

RDF STRUCTURE

- triple is a fundamental data structure of RDF
- set of triples can be represented as a graph
 - Subjects & objects : nodes
 - Predicates : edges
- two or more RDF graphs can merged into one graph
- namespaces are sets of names used to simplify encoding & prevent collisions of abbreviations

rdf:type relation between instances & classes

rdf:Property root type indicating an identifier as a property

rdfs:subClassOf everything of type X is also from type Y
 $X \rightarrow Y \rightarrow Z \Rightarrow X \rightarrow Z$

rdfs:subPropertyOf like a hierarchy
quasi inheritance (not clear)

ex: iPhone ex: madeBy ex: Apple
ex: madeBy rdfs:subPropertyOf ex: producedBy
 \Rightarrow ex: iPhone ex: producedBy ex: Apple

rdfs:domain X instance of class rdf:Property
Y instance of rdfs:Class
rdf:Property rdfs:subClassOf rdfs:Class

all subjects of triples, whose predicate X is of type Y

rdfs:range all objects of triples, whose predicate X is of type Y

Example

madeBy rdf:type rdf:Property
madeBy rdfs:domain ex:Product
madeBy rdfs:range ex:Company
NikonD3 madeBy Nikon

Product rdf:type rdfs:Class
NikonD3 rdf:type Product
Nikon rdf:type Company

VK Semantic Web

Name: _____

Matrikelnummer: _____

	a	b	c	d	Overall
RDF/RDFS					
SPARQL					
OWL/DL			--	--	

Please give short answers to the questions straight to the point.

1. RDF/RDFS

- a) (2+2+2+2)
 - i. How can you identify AAU on the Web? Give an example.
 - ii. Give a short description of the basic structure of an RDF document.
 - iii. Explain why RDF is defined on top of XML?
 - iv. Why XML alone is not an alternative to RDF?
- b) (3) What is a blank node? Encode the following text as two RDF triples: Tim knows someone who studies at AAU.
- c) (3) How can one express in RDF statements made by a third party? Give an example.
- d) (6) Use RDF deductive system to get three derivations from the following RDFS document.
Explain how results were obtained.

```
Ex:isWrittenBy rdfs:subPropertyOf ex:isAuthoredBy
      ex:isAuthoredBy rdfs:range ex:Book
      ex:isAuthoredBy rdfs:domain ex:Person
      ex:AI_Modern ex:isWrittenBy ex:Russel
```

2. SPARQL

- a) (3) Write a query that returns all triples of an RDF document ordered by predicates.
- b) (5) Return names of all persons that are listed in the document which do not have universal identifiers.
- c) (6) Count the number of persons who are known by Axel.
- d) (6) Find all persons that do not have last name in the FOAF attached.

3. OWL/DL

- a) (1+3+1+3) Model in OWL or DL
 - i. All birds are animals.
 - ii. Some of the birds are flying birds and some are not.
 - iii. Each jet has an engine.
 - iv. Vegetarian pizzas cannot have toppings including meat.
- b) (10) Find whether the following ontology is consistent. Present the solution in form of a tree as on the slides.

$$\begin{aligned} A &\sqsubseteq \exists s. \neg B \\ C &\sqsubseteq A \sqcap B \\ C(x) \\ C(y) \\ s(x,y) \end{aligned}$$

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.  
@prefix foaf: <http://xmlns.com/foaf/0.1/>.  
@prefix xml: <http://www.w3.org/XML/1998/namespace>.  
@prefix owl: <http://www.w3.org/2002/07/owl#>.  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.  
<callto://droxel> a <http://skype.com/>.  
  
<http://glo.net#me> a foaf:Person;  
    rdfs:seeAlso <http://glo.net/foaf.rdf>;  
    foaf:name "Giovanni Tummarello".  
  
<http://sw.deri.org/~aidanh/foaf/foaf.rdf#Aidan_Hogan> a foaf:Person;  
    rdfs:seeAlso <http://sw.deri.org/~aidanh/foaf/foaf.rdf>;  
    foaf:homepage <http://www.aidanhogan.com/>;  
    foaf:name "Aidan Hogan".  
  
<http://www.polleres.net/foaf.rdf#me> a foaf:Person;  
    rdfs:seeAlso <http://www.google.com/profiles/droxel>;  
    foaf:depiction <http://www.polleres.net/images/axel.jpg>;  
    foaf:family_name "Polleres";  
    foaf:familyName "Polleres";  
    foaf:firstName "Axel";  
    foaf:givenname "Axel";  
  
    foaf:knows <http://glo.net#me>,  
        _:bnode757172928,  
        _:bnode624461248;  
  
    foaf:lastName "Polleres".  
  
_:bnode624461248 a foaf:Person;  
    foaf:homepage <http://apassant.net/>;  
    foaf:name "Alexandre Passant".  
  
_:bnode757172928 a foaf:Person;  
    foaf:homepage <http://www.w3.org/People/Eric/>;  
    foaf:name "Eric Gordon Prud'hommeaux".
```

RDF / RDFS

1. RDF/RDFS

- a) (2+2+2+2)
 - i. How can you identify AAU on the Web? Give an example.
 - ii. Give a short description of the basic structure of an RDF document.
 - iii. Explain why RDF is defined on top of XML?
 - iv. Why XML alone is not an alternative to RDF?
- b) (3) What is a blank node? Encode the following text as two RDF triples: Tim knows someone who studies at AAU.
- c) (3) How can one express in RDF statements made by a third party? Give an example.
- d) (6) Use RDF deductive system to get three derivations from the following RDFS document.
Explain how results were obtained.

```
Ex:isWrittenBy rdfs:subPropertyOf ex:isAuthoredBy  
ex:isAuthoredBy rdfs:range ex:Book  
ex:isAuthoredBy rdfs:domain ex:Person  
ex:AI_Modern ex:isWrittenBy ex:Russel
```

SDAPOL

(a)

AAU can be identified with the URI, which is a combination of the URL & VAI

Browser is finding the document with the URI

URI: how & where a resource can be identified

URN: persistent, location independent identifier

"printed copies of book"

Resource can be accessed

Example

<http://www.uni-klu.ac.at?query=value#fragment>

- URN (Uniform Resource Name) persistent, location independent identifier

BASIC STRUCTURE RDF Document

it is a set of triples <Subject, Predicate, Object>

<rdf:RDF xmlns:rdf> → defining in the header of the XML

RDF on top of XML, why?

RDF-tag is defined in the root

can be also used with a namespace

RDF-doc is also XML-doc

- triple is a fundamental data structure of RDF
- set of triples can be represented as a graph
 - Subjects & Objects : nodes
 - Predicates : edges
- two or more RDF graphs can merged into one graph
- namespaces are sets of names used to simplify encoding & prevent collisions of abbreviations

Why XML alone not alternative to RDF?

- it is only for defining vocabularies & allowed structures
- it is used for metadata & an extension of HTML
- serialization language storing data in universal format
- relation between subject & objects in RDF well-defined
⇒ predicate is in the URI

RDF uses URIs which are globally unique, while XML URI's are just unique in the document

XML data model is a tree, while RDF is a graph

RDF is based on a schema, while is very flexible, also when there is an ontology

b) Blank node

existentially quantified variables are expressed as blank nodes

Tim knows someone who studies at AAU.

person: Tim	ex: knows	-: someone
-: someone	ex: studies	uni: AAU

there is no URI used as an identifier!

c) expressing RDF as a third party + example

A person is saying, that Kira has 3 children. One is a boy

ex: Kira	ex: has	-: children
-: children	a	ex: boy

Siehe Stackoverflow!

Now I tell this on

- c) (5) how can one express in RDF statements made by a third party? Give an example.
d) (6) Use RDF deductive system to get three derivations from the following RDFS document.

Explain how results were obtained.

```
Ex:isWrittenBy rdfs:subPropertyOf ex:isAuthoredBy
    ex:isAuthoredBy rdfs:range ex:Book
    ex:isAuthoredBy rdfs:domain ex:Person
    ex:AI_Modern ex:isWrittenBy ex:Russel
```

CDAP01

ex: AI_Modern	ex: isAuthoredBy	ex: Russel
ex: Book	ex: isAuthoredBy	ex: Person
ex: Russel	Rdf: type	ex: Person
ex: isAuthoredBy	Rdf: type	Rdf: Property
ex: Person	Rdf: type	Rdfs: Class

SPARQL

ex.al-modern ex.tswitterby ex.russet

2. SPARQL

- (3) Write a query that returns all triples of an RDF document ordered by predicates.
- (5) Return names of all persons that are listed in the document which do not have universal identifiers.
- (6) Count the number of persons who are known by Axel.
- (6) Find all persons that do not have last name in the FOAF attached.

2. QMUL/DL

a) `SELECT ?subject ?predicate ?object
ORDER BY (?predicate)`

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.  
@prefix foaf: <http://xmlns.com/foaf/0.1/>.  
@prefix xml: <http://www.w3.org/XML/1998/namespace>.  
@prefix owl: <http://www.w3.org/2002/07/owl#>.  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.  
<callto://droxel> a <http://skype.com/>.  
  
<http://glo.net#me> a foaf:Person;  
    rdfs:seeAlso <http://glo.net/foaf.rdf>;  
    foaf:name "Giovanni Tummarello".  
  
<http://sw.deri.org/~aidanh/foaf/foaf.rdf#Aidan_Hogan> a foaf:Person;  
    rdfs:seeAlso <http://sw.deri.org/~aidanh/foaf/foaf.rdf>;  
    foaf:homepage <http://www.aidanhogan.com/>;  
    foaf:name "Aidan Hogan".  
  
<http://www.polleres.net/foaf.rdf#me> a foaf:Person;  
    rdfs:seeAlso <http://www.google.com/profiles/droxel>;  
    foaf:depiction <http://www.polleres.net/images/axel.jpg>;  
    foaf:family_name "Polleres";  
    foaf:familyName "Polleres";  
    foaf:firstName "Axel";  
    foaf:givenname "Axel";  
  
    foaf:knows <http://glo.net#me>,  
        _:bnode757172928,  
        _:bnode624461248;  
  
    foaf:lastName "Polleres".  
  
_:bnode624461248 a foaf:Person;  
    foaf:homepage <http://apassant.net/>;  
    foaf:name "Alexandre Passant".  
  
_:bnode757172928 a foaf:Person;  
    foaf:homepage <http://www.w3.org/People/Eric/>;  
    foaf:name "Eric Gordon Prud'hommeaux".
```

c) `Select x; (COUNT (?knows) AS ?know)
where x ? foaf:knows ?knows`

d) `SELECT x ?
WHERE`

OWL / DL

3. OWL/DL

a) (1+3+1+3) Model in OWL or DL

- i. All birds are animals.
- ii. Some of the birds are flying birds and some are not.
- iii. Each jet has an engine.
- iv. Vegetarian pizzas cannot have toppings including meat.

b) (10) Find whether the following ontology is consistent. Present the solution in form of a tree as on the slides.

$$A \sqsubseteq \exists s. \neg B$$

$$C \sqsubseteq A \sqcap B$$

$$C(x)$$

$$C(y)$$

$$s(x, y)$$

(a)

i. Birds Subclassof Animals

not flying birds

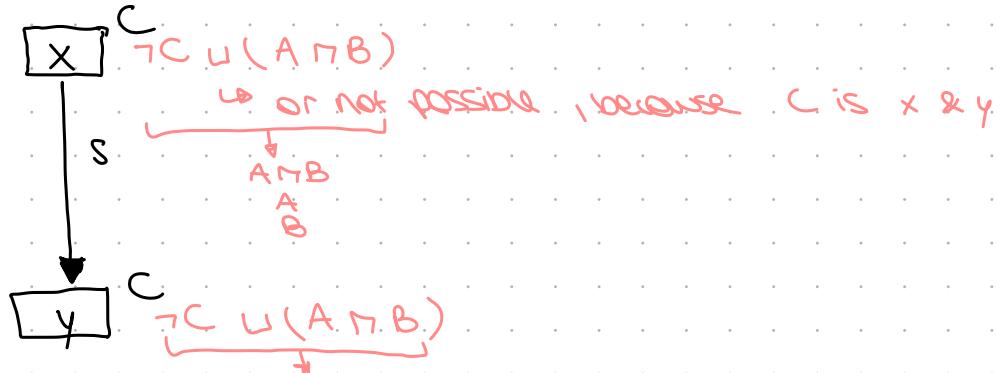
ii. flyingbirds Types Birds
not flyingbirds Types Birds

flyingbirds owl:complement

iii. Jet Subclassof Has Some Engine

iv. topping Subclassof Has Some (meat or vegetarian)
vegetarianpizza

(b)



We can move rule a in x or y , because it is not clear.

$$A \sqsubseteq \exists s. \neg B$$

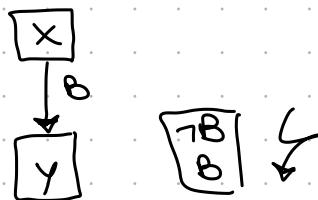


Ontology is consistent, because we don't have a clash everywhere

Existing is just checking the possibility, not existence

Example

$A \sqsubseteq HS, \neg B$



Question

so what is the value for X ? \rightarrow then C, A, B

How many nodes does the tree have?

3 or 4, depends on how to apply A

Example

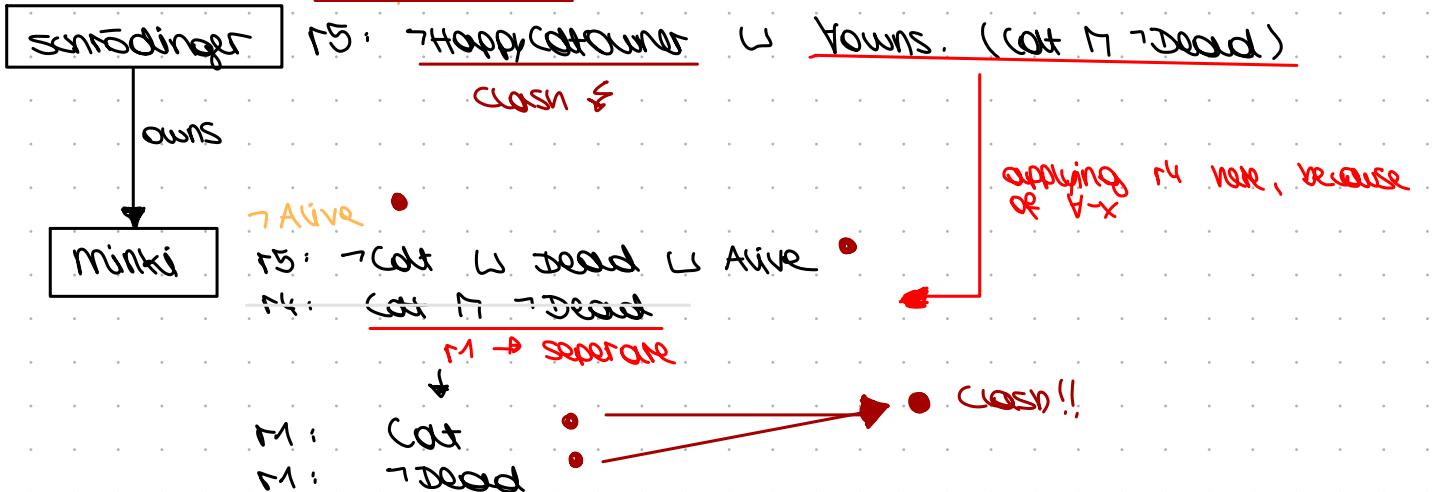
HappyCatOwner (Schrödinger)

owns (Schrödinger, Minki)

HappyCatOwner \sqsubseteq Towns. (Cat \sqcap \neg Dead)

Cat \sqsubseteq Dead \sqcup Alive

HappyCatOwner



Axioms OWL to RDF TURTLE

Human $\sqsubseteq \neg Alien$

ex: Human rdfs: subclassof [owl: complementof ex: Alien]

FatherWithDaughters $\sqsubseteq Man \sqcap hasChild . Woman$

ex: FatherWithDaughters rdfs: subclassof [owl: allValuesFrom ex: Woman
rdfs: subclassof [owl: someValuesFrom ex: Woman
owl: onProperty ex: hasChild]

Sibling $\sqsubseteq \exists parent . \exists hasChild . \neg Self$

ex: Sibling rdfs: subclassof [owl: onProperty ex: Parent ;
owl: someValuesFrom [owl: hasSelf false ;
owl: onProperty ex: Child]]