

OWL

ANNA QUERALT, OSCAR ROMERO
(FACULTAT D'INFORMÀTICA DE BARCELONA)

OWL

W3C recommendation (2003)

Based on OIL and DAML

OWL allows to express certain Description Logics in the form of Knowledge Graphs

- Uses RDF as underlying representation (and therefore, OWL creates a knowledge graph)
- There were three languages in OWL 1.0 (each mapping to a different description logic):
 - Lite
 - DL
 - Full
- OWL 2.0 eliminates OWL Full and Lite and adds three profiles: RL, QL, EL (all of them mapping, again, to a specific description logic)

2 flavor of OWL

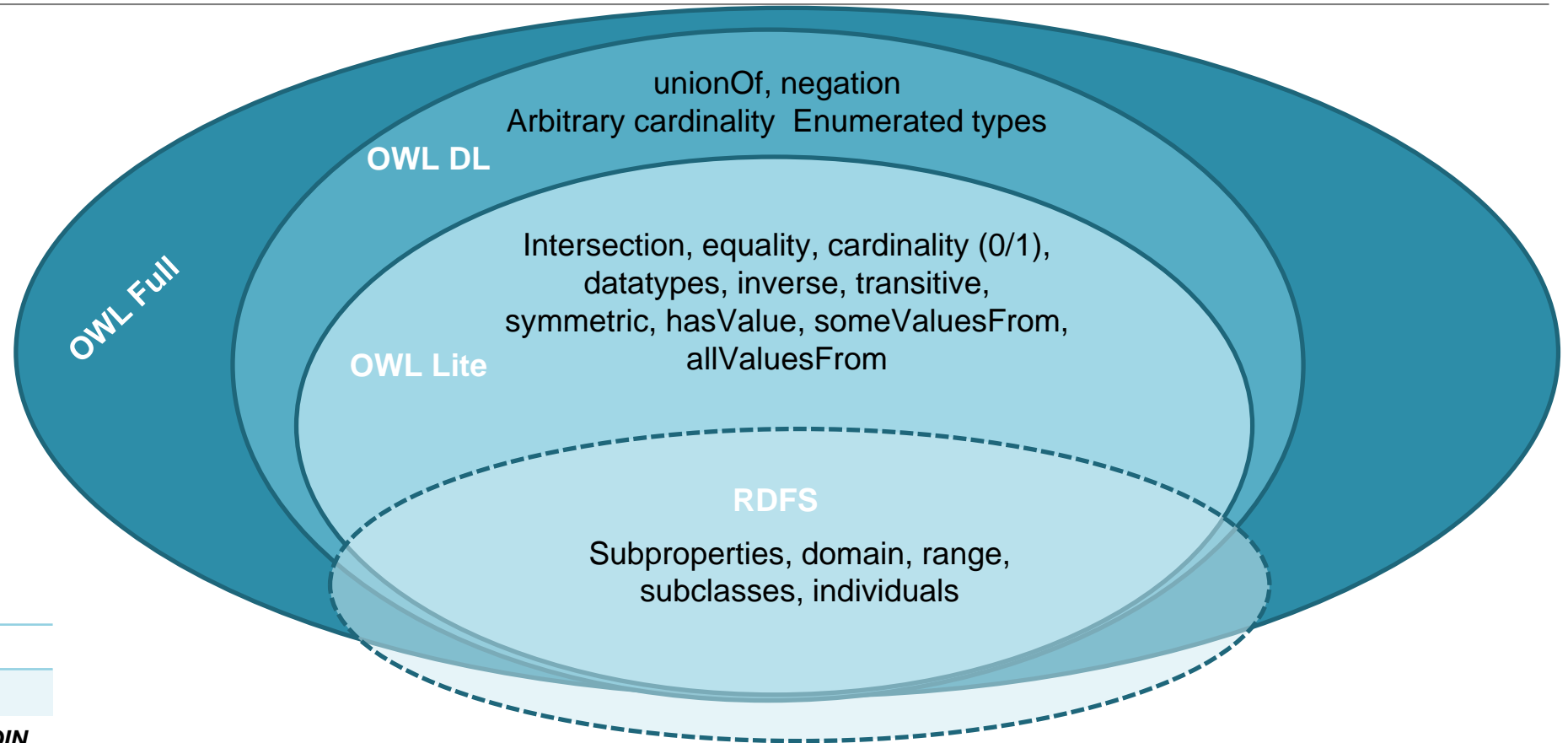
1. OWL 1

2. OWL 2

Thus, OWL is nothing else than syntactic sugar for Description Logics (!)

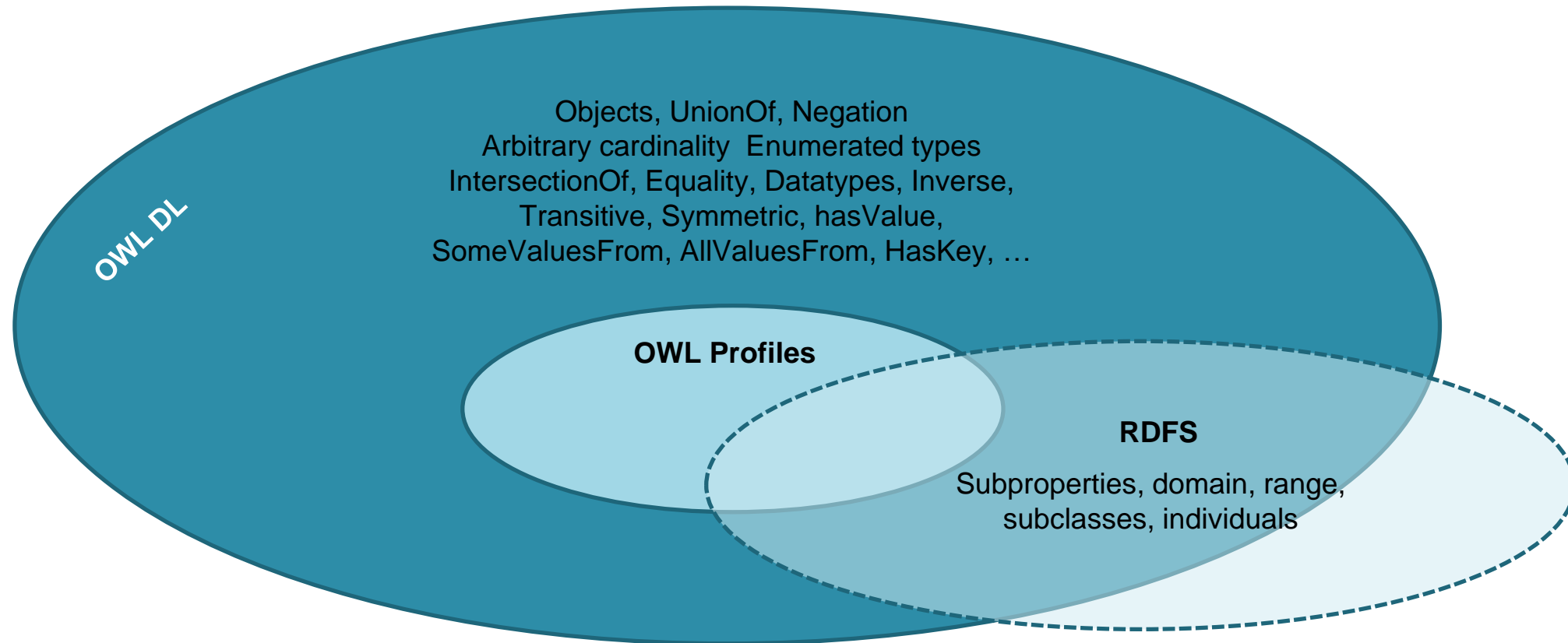
RDFS is not subset of OWL?
RDFS does not force metadata, metamodel, and instance
we can create metamodel of metamodel
but in owl, it is enforced, to have abox and tbox

OWL 1.0



Based on Sven Groppe's Data Management and Query Processing in Semantic Web Databases

OWL 2.0



<https://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/#Vocabulary>

OWL-DL *SROIQ*

Based on Sven Groppe's Data Management and Query Processing in Semantic Web Databases

OWL Versions

Criticism to OWL 1.0

- Complex reasoning over the ABOX (i.e., on data complexity)
- Integrity constraints are not allowed
- Limited support to data types

OWL 2.0

- Syntactic facilities
- *SROIQ* constructs
 - OWL 1.0 corresponds to *SHOIN*
- User defined data types
- Class attributes and primary keys

Important: OWL reuses `rdfs:subClassOf`, `rdfs:subPropertyOf` but it does not apply the rule-based inference we saw with RDFS but applies DL reasoning (!)

Syntax Example (Axioms)

OWL 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	$C_1 \sqsubseteq \neg C_2$	Man $\sqsubseteq \neg$ Female
sameIndividualAs	$\{a_1\} \equiv \{a_2\}$	{presBush} \equiv {G.W.Bush} <small>elements are equivalent</small>
differentFrom	$\{a_1\} \sqsubseteq \neg\{a_2\}$	{john} $\sqsubseteq \neg$ {peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	hasCost \equiv hasPrice
inverseOf	$P_1 \equiv P_2^-$	hasChild \equiv hasParent ⁻
transitiveProperty	$P^+ \sqsubseteq P$	ancestor ⁺ \sqsubseteq ancestor
functionalProperty	$\top \sqsubseteq (\leq 1 P)$	$\top \sqsubseteq (\leq 1 \text{ hasFather})$ <small>2 or above are not functional property</small>
inverseFunctionalProperty	$\top \sqsubseteq (\leq 1 P^-)$	$\top \sqsubseteq (\leq 1 \text{ hasSSN}^-)$

Syntax Example (Constructs)

OWL constructor	DL constructor	Example
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer elements of TBOX
complementOf	$\neg C$	\neg Male
oneOf	$\{a_1\} \sqcup \dots \sqcup \{a_n\}$	{john} \sqcup {mary} elements of ABOX
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor
someValuesFrom	$\exists P.C$	\exists hasChild.Lawyer
maxCardinality	$(\leq n P)$	$(\leq 1$ hasChild)
minCardinality	$(\geq n P)$	$(\geq 2$ hasChild)

Complex Constructs

Most OWL constructs are binary and translate to RDF as <subject, predicate, object>. However, constructs such as *owl:someValuesFrom*, *owl:allValuesFrom*, *owl:minCardinality*, *owl:maxCardinality* are N-ary relationships that require using blank nodes together with *owl:Restriction* (i.e., they use **reification** to express the construct)

Example:

```
:a rdfs:subClassOf owl:Restriction  
_:a owl:onProperty :Leads  
_:a owl:allValuesFrom :Professor
```

_:a is a complex concept

denotes the property used to create the complex concept

denotes the class expression related to the property

The class describing the set of individuals for which all range values of the property *:Leads* are instances of the class *:Professor* (i.e., *this is equivalent to the use of range in RDFS*)

:Department *rdfs:subClassOf* *_:a* would be equivalent in DL to:

:Department \sqsubseteq \forall *:Leads* . *:Professor*

Complex Constraints (II)

Cardinalities (via constrained roles):

```
_:a rdfs:subClassOf owl:Restriction
_:a owl:onProperty RegisteredTo
_:a owl:minCardinality 3
_:b rdfs:subClassOf owl:Restriction
_:b owl:onProperty RegisteredTo
_:b owl:maxCardinality 6

:Student rdfs:subClassOf _:a
:Student rdfs:subClassOf _:b
```

How would you express on OWL the following axiom?

$$C_1 \sqsubseteq \exists P.C$$

Complex Constraints (III)

Union and Intersection (functional syntax)

```
_:a  rdfs:subClassOf      owl:Restriction  
_:a  owl:onProperty      :TeachesTo  
_:a  owl:someValuesFrom  :Undergrad  
_:b  owl:unionOf         (:Professor, :Lecturer)  
_:a  rdfs:subClassOf      _:b
```

How do you express this constraint in DL?

OWL Implementation

- I. Uses RDF syntax (i.e., URIs and literals that conform valid triples)
- II. It reuses some URIs from RDFS (e.g., `rdfs:subClassOf`). However, be aware that the whole RDFS is NOT a subset of OWL and the RDFS rule-based inference does not apply in OWL EXAMPLE?
- III. OWL adds new properties and classes based on DL and defined at the OWL namespace:
<http://www.w3.org/2002/07/owl#>
`owl:Thing`, `owl:Restriction`, `owl:Ontology`

Example of OWL Syntax (II)

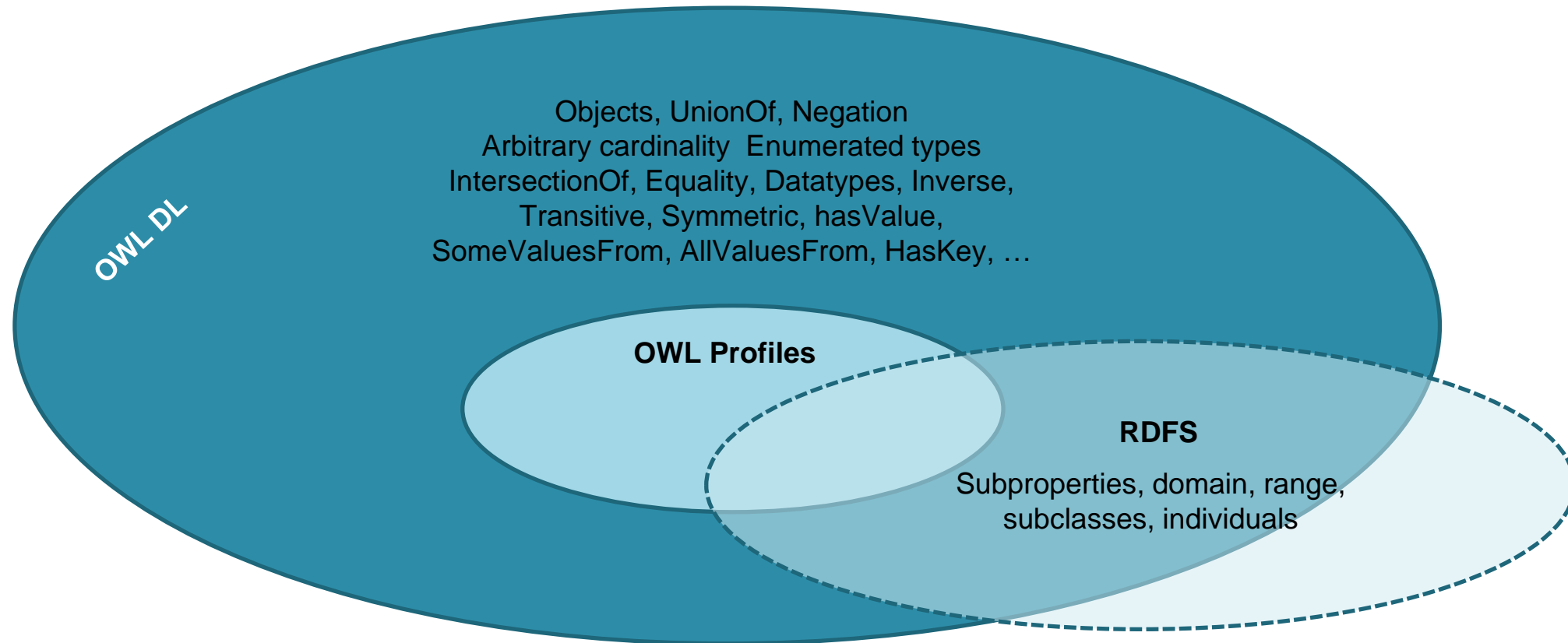


functional notation are not asked in exam

[Sample](#)

This example uses a new **functional notation** that avoids the overloading owl:Restriction notation and XML

OWL 2.0



<https://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/#Vocabulary>

OWL-DL SROIQ

Based on Sven Groppe's Data Management and Query Processing in Semantic Web Databases

OWL 2 Profiles

interesting

↑ reasoning can be reduced to log space
abox is stored in relational database
used much in data world

OWL 2 EL:

Based on *EL++*

Large number of
properties / classes

Reasoning:
Polynomial with
regard to the
ontology TBOX

used in bioinformatics
used to represent DL with lots of
classes and properties
but just taxonomies

OWL 2 QL:

Based on DL-Lite

Captures (most of)
ER and UML
expressive power

Reasoning:
Reducible to
LOGSPACE (i.e.,
DBs)



The expressivity required to capture UML schemas (like the exercise from the previous session) is contained in **OWL 2 QL**!

OWL 2 RL:

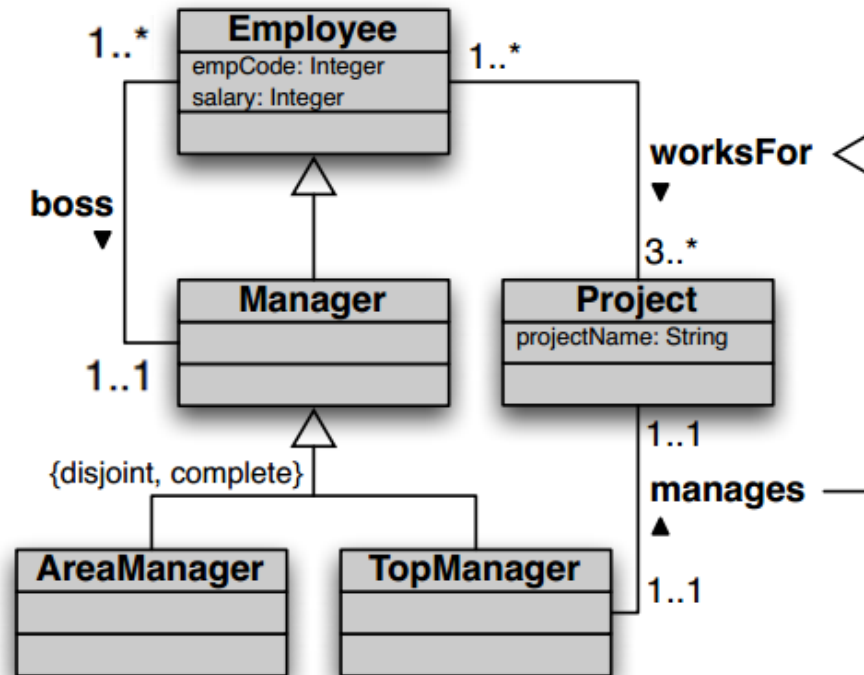
Based on Description
Logic programs

Scalable reasoning
without sacrificing
much expressivity

Reasoning:
Polynomial with
regard to the size of
the ontology

Exercise

Create a OWL Knowledge Graph capturing as much constraints as possible from the following schema:



Summary

Description Logics

- TBOX
 - Constructs
 - Formal Semantics
- ABOX
- Reasoning

OWL

- Languages
- Profiles