Chapter 3:

a significant* chapter

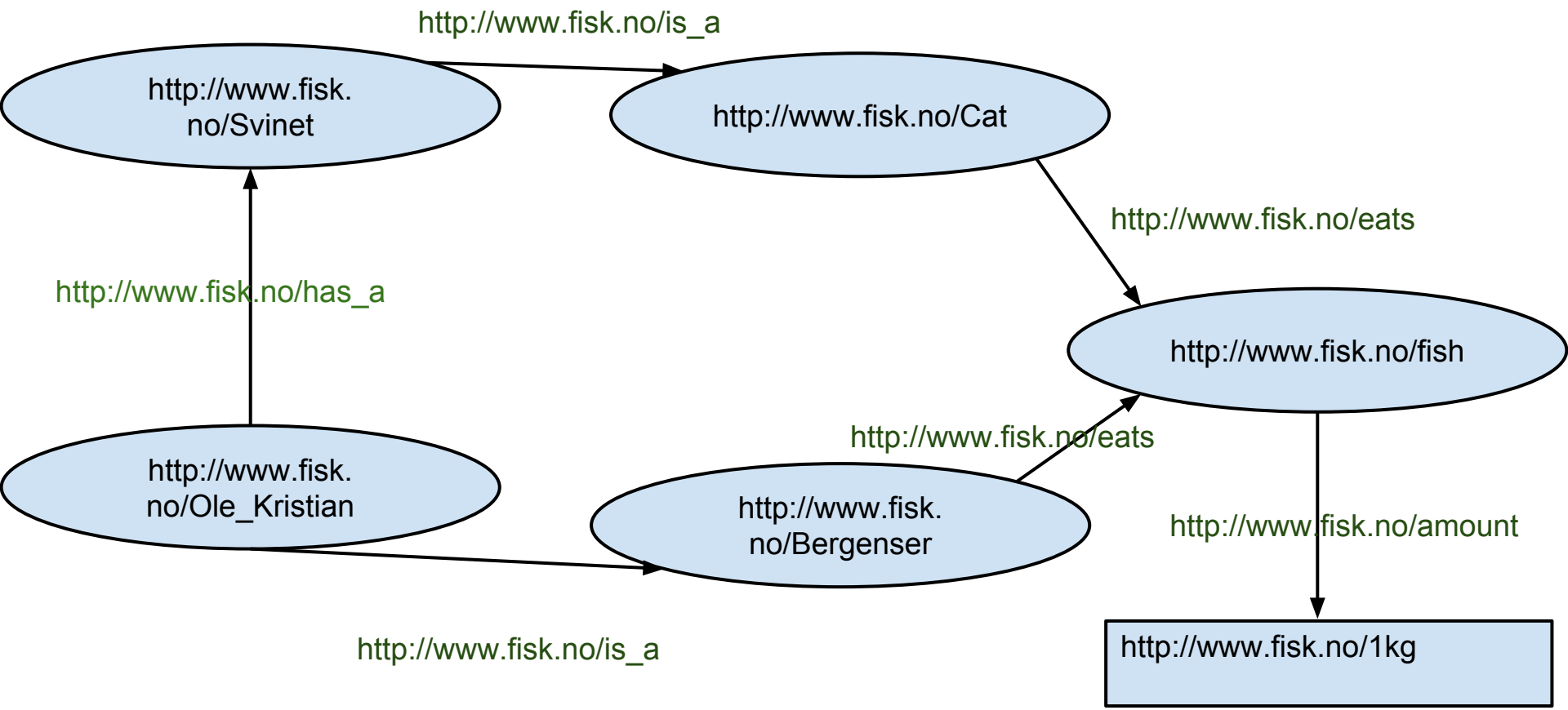
*significant (eng) <-- semantic (gr)

Why this significance?

- Provide a meaning (interpretation), e.g. when our cat "Svinet" smells the odour of a fish
  - Is a cat an animal?
  - Is a fish something a cat may enjoy eating?
  - Would I become angry if Svinet (a cat) sits on the table eating it?

- We will now show you an interpretation of a small example

A graphical view



Interpretation

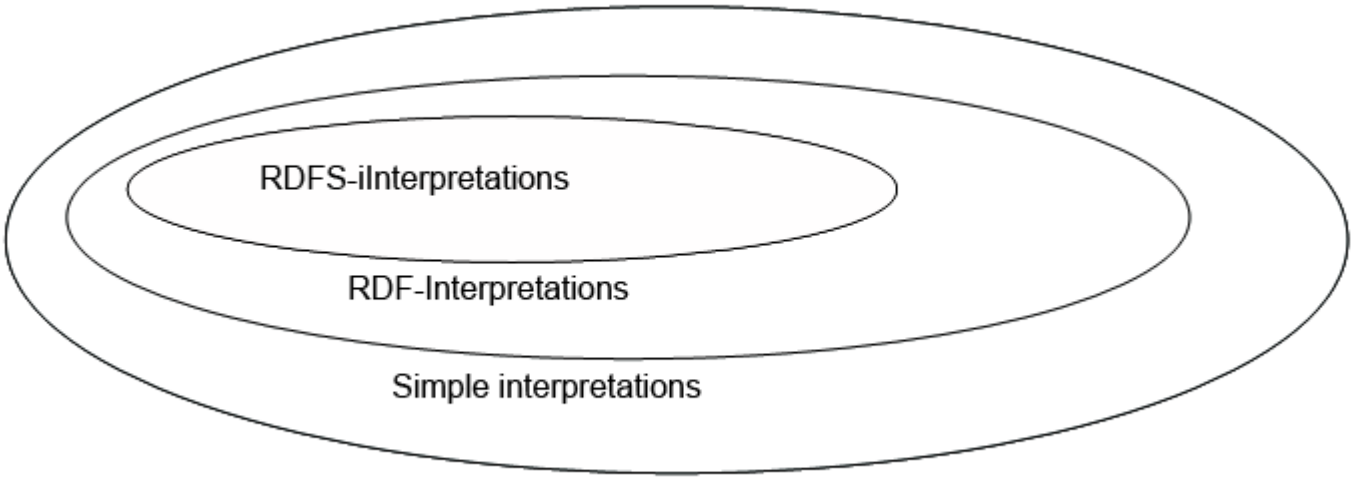
- An Interpretation contains resources and properties and how they are connected
  - Example: Svinet isA Cat
  - Resources: Svinet and Cat
  - Properties isA
- Entailing knowledge into compressed representation;
- Interpreting the knowledge to correctly answer questions using our data
  - "A (is a) C" entails "{A (is a) B; B (is a) C}"
  - Notation: Propositions P = [p3=”Svinet eats fish”, p1=“Svinet is a cat”, p2=“A cat eats fish”} gives: “{p1, p2} |= p3”

Observations -> ?(knowledge) ->mess

- Applying tacit knowledge about Svinet, I added > 10,000 facts
- How can we take "shortcuts" in our universe?
- Syntactic reasoning required
  - Deduction rules if(p1.....pn) is valid, 'p' is true...
    - aka. "inference rules".....aka. P |- P'
  - The complete set: "deduction calculus"

Interpretation figure

- Every RDFS interpretation is a valid RDF interpretation and every RDF interpretation is valid Simple interpretation.



Interpretation extension

- RDF support formalisms, but limit the exactness
  - Ole Kristian (rdf: type) Bergenser
- RDF(s) makes it possible adding terminological knowledge as triples
  - Extending the class-specification
  - "Svinet (is a) cat" = "Svinet (rdfs: subClassOf) cat"
  - "Hard drive (is property of) a computer = "Hard Drive (rdfs: subProperyOf) computer"
  - "Cat (is in domain) animals" = "Cat (rdfs:domain) animals.

Deduction syntax

- A deduction is normally given in the form:
  - $((p_1 \ldots p_n) / p)$
- The letters used when describing the rules
  - *a* and *b* can refer to arbitrary URIs (anything admissible for the predicate position in a triple)
  - _:n will be used for the ID of a blank node
  - *u* and *v* refer to arbitrary URIs or blank node IDs (any possible subject of a triple)
  - *x* and *y* can be used for arbitrary URIs, blank node IDs or literals (i.e. anything admissible for the object position in a triple)
  - *l* may be any literal

Deduction for simple entailment

- Two different rules
  - (u a x .) / (u a _:n .) se1
  - example: Per is Ole Kristians brother will be reduced to Per has a brother.
  - (u a x .) / (_:n a x .) se2
  - example: Ole Kristian is the owner of Svinet. Will be reduced to: Svinet has an ower.
  - From these deductions we can extract following sentence: ***Per is the brother of Svinets owner***

Simple Entailment Deduction Theorem

A graph G_1 simply entails G_2, if G_1 can be extended to a graph G'_1 by virtue of the rules se1 and se2 such that G_2 is contained in G'_1

RDF and RDFS - Rules

- RDF has 4 rules
- RDFS has 13 rules
  - We will explain 4 of them
  - Everyone follows the same logic

RDF deduction rules

In RDF-entailment we have 4 new rules.

- /v a x (rdfax)
- v a l . / v a _:n . (lg)
- v a y . / a rdf:type rdf:Property . (rdf1)
- v a l . / _:n rdf:type rdf:XMLLiteral (rdf2)

RDF Deduction - Theorem

A graph G_1 RDF entails a graph G_2 if and only if there is a graph G'_1 that can be derived from G_1 by virtue of the rules lg, rdf1 rdf2, as well as rdfax such that G'_1 simply entails G_2

RDFS - Rules - Subclasses

- u rdf:type rdfs:Class . / u rdfs:subClassOf rdfs:Resource . (rdfs8)
- u rdfs:subClassOf x . v rdf:type u. / v rdf:type x . (rdfs9)
- u rdf:type rdfs:Class . / u rdfs:subClassOf u. (rdfs10)
- u rdfs:subClassOf v . v rdfs:subClassOf x . / u rdfs:subClassOf x . (rdfs11)