

Practical Reasoning with OWL and DL-Safe Rules

Peter Haase¹

Pascal Hitzler¹

Markus Krötzsch¹

Jürgen Angele²

Rudi Studer¹²³

¹ Institute AIFB, Universität Karlsruhe

² ontoprise GmbH, Karlsruhe

³ FZI Karlsruhe

- W3C recommendation since April 2004
- conceived as extension of RDFS
- three different flavours:
 - OWL Lite: based on a fragment of first-order logic (FOL)
 - OWL DL: larger fragment of FOL
 - OWL Full: “combination” of OWL DL with *all* features of RDFS (reification, ...)

↪ Here: **focus on OWL DL**

- OWL DL is based on the **description logic** $SHOIN(D)$
- typical reasoning tasks for OWL DL are **decidable**
↪ relevant for many applications
- implementing OWL DL reasoning is difficult
- largely **compatible with RDFS**
(but not containing all of RDFS)



- 1 Description logics
- 2 The KAON2 reasoner
- 3 Conjunctive queries
- 4 Semantic Web rules
- 5 Outlook

DLs are logical formalisms that correspond to certain fragments of first-order logic

- used for describing knowledge in a precise and well-defined way
 \rightsquigarrow suitable as ontology languages
- formal semantics allows for unambiguous interpretation
- reasoning with DLs typically decidable
 \rightsquigarrow fully automatic processing of knowledge
- there are many DLs, defined by their expressive features:
 expressivity vs. complexity

Three types of modelling primitives:

- 1 **Individuals**, describing single elements of the domain.
E.g. *eswc2006*, *markus*, ...
- 2 **Concepts**, describing sets of individuals.
E.g. *Conference*, *Human*, ...
- 3 **Roles**, describing binary relations between individuals.
E.g. *loves*, *participates_in*, ...

Simple assertions about individuals, e.g.

eswc2004 : *Conference*

(*markus*, *eswc2004*) : *participates_in*

↪ **ABox** of assertional axioms

Applications involve **complex classes** and relationships between them

↪ combine roles and classes with logical operators

A simple DL: \mathcal{ALC}

- \mathcal{ALC} : “Attribute Language with Complement”
- Building concepts:

$C \sqcap D$	individuals in C and D
$C \sqcup D$	individuals in C or D
$\neg C$	individuals not in C
$\exists R.C$	individuals with some relation R to C
$\forall R.C$	individuals with all relations R to C

- Stating relationships between concepts

$C \sqsubseteq D$	all individuals of C are also in D
$C \equiv D$	the individuals of C and D are the same

\rightsquigarrow **TBox** of terminological axioms

Knowledge base = ABox + TBox

Conference \sqsubseteq *Event*

Every conference is an event.

Conference $\sqsubseteq \forall$ *participant*.*Person*

Everybody who participates in a conference is a person.

Person \sqsubseteq *Female* \sqcup *Male*

Persons are female or male.

eswc2006 : *Conference*

ESWC2006 is a conference.

(*eswc2006*, *markus*) : *participant*

Markus participates in ESWC.

We would like to **conclude** also that

markus : *Person*

Markus is a person.

Classical tasks for reasoning with description logics:

- **Instance checking.** Is individual a in concept C ?
- **Concept subsumption.** Is concept C more general than D ?
- **Concept satisfiability.** Does the definition of concept C allow any instances of this concept?
- **Knowledge base satisfiability.** Is the combined knowledge of TBox and ABox free of contradictions?

Checking knowledge base satisfiability suffices

- Individual a in concept C , if $a : \neg C$ leads to a contradiction
- C is more general than D , if $x : C \sqcap \neg D$ leads to a contradiction
- C is unsatisfiable if $x : C$ leads to a contradiction
(with x is some hitherto unused individual)

Example – drawing conclusions

Conference \sqsubseteq *Event*

Every conference is an event.

Conference $\sqsubseteq \forall$ *participant.Person*

Everybody who participates in a conference is a person.

markus : *Person*

Markus is a person.

(markus, eswc2006) : *participant*

Oops! ESWC participates in Markus?

Isn't this a contradiction since ESWC is no person?

eswc2006 : *Person*

Uh oh ... ESWC is a person.

What is missing?

Person $\sqsubseteq \neg$ *Events*

Persons are not events.

eswc2006 : *Conference*

ESWC is a conference.

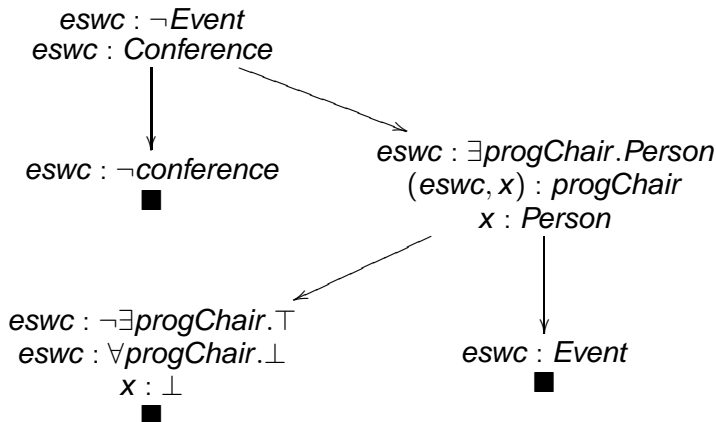
Most common reasoning method: **tableau calculus**

- closely related to tableaux in FOL and modal logics
- approach: try to construct a valid model for a knowledge base
 \rightsquigarrow determine whether knowledge base is satisfiable
- possible results:
 - 1 model constructed successfully \rightsquigarrow satisfiable
 - 2 *all* attempts of model construction fail \rightsquigarrow unsatisfiable
 - 3 the algorithms fails to halt

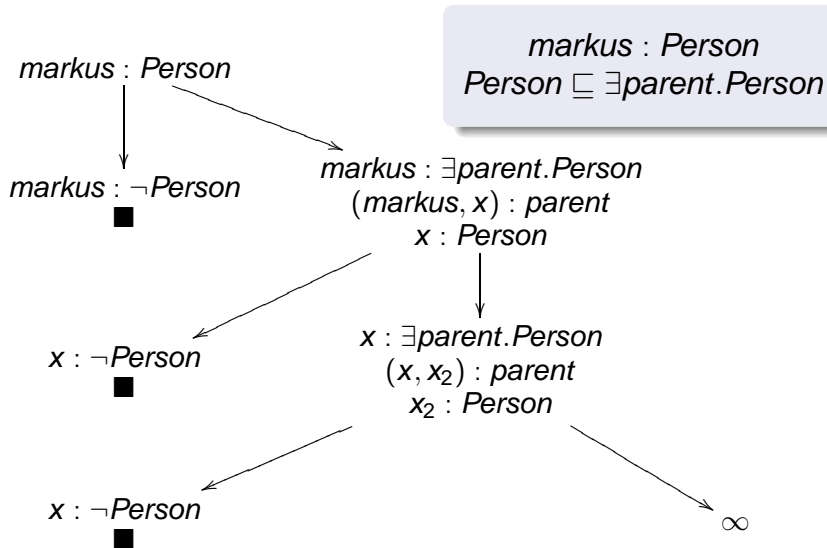
The tableau calculus

$eswc : Conference$
 $Conference \sqsubseteq \exists progChair. Person$
 $\exists progChair. \top \sqsubseteq Event$

Can we derive
 $eswc : Event$?



Termination



⇒ sophisticated **blocking techniques** needed to ensure termination

More expressive DLs

Concepts

\mathcal{ALC}	Boolean operators, modalities: $\sqcap, \sqcup, \neg, \forall R, \exists R$	
\mathcal{N}	Number restrictions	$\geq 2 \text{ has_child}$ $\leq 1 \text{ has_mother}$
\mathcal{Q}	Qualified number restr.	$\geq 2 \text{ has_child.Doctor}$
\mathcal{O}	Nominals	$\{\text{peter, pascal, markus}\}$

Individuals

\approx	Same	$\text{markus} \approx m_krötzsch$
\neq	Different	$\text{peter} \neq \text{markus}$

Roles

\mathcal{H}	Subrole hierarchy	$\text{has_mother} \sqsubseteq \text{has_ancestor}$
\mathcal{I}	Inverse roles	$\text{has_ancestor}^{-1} \sqsubseteq \text{has_offspring}$
\mathcal{S}	(\mathcal{ALC} +) role transitivity	$\text{Trans}(\text{has_ancestor})$

OWL DL: \mathcal{SHOIN} + concrete domains (datatypes)

Recall: $P \subseteq NP \subseteq PSPACE \subseteq EXPTIME \subseteq NEXPTIME$

Reasoning with (many) DLs is computationally **hard**:

- *ALC* with empty TBox: PSPACE
- *ALC*: EXPTIME
- *SHOIN* (OWL DL): NEXPTIME

However, **worst case \neq average case!**

- Highly optimized (practically efficient) reasoners exist.
- Some more restricted DLs are tractable (= decidable in P)

Conference $\sqsubseteq \forall \text{participant}.(\text{Female} \sqcup \text{Male})$

is expressed in OWL/RDF as:

```
<owl:Class rdf:ID="Conference">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="participant"/>
      <owl:allValuesFrom>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="Female"/>
          <owl:Class rdf:about="Male"/>
        </owl:unionOf>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

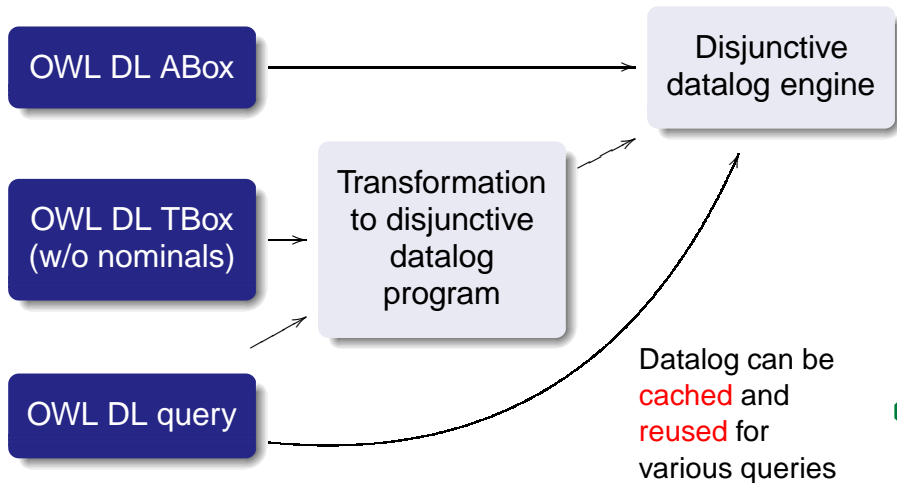
KAON2: OWL reasoner **and** ontology management API

KAON2 reasoner:

- not based on tableau methods
- based on first-order resolution calculus
- goal: efficient reasoning for large ABoxes
- binaries available from <http://kaon2.semanticweb.org>

Reasoning in KAON2: overview

KAON2 answers queries in two processing steps:

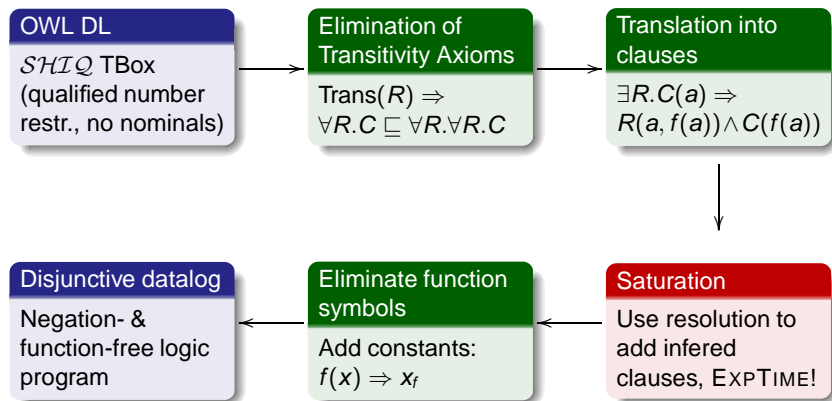


OWL can be translated into first-order logic:

$$\begin{array}{ll} C \sqsubseteq D & \mapsto \quad \forall x. C(x) \rightarrow D(x) \\ a : \exists R.C & \mapsto \quad \exists x. (R(a, x) \wedge C(x)) \\ a : \leq 1 R & \mapsto \quad \forall x. \forall y. ((R(a, x) \wedge R(a, y)) \rightarrow x \approx y) \\ \dots & \mapsto \quad \dots \end{array}$$

\rightsquigarrow application of first-order reasoning techniques possible

KAON2 transformation in detail



Example

Person $\sqsubseteq \exists \text{childOf}.\text{Person}$
NoSiblings $\sqsubseteq \text{Person} \sqcap \forall \text{childOf}.\leq 1 \text{hasChild}.\top$
Parent $\equiv \exists \text{hasChild}.\top$
childOf $\equiv \text{hasChild}^{-1}$

childof(X, X_{f_0}) : - person(X), kaon2s $_{f_0}$ (X, X_{f_0}).

person(X_{f_0}) : - person(X), kaon2s $_{f_0}$ (X, X_{f_0}).

person(X) : - nosiblings(X).

kaon2equal(Y_1, Y_2) : - nosiblings(X), childof(X, Z),
haschild(Z, Y_1), haschild(Z, Y_2).

haschild(X, X_{f_1}) : - parent(X), kaon2s $_{f_1}$ (X, X_{f_1}).

parent(X) : - haschild(X, Y).

haschild(Y, X) : - childof(X, Y).

childof(Y, X) : - haschild(X, Y).

- 1 Process query and TBox to obtain disjunctive datalog \rightsquigarrow EXPTIME
- 2 Add ABox
- 3 Use Datalog reasoner for query answering \rightsquigarrow NP

Features

- TBox translation not necessary for every query
- Datalog-reasoning exploits well-known optimisation strategies (e.g. *magic sets*)
- Data complexity is NP
- Overall algorithm is worst-case optimal (EXPTIME)

Problem: comparison to other reasoners difficult

due to different architectures, caching mechanisms, pre-processing, ...

- best results for large ABoxes, medium complexity TBoxes
 ↪ generally superior to tableaux algorithms
- still able to solve non-trivial TBox problems
 ↪ generally inferior to tableaux algorithms
- no support for nominals
- additional reasoning features: see below
- powerful API and ontology management tools: see second half of tutorial

Conjunctive queries

The knowledge base can be queried for conjunctions of

- terms $A(x)$ and $\neg A(x)$ where A is a concept name, and
- terms $R(x, y)$ where R is a *simple* role (one without transitive subroles).

The conjunctive query asks for concrete **individuals** that are valid fillers for the **distinguished** variables.

Example

$\exists y, z : \text{Conference}(x) \wedge \text{location}(x, y) \wedge \text{weather}(y, z) \wedge \neg \text{rainy}(z):$

“Find **known** conferences at some (possibly unknown) location where the weather is no rainy.”

\rightsquigarrow **additional query expressivity** to extend DL reasoning.

Expressiveness of OWL is limited

Example: uncles in OWL

Given DL roles *parent*, *brother*, and *uncle*, one cannot describe their exact relationship, i.e.

“Someones uncle is the brother of her parent”

cannot be expressed in OWL.

⇨ Rules might add additional expressiveness:

$$parent(x, y) \wedge brother(y, z) \rightarrow uncle(x, z)$$

⇨ **Semantic Web Rule Language** (SWRL)

Problem

OWL DL + SWRL is not decidable anymore.

↪ restriction of SWRL rules

Safety condition

Every variable appears in a non-DL atom in the rule condition.

Example:

$$O(x), O(y), O(z), \textit{parent}(x, y) \wedge \textit{brother}(y, z) \rightarrow \textit{uncle}(x, z)$$

where O is not a concept from the DL knowledge base.

$O(x), O(y), O(z), \text{parent}(x, y) \wedge \text{brother}(y, z) \rightarrow \text{uncle}(x, z)$

What means $O(x)$?

\rightsquigarrow Add fact $O(a)$ for every known individual a .

Intuition: DL-safe rules are SWRL rules that are **restricted to known individuals**.

In KAON2:

- DL-safe rules can be added to the disjunctive datalog output.
- No additional pre-processing required.
- Complexity of Datalog reasoning still NP.

- **Further extension of OWL DL:** OWL 1.1

↪ additional expressivity with similar complexity

[http://www-db.research.bell-labs.com
/user/pfps/owl/overview.html](http://www-db.research.bell-labs.com/user/pfps/owl/overview.html)

- **Rule languages:** W3C working group “RIF”

<http://owl-workshop.man.ac.uk/Tractable.html>

- **Tractable fragments of OWL:** interesting DLs with polynomial decision problems ↪ e.g. Horn-*SHIQ*, EL++, DL-Lite, ...

<http://www.w3.org/2005/rules/wg.html>