in a Abox with respect to a Tbox
Example ontology

Male ⊆ Person
Female ⊆ Person
Person ⊆ Male ∪ Female
Article ⊆ Publication
Prof ⊆ Person
Chair ⊆ Prof ∩ ∃headOf.Department
Prof ∩ ∃headOf.Department ⊆ Chair
⊤ ⊆ ∀hasAuthor¯.Publication
⊤ ⊆ ∀takesCourse.Course

headOf ⊆ worksFor
- Prof(ralf)
- Prof(sibylle)
- headOf(sibylle, sts)
- Department sts
- worksFor(ralf, sts)
- takesCourse(amy, db)
- Course db
- hasFriend(amy, luis)
- hasAuthor(pub1, ralf)
- hasAuthor(pub1, sibylle)
- Publication pub1
- Person(luis)
Instance Checking Optimization

- Assume a query: $\text{KB} \models \text{Chair}(\text{sybille})$
- Looking at our ontology, the important Tbox axiom is

$$\text{Prof} \sqcap \exists \text{headOf}.\text{Department} \sqsubseteq \text{Chair}$$

- which is "equivalent" to a disjunction:

$$\neg \text{Prof} \sqcap \forall \text{headOf}.\neg \text{Department} \sqsubseteq \text{Chair}$$

Informally speaking, the headOf-role can be used to propagate "negative" Department-information to successor-nodes of sybille.

Important to note: all the explicit headOf-successors of sybille are Departments already! Thus we will only propagate "obvious" information for this particular assertion.
Taking all axioms in the TBox into account, we obtain the following subgraph of the ABox relevant for reasoning about sybille:

This subset of our ABox suffices to show KB |= Chair(sybille)
Instance Checking Optimization

- To sum up:
  - Axioms can be analyzed offline
  - Depending on these forall-constraints we can determine on the fly a (usually small) subset of ABox assertions relevant for a given individual
- Partition structure can be incrementally (re)computed
Example: Find all named individuals X, such that KB|= Chair(X)

The naive approach is to perform instance checkings for all named individuals in the ABox

=> We have 7 named individuals, only sybille turns out to be a Chair

The question is:

*Given an ABox instance query, can we apply some preselection techniques to filter obvious solutions and obvious non-solutions to reduce the number of instance checkings?*

*The answer is: sometimes*
Instance Retrieval Optimization

- Try to find an approximation of the ontology in a "weaker" DL-language, which enables unsound + complete + more efficient reasoning
- Approximation has been dealt with before (e.g., [Hitzler et al. 08], [Pan et al. 07/09])
- Here: conversion to DL-Lite
The only critical (impossible to rewrite in an equivalence preserving way) axiom is

\[ \text{Prof} \cap \exists \text{headOf.Department} \subseteq \text{Chair} \]

We could replace it by:

\[ \text{Prof} \subseteq \text{Chair} \]

What happens?
The resulting DL-Lite ontology is still consistent (which is important for reasoning)

Now all Professors become instances of Chair

An instance retrieval query on the DL-Lite ontology yields the candidate individuals ralf and sybille

Only two (more costly) instance checking tests left to remove unsound answers (here: ralf)
The complete picture…

query Q (UCQ) → Query expander → TBox' → Approximation

query Q' (SQL) → DBMS → ABox → Partitioning

answers to Q' (candidates) → Compressor → ABox

Candidate Eliminator

fewer candidates → DL-System

Partitions

answers to Q