Slovník RDFS dokončenie

Spracované podľa prednášok TUD

• kapitola Lists v www.inf.tu-dresden.de/content/institutes/ki/cl/study/summer14/fswt/slides/FSWT2014-L2-IntroRDF.pdf

• http://www.inf.tu-dresden.de/content/institutes/ki/cl/study/summer14/fswt/slides/FSWT2014-L3-RDFS.pdf

a **dokumentov W3C**

• http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/#section-vocabulary - neformálny úvod

• http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

• http://www.w3.org/TR/rdf-schema/ kompletná špecifikácia

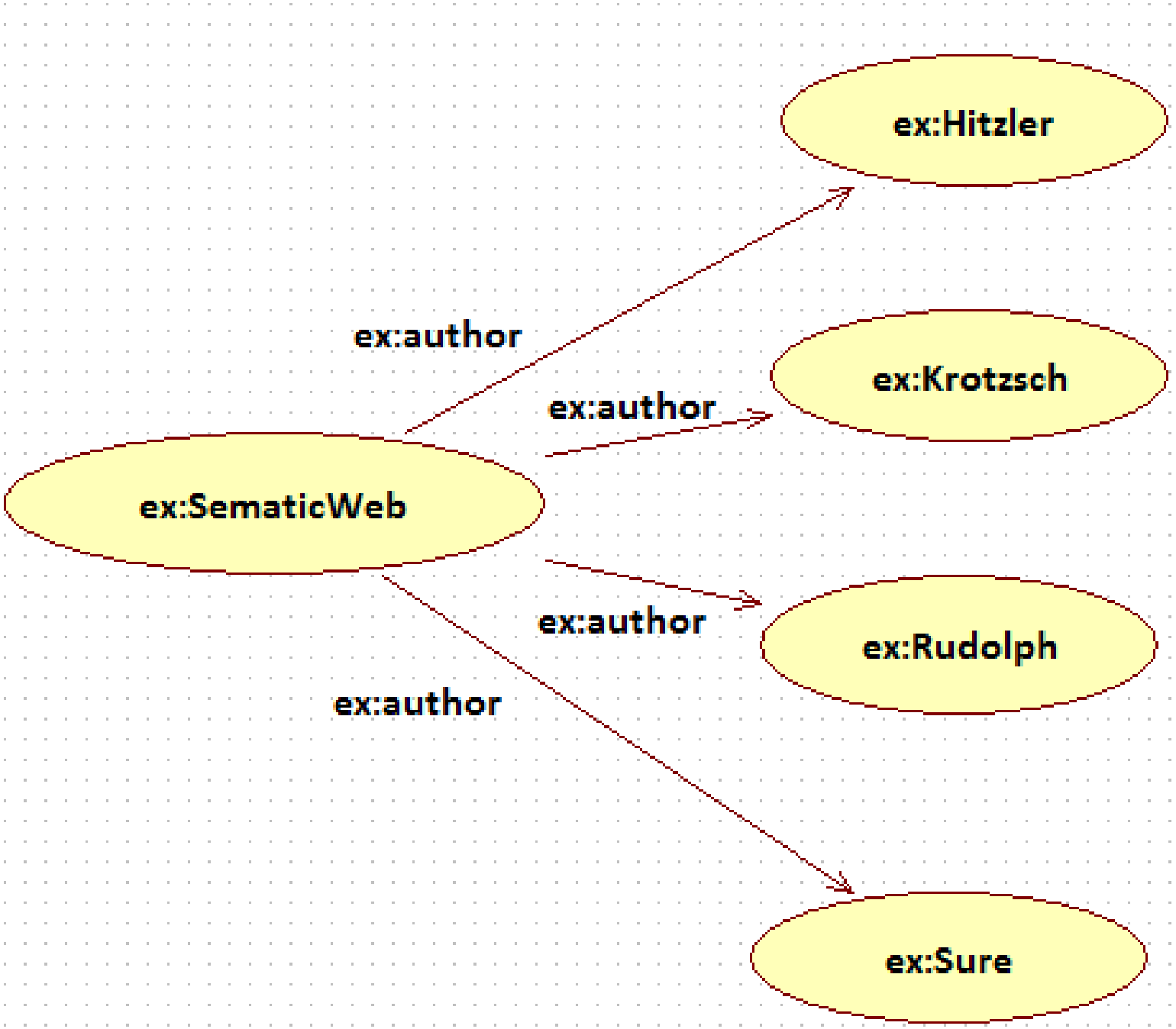
• https://www.w3.org/2007/02/turtle/primer/

Téma nasledujúcich prednášok – sémantika podľa

• http://www.w3.org/TR/rdf11-mt/

• http://www.inf.tu-dresden.de/content/institutes/ki/cl/study/summer14/fswt/slides/FSWT2014-L4-RDSF-Semantics.pdf

Viacnasobné vzťahy



Zoznam - Lists

Zoznamy sú dátové štruktúry, ktoré umožňujú vymenovať a **odkazovať na viacero resoursov spolu** (pričom obecne záleží aj na poradí jednotlivých resoursov).

RDF rozlišuje dva druhy zoznamov

- Ovtvorené - **kontainery**
  - Možno do nich pridávať dalšie členy
- Uzavreté - **kolekcie**
  - Nie je možné pridávať daľšie členy

Open Lists (container)

Slovník RDF definuje URI pre **triedu kontainerov rdfs:Container**

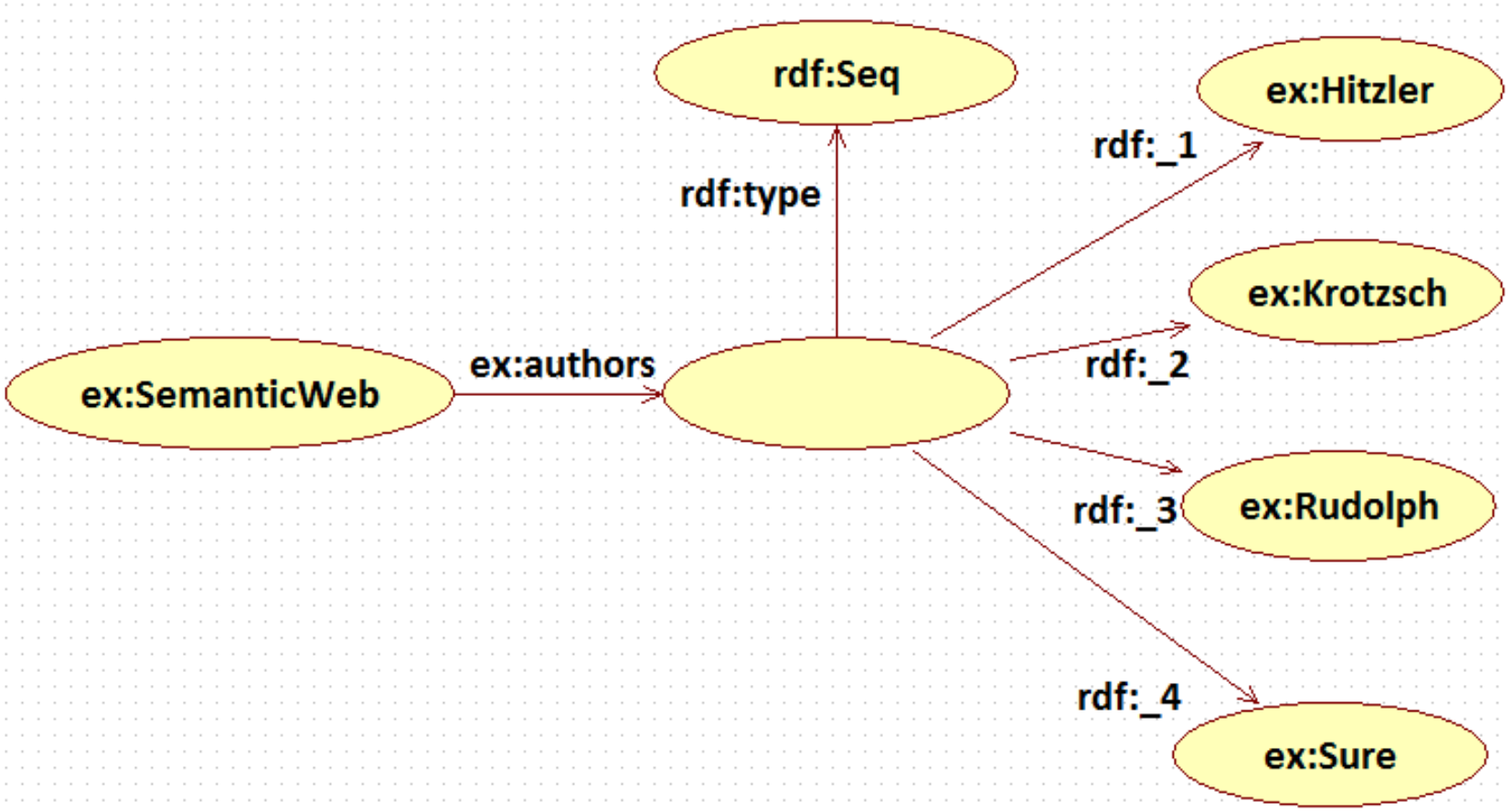
a 3 podtriedy kontainerov:

- **rdf:Seq** – usporiadaná postupnosť
- **rdf:Bag** – neusporiadaná monžina
- **rdf:Alt** – množina alternatív / sinoným

Ďalej definuje URI pre **vzťahy členstva** vyjadrujúce, že resource je v zozname, konkrétne:

- pre každé číslo 1,2,3... URI **rdf:_1, rdf:_2, rdf:_3** ...

Open Lists (container)



Turtle syntax

```
ex:SemanticWeb ex:authors [a rdf:Seq ;
                           rdf:_1 ex:Hitzler ;
                           rdf:_2 ex:Krotzsch ;
                           rdf:_3 ex:Rudolph ;
                           rdf:_4 ex:Sure].
```

Open Lists (container)

Čísla v vo vzťahoch členstva **rdf:_1, rdf:_2, rdf:_3** ...., slúžia iba na vytvorenie identifikátora.

Pre sekvenciu rdf:Seq **určujú aj poradie členov v kontainery** ale nedajú sa chápať ako poradové číslo.

T.j. rdf:_3 určite bude pred rdf:_5. Ale neznamená to, že je nutne 3. členom v kontainery, môže byť hoci aj prvý.

RDF definuje tiež všeobecný vzťah členstva

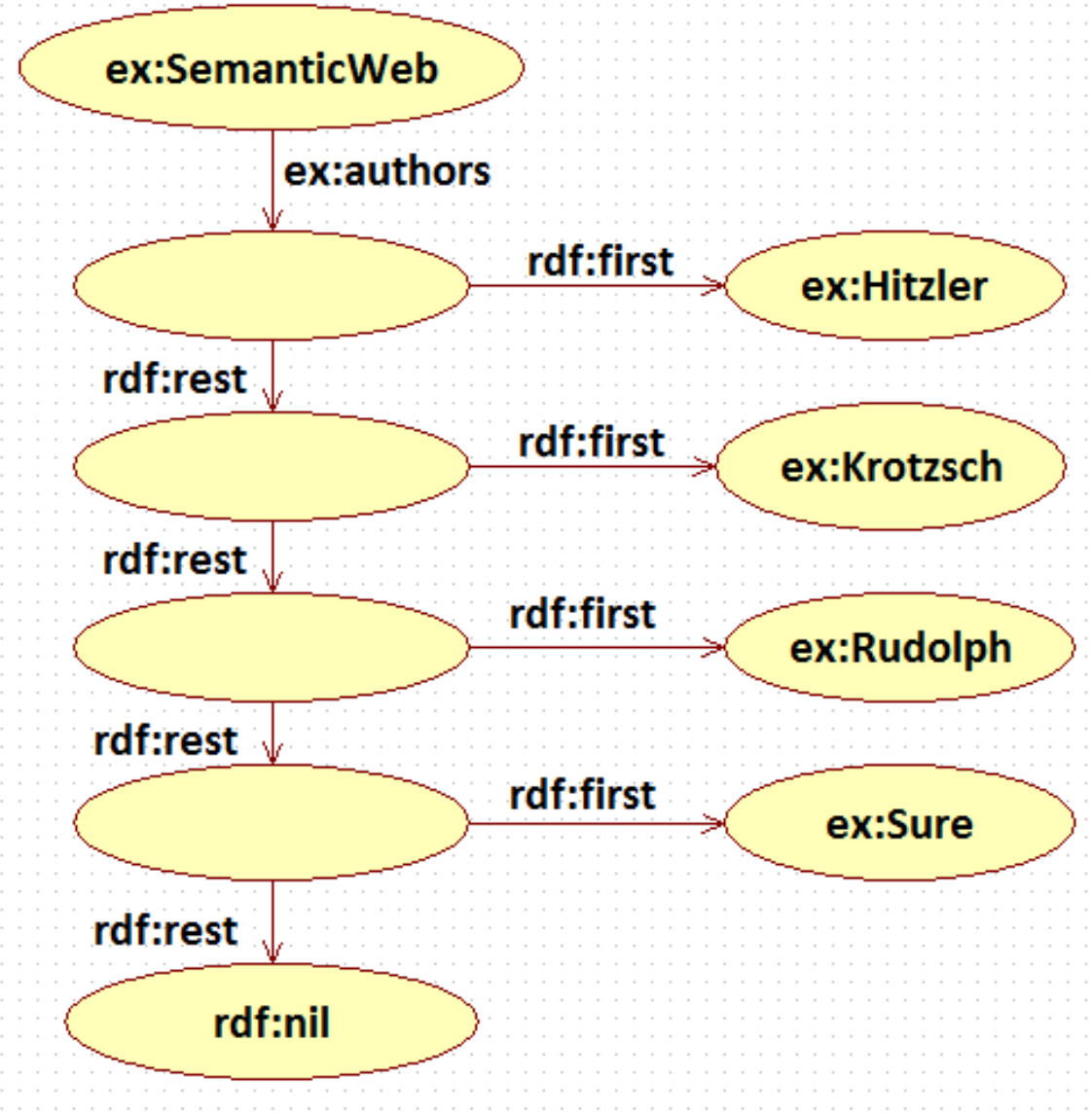
rdf:member

ktorý vyjadruje, že resource je v zozname ale **bez akejkoľvek informácie o poradí**. Je to superproperty všetkým ostatným membership-property

Všetky **vzťahy vyjadrujúce členstvo** v kontajnery sú property typu

rdfs:ContainerMembershipProperty

Closed Lists (collections)



Closed Lists (collections)

Turtle syntax

```
_:autos12 rdf:first ex:Hitzler;
          rdf:rest _:autos13.
_:autos13 rdf:first ex:Krotzsch;
          rdf:rest _:autos14.
_:autos14 rdf:first ex:Rudolph;
          rdf:rest _:autos15.
_:autos15 rdf:first ex:Sure;
          rdf:rest rdf:nil.
ex:SemanticWeb ex:authors _:autos12.
```

Skrátená Turtle syntax

```
ex:SemanticWeb ex:authors
    ( ex:Hitzler ex:Krotzsch ex:Rudolph ex:Sure).
```

Reification - zpredmetnenie

Tvrdenia o tvrdeniach ...povedal, že ...

- problematic in RDF(S): model propositions about proposition (in natural language, such propositions can be identified by a leading "that"), e.g.: "The detective suspects that the butler killed the gardener."
- first modeling attempt:

```
ex:detektive ex:suspects "The butler killed the
gardener." .
```

  - Suboptimal: the literal object cannot be easily referenced in other triples.

- second modeling attempt:

```
ex:detektiv ex:suspects ex:theButlerKilledTheGardener
.
```

  - Suboptimal: we lose the inner structure of the talked about proposition

Reification

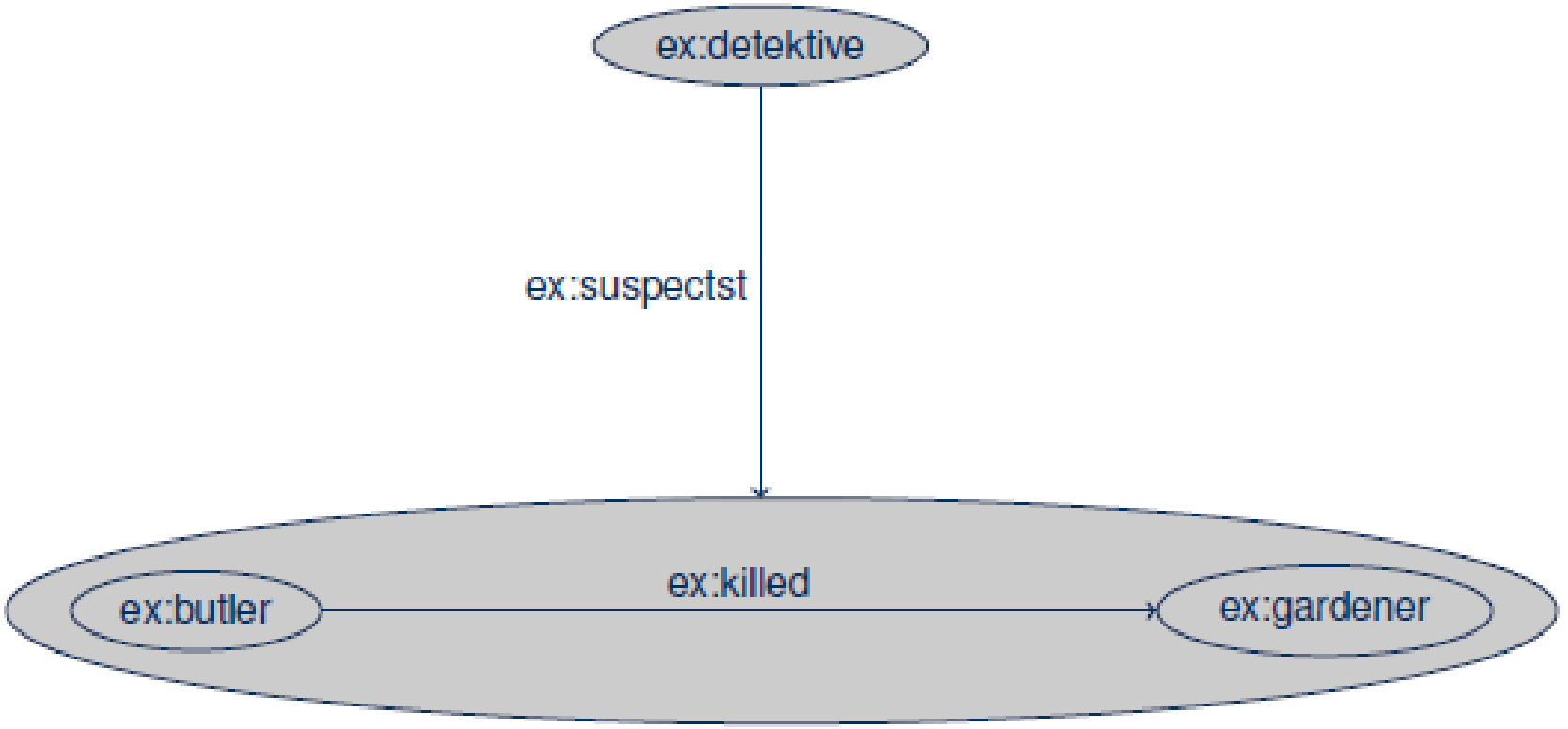
- problematic in RDF(S): model propositions about proposition (in natural language, such propositions can be identified by a leading “that”), e.g.: “The detective suspects that the butler killed the gardener.”
- Out of context, proposition can be easily modeled in RDF:

```
ex:butler ex:killed ex:gardener .
```

- desirable: this whole triple should occur as an object of another triple, however, this is not valid RDF

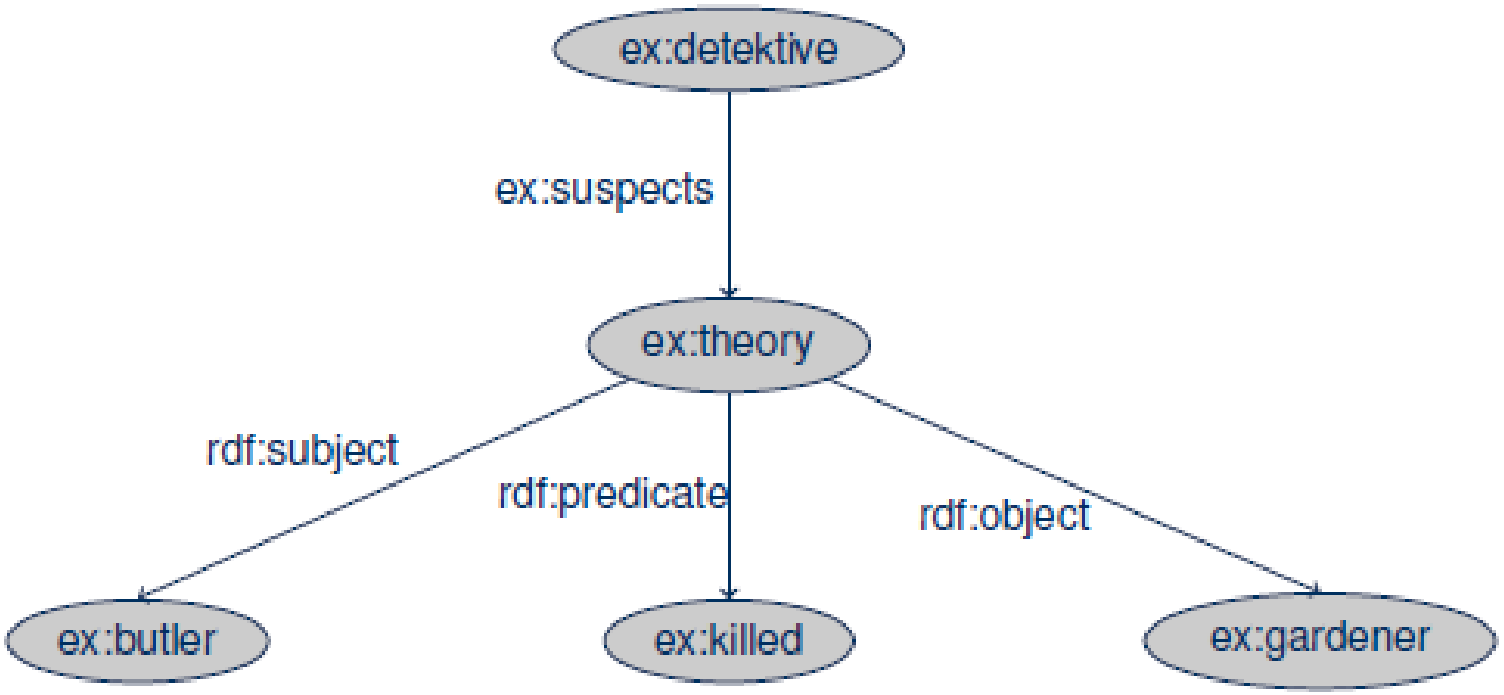
Reification

solution (similar to multi-valued relationships): introduce auxiliary nodes representing the nested proposition:



Reification

solution (similar to multi-valued relationships): introduce auxiliary nodes representing the nested proposition:



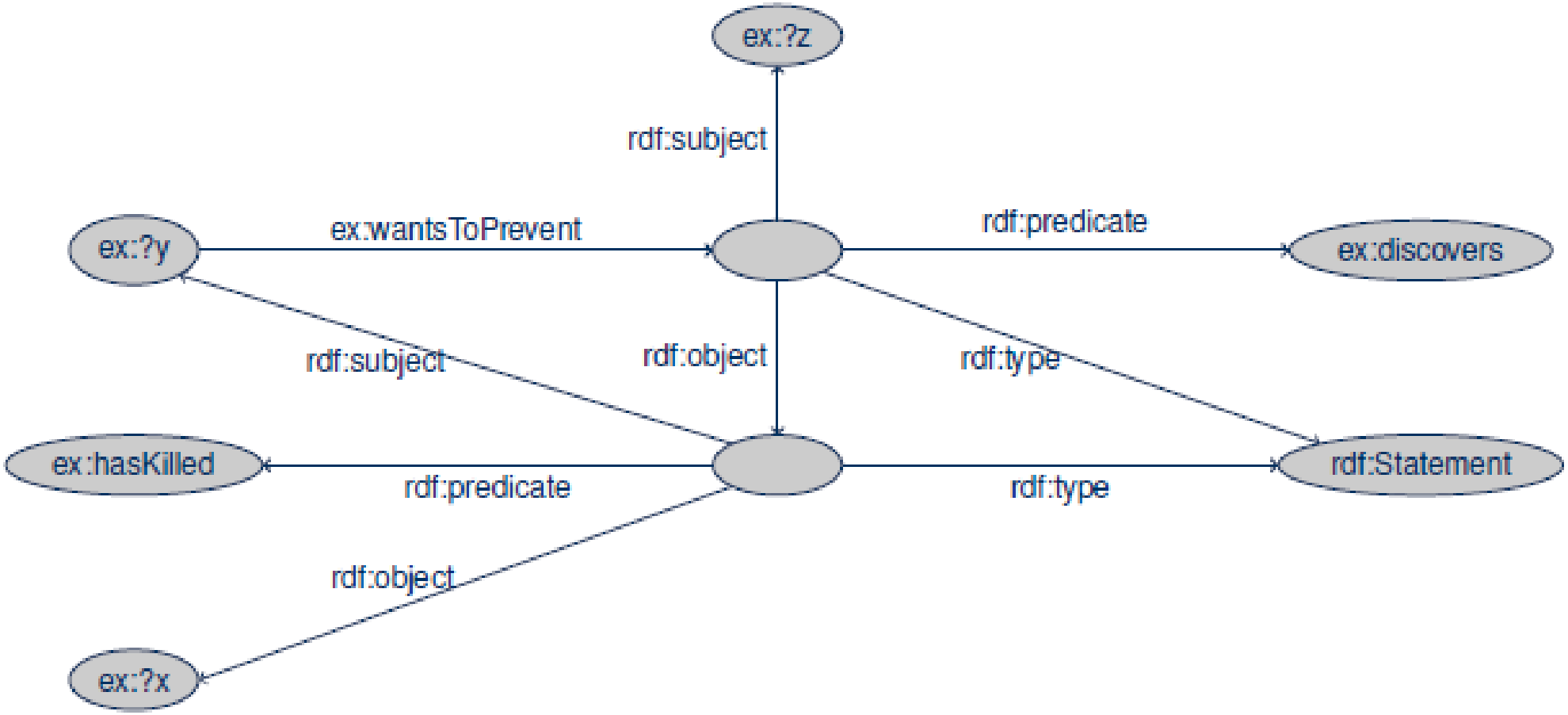
Reification

- caution: reified triple does not need to hold (would not be always sensible either, cf. propositions like: “The detective has doubts that the butler killed the gardener.”)
- if this is wanted, the original (un-reified) triple has to be added to the RDF document
- the class `rdf:Statement` is used to mark nodes which represent reified propositions
- in case this proposition is not referred to from the “outside”, the auxiliary node may be a bnode

Prečítajte

Reification

A small reification riddle: another criminal story...



Additional informations

- like with programming languages, one sometimes wants to add comments (without changing the semantics)
- purpose: increase understandability for human users
- it is to be preferred to model this knowledge in a graph-based way (e.g., due to compatibility reasons)
- thus: defined set of properties that serve this purpose

rdfs:label

`rdfs:label`

- property that assigns a name (Literal) to an arbitrary resource
- often, URIs themselves are difficult to read, or “bulky” at best
- names provided via `rdfs:label` are often used by tools that graphically represent the data

RDF/XML syntax

```
13   <rdf:Description rdf:about="&rzz;KeltRadler">
14     <rzz:alcContent>2.5 %</rzz:alcContent>
15     <rdfs:label xml:lang="sk">Kelt radler</rdfs:label>
16   </rdf:Description>
```

Turtle syntax

```
7 rzz:KeltRadler rzz:alcContent "2.5 %";
8                rdfs:label "Kelt radler"@sk.
```

Additional Information

`rdfs:comment`

- property assigning an extensive comment (literal) to an arbitrary resource
- may e.g. contain the natural language description of a newly introduced class – this facilitates later usage

`rdfs:seeAlso`, `rdfs:definedBy`

- properties giving resources (URIs!) where one can find further information or a definition of the subject resource

RDF/XML syntax

```
16   <rdf:Description rdf:about="&ex;Primates">
17     <rdfs:label xml:lang="de">Primaten</rdfs:label>
18     <rdfs:seeAlso rdf:resource="&wikipedia;Primate"/>
19   </rdf:Description>
```

Turtle syntax

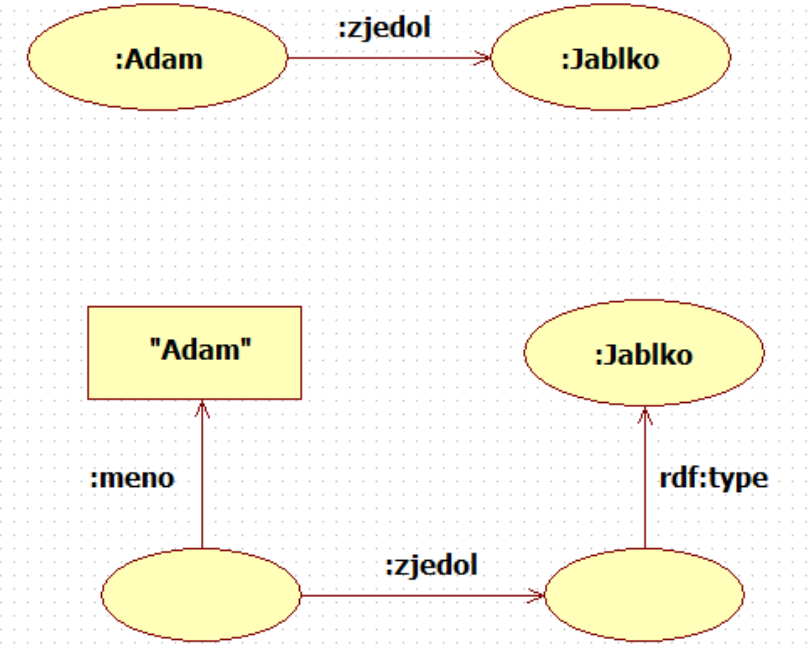
```
8 <http://example.org/Primates> rdfs:label "Primaten"@de;
9                               rdfs:seeAlso wikipedia:Primate.
```

Poznámka k rdf/xml:
rdf:resource, **rdf:about** *a ďalšie mená atribútov v RDF/XML dokumente sú definovane XML/schémou jazyka RDF/XML. Nepatria teda do slovnika jazyka RDF ani RDFS.*
Treba teda rozlišovať medzi:

- **rdfs:Resource** – *URI triedy všetkých resourcov, patrí do slovníka RDFS*
- **rdf:resource** - *meno xml-atribútu ktorý sa používa v RDF-XML dokumentoch.*

Ukážka úloh na precvičenie

Uloha 1. Precitajte rdf-grafy:



Vysvetlite v čom je rozdiel v ich interpretácii.

Uloha 2. Nakreslite rdf-graf, tvrdeni spominaných v starom zákone:

- Adam zjedol jablko, ktore odtrhla Eva.
- Adam zjedol jablko, ktore mu podala Eva.

Uloha 3.

Grafy z predchadzajucich uloh zapiste pomocou syntaxe turtle a rdf/xml. Nezabudnite deklarovat a pouzit vhodne prefixy (menne priestory)

Schema Knowledge with RDFS

- also desirable: specification of logical interdependencies between individuals, classes and relationships, in order to capture as much of the semantics of the described domain as possible, e.g.:
  "Publishers are Organizations."
  "Only persons write books."
- in database speak: schema knowledge

Schema Knowledge with RDFS

RDF Schema (RDFS):

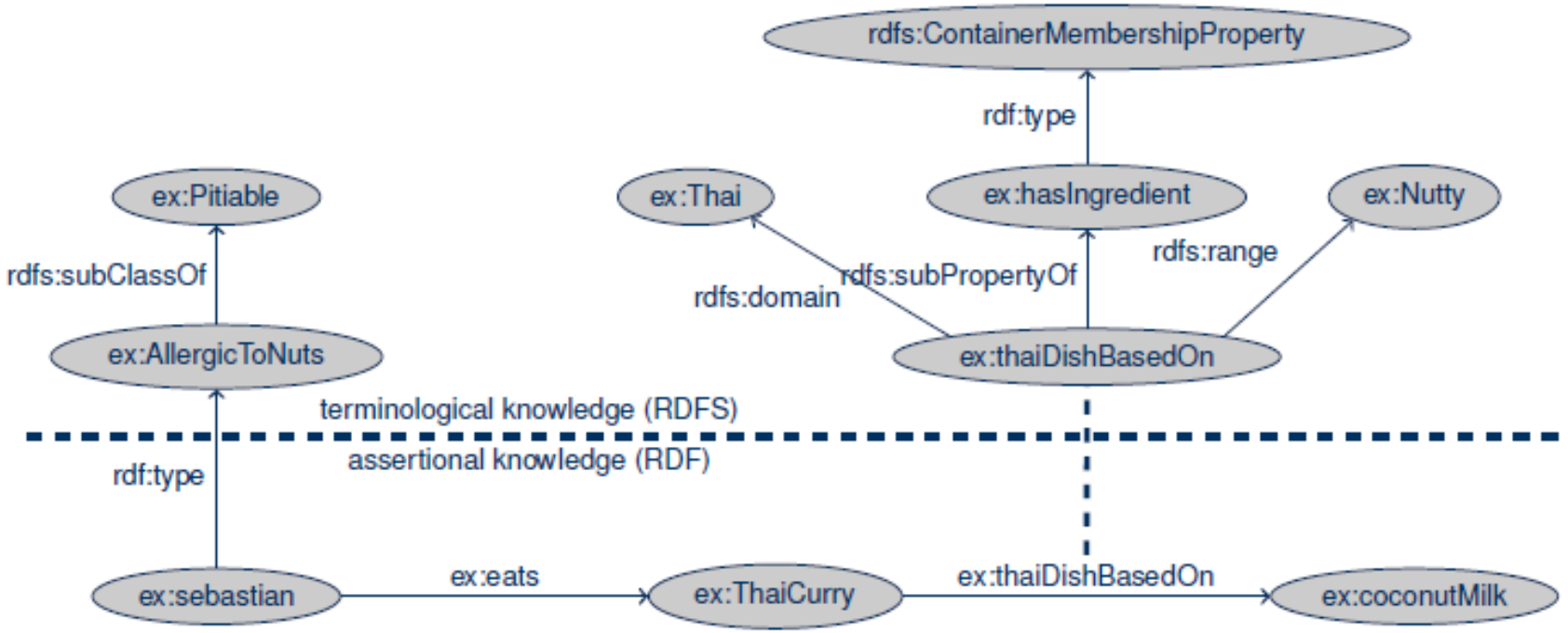
- yet: vocabulary not domain-specific (like, e.g., with FOAF), but generic
- allows for specifying (parts of) the semantics of arbitrary RDF vocabularies (could thus be called a "meta vocabulary")
- advantage: every RDFS-compliant software faithfully supports every vocabulary that has been defined through RDFS
- this functionality makes RDFS an ontology language for lightweight ontologies
- "A little semantics goes a long way."

Simple Ontologies

- By means of the modeling features of RDFS, important aspects of many domains can already be captured semantically.
- Based on the RDFS semantics, a certain amount of implicit knowledge can be derived.
- Consequently, RDFS can be seen as a (though not overly expressive) ontology language.

Simple Ontologies - Example

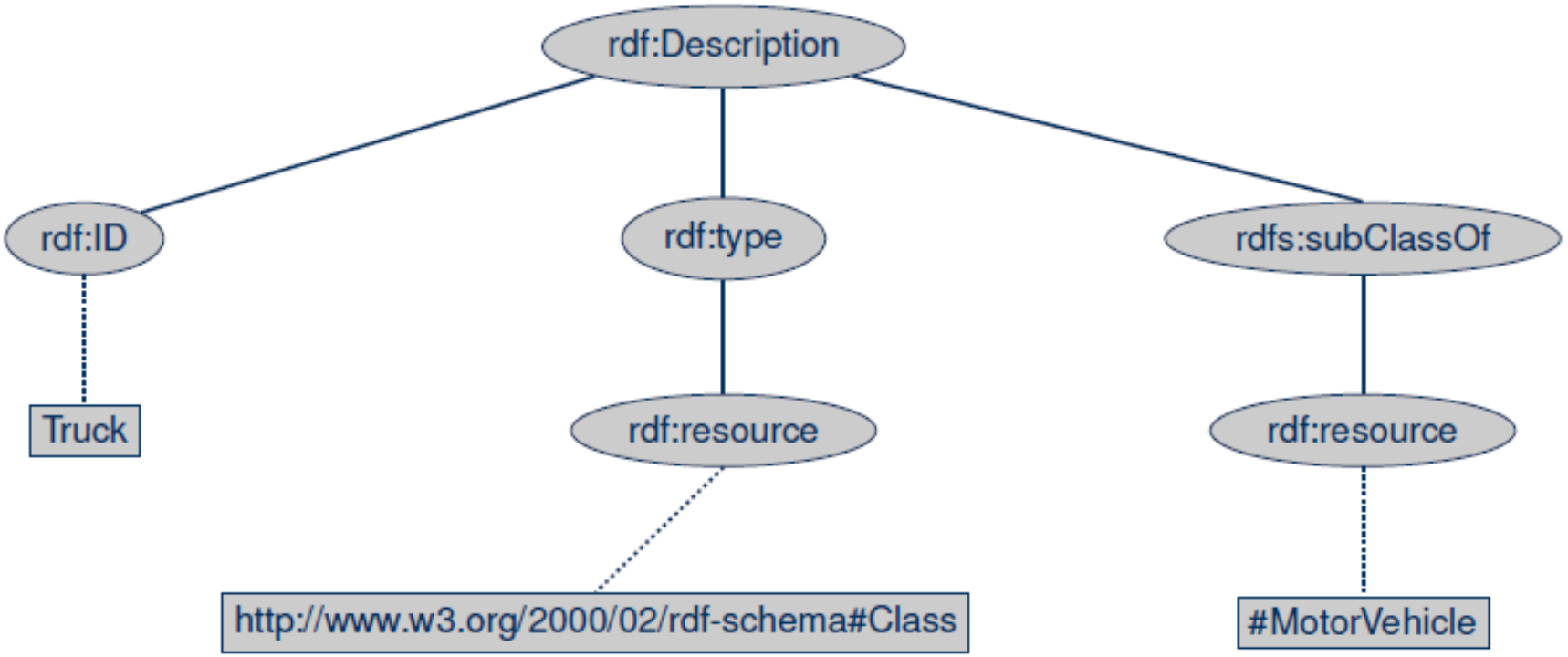
```
ex:vegetableThaiCurry  ex:thaiDishBasedOn    ex:coconutMilk .
ex:sebastian           rdf:type              ex:AllergicToNuts .
ex:sebastian           ex:eats               ex:vegetableThaiCurry .
ex:AllergicToNuts      rdfs:subClassOf       ex:Pitiable .
ex:thaiDishBasedOn     rdfs:domain           ex:Thai .
ex:thaiDishBasedOn     rdfs:range            ex:Nutty .
ex:thaiDishBasedOn     rdfs:subPropertyOf    ex:hasIngredient .
ex:hasIngredient       rdf:type    rdfs:ContainerMembershipProperty.
```



1 Document - 3 Interpretations

```
<rdf:Description rdf:ID="Truck">
  <rdf:type rdf:resource=
    "http://http://www.w3.org/2000/02/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
```

Interpretation as XML:



1 Document - 3 Interpretations

```
<rdf:Description rdf:ID="Truck">
  <rdf:type rdf:resource=
    "http://http://www.w3.org/2000/02/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
```

Interpretation as RDF:

- another data model
- `rdf:Description`, `rdf:ID` and `rdf:resource` have a fixed meaning

subject	predicate	object
`#Truck`	`rdf:type`	`rdfs:Class`
`#Truck`	`rdfs:subClassOf`	`#Motorvehicle`

1 Document - 3 Interpretations

```
<rdf:Description rdf:ID="Truck">
  <rdf:type rdf:resource=
    "http://http://www.w3.org/2000/02/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
```

Interpretation as RDF Schema:

- yet another data model
- `rdf:type` and `rdf:subClassOf` have a specific interpretation