Practical Reasoning with OWL and DL-Safe Rules

Peter Haase[1] Pascal Hitzler[1] Markus Krötzsch[1]

Jürgen Angele[2] Rudi Studer[123]

[1] Institute AIFB, Universität Karlsruhe
[2] ontoprise GmbH, Karlsruhe
[3] FZI Karlsruhe

Web Ontology Language OWL

- 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, . . .)

$\leadsto$ Here: focus on OWL DL

OWL DL

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

Overview

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

Description logics (DLs)

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

Simple description logics

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.

$$\begin{aligned} \mathit{eswc2004} &: \mathit{Conference} \\ (\mathit{markus}, \mathit{eswc2004}) &: \mathit{participates\_in} \end{aligned}$$

$\leadsto$ ABox of assertional axioms

Applications involve complex classes and relationships between them
$\leadsto$ 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 individucals of C and D are the same

$\rightsquigarrow$ TBox of terminological axioms

Knowledge base = ABox + TBox

Example

$\mathit{Conference} \sqsubseteq \mathit{Event}$	Every conference is an event.
$\mathit{Conference} \sqsubseteq \forall \mathit{participant}.\mathit{Person}$	Everybody who participates in a conference is a person.
$\mathit{Person} \sqsubseteq \mathit{Female} \sqcup \mathit{Male}$	Persons are female of male.
$\mathit{eswc2006} : \mathit{Conference}$	ESWC2006 is a conference.
$(\mathit{eswc2006}, \mathit{markus}) : \mathit{participant}$	Markus participates in ESWC.

We would like to conclude also that

$\mathit{markus} : \mathit{Person}$	Markus is a person.

Reasoning tasks

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 satisfibility. Does the definition of concept C allow any instances of this concept?
- Knowledge base satisfibility. 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.
$(markus, escw2006) : participant$	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.

Reasoning

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
  $\leadsto$ determine whether knowledge base is satisfiable
- possible results:
  1. model constructed successfully $\leadsto$ satisfiable
  2. *all* attempts of model construction fail $\leadsto$ 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 $eswc : Event$?

$$eswc : \neg Event$$
$$eswc : Conference$$

$$eswc : \neg conference$$
$\blacksquare$

$$eswc : \exists progChair.Person$$
$$(eswc, x) : progChair$$
$$x : Person$$

$$eswc : \neg\exists progChair.\top$$
$$eswc : \forall progChair.\bot$$
$$x : \bot$$
$\blacksquare$

$$eswc : Event$$
$\blacksquare$

Termination

$$markus : Person$$
$$Person \sqsubseteq \exists parent.Person$$

$markus : Person$

$markus : \neg Person$ ■

$markus : \exists parent.Person$
$(markus, x) : parent$
$x : Person$

$x : \neg Person$ ■

$x : \exists parent.Person$
$(x, x_2) : parent$
$x_2 : Person$

$x : \neg Person$ ■

∞

$\leadsto$ 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	≥ 2 *has_child* ≤ 1 *has_mother*
$\mathcal{Q}$	Qualified number restr.	≥ 2 *has_child.Doctor*
$\mathcal{O}$	Nominals	$\{peter, pascal, markus\}$

Individuals

$\approx$	Same	*markus* $\approx$ *m_krötzsch*
$\not\approx$	Different	*peter* $\not\approx$ *markus*

Roles

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

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

Complexity and efficiency

Recall: P $\subseteq$ NP $\subseteq$ PSpace $\subseteq$ ExpTime $\subseteq$ NExpTime

Reasoning with (many) DLs is computationally hard:

- $\mathcal{ALC}$ with empty TBox: PSpace
- $\mathcal{ALC}$: ExpTime
- $\mathcal{SHOIN}$ (OWL DL): NExpTime

However, worst case $\neq$ average case!

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

Back to OWL DL

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

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 ontology management suite

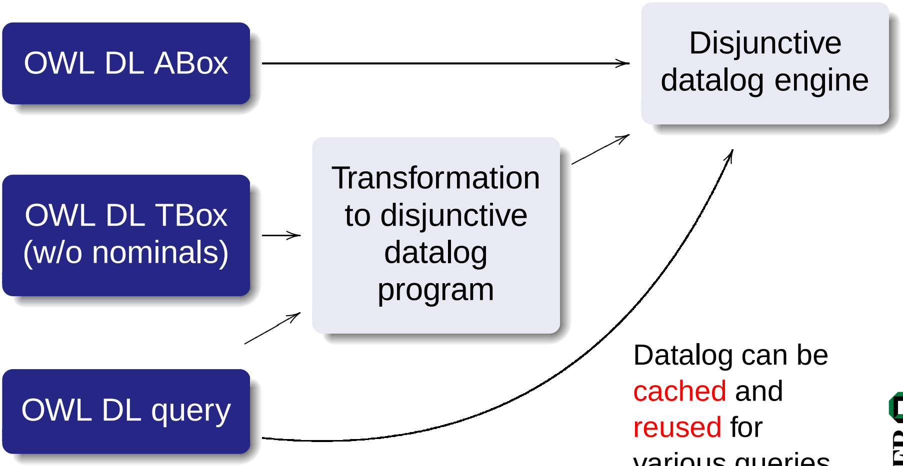
KAON2: OWL reasoner and ontology management API

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:



Datalog can be cached and reused for various queries

OWL as a fragment of FOL

OWL can be translated into first-order logic:

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

$\rightsquigarrow$ application of first-order reasoning techniques possible

KAON2 transformation in detail

OWL DL

$\mathcal{SHIQ}$ TBox (qualified number restr., no nominals)

→

Elimination of Transitivity Axioms

Trans(R) $\Rightarrow$ $\forall R.C \sqsubseteq \forall R.\forall R.C$

→

Translation into clauses

$\exists R.C(a) \Rightarrow R(a, f(a)) \wedge C(f(a))$

↓

Saturation

Use resolution to add infered clauses, ExpTime!

←

Eliminate function symbols

Add constants: $f(x) \Rightarrow x_f$

←

Disjunctive datalog

Negation- & function-free logic program

Example

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

$$\begin{aligned}
\text{childof}(X, X_{f0}) &\text{ :- } \text{person}(X), \text{kaon2s}_{f0}(X, X_{f0}). \\
\text{person}(X_{f0}) &\text{ :- } \text{person}(X), \text{kaon2s}_{f0}(X, X_{f0}). \\
\text{person}(X) &\text{ :- } \text{nosiblings}(X). \\
\text{kaon2equal}(Y_1, Y_2) &\text{ :- } \text{nosiblings}(X), \text{childof}(X, Z), \\
&\qquad \text{haschild}(Z, Y_1), \text{haschild}(Z, Y_2). \\
\text{haschild}(X, X_{f1}) &\text{ :- } \text{parent}(X), \text{kaon2s}_{f1}(X, X_{f1}). \\
\text{parent}(X) &\text{ :- } \text{haschild}(X, Y). \\
\text{haschild}(Y, X) &\text{ :- } \text{childof}(X, Y). \\
\text{childof}(Y, X) &\text{ :- } \text{haschild}(X, Y).
\end{aligned}$$

Complexity and efficiency

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)

KAON2: strengths and limitations

Problem: comparison to other reasoners difficult

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

- best results for large ABoxes, medium complexity TBoxes
  $\rightsquigarrow$ generally superior to tableaux algorithms
- still able to solve non-trivial TBox problems
  $\rightsquigarrow$ 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

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

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

$\leadsto$ additional query expressivity to extend DL reasoning.

Semantic Web rules

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.

$\rightsquigarrow$ Rules might add additional expressiveness:

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

$\rightsquigarrow$ Semantic Web Rule Language (SWRL)

DL-safe rules

Problem

OWL DL + SWRL is not decidable anymore.

$\rightsquigarrow$ restriction of SWRL rules

Safety condition

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

Example:

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

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

DL-safe rules in KAON2

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

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.

OWL: future development

- Further extension of OWL DL: OWL 1.1
  $\rightsquigarrow$ additional expressivity with similar complexity
  `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: intersting DLs with polynomial descision problems $\rightsquigarrow$ e.g. Horn-$\mathcal{SHIQ}$, EL++, DL-Lite, ...
  `http://www.w3.org/2005/rules/wg.html`