Unit 5: OWL2 (in a nutshell) & OWL Reasoning for Linked Data

Overview

- What’s new in OWL2 (2009, edited REC 2012)?
- OWL for Linked Data (summary of own works on over the past view years, with various co-authors, e.g. **Aidan Hogan, Jürgen Umbrich, Stefan Bischof, Andreas Harth, Birte Glimm, Markus Krötzsch**, etc.)

SIEMENS

Why OWL1 is Not Enough

Too expensive to reason with

- High complexity: NEXPTIME-complete
- Some ontologies only use some limited expressive power; e.g. The SNOMED (Systematised Nomenclature of Medicine) ontology

Not expressive enough; e.g.

- No user defined datatypes
- No metamodeling support
- Limited support for describing properties

From OWL1 to OWL2

OWL2: A new version of OWL in 2009

Main goals:

1. To define "profiles" of OWL that are:

- smaller, easier to implement and deploy
- cover important application areas and are easily understandable to non-expert users

2. To add a few extensions to current OWL that are useful, and is known to be implementable

- many things happened in research since 2004

SIEMENS

New expressive power

- user defined datatypes, e.g.:

```
:personAge owl:equivalentClass _:x .
_:x rdf:type rdfs:Datatype .
_:x owl:onDatatype xsd:integer .
_:x owl:withRestrictions (_:y1 _:y2) .
_:y1 xsd:minInclusive "0"^^xsd:integer .
_:y2 xsd:maxInclusive "150"^^xsd:integer.
```

- punning (metamodeling), e.g.:

```
:John rdf:type :Father .
:Father rdf:type :SocialRole .
```

New Expressiveness in OWL 2

- qualified cardinality restrictions, e.g.:

```
_:x rdf:type owl:Restriction .
_:x owl:onProperty foaf:knows .
_:x owl:minQualifiedCardinality "2"^^xsd:nonNegativeInteger .
_:x owl:onClass Scottish .
```

- property chain inclusion axioms, e.g.:

```
foaf:nick owl:propertyChainAxiom (foaf:holdsAccount sioc:name).
```

- local reflexivity restrictions, e.g.:

```
_:x rdf:type owl:Restriction .
_:x owl:onProperty :like .
_:x owl:hasSelf "true"^^xsd:boolean
```

[for narcissists]

- reflexive, irreflexive, symmetric, and antisymmetric properties, e.g.:

```
foaf:knows rdf:type owl:ReflexiveProperty .
rel:childOf rdf:type owl:IrreflexiveProperty .
```

- disjoint properties, e.g.:

```
rel:childOf owl:propertyDisjointWith rel:parentOf .
```

- keys, e.g.:

```
foaf:OnlineAccount owl:hasKey
      (foaf:accountName foaf:accountServiceHomepage) .
```

New Expressiveness in OWL 2

Syntactic sugar (make things easier to say)

- Disjoint unions, e.g.:

```
child owl:disjointUnionOf (boy girl) .
```

- Disjoint classes, e.g.:

```
_:x rdf:type owl:AllDisjointClasses .
_:x owl:members (boy girl) .
```

- Negative assertions, e.g.:

```
_:x rdf:type owl:NegativePropertyAssertion .
_:x owl:sourceIndividual :John .
_:x owl:assertionProperty foaf:knows .
_:x owl:targetIndividual :Mary .
```

SIEMENS

OWL 2 and Description Logics (DL)

$\mathcal{R}$ often used for $\mathcal{ALC}$ extended with property chain inclusion axioms

- following the notion introduced in $\mathcal{RIQ}$ [Horrocks and Sattler, 2003]
- including transitive property axioms

Additional letters indicate other extensions, e.g.:

- $\mathcal{S}$ for property characteristics (e.g., reflexive and symmetric)
- $\mathcal{O}$ for **nominals**/singleton classes
- $\mathcal{I}$ for inverse roles
- $\mathcal{Q}$ for qualified number restrictions

property characteristics ($\mathcal{S}$) + $\mathcal{R}$ + nominals ($\mathcal{O}$) + inverse ($\mathcal{I}$) + qualified number restrictions($\mathcal{Q}$) = $\mathcal{SROIQ}$

$\mathcal{SROIQ}$ [Horrocks et al., 2006] is the basis for OWL 2 DL

OWL 2 Profiles and Reasoning Services

SIEMENS

Rationale:

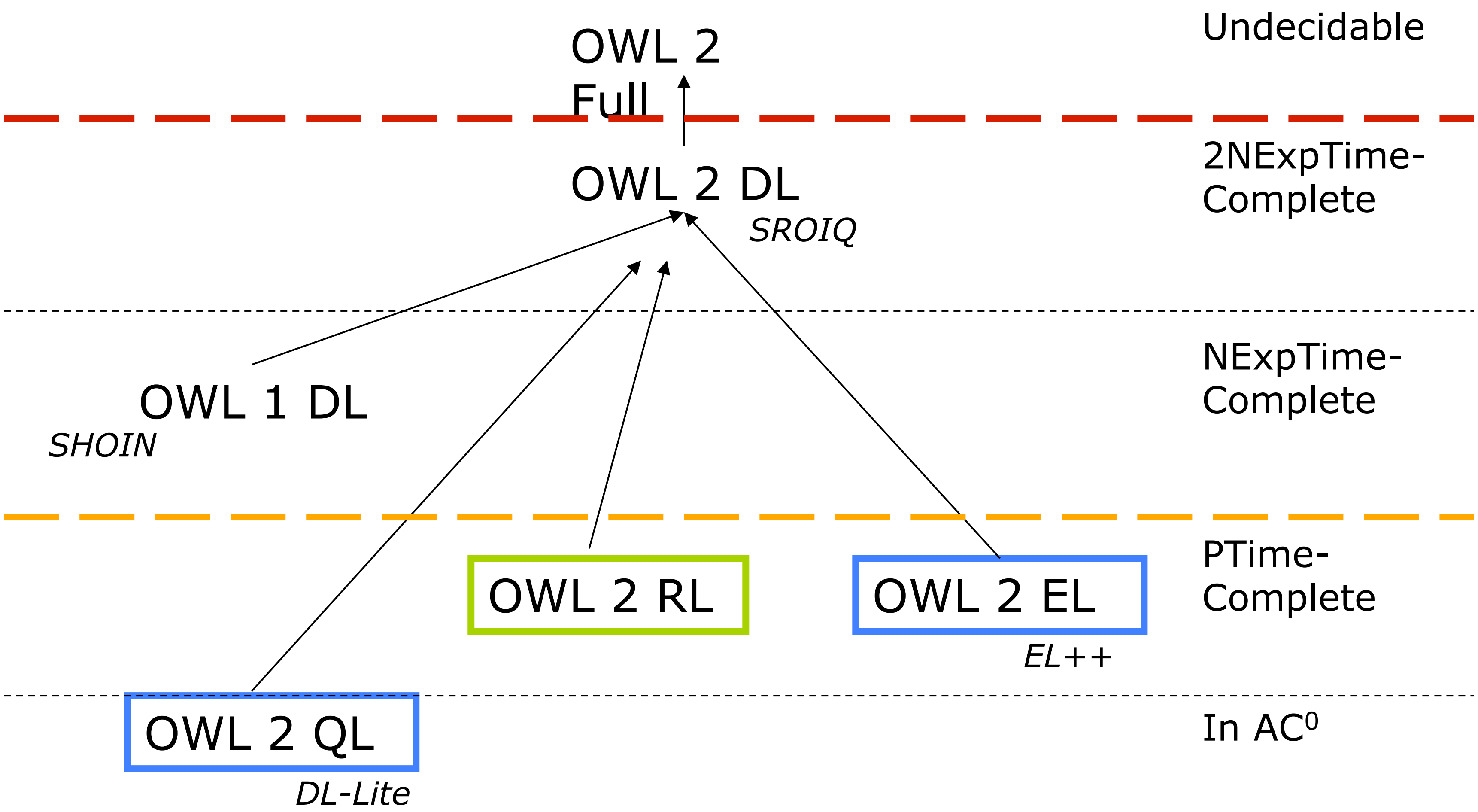
- Tractable
- Tailored to specific reasoning services

Popular reasoning services

- TBox reasoning: OWL 2 EL
- ABox reasoning: OWL 2 RL
- Query answering: OWL 2 QL

Specification: http://www.w3.org/TR/owl2-profiles/

The family tree



OWL 2 QL

SIEMENS

A (near maximal) fragment of OWL 2 such that

- Data complexity of conjunctive query answering in **AC^0**

Based on **DL-Lite** family of description logics [Calvanese et al. 2005; 2006; 2008]

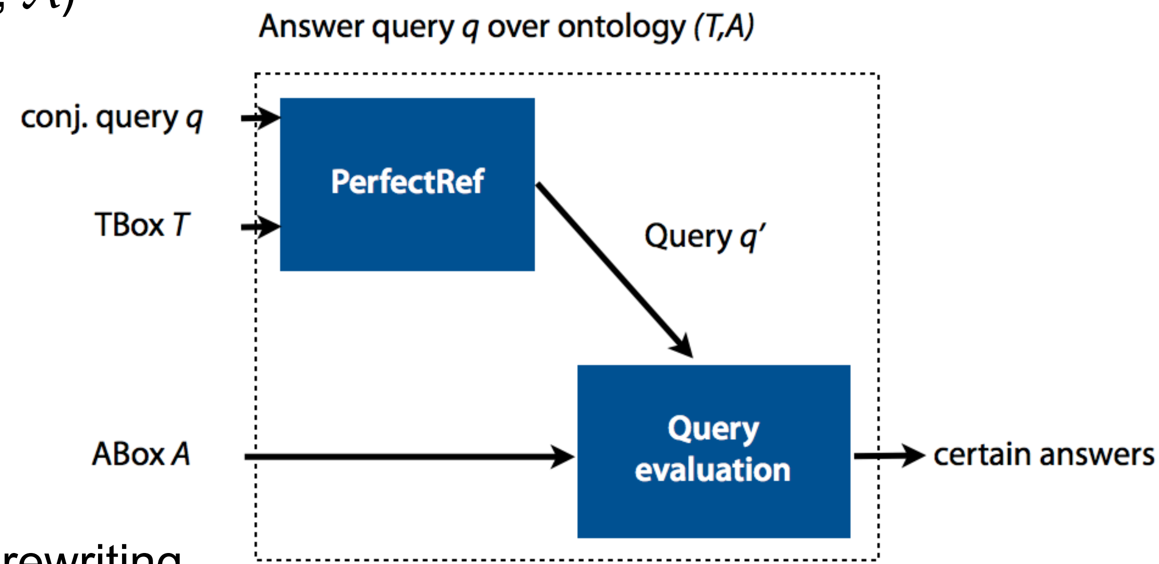
Can exploit **query rewriting** based reasoning technique

- Computationally optimal
- Data storage and query evaluation can be delegated to standard RDBMS
- Novel technique to prevent exponential blowup produced by rewritings [Kontchakov et al. 2010, Rosati and Almatelli 2010]
- Can be extended to more expressive languages (beyond AC^0) by delegating query answering to a Datalog engine [Perez-Urbina et al. 2009]

Query Rewriting Technique (basics)

SIEMENS

Given ontology $\mathcal{O}$ (TBox) and a conjunctive query q use $\mathcal{O}$ to rewrite q to q' $\mathcal{Q}'$ s.t., for any set of ground facts $\mathcal{A}$:
ans(q, $\mathcal{O}$, $\mathcal{A}$) = ans(q', $\emptyset$, $\mathcal{A}$)



Resolution based query rewriting

- **Clausify** ontology axioms
- **Saturate** (clausified) ontology and query using resolution
- **Prune** redundant query clauses

Query Rewriting Technique (basics)

SIEMENS

Example:

```
:Doctor rdfs:subClassOf [ a owl:Restriction;
    owl:onProperty :treats ; owl:someValuesFrom :Patient ].

:Consultant rdfs:subClassOf :Doctor .
```

```
SELECT ?x  WHERE {
?x :treats ?y . ?y a :Patient. }
```

Query Rewriting Technique (basics)

SIEMENS

Example:

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$

$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$

Query Rewriting Technique (basics)

SIEMENS

Example (clausify):

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \land \mathsf{Patient}(y)$$

SIEMENS

Query Rewriting Technique (basics)

Example (saturate):

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$

Query Rewriting Technique (basics)

SIEMENS

Example (saturate):

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$

$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$

$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$

$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$

$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$

$$Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)$$

Query Rewriting Technique (basics)

SIEMENS

Example (saturate):

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$
$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$
$$Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x)$$

Query Rewriting Technique (basics)

SIEMENS

Example (saturate):

$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$

$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

$$\mathsf{treats}(x, f(x)) \leftarrow \mathsf{Doctor}(x)$$

$$\mathsf{Patient}(f(x)) \leftarrow \mathsf{Doctor}(x)$$

$$\mathsf{Doctor}(x) \leftarrow \mathsf{Consultant}(x)$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$

$$Q(x) \leftarrow \mathsf{Doctor}(x) \wedge \mathsf{Patient}(f(x))$$

$$Q(x) \leftarrow \mathsf{treats}(x, f(x)) \wedge \mathsf{Doctor}(x)$$

$$Q(x) \leftarrow \mathsf{Doctor}(x)$$

$$Q(x) \leftarrow \mathsf{Consultant}(x)$$

Query Rewriting Technique (basics)

SIEMENS

Example (prune):

$$\text{Doctor} \sqsubseteq \exists\text{treats}.\text{Patient}$$
$$\text{Consultant} \sqsubseteq \text{Doctor}$$

Fo

$$\text{treats}(x, f(x)) \leftarrow \text{Doctor}(x)$$
$$\text{Patient}(f(x)) \leftarrow \text{Doctor}(x)$$
$$\text{Doctor}(x) \leftarrow \text{Consultant}(x)$$

$$Q(x) \leftarrow \text{treats}(x, y) \wedge \text{Patient}(y)$$
~~$Q(x) \leftarrow \text{Doctor}(x) \wedge \text{Patient}(f(x))$~~
~~$Q(x) \leftarrow \text{treats}(x, f(x)) \wedge \text{Doctor}(x)$~~
$$Q(x) \leftarrow \text{Doctor}(x)$$
$$Q(x) \leftarrow \text{Consultant}(x)$$

$$Q(x) \leftarrow (\text{treats}(x, y) \wedge \text{Patient}(y)) \vee \text{Doctor}(x) \vee \text{Consultant}(x)$$

Query Rewriting Technique (basics)

SIEMENS

```
SELECT ?x  WHERE {
?x :treats ?y . ?y a :Patient. }
```

$$\Updownarrow$$

$$Q(x) \leftarrow \mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)$$

$$\Updownarrow$$

$$Q(x) \leftarrow (\mathsf{treats}(x, y) \wedge \mathsf{Patient}(y)) \vee \mathsf{Doctor}(x) \vee \mathsf{Consultant}(x)$$

$$\Updownarrow$$

```
SELECT ?x  WHERE {
{ ?x :treats ?y . ?y a :Patient. }
UNION
{ ?x a :Doctor }
UNION
{ ?x a :Consultant} }
```

OWL 2 EL

SIEMENS

A (near maximal) fragment of OWL 2 such that

- Satisfiability checking is in PTime (**PTime-Complete**)
- Data complexity of query answering also PTime-Complete

Based on $\mathcal{EL}$ family of description logics [Baader et al. 2005]

Can exploit **saturation** based reasoning techniques

- Computes complete classification in “one pass”
- Computationally optimal (PTime for EL)

Saturation-based Technique (basics)

SIEMENS

Normalise ontology axioms to standard form:

$$A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$$

Saturate using inference rules:

$$\frac{A \sqsubseteq B \quad B \sqsubseteq C}{A \sqsubseteq C} \qquad \frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

Extensions to Horn fragment of OWL DL [Kazakov 2009] requires (many) more rules

Saturation-based Technique (basics)

SIEMENS

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$
$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$
$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

Saturation-based Technique (basics)

SIEMENS

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$
$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$
$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

Saturation-based Technique (basics)

SIEMENS

Example:

$$\mathsf{OrganTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Organ}$$
$$\mathsf{HeartTransplant} \equiv \mathsf{Transplant} \sqcap \exists \mathsf{site}.\mathsf{Heart}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

$$\mathsf{OrganTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{OrganTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Organ}$$
$$\exists \mathsf{site}.\mathsf{Organ} \sqsubseteq \mathsf{SO}$$
$$\mathsf{Transplant} \sqcap \mathsf{SO} \sqsubseteq \mathsf{OrganTransplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{Transplant}$$
$$\mathsf{HeartTransplant} \sqsubseteq \exists \mathsf{site}.\mathsf{Heart}$$
$$\exists \mathsf{site}.\mathsf{Heart} \sqsubseteq \mathsf{SH}$$
$$\mathsf{Transplant} \sqcap \mathsf{SH} \sqsubseteq \mathsf{HeartTransplant}$$
$$\mathsf{Heart} \sqsubseteq \mathsf{Organ}$$

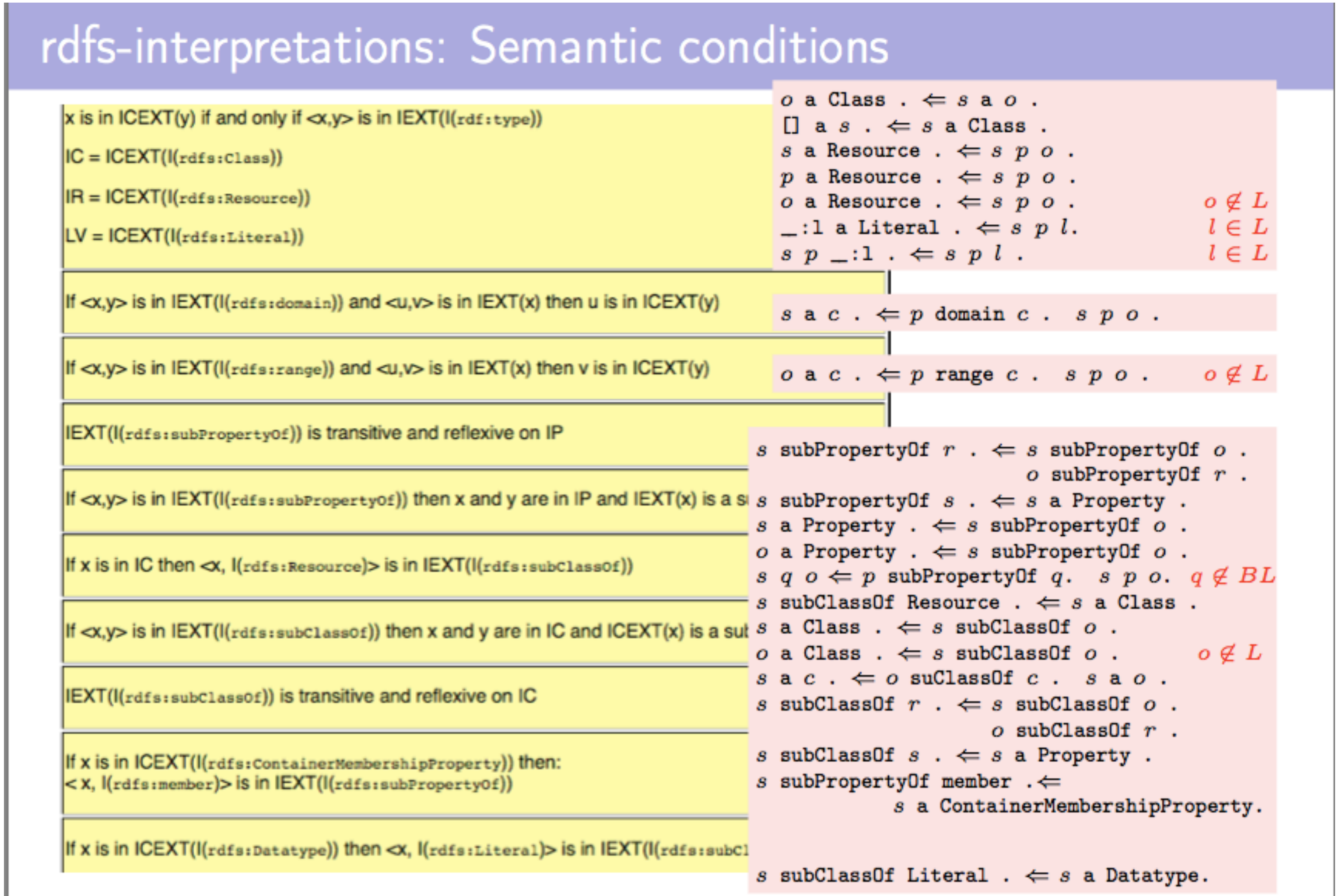
$$\mathsf{HeartTransplant} \sqsubseteq \mathsf{SO}$$

OWL 2 RL

SIEMENS

A (near maximal) fragment of OWL 2 such that

- ABox reasoning can be implemented using **forward-chaining rules**
- Expands on the idea of inference rules for RDFS (recall Unit 3):



SIEMENS

OWL 2 RL

A (near maximal) fragment of OWL 2 such that

- ABox reasoning can be implemented using **forward-chaining rules**
- Expands on the idea of inference rules for RDFS (recall Unit 3).

- Some additional rules for OWL 2:

```
{ ?O ?Q ?S . } :- { ?S ?P ?O .   ?Q owl:inverseOf ?P. }

{ ?O ?P ?S . } :- { ?S ?P ?O .   ?P a owl:SymmetricProperty. }
```

OWL 2 RL

Further examples:

e.g. inverseFunctionalProperty can also (partially) be expressed by Rules:

```
{ ?S1 owl:sameAs ?S2 } :-
           { ?S1 ?P ?O . ?S2 ?P ?O . ?P rdf:type owl:InverseFunctionalProperty }
```

also owl:sameAs

```
{ ?X owl:sameAs ?Z . } :- { ?X owl:sameAs ?Y . ?Y owl:sameAs ?Z . }
{ ?Y ?P ?O } :- { ?X owl:sameAs ?Y . ?X ?P ?O }
{ ?S ?Y ?O } :- { ?X owl:sameAs ?Y . ?S ?X ?O }
{ ?S ?P ?Y } :- { ?X owl:sameAs ?Y . ?S ?P ?X }
```

…

Details:
http://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules

Now can OWL(2) Reasoning be done on Linked Data?

Slides: OWLED 2013 keynote,

http://www.polleres.net/presentations//20130527OWLED2013_Invited_talk.pdf

Next time: Discuss assignment 3 & Q/A

Anything in particular you want repeated/answered?

Note:

1) OPTIONAL in Virtuoso sometimes causes troubles (please submit still what you have, we'll check that next time in detail)

2) Queries with FROM/FROM NAMED probably don't run on a remote endpoint, use ARQ for those.

Student presentations:

Who has sent me a topic suggestion already?

- SPARQL GUIs (F. J. Ekaputra)
- Good Relations Ontology and use (B.Ege)
- Web of Needs (A.Tus)
- SPARQL1.1 Property Paths (K. Bui, J. Petttersson)
- RDB2RDF **or** OWL/DL Modeling? (B.Doenz)
- SPARQL1.1 Entailment Regimes & Update (L. Agatic, V. Viljanic)
- SPARQL Benchmarking (E. Rut)
- OWL 2 and metamodeling (F. Leberl)
- Schema.org and RDFa (M.Suchi)
- A SW application based on Dbpedia (S. Belk)
- Using Semantics for resource allocation (D. Drenjanac)
- W3C LDP (O. Zhukova)
- XBRL & Linked Data (L. Madlberger)
- RDF/OWL wrapper for TISS (N. Frohner, S.Lindner)
- SKOS (B. Fazekas)

Anybody who has NOT suggested a topic yet?

Student presentations schedule

Stefan Belk	SW application using DBPEdia	24.6.2013 09:00
Olga Zhukova	W3C Linked Data Platform	24.6.2013 09:30
Simon Lindner, Nikolaus Frohner	OWL/RDF for TISS	24.6.2013 10:00
Botond Fazekas	SKOS	24.6.2013 10:30
Elias Rut	Berlin SPARQL Benchmark	24.6.2013 11:00
Leon Agatic, Vanda Viljanac	SPARQL1.1 Entailment Regimes and Update	24.6.2013 11:30
Boertecin Ege	Good Relations	24.6.2013 15:00
Franz Leberl	OWL2 Metamodeling	24.6.2013 15:30
Domagoj Drenjanac	Semantics for resource allocation	24.6.2013 16:00
Benjamin Doenz	RDB2RDF	24.6.2013 16:30
Ekaputra Fajar Juang	SPARQL GUIs	24.6.2013 17:00
Lisa Madlberger	XBRL & Linked Data	24.6.2013 17:30
Markus Suchi	Schema.org and RDFa	25.6.2013 09:00
Alan Tus	Web of Needs project	25.6.2013 09:30
Jonas Petterson, Kien Bui	SPARQL1.1 Property Paths	25.6.2013 10:00
Florian Wieser	??	25.6.2013 10:30?
Soroosh Mortezapoor	??	25.6.2013 11:00?

Some suggested topics (which we can assign now already – first come, first serve:

- W3C RDF1.1 WG – status semantics, changes, semantics for named graphs, etc.

More own topics suggestions welcome!

SIEMENS

Presentations

First slot: 24/06/2013
Second slot: 25/06/2013

Send me the slides at least 1 week in advance per email!
→ You should start to work on the topic soon!

Axel Polleres