Introduction to Computational Linguistics

Ontologies

Lecture: Monday, May 18, 2009
Exercise (Christian Federmann): Thursday, May 28, 2009

Hans-Ulrich Krieger

Language Technology Lab
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
krieger@dfki.de

- lecture & exercise will be put on course homepage tomorrow
- deadline for returning YOUR submission: Tuesday, May 26
- email YOUR solution as a **single** TEXT, PS, or PDF file
- MY solution will be put on course page

Overview

- What is an Ontology?
- Examples
- Digging Deeper
  - OWL
  - SWRL
  - OWLIM
- Outlook

What is an Ontology?

What is an Ontology?

Ontology [Greek]: most fundamental branch of general metaphysics, dealing with the study of existence (science of being; Aristotle, 384BC–322BC)

first occurrence of the term *ontologia* as we use it today by Jacob Lorhard (1561–1609; Jacobo Lorhardo, Jacobus Lorhardus) in first edition of *Ogdoas Scholastica* (1606)

discipline can be subdivided into

- formal ontology (or universal science)
- material ontology

Formal Ontology

question: what are the truth-determining foundations of general metaphysics, i.e., what are the most general rules directing our decisions, leading to more specialized rules (e.g., in medicine): *first principles*

- Law of Identity
  $A = A$: an axiom in most logics

- Law of Excluded Middle
  either P or $\neg P$

- Law of Non-Contradiction
  proof by contradiction: $(\neg P \Rightarrow (R \wedge \neg R)) \Rightarrow P$

Material Ontology

what are the fundamental categories of being? (Aristotle)
more general view: find out what entities and what types of entities exist!

similar to the idea of first principles: start with *Being* (does not need any definition), and add subcategories, such as *Substance*

what does it mean for an entity to be member of a certain category?
sharing prototypical values for category-specific properties!

Reappearance of the Wheel

Aristotle's theory of categories and classification "reappears" in philosophy and many other scientific disciplines:

- biology
- ...
- CL, AI, CS, LT, ...
  - (computational) linguistics
  - artificial intelligence
  - computer science
  - information science, lexicography, semantic web, ...

What is an Ontology: Tom Gruber (1993)

A conceptualization is an *abstract*, simplified view of the world that we wish to represent for some purpose. . . . An **ontology** is an *explicit specification* of a conceptualization. . . . When the knowledge of a domain is represented in a *declarative formalism*, the set of objects that can be represented is called the universe of discourse. This set of *objects*, and the describable *relationships* among them, are reflected in the *representational vocabulary* with which a knowledge-based program represents knowledge.

What is an Ontology (Gruber)

an ontology is a description of objects (categories & individuals) and relationships between objects

1+is-a relation: *taxonomy*; 1+2: *thesaurus*

1. categories/concepts/classes/types: `Man`
2. (built-in) relations between categories: `Man subclassOf Human`
3. individuals/instances: `peter`, `mary`
4. relations/roles between individuals: `peter isMarriedTo mary`

what is missing here? **semantics!** (later)

Why are we interested in Ontologies?

- pure **epistemological** aspects—no practical interest in running systems
  - build models of (specific parts of) the world
  - find encoding that conforms with taken observations
  - good model should predict facts not encountered so far
  - questions:
    - what can be encoded in the representational vocabulary and what can not?
    - what is the computational complexity of the representation language?
    - is the language decidable?
- very **practical** aspects $\longrightarrow$ next slide

Application Areas

- query expansion in IR & QA
- DB access & ontology retrieval
- word sense disambiguation
- ontology population through IE
- language-specific inferences on lexical semantic representation
- general inferences dealing with world knowledge

Examples

Examples

- thesauri
- WordNet
- FrameNet
- SUMO/MILO
- description logics & OWL

Merriam-Webster Online Thesaurus

Word: human

Function: adjective

Text: relating to or characteristic of human beings (it's human nature to care about what people think of us)

Synonyms: mortal, natural

Related Words: anthropoid, hominid, humanlike, humanoid

Near Antonyms: angelic (or angelical), divine, godlike, superhuman, supernatural; immortal, omnipotent, omniscient; animal, beastly, bestial, brute; inhuman, robotic

Antonyms: nonhuman

Merriam-Webster Online Thesaurus, cont.

Word: human

Function: noun

Text: a member of the human race (humans are the only mammals not endowed with a natural defense against the elements, such as fur or a thick hide)

Synonyms: being, bird, body, creature, customer, devil, guy, head, individual, life, man, mortal, party, person, scout, sort, soul, specimen, thing, wight

Related Words: hominid, homo, humanoid; brother, fellow, fellowman, neighbor; celebrity, personage, personality, self, somebody

Near Antonyms: animal, beast, brute

WordNet—Hypernyms of Human

WN *hierarchically* organizes nouns, verbs, adjectives, and adverbs into synonym sets which refer to lexical concepts (155,327 unique strings & 117,597 synsets in WordNet 3.0)

Sense 1/noun: *a human being*

```
person, individual, someone, somebody, mortal, human, soul
       => organism, being
           => living thing, animate thing
               => object, physical object
                   => entity
       => causal agent, cause, causal agency
           => entity
```

WordNet—Hypernyms of Human, cont.

Sense 2/noun: *any living or extinct member of the family Hominidae*

```
homo, man, human being, human
       => hominid
           => primate
               => placental, placental mammal, eutherian, eutherian mammal
                   => mammal
                       => vertebrate, craniate
                           => chordate
                               => animal, animate being, beast, brute, ....
                                   => organism, being
                                       => living thing, animate thing
                                           => object, physical object
                                               => entity
```

Relations we are interested in w.r.t. Concept C

- **synonyms** concepts having the same meaning as C
- **antonyms** concepts that do not share any properties with C
- **hypernyms** concepts that are more general than C
- **hyponyms** concepts that are more specific than C
- **holonyms** concepts that contain C as a part
- **meronyms** concepts that are part of C

FrameNet—Human, Again

FN lists semantic and syntactic combinatory possibilities (valences) of each word in each of its senses ($>$ 10,000 lexical units; $\approx$ 800 hierarchical semantic frames)

two lexical units for *human*: `human_being.n` and `human.n`

but semantic frame is `People`

several "subclasses" of `People`, e.g., `People_by_age`

binary relations, connecting frames: `Inherits_From`, `Uses`, ...

example: `People_by_age Inherits_From People` ("specialization")
`People_by_age Uses Age` ("properties")

SUMO & MILO

Suggested Upper Merged Ontology: very basic concepts & axioms (similar upper ontologies: DOLCE, PROTON)

higher-order LISPish specification language SUO-KIF

```
(instance instance BinaryPredicate)
(subrelation immediateInstance instance)
(instance immediateInstance AsymmetricRelation)

(=> (immediateInstance ?ENTITY ?CLASS)
    (not (exists (?SUBCLASS)
             (and (subclass ?SUBCLASS ?CLASS)
                  (not (equal ?SUBCLASS ?CLASS))
                  (instance ?ENTITY ?SUBCLASS)))))
```

SUMO & MILO, cont.

MId-Level Ontology: bridges between the abstract content of SUMO and various domain ontologies

all ontologies together: 20,000 terms and 60,000 axioms

partial inference support via Vampire

```
(subclass HumanSlave Human)

(=> (instance ?SLAVE HumanSlave)
    (exists (?PERSON)
      (and (instance ?PERSON Human)
           (not (equal ?PERSON ?SLAVE))
           (possesses ?PERSON ?SLAVE))))
```

SUMO & MILO—That Human Thing, Again

mappings of concepts to WordNet lexicon
example *human*: found the two senses from WordNet

```
(partition Human Man Woman)
(subclass Human CognitiveAgent)
(subclass Human Hominid)
(subclass Man Human)

(<=> (attribute ?PERSON Unemployed)
     (and (instance ?PERSON Human)
          (forall (?ORG)
            (not (employs ?ORG ?PERSON)))))
```

Description Logics

family of logic-based knowledge representation formalisms
DL example: OWL (later!)

descendants of semantic networks and KL-ONE

describe domain in terms of concepts, roles, and individuals

complex expressions through concept-forming constructors

$\texttt{HumanSlave} \equiv$

$\quad \texttt{Human} \sqcap \exists\ \texttt{possesses}^{-1}.\ (\texttt{Human} \sqcap \neg\ \texttt{Slave})$

$\texttt{HappyFather} \equiv$

$\quad \texttt{Man} \sqcap \forall\ \texttt{hasChild}.\ (\texttt{Doctor} \sqcap \exists\ \texttt{hasFriend}.\ (\texttt{Rich} \sqcup \texttt{Famous}))$

Description Logics, cont.

model-theoretic semantics (decidable 2-var fragment of FOL)

sound & complete decision procedures

highly optimized implemented systems

increasing importance for

- Tim Berners-Lee's vision of a Semantic Web
- language technology (ontology-based information systems)
- artificial intelligence (multi-agent systems, user modeling)
- computer science (deductive, object-oriented data bases)

Recap: What is an Ontology

similarities between examples indicate that

- I take a liberal stance here what an ontology is
- we always construct ontologies when conceptualizing a domain
  1. categories/concepts/classes/types
  2. distinguished sub/super relationship
  3. individuals/instances/entities
  4. relations/roles/properties/attributes
- but: formal ontology languages must address
  - semantics: well-defined (yes)
  - decidability: sound (yes) & complete calculus (yes .. no)
  - tractability: average-case problems (yes .. no)

OWL

The Semantic Web Vision

(syntactic) Web made possible through established standards:
TCP/IP, HTTP, HTML, ...

1st generation: mostly handwritten HTML pages

2nd generation: very often machine-generated active pages

next generation (we're just here!): resources should be more accessible to automated processes

- to be achieved via semantic markup
- metadata annotations, describing content/function

coincides with Tim Berners-Lee's vision of a Semantic Web

Semantic Web & Ontologies

semantic markup must be meaningful to automated processes

ontologies will play a key role here

- source of precisely defined terms (vocabulary)
- can be shared across applications and humans

increased formality facilitates machine understanding

very important: standards!

long road:
XML, URI, **RDF**, **RDFS**, DAML & OIL, **OWL**, **SWRL**, ...

RDF: Resource Description Framework

- general-purpose language for representing information
- provides a lightweight ontology system
- enabling technology for the Semantic Web
- XML exchange syntax (but also N3, N-Triples)
- RDF data model: triple
- idea: everything can be represented as a triple

RDF, cont.

- **triple**: $\langle$ subject, predicate, object $\rangle$
- subject, predicate, object: URIs or XSD literals
  (or again triples: reification)
- **URI**: Uniform Resource Identifier ($\approx$ Web identifier)
  e.g., http://www.w3.org/2002/07/owl#intersectionOf
- **XSD**: XML Schema Datatypes
  typed **literals**, e.g., "2.4"^^xsd:decimal

RDFS: RDF Schema

- describes how to use RDF to describe RDF vocabularies
- defines other built-in RDF vocabulary (domain, subClassOf)
- class & property system similar to OOPL (e.g., Java)
- RDF(S) semantics via **axiomatic triples** & **entailment rules** (Hayes 2004), e.g.,
  - $\langle$rdf:type, rdf:type, rdf:Property$\rangle$
    $\langle$rdfs:subPropertyOf, rdfs:subPropertyOf, rdfs:subPropertyOf$\rangle$
  - $\langle ?p$, rdfs:domain, $?d\rangle \wedge \langle ?s, ?p, ?o\rangle \Rightarrow \langle ?s$, rdf:type, $?d\rangle$
    $\langle ?i$, rdf:type, $?d\rangle \wedge \langle ?d$, rdfs:subClassOf, $?c\rangle \Rightarrow \langle ?i$, rdf:type, $?c\rangle$

OWL

decidable instance of the description logics family (FOL fragment)

well-founded set-theoretical semantics

outcome of the DAML+OIL W3C standardization

de facto standard today to specify ontologies

RDFS-based syntax and ontological primitives
e.g., `rdfs:subClassOf`

fine-grained, more complex means as in RDFS
e.g., `owl:intersectionOf`

uses XML/RDF exchange syntax

ontology is a set of axioms describing classes and properties

Sublanguages of OWL

three increasingly expressive sublanguages

- base: $\mathcal{ALC}_{R^+} = \mathcal{S}$
- **OWL Lite**: sound & complete, decidable
  reasoning services: $\textsc{ExpTime}$ (worst case)
  optimized implementations: tableaux algorithms
- **OWL DL**: sound & complete, decidable ($\textsc{NExpTime}$)
  extends OWL Lite with disjunction & negation, cardinality constraints, and nominals
- **OWL Full**: reasoning usually undecidable

Class vs. Instance

classes & class properties (KL-ONE: **TBox**; DB: Schema)

`owl:Class`

`owl:equivalentClass rdfs:subClassOf`

`owl:intersectionOf owl:unionOf owl:complementOf`

`owl:disjointWith`

`owl:ObjectPoperty owl:DatatypeProperty`

`rdfs:subPropertyOf owl:equivalentProperty`

`rdfs:domain rdfs:range`

instances or individuals (**ABox**; DB: complete knowledge)

`owl:sameAs owl:differentFrom owl:AllDifferent`

plus instantiated object/datatype properties

plus: `rdf:type`, ...

ontology = TBox + ABox (+ RBox)

Properties

property characteristics

`TransitiveProperty`
`SymmetricProperty`
`FunctionalProperty`
`inverseOf`
`InverseFunctionalProperty`

property restrictions

`allValuesFrom`
`someValuesFrom`
`cardinality minCardinality maxCardinality`
`hasValue`

OWL 1.1: reflexive, irreflexive & asymmetric properties

OWL Class Constructors

Constructor	DL Syntax	Example
`Thing`, `Nothing`	$\top, \bot$	
`intersectionOf`	$C_1 \sqcap \ldots \sqcap C_n$	Human $\sqcap$ Male
`unionOf`	$C_1 \sqcup \ldots \sqcup C_n$	Doctor $\sqcup$ Lawyer
`complementOf`	$\neg C$	$\neg$Male
`oneOf`	$\{x_1, \ldots, x_n\}$	{john, mary}
`someValuesFrom`	$\exists P\,.\,C$	$\exists$hasChild . Lawyer
`allValuesFrom`	$\forall P\,.\,C$	$\forall$hasChild . Doctor
`maxCardinality`	$\leq nP$	≤ 1 hasChild
`minCardinality`	$\geq nP$	≥ 2 hasChild

XMLS datatypes possible in $\forall P\,.\,C$ and $\exists P\,.\,C$
 e.g., $\exists$hasAge . nonNegativeInteger

OWL Semantics

model theory relates expressions to interpretations $\mathcal{I} = \langle \mathcal{U}, \cdot^{\mathcal{I}} \rangle$
note: $\mathcal{U} = \top^{\mathcal{I}}$

- classes/concepts: subsets of $\mathcal{U}$
- object properties/roles: subsets of $\mathcal{U} \times \mathcal{U}$
- instances/individuals: elements of $\mathcal{U}$
- separation between object classes and datatypes (XMLSD): $\mathcal{U} \cap \mathcal{U}_D = \emptyset$
  - datatypes structured by built-in predicates
  - not possible to form new datatypes using ontology language
  - datatype properties: subsets of $\mathcal{U} \times \mathcal{U}_D$

OWL Semantics, cont.

extend interpretation function $\cdot^{\mathcal{I}}$ to concept expressions

- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $(\neg C)^{\mathcal{I}} = \mathcal{U} \setminus C^{\mathcal{I}}$
- $(\{x_1, \ldots, x_n\})^{\mathcal{I}} = \{x_1^{\mathcal{I}}, \ldots, x_n^{\mathcal{I}}\}$
- $(\exists P \,.\, C)^{\mathcal{I}} = \{x \mid \exists y \,.\, (x, y) \in P^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
- $(\forall P \,.\, C)^{\mathcal{I}} = \{x \mid \forall y \,.\, (x, y) \in P^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$
- $(\leq nP)^{\mathcal{I}} = \{x \mid \#\{y \mid (x, y) \in P^{\mathcal{I}}\} \leq n\}$
- $(\geq nP)^{\mathcal{I}} = \{x \mid \#\{y \mid (x, y) \in P^{\mathcal{I}}\} \geq n\}$

OWL Axioms

Axiom	DL Syntax	Example
`subClassOf`	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
`equivalentClass`	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
`disjointWith`	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq$ $\neg$Female
`sameAs`	$\{x_1\} \equiv \{x_2\}$	{president_bush} $\equiv$ {g_w_bush}
`differentFrom`	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{John} $\sqsubseteq$ $\neg${Peter}
`subPropertyOf`	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
`equivalentProperty`	$P_1 \equiv P_2$	cost $\equiv$ price
`inverseOf`	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
`transitiveProperty`	$P^+ \sqsubseteq P$	anchestor$^+$ $\sqsubseteq$ anchestor

- $\mathcal{I}$ satisfies $C_1 \equiv/\sqsubseteq C_2$ iff $C_1^{\mathcal{I}} =/\subseteq C_2^{\mathcal{I}}$ (same for properties)
- $\mathcal{I}$ satisfies ontology $\mathcal{O}$/is a *model* of $\mathcal{O}$ ($\mathcal{I} \models \mathcal{O}$) iff $\mathcal{I}$ satisfies *every* axiom in $\mathcal{O}$

Open-World Semantics & Non-Unique Name Assumption

OWL must allow for distributed information (Semantic Web!); information can be added incrementally: monotonicity; i.e., new information can NOT retract old; old can NOT be deleted

open-world assumption
what can NOT proven to be true is NOT believed to be false

example ontology:
$\{$Woman(alice), hasChild(alice, doris), hasChild(alice, boris)$\}$
question: $\{$alice$\} \sqsubseteq\ \leq 2$ hasChild vs. $\{$alice$\} \sqsubseteq\ \geq 2$ hasChild
at most: **don't know** at least: **yes** (but ...)

non-unique name assumption
individuals sharing different names need not be different/might be equal

Basic Inference Problems

consistency: check if knowledge is meaningful
is $\mathcal{O}$ consistent $\Longleftrightarrow$ there exists some model $\mathcal{I}$ of $\mathcal{O}$
is C consistent $\Longleftrightarrow$ $C^{\mathcal{I}} \neq \emptyset$ in some model $\mathcal{I}$ of $\mathcal{O}$

subsumption: structure knowledge, compute taxonomy
$C \sqsubseteq_{\mathcal{O}} D \Longleftrightarrow C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{O}$

equivalence: check whether two classes have same denotation
$C \equiv_{\mathcal{O}} D \Longleftrightarrow C^{\mathcal{I}} = D^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{O}$

NOTE: all problems are either reducible to consistency/satisfiability or subsumption

Reasoning With OWL

well-defined model-theoretic semantics

sound, complete & decidable algorithms for basic problems

highly optimized DL systems, e.g., FaCT, RACER, Pellet

why reasoning?

- design, maintenance & integration of ontologies
- querying class and instance data w.r.t. ontologies

Limitations of OWL
OWL & Rules:
SWRL & OWLIM

Good Things about DL/OWL

- strong concept language (but weak role language)
- sub-/supertype relationships between classes: easy
- implication & equivalence in class axioms: easy
- domain & range restriction on properties: easy
- certain property characteristics: easy
- cardinality constraints: easy

Limitations of OWL

- only unary and binary relations
- no role constructors & role composition
- missing: **rules** (weak property language)
- missing: counting & dynamic DSs/individuals
- missing: constraints
- missing: knowledge revision
- missing: handling of inconsistent knowledge

A W3C Proposal: SWRL

- SWRL = Semantic Web Rule Language
- combines OWL DL & RuleML
- combination loses decidability (existential quantifiers plus recursive rules)
- rules expressed in terms of classes, properties & individuals
- Horn-like rules
  $hasParent(?x, ?y) \wedge hasBrother(?y, ?z) \rightarrow hasUncle(?x, ?z)$

SWRL Abstract Syntax

- extends OWL DL abstract syntax by further axiom:
  `<axiom> ::= <rule>`
- rule is interpreted as an implication, consisting of a LHS (antecedent or body) and a RHS (consequent or head)
- LHS and RHS consist of a sequence of atoms, interpreted conjunctively
- atoms are of the form
  - $C(x)$
  - $p(x, y)$
  - sameAs(x, y)
  - differentFrom(x, y)
  - builtIn$(r, x, \ldots)$

  where C is an OWL class, P a property, r a built-in relation, and $x, y, \ldots$ either **variables** (new!), individuals, or data values

Extended Satisfaction Relation $\models$

interpretation $\mathcal{I}$ can be used to define a *satisfaction relation* $\models$ on syntactically well-formed class expressions and axioms

$\models$ can be straightforwardly be extended to cover the semantics of SWRL rules, as is done in FOL and Prolog

need valuation or assignment function $\alpha : V \mapsto \mathcal{U}$

rules are satisfied by $\mathcal{I}$ iff every variable binding satisfying the antecedent also satisfies the consequent

further requirement (safety): variables in the head have to be bound in the body

$\models$, cont.

- $\mathcal{I}, \alpha \models B \rightarrow H$ iff $\mathcal{I}, \alpha \models B$ implies $\mathcal{I}, \alpha \models H$
- body and head are a conjunction of atoms:
- $\mathcal{I}, \alpha \models A_1 \wedge \ldots \wedge A_n$ iff $\mathcal{I}, \alpha \models A_1$ and $\ldots$ and $\mathcal{I}, \alpha \models A_n$
- atoms are either unary or binary relations
- class/concept: $\mathcal{I}, \alpha \models C(t)$ iff $t^{\mathcal{I},\alpha} \in C^{\mathcal{I}}$
- property/role: $\mathcal{I}, \alpha \models p(t_1, t_2)$ iff $\langle t_1^{\mathcal{I},\alpha}, t_2^{\mathcal{I},\alpha} \rangle \in p^{\mathcal{I}}$
- terms are either variables or constants/individuals from $\top$
- $x^{\mathcal{I},\alpha} = \alpha(x)$
- $c^{\mathcal{I},\alpha} = c^{\mathcal{I}}$
- NO function symbols as in FOL (variant of Datalog)

Implementations

only partial (safe) SWRL implementation available yet:
Pellet, RACER, KAON2

- specialized tableaux algorithms for DL can NOT be easily extended to cover rules (hard-wired/built-in semantics)
- alternative 1: implement OWL semantics via axiomatic tuples (triples!) and entailment rules à la Hayes (2004) and ter Horst (2005)
  examples: **OWLIM**, Jena: **forward chaining** (data-driven inference)
- alternative 2: apply offline transformation into typed logic language
  examples: Flora2, Ontobroker (FLogic): backward chaining (goal-driven inference)

Forward Chaining

way to carry out all inferences at compile time

even useless inferences w.r.t. application

querying at run time reduces to an indexing problem

compute assertions entailed by a set of ground atoms/triples & a set of universally quantified implications $\{B_i \rightarrow H_i \mid i \in \mathbf{N}\}$

antecedent and consequent consist of constants and variables

Basic Naïve Algorithm

input R: set of if-then rules, T: set of RDF triples

repeat

$T' := T$

for each $r \in R$

for each binding $b \in match(body(r), T')$

$T := T \cup \{instantiate(head(r), b)\}$

until $T' = T$

Problems with Forward Chaining Approach

potentially large deductive closure, but total materialization usually not needed (compare: tabled backward chaining)

counting & dynamic data structures require introduction of new individuals; problem termination

cardinality constraints (counting!)

negation conflicts with order-independence of rules

Advantages of Forward Chaining Approach

basic idea easy to implement

no inference at run time, only indexing

fast

terminating (finite model property)
finite closure iff functions on RHS are NOT involved
functions usually introduce new material (URIs and XSD literals)

storage/access layer: from in-memory, XML-DBs, RDMS, AllegroGraph, ...

scales up well in practice

OWLIM

essentially Datalog ("function-free" Prolog)

support for RDF(S) & OWL through *axiomatic facts* and *entailment rules* à la Hayes (2004) and ter Horst (2005)

not even full OWL Lite

at the same time, rule language provides extensions not covered by OWL DL

predefined rule sets of increasing complexity

custom rule sets on top of RDFS/OWL support

developed by Ontotext (www.ontotext.com)

Axiomatic Triples and Entailment Rules for OWL OWLIM Syntax

```
<rdf:type> <rdf:type> <rdf:Property>
<rdfs:domain> <rdfs:domain> <rdf:Property>
<rdf:type> <rdfs:subPropertyOf> <rdf:type>
<rdfs:subPropertyOf> <rdfs:subPropertyOf> <rdfs:subPropertyOf>
```

```
s p o
p <owl:inverseOf> q
-------------------
o q s
```

```
p <rdf:type> <owl:TransitiveProperty>
x p y
y p z
-------------------------------------
x p z
```

```
x <owl:sameAs> y
x p z
----------------
y p z
```

```
x <owl:sameAs> y
x <owl:differentFrom> y
--------------------------
x <rdf:type> <owl:Nothing>
y <rdf:type> <owl:Nothing>
```

Variables in OWLIM: Termination

variables in antecedent of rule are **universally** quantified

free variables in consequent are interpreted **existentially** through the introduction of anonymous individuals (RDF: blank nodes)

potential effect of existential variables: forward chaining is not guaranteed to terminate

example: axiomatize that time is arbitrarily dense

```
i1 <rdf:type> Interval
----------------------
i2 <rdf:type> Interval
i2 <inside> i1            [Constraint i1 != i2]
```

this OWLIM rule will NOT lead to a finite deductive closure, i.e., closure computation will not terminate

Outlook: A Lot To Do

Decidability of Formalism

XML syntactic transport layer

RDF(S) basic relational language & simple ontological primitives

OWL DL decidable, but for many problems still to weak

further (rule) layers may/will extend OWL (e.g., SWRL)

BUT: will definitely be undecidable

how do we cope with this fact in practice?

- organize axioms in contexts that will not interact
- give up completeness of formalism (but not soundness!?)
- limit deductions only to a few steps

full answer to question is still missing (since early days of AI)

Inconsistency of Information

OWL (and other formalism) provide a monotonic framework

need mechanisms

- to retract outdated (entailed) information (AI: RMS)
- to cope with uncertainty, belief, trust, ...

only toy implementations and applications at the moment

Amount of Information

ontologies with $\approx$ 10K classes and $\approx$ 100K instances can be handled, e.g., can be checked for consistency

even larger ontologies can be queried

what to do with larger ontologies/vision of Semantic Web?

- deductive closure in forward chaining will become too large
- backward chaining inference will become too slow
- will probably need a mixture (partial materialization & tabling)
- give up logical completeness (!?)

Links & Books

Tom Gruber's article: `www.-ksl.stanford.edu/kst/what-is-an-ontology.html`

RDF & OWL recommendations of W3C: `www.w3.org/2004/01/sws-pressrelease`

Resource Description Framework: `www.w3.org/RDF/`

RDF Schema: `www.w3.org/TR/rdf-schema/`

OWL: `www.w3.org/2004/OWL/`

WordNet: `wordnet.princeton.edu/`

FrameNet: `framenet.icsi.berkeley.edu/`

SUMO & MILO ontology: `www.ontologyportal.org/`

PROTON ontology: `proton.semanticweb.org`

DOLCE ontology: `dolce.semanticweb.org`

Protégé: `protege.stanford.edu/`

OWLIM: `www.ontotext.com/owlim`

SWRL: `www.w3.org/Submission/SWRL/`

Ontology resources: `www-ksl.stanford.edu/kst/ontology-sources.html`

more resources: `protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary`

OWL-Time: `www.w3.org/TR/owl-time/`

FaCT: `www.cs.man.ac.uk/~horrocks/FaCT/`

RACER: `www.sts.tu-harburg.de/%7Er.f.moeller/racer/`

Pellet: `http://clarkparsia.com/pellet/`

Knowledge Interchange Format: `logic.stanford.edu/kif/`

Cyc: www.cyc.com/

Description Logics homepage: http://www.dl.kr.org/

W3C group Semantic Web: www.w3.org/2001/sw/

Web Ontology Working Group: www.w3.org/2001/sw/WebOnt/

T. Berners-Lee et al.: The Semantic Web, Scientific American.
www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

F. Baader et al.: Description Logic Handbook, Cambridge University Press; see also www.inf.unibz.it/~franconi/dl/course/.

P. Hayes: RDF Semantics, 2004 (http://www.w3.org/TR/rdf-mt/).

H. ter Horst: Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. ISWC 2005, 668–684.

J.W. Lloyd: Foundations of Logic Programming, Springer.

M. Huth & M. Ryan: Logic in Computer Science, Cambridge University Press.

M. Tarnowski: Mathematische Grundlagen der formalen Linguistik, IWBS Report 174, IBM.

B.H. Partee et al.: Mathematical Methods in Linguistics, Kluwer.

G. Smolka: Logische Programmierung.
`www.ps.uni-sb.de/courses/lp-course93.html`