
Integrating OWL-DL with Rule-based Reasoners

Boris Motik

Rudi Studer

FZI (<http://www.fzi.de/>)

AIFB (<http://www.aifb.uni-karlsruhe.de/>)

University of Karlsruhe, Germany



AIFB 

The AIFB logo features the letters 'AIFB' in a bold, black, serif font, followed by a green square containing a white stylized 'O' shape.

Scalability Problems of DL Reasoning

- Main reasoning problem: query answering
 - assuming the number of instances is large

In a Tableaux Calculus...

- ...when I ask a query $KB \models C(X)$, I need to check $KB \models C(\alpha)$ for each α
 - not a very efficient algorithm
- ...it is difficult to isolate the part of the ABox relevant to the query
 - usually only a subset of the ABox is relevant to a query
- ...it is difficult to apply optimizations
 - DB statistic was shown to be crucial in practice

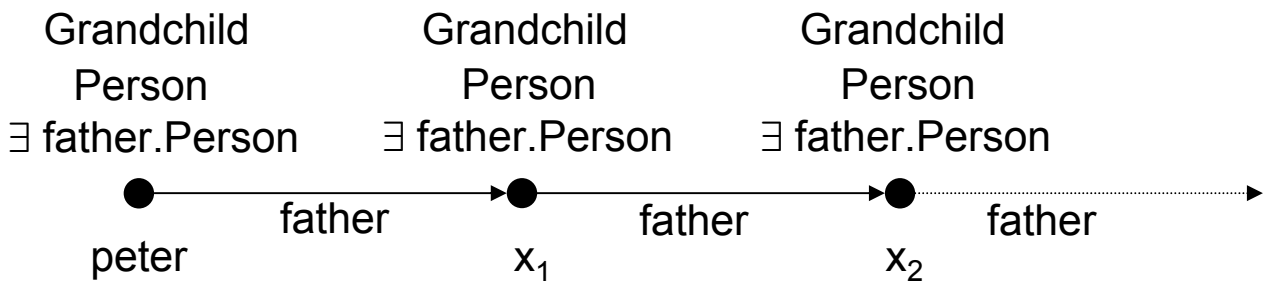
In a Deductive Database...

- ...query answer is computed in one pass bottom-up
- ...techniques exist to propagate bindings
 - magic sets propagate bindings to select the relevant ABox part
- ...efficient statistics-based optimizations exist

So What is the Main Problem?

$Person(Peter)$	Peter is a person.
$Person \sqsubseteq \exists father.Person$	Each person has a father who is a person.
$\exists father.(\exists father.Person) \sqsubseteq Grandchild$	Things having a father of a father who is a person are grandchildren.

- Straightforward application of 'rules' **does not** terminate:



- The problem is reflected in the clausal form:
 - contains **function symbols** obtained through skolemization

$Person \sqsubseteq \exists parent.Person$	\rightsquigarrow	$Person(f(x)) \leftarrow Person(x)$
		$parent(x, f(x)) \leftarrow Person(x)$

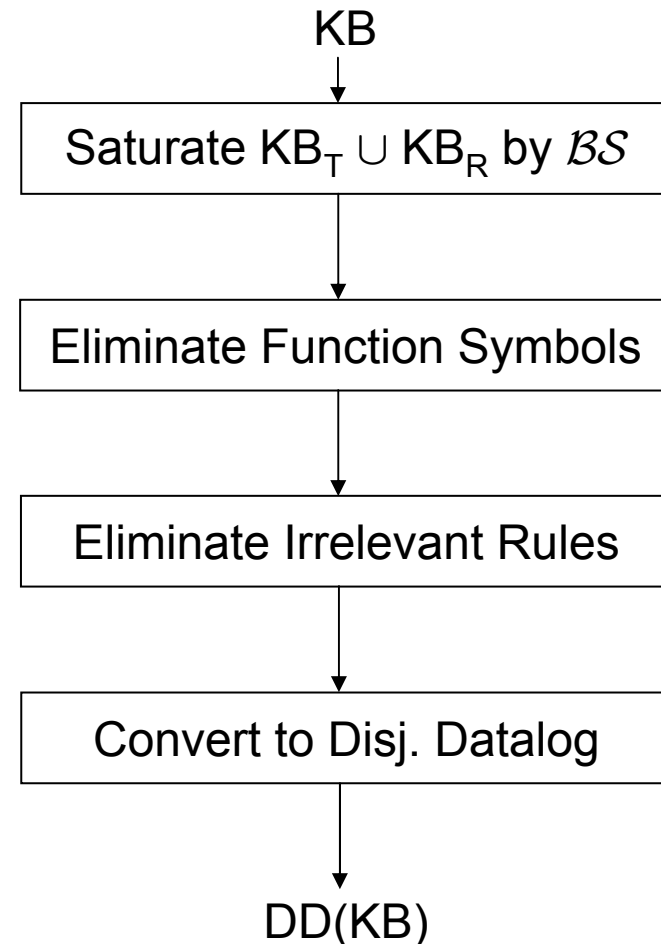
Solution: Eliminate Function Symbols

For a KB, compute a disjunctive datalog program $DD(KB)$ such that

$$KB \models \alpha \text{ iff } DD(KB) \models \alpha$$

(α is a ground fact $C(a)$ or $R(a,b)$)

- Handles $SHIQ(\mathbf{D})$ (no nominals)
- Principle:
 - find a calculus \mathcal{C} for deciding satisfiability of KB
 - create $DD(KB)$ such that:
 - each refutation by \mathcal{C} in KB can be reduced to a refutation in $DD(KB)$
 - each refutation in $DD(KB)$ can be reduced to a refutation in KB
 - since refutations can be reduced in both ways, KB and $DD(KB)$ are equisatisfiable



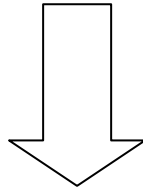
Complexity

- Algorithm runs in ExpTime
 - worst-case optimal
 - tableaux runs in $2N$ ExpTime
- **Data complexity** is much better:
 - assume that TBox constant; measure complexity in $|ABox|$
 - NP-complete in general
 - for Horn-*SHIQ* it is P-complete
 - does not support disjunctive reasoning (reasoning-by-cases)
- Theoretical hope for providing reasoners capable of dealing with large ABoxes

DL-safe Subset of SWRL

- SWRL is **undecidable**
 - benefits of DL systems, such as optimized algorithms are lost
- DL-safe rules: decidable fragment of SWRL
 - each variable must occur in one literal with predicate outside KB in body
 - this restricts the applicability of rules to explicit individuals
 - rules can be appended to DD(KB) → **practical reasoning algorithm**

Homeworker(x) \leftarrow Person(x), livesAt(x,y), worksAt(x,y)



Not DL-safe, since x and y occur only in DL-atoms.

Homeworker(x) \leftarrow Person(x), livesAt(x,y), worksAt(x,y), $\mathcal{O}(x)$, $\mathcal{O}(y)$

We assume there is a fact $\mathcal{O}(\alpha)$ for each individual α in the ABox.

Global Picture

- Relationship between DLs and DDs has three axes:
 - each axis imposes requirements on the DD system
- 1. Is KB Horn, i.e. is reasoning by cases allowed?
 - a non-disjunctive system suffices
 - + must use a disjunctive system
- 2. Does KB use number restrictions?
 - an ordinary engine suffices
 - + must use an engine capable of handling equality
- 3. Does KB use existential quantifiers?
 - a function-free program is obtained
 - + existential quantifiers must be handled:
 - simple skolemization may lead to undecidability
 - saturation can be used to eliminate function symbols and produce a function-free program