TECHNISCHE UNIVERSITÄT DRESDEN

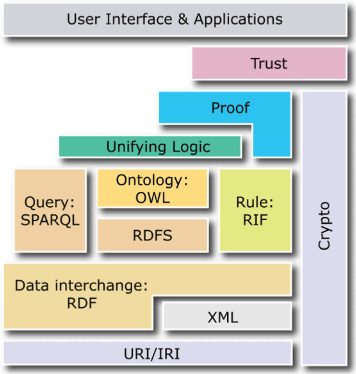
FOUNDATIONS OF SEMANTIC WEB TECHNOLOGIES

OWL 2 – Syntax and Semantics

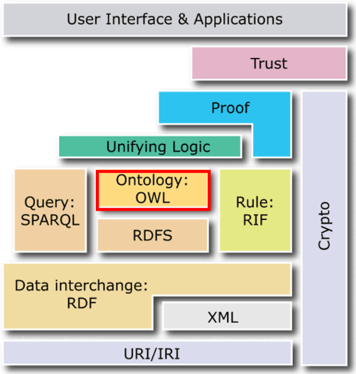
Sebastian Rudolph

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

OWL


User Interface & Applications
Trust
Proof
Unifying Logic
Query: SPARQL
Ontology: OWL
RDFS
Rule: RIF
Crypto
Data interchange: RDF
XML
URI/IRI

OWL


User Interface & Applications
Trust
Proof
Unifying Logic
Query:
SPARQL
Ontology:
OWL
RDFS
Rule:
RIF
Crypto
Data interchange:
RDF
XML
URI/IRI

Agenda

- Recap OWL & Overview OWL 2
- The Description Logic $\mathcal{SROIQ}$
- Inferencing with $\mathcal{SROIQ}$
- OWL 2 DL
- OWL 2 Profiles
- OWL 2 Full
- Summary

Agenda

- Recap OWL & Overview OWL 2
- The Description Logic $\mathcal{SROIQ}$
- Inferencing with $\mathcal{SROIQ}$
- OWL 2 DL
- OWL 2 Profiles
- OWL 2 Full
- Summary

Insufficiencies of OWL

OWL still too weak for certain tasks

Insufficiencies of OWL

OWL still too weak for certain tasks

- OWL insufficient as query language
  $\rightsquigarrow$ conjunctive queries, SPARQL for OWL

Insufficiencies of OWL

OWL still too weak for certain tasks

- OWL insufficient as query language
  $\rightsquigarrow$ conjunctive queries, SPARQL for OWL
- OWL insufficient as ontology language
  $\rightsquigarrow$ FOL-based rule extensions, SWRL & RIF

Insufficiencies of OWL

OWL still too weak for certain tasks

- OWL insufficient as query language
  $\rightsquigarrow$ conjunctive queries, SPARQL for OWL
- OWL insufficient as ontology language
  $\rightsquigarrow$ FOL-based rule extensions, SWRL & RIF

Should the OWL standard itself be extended?

Insufficiencies of OWL

OWL still too weak for certain tasks

- OWL insufficient as query language
  $\leadsto$ conjunctive queries, SPARQL for OWL
- OWL insufficient as ontology language
  $\leadsto$ FOL-based rule extensions, SWRL & RIF

Should the OWL standard itself be extended?
$\leadsto$ OWL 2

Development of OWL 2

OWL 2 as “next version” of OWL

extensions due to practical experiences with OWL 1.0:

- additional expressivity due to new ontological axioms
- extralogical extensions (syntax, metadata, . . .)
- revision of the OWL variants (Lite/DL/Full)

goals:

- most extensive compatibility with the existing OWL standard
- preservation of decidability of OWL DL
- correction of problems in the OWL 1.0 standard

Agenda

- Recap OWL & Overview OWL 2
- The Description Logic $\mathcal{SROIQ}$
- Inferencing with $\mathcal{SROIQ}$
- OWL 2 DL
- OWL 2 Profiles
- OWL 2 Full
- Summary

From $\mathcal{SHOIN}$ to $\mathcal{SROIQ}$

OWL DL based on DL $\mathcal{SHOIN}(D)$:

- axioms:
  - TBox: subclass relationships $C \sqsubseteq D$
  - RBox: subrole relationships $R \sqsubseteq S$ ($\mathcal{H}$), inverse roles R^- ($\mathcal{I}$), transitivity
  - ABox: class assertions $C(a)$, role assertions $R(a, b)$, equality $a \approx b$, inequality $a \not\approx b$
- class constructors:
  - conjunction $C \sqcap D$, disjunction $C \sqcup D$, negation $\neg C$ of classes
  - role restrictions: universal $\forall R.C$ and existential $\exists R.C$
  - number restrictions ($\mathcal{N}$): $\leq n\, R$ and $\geq n\, R$ (n non-negative integer)
  - nominals ($\mathcal{O}$): $\{a\}$
- datatypes (D)

OWL 2 extends this to $\mathcal{SROIQ}(D)$

TECHNISCHE UNIVERSITÄT DRESDEN

ABox

$\mathcal{SHOIN}$ supports different ABox assertions:

- class membership $C(a)$ (C complex class),
- special case: negated class membership $\neg C(a)$ (C complex class),
- equality $a \approx b$,
- inequality $a \not\approx b$
- role membership $R(a, b)$

ABox

$\mathcal{SHOIN}$ supports different ABox assertions:

- class membership $C(a)$ (C complex class),
- special case: negated class membership $\neg C(a)$ (C complex class),
- equality $a \approx b$,
- inequality $a \not\approx b$
- role membership $R(a, b)$
- negated role membership?

ABox

$\mathcal{SHOIN}$ supports different ABox assertions:

- class membership $C(a)$ (C complex class),
- special case: negated class membership $\neg C(a)$ (C complex class),
- equality $a \approx b$,
- inequality $a \not\approx b$
- role membership $R(a, b)$
- negated role membership?

$\rightsquigarrow$ $\mathcal{SROIQ}$ allows negated roles in der ABox: $\neg R(a, b)$

Number Restrictions

$\mathcal{SHOIN}$ supports only unqualified number restrictions ($\mathcal{N}$):

$$\text{Person} \sqcap \geq 3\, \text{hasChild}$$

"‘class of all persons with 3 or more children"’

Number Restrictions

$\mathcal{SHOIN}$ supports only unqualified number restrictions ($\mathcal{N}$):

$$\text{Person} \sqcap \geq 3\, \text{hasChild}$$

"‘class of all persons with 3 or more children"’ $\rightsquigarrow$ $\mathcal{SROIQ}$ also allows qualified number restrictions ($\mathcal{Q}$):

$$\text{Person} \sqcap \geq 3\, \text{hasChild}.(\text{Woman} \sqcap \text{Professor})$$

"‘class of all persons with 3 or more daughters who are professors"’

The Self “Concept”

modeling task: "‘Every human knows himself/herself."’

The Self "Concept"

modeling task: "'Every human knows himself/herself."'

- $\mathcal{SHOIN}$:

$$\text{knows}(\text{tom},\text{tom}) \quad \text{knows}(\text{tina},\text{tina}) \quad \text{knows}(\text{udo},\text{udo}) \quad \ldots$$

The Self "Concept"

modeling task: "'Every human knows himself/herself."'

- $\mathcal{SHOIN}$:

$$\mathsf{knows}(\mathsf{tom},\mathsf{tom}) \quad \mathsf{knows}(\mathsf{tina},\mathsf{tina}) \quad \mathsf{knows}(\mathsf{udo},\mathsf{udo}) \quad \ldots$$

  $\leadsto$ not generally applicable

- $\mathcal{SROIQ}$: specific notation Self

$$\mathsf{Human} \sqsubseteq \exists\mathsf{knows}.\mathsf{Self}$$

Role Axioms in $\mathcal{SHOIN}$

$\mathcal{SHOIN}$ provides few role axioms:

- Trans(r), `owl:TransitiveProperty`: r is transitive
  Example: Trans(locatedIn)

Role Axioms in $\mathcal{SHOIN}$

$\mathcal{SHOIN}$ provides few role axioms:

- $\mathsf{Trans}(r)$, `owl:TransitiveProperty`: r is transitive
  Example: $\mathsf{Trans}(\mathsf{locatedIn})$
- $\mathsf{Sym}(r)$, `owl:SymmetricProperty`: r is symmetric
  Example: $\mathsf{Sym}(\mathsf{relativeOf})$
  also: $r \sqsubseteq r^-$

Role Axioms in $\mathcal{SHOIN}$

$\mathcal{SHOIN}$ provides few role axioms:

- Trans(r), `owl:TransitiveProperty`: r is transitive
  Example: Trans(locatedIn)
- Sym(r), `owl:SymmetricProperty`: r is symmetric
  Example: Sym(relativeOf)
  also: $r \sqsubseteq r^-$
- Func(r), `owl:FunctionalProperty`: r is functional
  Example: Func(hasFather)
  also: $\top \sqsubseteq \leqslant 1 r$

TECHNISCHE UNIVERSITÄT DRESDEN

Role Axioms in $\mathcal{SHOIN}$

$\mathcal{SHOIN}$ provides few role axioms:

- Trans(r), `owl:TransitiveProperty`: r is transitive
  Example: Trans(locatedIn)
- Sym(r), `owl:SymmetricProperty`: r is symmetric
  Example: Sym(relativeOf)
  also: $r \sqsubseteq r^-$
- Func(r), `owl:FunctionalProperty`: r is functional
  Example: Func(hasFather)
  also: $\top \sqsubseteq \leqslant 1 r$
- InvFunc(r), `owl:InverseFunctionalProperty`: r is inverse functional
  Example: InvFunc(isFatherOf)
  also $\top \sqsubseteq \leqslant 1 r^-$ or Func(r^-)

Role Axioms in $\mathcal{SROIQ}$

$\mathcal{SROIQ}$ provides additional statements about roles:

- Ref(r), `owl:ReflexiveProperty`: r is reflexive, $(x, x) \in r^{\mathcal{I}}$ for all domain individuals x
  Example: Ref(knows)

Role Axioms in $\mathcal{SROIQ}$

$\mathcal{SROIQ}$ provides additional statements about roles:

- Ref(r), `owl:ReflexiveProperty`: r is reflexive, $(x, x) \in r^{\mathcal{I}}$ for all domain individuals x
  Example: Ref(knows)
- Irr(r), `owl:IrreflexiveProperty`: r is irreflexive, $(x, x) \notin r^{\mathcal{I}}$ for all domain individuals x
  Example: Irr(hasChild)

Role Axioms in $\mathcal{SROIQ}$

$\mathcal{SROIQ}$ provides additional statements about roles:

- Ref(r), `owl:ReflexiveProperty`: r is reflexive, $(x, x) \in r^{\mathcal{I}}$ for all domain individuals x
  Example: Ref(knows)
- Irr(r), `owl:IrreflexiveProperty`: r is irreflexive, $(x, x) \notin r^{\mathcal{I}}$ for all domain individuals x
  Example: Irr(hasChild)
- Asym(r), `owl:AsymmtericProperty`: r is asymmetric, $(x, y) \in r^{\mathcal{I}}$ implies $(y, x) \notin r^{\mathcal{I}}$
  Example: Asym(hasChild)

Role Axioms in $\mathcal{SROIQ}$

$\mathcal{SROIQ}$ provides additional statements about roles:

- Ref(r), `owl:ReflexiveProperty`: r is reflexive, $(x, x) \in r^{\mathcal{I}}$ for all domain individuals x
  Example: Ref(knows)
- Irr(r), `owl:IrreflexiveProperty`: r is irreflexive, $(x, x) \notin r^{\mathcal{I}}$ for all domain individuals x
  Example: Irr(hasChild)
- Asym(r), `owl:AsymmtericProperty`: r is asymmetric, $(x, y) \in r^{\mathcal{I}}$ implies $(y, x) \notin r^{\mathcal{I}}$
  Example: Asym(hasChild)
- Dis(r, s), `owl:propertyDisjointWith`, , `owl:AllDisjointProperties`: r and s are disjoint, $(x, y) \notin r^{\mathcal{I}} \cap s^{\mathcal{I}}$ for all x, y
  Example: Dis(hasFather, hasSon)

The Universal Role

$\mathcal{SROIQ}$ provides the universal role:

- universal role U (`owl:TopObjectProperty`):
  $(x, y) \in U^{\mathcal{I}}$ for all x, y

Example

$\top \sqsubseteq \leqslant 7\,000\,000\,000\, U.$Human
(not recommended!)

- ⇝ U is mainly comfortable as a counterpart for $\top$, e.g. as root element in a graphically displayed role hierarchy
- the converse `owl:BottomObjectProperty` has been introduced in OWL, but has no corresponding syntactic element in DLs
- for datatype properties analog `owl:TopDataProperty` and `owl:BottomDataProperty`

Complex Role Inclusion

"‘The friends of my friends are my friends."’

$\leadsto$ can be expressed in $\mathcal{SHOIN}$:
hasFriend is transitive

"‘The enemies of my friends are my enemies."’

$\leadsto$ Cannot be expressed in $\mathcal{SHOIN}$!

Complex Role Inclusion

"‘The friends of my friends are my friends."’

$\rightsquigarrow$ can be expressed in $\mathcal{SHOIN}$:
hasFriend is transitive

"‘The enemies of my friends are my enemies."’

$\rightsquigarrow$ Cannot be expressed in $\mathcal{SHOIN}$!

complex role inclusion

- RBox-expressions of the form $r_1 \circ r_2 \circ \ldots \circ r_n \sqsubseteq s$
- semantics: $(x_0, x_1) \in r_1^\mathcal{I}$, $(x_1, x_2) \in r_2^\mathcal{I}$, $\ldots$, $(x_{n-1}, x_n) \in r_n^\mathcal{I}$,
  implies $(x_0, x_n) \in s^\mathcal{I}$

Complex Role Inclusions – Example

Example

hasFriend $\circ$ hasEnemy $\sqsubseteq$ hasEnemy:
if $(x, y) \in$ hasFriend$^{\mathcal{I}}$ and $(y, z) \in$ hasEnemy$^{\mathcal{I}}$,
then also holds $(x, z) \in$ hasEnemy$^{\mathcal{I}}$

further examples

partOf $\circ$ belongsTo $\sqsubseteq$ belongsTo

hasBrother $\circ$ hasChild $\sqsubseteq$ isUncleOf

Expressivity of Complex Role Inclusions

How complicated are complex role inclusions?

RBoxes allow for encoding formal languages:

grammar for language of words `ab`, `aabb`, `aaabbb`, . . . :

```
L ::= ab
L ::= aLb
```

becomes the following RBox

$$\begin{aligned} r_a \circ r_b &\sqsubseteq \ell \\ r_a \circ \ell \circ r_b &\sqsubseteq \ell \end{aligned}$$

In fact, this way, all context-free languages can be encoded. This even enables us to encode the emptiness problem for intersection of two context-free languages into KB satisfiability.

$\leadsto$ OWL with (unrestricted) role inclusions is undecidable.

Regular RBoxes

Can complex role inclusion be restricted in order to retain decidability?

- RBoxes correspond to grammars for context-free languages
- intersection of these problematic

$\leadsto$ restriction to regular languages!

Regularity Conditions for RIAs

in order to guarantee decidability of inferenceing, the set of role inclusions has to be regular

- there has to be a strict linear order $\prec$ over the roles such that every RIA has one of the following forms (with $s_i \prec r$ for all $1 \leq i \leq n$):

  - $r \circ r \sqsubseteq r$
  - $r^{-} \sqsubseteq r$
  - $s_1 \circ s_2 \circ \ldots \circ s_n \sqsubseteq r$
  - $r \circ s_1 \circ s_2 \circ \ldots \circ s_n \sqsubseteq r$
  - $s_1 \circ s_2 \circ \ldots \circ s_n \circ r \sqsubseteq r$

Regularity Conditions for RIAs

- Example 1: $r \circ s \sqsubseteq r \quad s \circ s \sqsubseteq s \quad r \circ s \circ r \sqsubseteq t$

Regularity Conditions for RIAs

- Example 1: $r \circ s \sqsubseteq r \quad s \circ s \sqsubseteq s \quad r \circ s \circ r \sqsubseteq t$

  $\rightsquigarrow$ regular with order $s \prec r \prec t$
- Example 2: $r \circ t \circ s \sqsubseteq t$

Regularity Conditions for RIAs

- Example 1: $r \circ s \sqsubseteq r \quad s \circ s \sqsubseteq s \quad r \circ s \circ r \sqsubseteq t$
  - $\rightsquigarrow$ regular with order $s \prec r \prec t$
- Example 2: $r \circ t \circ s \sqsubseteq t$
  - $\rightsquigarrow$ not regular, form not allowed
- Example 3: $r \circ s \sqsubseteq s \quad s \circ r \sqsubseteq r$

Regularity Conditions for RIAs

- Example 1: $r \circ s \sqsubseteq r \quad s \circ s \sqsubseteq s \quad r \circ s \circ r \sqsubseteq t$
  - $\rightsquigarrow$ regular with order $s \prec r \prec t$
- Example 2: $r \circ t \circ s \sqsubseteq t$
  - $\rightsquigarrow$ not regular, form not allowed
- Example 3: $r \circ s \sqsubseteq s \quad s \circ r \sqsubseteq r$
  - $\rightsquigarrow$ not regular, since no appropriate order exists

Revisiting the Definition of Simple Roles

- simple roles in $\mathcal{SHOIN}$ = roles without transitive subroles
- in $\mathcal{SROIQ}$ we need to take RIAs into account

Revisiting the Definition of Simple Roles

simple roles are all roles. . .

- that do not occur on the rhs of a role inclusion,
- that are inverses of other simple roles,
- that occur only on the rhs of RIAs where the lhs consists of a length-one chain with a simple role.

(Caution: inductive definition)
$\rightsquigarrow$ non-simple are roles that can be derived from a chain of roles with length at least 2

Revisiting the Definition of Simple Roles

simple roles are all roles...

- that do not occur on the rhs of a role inclusion,
- that are inverses of other simple roles,
- that occur only on the rhs of RIAs where the lhs consists of a length-one chain with a simple role.

(Caution: inductive definition)
$\rightsquigarrow$ non-simple are roles that can be derived from a chain of roles with length at least 2

Expressions $\leq n\, r.C$, $\geq n\, r.C$, $\mathsf{Irr}(r)$, $\mathsf{Dis}(r,s)$, $\exists r.\mathsf{Self}$, $\neg r(a,b)$
are only allowed for simple roles r and s!
(Reason: ensure decidability)

Overview $\mathcal{SROIQ}$ – TBoxes

class expressions	
class names	A, B
conjunction	$C \sqcap D$
disjunction	$C \sqcup D$
negation	$\neg C$
existential role restriction	$\exists r.C$
universal role restriciton	$\forall r.C$
Self	$\exists s.\text{Self}$
atleast restriction	$\geqslant n\, s.C$
atmost restriction	$\leqslant n\, s.C$
nominals	$\{a\}$

TBox (class axioms)	
inclusion	$C \sqsubseteq D$
equivalence	$C \equiv D$

Overview $\mathcal{SROIQ}$ – RBoxes & ABoxes

Roles

roles	r, s, t
simple roles	s, t
universal role	u

ABox (assertions)

class membership	$C(a)$
role membership	$r(a, b)$
neg. role membership	$\neg s(a, b)$
equality	$a \approx b$
inequality	$a \not\approx b$

RBox (role axioms)

inclusion	$r_1 \sqsubseteq r_2$
complex role inclusion	$r_1 \circ \ldots \circ r_n \sqsubseteq r$
transitivity	$\mathsf{Trans}(r)$
symmetry	$\mathsf{Sym}(r)$
reflexivity	$\mathsf{Ref}(r)$
irreflexivity	$\mathsf{Irr}(s)$
disjointness	$\mathsf{Dis}(s, t)$

Agenda

- Recap OWL & Overview OWL 2
- The Description Logic $\mathcal{SROIQ}$
- Inferencing with $\mathcal{SROIQ}$
- OWL 2 DL
- OWL 2 Profiles
- OWL 2 Full
- Summary

How complicated is $\mathcal{SROIQ}$?

recap: $\mathcal{SHOIN}$ (OWL DL) is very complex (NExpTime)

How complicated is $\mathcal{SROIQ}$?

recap: $\mathcal{SHOIN}$ (OWL DL) is very complex (NExpTime)
observation: some modeling features are not really necessary (“syntactic sugar”)

- Trans(r) can be expressed as $r \circ r \sqsubseteq r$
- Sym(r) can be expressed as $r^- \sqsubseteq r$
- Asym(r) can be expressed as Dis(r, r^-)
- Irr(s) can be expressed as $\top \sqsubseteq \neg\exists S.$Self
- ABox can be represented by TBox axioms with nominals, e.g. $r(a, b)$ becomes $\{a\} \sqsubseteq \exists r.\{b\}$

qualifizierte number restrictions do not cause problems (known and implemented before)

$\rightsquigarrow$ main problem: role axioms (RBox)

Role Inclusions , Languages, Automata

How to deal with RBoxes?

- RBox inclusions resemble formal grammars
- every role r defines a regular language:
  the language of role chains from which it follows
- regular languages $\equiv$ regular Expressions $\equiv$ finite automata

$\rightsquigarrow$ approach: tableau methods are extended by "‘RBox automata"’

Decidability of $\mathcal{SROIQ}$

tableau method for $\mathcal{SROIQ}$ shows decidability

- algorithm has a good adaptation behaviour: modeling features that are not used do hardly impede computation (“pay as you go”)
- tableau method not useful for complexity considerations
- $\mathcal{SROIQ}$ 2-NExpTime-complete
  - $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are Harder than $\mathcal{SHOIQ}$. Yevgeny Kazakov. In Gerhard Brewka and Jérôme Lang, editors, KR 2008. Pages 274-284. AAAI Press. 2008
  - lower bound: encoding of a 2Exp tiling problem
  - upper bound: exponential translation into the 2-variable fragment of FOL with counting quantifiers, $\mathcal{C}_2$, for which satisfiability checking is known to be NExpTime-complete)

Agenda

- Recap OWL & Overview OWL 2
- The Description Logic $\mathcal{SROIQ}$
- Inferencing with $\mathcal{SROIQ}$
- OWL 2 DL
- OWL 2 Profiles
- OWL 2 Full
- Summary

OWL 2 DL: Further Aspects

$\mathcal{SROIQ}$ is “only” logical foundation of OWL 2 DL

further non-logical aspects:

- Syntax (extension necessary)
- datatype declarations and datatype functions, new datatypes?
- metamodeling: “punning”
- comments and ontological metadata
- invers-functional conkrete roles (datatype properties): Keys?
- mechanisms for ontology import?
- ...

$\leadsto$ diverse smaller changes

Metamodeling

Metamodeling

specification of ontological knowledge about elements of the ontology (including classes, roles, axioms).

Examples:

- "The class Person was created on the 30.1.2008 by bglimm."
- "For the class City, we recommend the property numberOfCitizens."
- "The statement ‚Dresden was founded in 1206' was extracted automatically with a confidence of 85%."

(Compare Reification in RDF Schema)

Punning in OWL

Metamodeling in expressive logics is dangerous and expensive!

OWL 2 currently supports the simples form of metamodeling:

Punning

- the names for classes, roles, individuals do not have to be disjoint
- no logical relationship between class, individual and role of the same name
- only relevant for pragmatic interpretation

Example:

$$\text{Person}(\text{Birte}) \quad \text{classCreatedBy}(\text{Person}, \text{bglimm})$$

TECHNISCHE UNIVERSITÄT DRESDEN

Comments and Metadata

punning supports simple metadata with (weak) semantic meaning

How can one make purely syntactic comments in an ontology?

- comments in XML files: `<!-- comment -->`

Comments and Metadata

punning supports simple metadata with (weak) semantic meaning

How can one make purely syntactic comments in an ontology?

- comments in XML files: `<!-- comment -->`
  $\leadsto$ no relation to the OWL axioms in this file
- non-logical annotations in OWL 2: `owl:AnnotationProperty`

Comments and Metadata

punning supports simple metadata with (weak) semantic meaning

How can one make purely syntactic comments in an ontology?

- comments in XML files: `<!-- comment -->`
  ⇝ no relation to the OWL axioms in this file
- non-logical annotations in OWL 2: `owl:AnnotationProperty`
  ⇝ attached to (semantic) ontological element

Syntactic Aspects

new/extended syntaxes:

- RDF/XML: extension by OWL 2 elements
- functional-style syntax: replaces “abstract syntax” in OWL 1
- OWL/XML: syntax for simpler processing in XML tools
- Turtle: RDF triple syntax
- Manchester syntax: syntax that is easier to read for humans

Quo vadis, OWL Lite?

Quo vadis, OWL Lite?

OWL Lite as a Failure:

- almost as complex as OWL DL
- complicated syntax that does not provide direct access to actual modeling power
- use in ontologies today only "‘by accident"’, not deliberately

original goal:
capture the part of OWL that is easy and efficiently implementable

Quo vadis, OWL Lite?

OWL Lite as a Failure:

- almost as complex as OWL DL
- complicated syntax that does not provide direct access to actual modeling power
- use in ontologies today only "‘by accident"’, not deliberately

original goal:
capture the part of OWL that is easy and efficiently implementable

$\rightsquigarrow$ OWL 2 Profiles

Agenda

- Recap OWL & Overview OWL 2
- The Description Logic $\mathcal{SROIQ}$
- Inferencing with $\mathcal{SROIQ}$
- OWL 2 DL
- OWL 2 Profiles
- OWL 2 Full
- Summary

OWL 2 Profiles

OWL 2 defines three fragments where automated inferencing can be done in PTime

- OWL EL
  - computation of the class hierarchy (all subclass relationships) in PTime

OWL 2 Profiles

OWL 2 defines three fragments where automated inferencing can be done in PTime

- OWL EL
  - computation of the class hierarchy (all subclass relationships) in PTime
- OWL QL
  - conjunctive queries in AC_0 (data complexity) $\rightsquigarrow$ reducible to SQL

OWL 2 Profiles

OWL 2 defines three fragments where automated inferencing can be done in PTime

- OWL EL
  - computation of the class hierarchy (all subclass relationships) in PTime
- OWL QL
  - conjunctive queries in AC_0 (data complexity) $\rightsquigarrow$ reducible to SQL
- OWL RL
  - can be used as an extension of RDFS or as a fragment of OWL DL (OWL Direct Semantics)
  - complexity PTime

OWL 2 EL

- An (almost maximal) fragment of OWL 2 such that
  - satisfiability can be checked in PTime (PTime-complete)
  - data complexity for ABox queries also PTime-complete
- class hierarchy (all subsumption relationships between atomic classes) can be computed in one pass
- uses a saturation method that was developed for the description logic $\mathcal{EL}$

OWL 2 EL

- allowed:
  - subclass axioms with conjunction, existential restriction, $\top$, $\bot$, singleton nominals
  - complex RIAs, range restrictions (under certain conditions)
- not allowed:
  - negation, disjunction, universal restrictions, inverse roles

OWL 2 QL

- an (almost maximal) fragment of OWL 2 such that
  - data complexity of conjunctive query answering is in AC^0
- queries can be rewritten such that no terminological knowledge has to be taken into account
  - $\Rightarrow$ standard RDBMS can be used for storage and querying

OWL 2 QL

- allowed:
  - simple role hierarchies, domain & range axioms
  - subclass axioms with
    - lhs: class name or existential restriction with $\top$
    - rhs: conjunction of class names, existential restriction and negation of lhs expressions
- supports RDFS with “well-formed” graphs

OWL 2 RL

- An (almost maximal) fragment of OWL 2 such that
  - automated inferencing is PTime-complete (consistency, satisfiability of classes, subsumption, class membership checks)
  - automated inferencing is correct (sound & complete) if the given RDF graph satisfies certain requirements
  - otherwise the automated reasoning may be be sound but incomplete.
- can operate directly on RDF triples in order to enrich instance data (materialization, forward chaining for facts)
- automated inferencing can be implemented via a set of rules (using a rule engine that supports equality)

Agenda

- Recap OWL & Overview OWL 2
- The Description Logic $\mathcal{SROIQ}$
- Inferencing with $\mathcal{SROIQ}$
- OWL 2 DL
- OWL 2 Profiles
- OWL 2 Full
- Summary

What to do with OWL Full?

Goal of OWL 2 DL: make many OWL Full 1.0 ontologies interpretable as OWL DL (cf. e.g. punning)

What to do with OWL Full?

Goal of OWL 2 DL: make many OWL Full 1.0 ontologies interpretable as OWL DL (cf. e.g. punning)

- extension of OWL Full by OWL 2 features is required by a few practitioners
- allows to work on all kinds of RDF graphs
- despite undecidability: many FOL verification tools do not guarantee termination and are still useful
- alternative implementation techniques can be used, which might be faster (but do not guarantee termination)

Crucial Differences in the Semantics

- annotations do not have a semantics in the direct semantics (which is used for OWL DL), but they do in the RDF-based semantics (which is used for OWL Full)
- import commands are only parser commands in the direct semantics, but do have a presence as triple in the RDF-based Semantics
- in the RDF-based semantics, classes are individuals, that are endowed with an extension $\rightsquigarrow$ semantic conditions are only applicable to those classes that have an individual representant

Crucial Differences in the Semantics

Example

- C(a)
- query for all instances of the class $C \sqcup D$

Crucial Differences in the Semantics

Example

- C(a)
- query for all instances of the class $C \sqcup D$
- RDF-based semantics: $\emptyset$, direct semantics: a

Crucial Differences in the Semantics

Example

- $C(a)$
- query for all instances of the class $C \sqcup D$
- RDF-based semantics: $\emptyset$, direct semantics: a

$\rightsquigarrow$ under the RDF-based semantics, we only have the guarantee that the union of the extensions of C and D do exist as subsets of the domain, however it is not ensured that an element exists which has this set as extension.

$\rightsquigarrow$ contrarily, in the direct semantics class names "directly" represent sets and not domain elements

$\rightsquigarrow$ the answer coincides for both semantics after adding $E \equiv C \sqcup D$

TECHNISCHE UNIVERSITÄT DRESDEN

Agenda

- Recap OWL & Overview OWL 2
- The Description Logic $\mathcal{SROIQ}$
- Inferencing with $\mathcal{SROIQ}$
- OWL 2 DL
- OWL 2 Profiles
- OWL 2 Full
- Summary

Summary

OWL 2 as first extension of the OWL standard

- standardized 27.10.2009
- logical extension based on description logic $\mathcal{SROIQ}$
- new modeling features, most notably complex RIAs, qualified number restrictions
- non-logical extensions: punning, comments, datatypes, etc.
- profiles with polynomial reasoning procedures