

Sémantický web

Petr Křemen

Katedra kybernetiky, FEL ČVUT

Co nás čeká

- 1 Úvod
- 2 RDF(S)
- 3 OWL

Jak vypadá Web dnes ?

- semistrukturovaná HTML nebo XML data. Existuje velké množství vyhledávacích enginů jako Google, Yahoo, MSN, atd. Přestože již dnes je jejich význam obrovský, odpovědi většinou obsahují obrovské množství irelevantních výsledků.

Jak vypadá Web dnes ?

- semistrukturovaná HTML nebo XML data. Existuje velké množství vyhledávacích enginů jako Google, Yahoo, MSN, atd. Přestože již dnes je jejich význam obrovský, odpovědi většinou obsahují obrovské množství irelevantních výsledků.
- Jak tento problém řešit ?

Jak vypadá Web dnes ?

- semistrukturovaná HTML nebo XML data. Existuje velké množství vyhledávacích enginů jako Google, Yahoo, MSN, atd. Přestože již dnes je jejich význam obrovský, odpovědi většinou obsahují obrovské množství irelevantních výsledků.
- Jak tento problém řešit ?
 - jsou třeba expresivnější datové struktury pro ukládání dat než XML – např. v současné době RDF(S) a OWL.

Jak vypadá Web dnes ?

- semistrukturovaná HTML nebo XML data. Existuje velké množství vyhledávacích enginů jako Google, Yahoo, MSN, atd. Přestože již dnes je jejich význam obrovský, odpovědi většinou obsahují obrovské množství irelevantních výsledků.
- Jak tento problém řešit ?
 - jsou třeba expresivnější datové struktury pro ukládání dat než XML – např. v současné době RDF(S) a OWL.
 - je třeba změnit vyhledávací algoritmy, aby uměly s těmito strukturami zacházet – vývoj inferenčních technik (vycházejících z inference v deskripčních logikách).

Jak vypadá Web dnes ?

- semistrukturovaná HTML nebo XML data. Existuje velké množství vyhledávacích enginů jako Google, Yahoo, MSN, atd. Přestože již dnes je jejich význam obrovský, odpovědi většinou obsahují obrovské množství irelevantních výsledků.
- Jak tento problém řešit ?
 - jsou třeba expresivnější datové struktury pro ukládání dat než XML – např. v současné době RDF(S) a OWL.
 - je třeba změnit vyhledávací algoritmy, aby uměly s těmito strukturami zacházet – vývoj inferenčních technik (vycházejících z inference v deskripčních logikách).
 - zkvalitnit rozhraní vyhledávacích strojů – vývoj expresivnějších dotazovacích jazyků, např. RDQL, SPARQL, atd.

Jak vypadá Web dnes ?

- semistrukturovaná HTML nebo XML data. Existuje velké množství vyhledávacích enginů jako Google, Yahoo, MSN, atd. Přestože již dnes je jejich význam obrovský, odpovědi většinou obsahují obrovské množství irelevantních výsledků.
- Jak tento problém řešit ?
 - jsou třeba expresivnější datové struktury pro ukládání dat než XML – např. v současné době RDF(S) a OWL.
 - je třeba změnit vyhledávací algoritmy, aby uměly s těmito strukturami zacházet – vývoj inferenčních technik (vycházejících z inference v deskripčních logikách).
 - zkvalitnit rozhraní vyhledávacích strojů – vývoj expresivnějších dotazovacích jazyků, např. RDQL, SPARQL, atd.
- Na druhou stranu, čím expresivnější jazyk používáme, tím inteligentnější nástroje na tvorbu znalostních bází v tomto jazyce potřebujeme.

Myšlenka sémantického webu

- komunita W3C - <http://www.w3.org/2001/sw>

Myšlenka sémantického webu

- komunita W3C - <http://www.w3.org/2001/sw>
- formát dat bude buď RDF(S), nebo OWL,

Myšlenka sémantického webu

- komunita W3C - <http://www.w3.org/2001/sw>
- formát dat bude buď RDF(S), nebo OWL,
- Inferenční stroje pro RDF(S) bude moci použít pro (alespoň částečné odvozování) v OWL,

Myšlenka sémantického webu

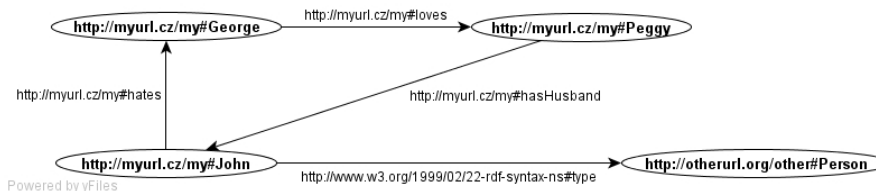
- komunita W3C - <http://www.w3.org/2001/sw>
- formát dat bude buď RDF(S), nebo OWL,
- Inferenční stroje pro RDF(S) bude moci použít pro (alespoň částečné odvozování) v OWL,
- Inferenční stroje pro OWL bude moci použít pro částečné odvozování v RDF(S).

RDF

- <http://www.w3c.org/TR/rdf-primer>

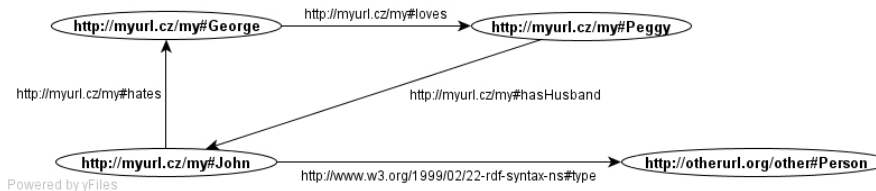
RDF

- <http://www.w3c.org/TR/rdf-primer>
- RDF je graf :



RDF

- <http://www.w3c.org/TR/rdf-primer>
- RDF je graf :



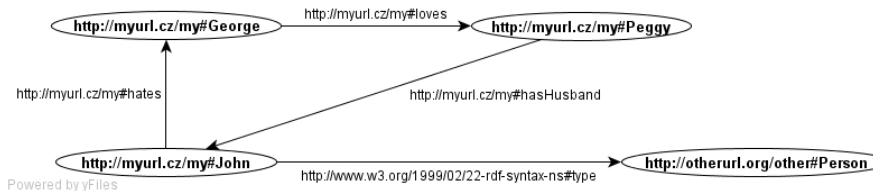
- N3 syntax :

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix my: <http://www.myurl.cz/my#> .
@prefix other: <http://www.otherurl.org/other#> .
my:George my:loves my:Peggy
my:Peggy my:hasHusband my:John
my:John my:hates my:George
my:John rdf:type other:Person
  
```

RDF

- <http://www.w3.org/TR/rdf-primer>
- RDF je graf :



- RDF/XML syntax :

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:base="http://myurl.cz/my"
  xmlns:my="http://myurl.cz/my">
  <rdf:Description rdf:ID="George">
    <my:loves rdf:about="http://myurl.cz/my#Peggy"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Peggy">
    <my:hasHusband rdf:about="http://myurl.cz/my#John"/>
  </rdf:Description>
  <rdf:Description rdf:ID="John">
    <rdf:type rdf:about="http://otherurl.org/other#Person"/>
  </rdf:Description>
</rdf:RDF>
```


RDF (2)

- ... "vše je zdroj (resource)" – nerozlišují se třídy a instance

RDF (2)

- ... "vše je zdroj (resource)" – nerozlišují se třídy a instance
- je množina trojic (hran grafu) ve tvaru S (subjekt) – P (predikát) – O (objekt)

RDF (2)

- ... "vše je zdroj (resource)" – nerozlišují se třídy a instance
- je množina trojic (hran grafu) ve tvaru S (subjekt) – P (predikát) – O (objekt)
- uzel v grafu je buď (1) zdroj (ovál s labelem), nebo (2) literál (obdélník), nebo (3) prázdný uzel (prázdný ovál)

RDF (2)

- ... "vše je zdroj (resource)" – nerozlišují se třídy a instance
- je množina trojic (hran grafu) ve tvaru S (subjekt) – P (predikát) – O (objekt)
- uzel v grafu je buď (1) zdroj (ovál s labelem), nebo (2) literál (obdélník), nebo (3) prázdný uzel (prázdný ovál)
- namespace <http://www.w3.org/1999/02/22-rdf-syntax-ns> reprezentuje slovník RDF (dále jen *rdf*). Tento slovník definuje zdroje jako :rdf:ID, rdf:about, rdf:Description, rdf:Property a mnoho dalších.

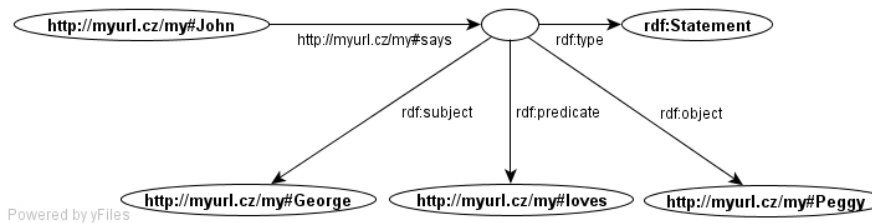
Prázdné uzly (blank nodes)

blank nodes označují existenciálně kvantifikované proměnné (). Využívají se pro :

Prázdné uzly (blank nodes)

blank nodes označují existenciálně kvantifikované proměnné (). Využívají se pro :

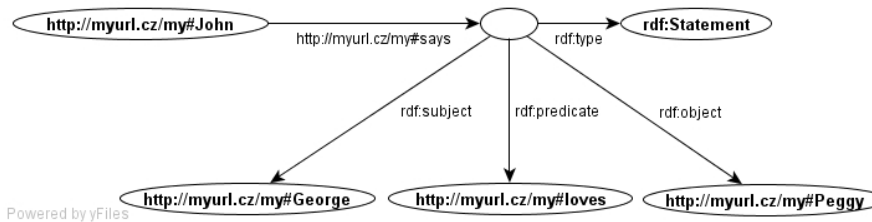
- popis tvrzení vyšších řádů



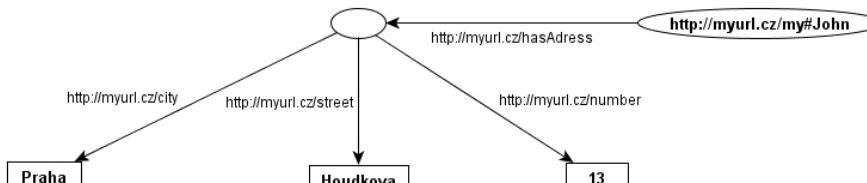
Prázdné uzly (blank nodes)

blank nodes označují existenciálně kvantifikované proměnné (). Využívají se pro :

- popis tvrzení vyšších řádů



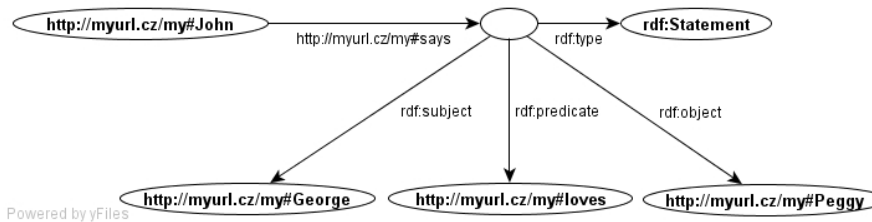
- vyjádření komplexních hodnot



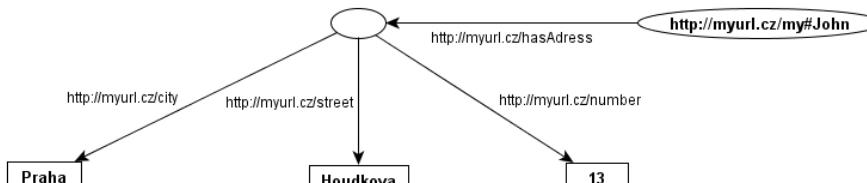
Prázdné uzly (blank nodes)

blank nodes označují existenciálně kvantifikované proměnné (). Využívají se pro :

- popis tvrzení vyšších řádů



- vyjádření komplexních hodnot



RDF kontejnery

- `rdf:Bag` označuje neuspořádanou množinu opakujících se prvků (multimnožina)

RDF kontejnery

- `rdf:Bag` označuje neuspořádanou množinu opakujících se prvků (multimnožina)
- `rdf:Seq` označuje uspořádanou n-tici - sekvenci

RDF kontejnery

- `rdf:Bag` označuje neuspořádanou množinu opakujících se prvků (multimnožina)
- `rdf:Seq` označuje uspořádanou n-tici - sekvenci
- `rdf:Alt` označuje výběr z daných zdrojů/literálů

RDF kontejnery

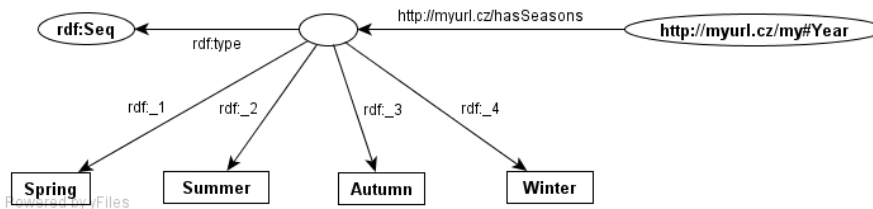
- `rdf:Bag` označuje neuspořádanou množinu opakujících se prvků (multimnožina)
- `rdf:Seq` označuje uspořádanou n-tici - sekvenci
- `rdf:Alt` označuje výběr z daných zdrojů/literálů
- Prvky kontejnerů lze adresovat pomocí vlastností `rdf:x`, kde 'x', je kladné číslo.

RDF kontejnery

- `rdf:Bag` označuje neuspořádanou množinu opakujících se prvků (multimnožina)
- `rdf:Seq` označuje uspořádanou n-tici - sekvenci
- `rdf:Alt` označuje výběr z daných zdrojů/literálů
- Prvky kontejnerů lze adresovat pomocí vlastností `rdf:x`, kde 'x', je kladné číslo.
- Kontejnery nejsou uzavřené – prvky, o kterých tvrdíme, že patří do daného kontejneru nemusí být jediné.

RDF kontejnery

- `rdf:Bag` označuje neuspořádanou množinu opakujících se prvků (multimnožina)
- `rdf:Seq` označuje uspořádanou n-tici - sekvenci
- `rdf:Alt` označuje výběr z daných zdrojů/literálů
- Prvky kontejnerů lze adresovat pomocí vlastností `rdf:x`, kde 'x', je kladné číslo.
- Kontejnery nejsou uzavřené – prvky, o kterých tvrdíme, že patří do daného kontejneru nemusí být jediné.
- Kontejnery se modelují pomocí prázdných uzlů

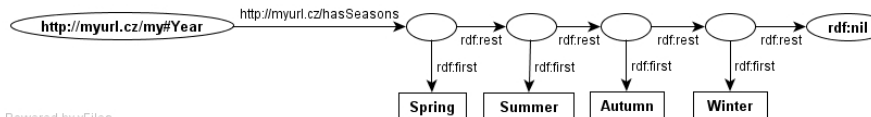


RDF kolekce

- představují "uzavíratelné" kontejnery, podobně jako seznamy v LISPu/Prologu

RDF kolekce

- představují "uzavíratelné" kontejnery, podobně jako seznamy v LISPu/Prologu
- `rdf:List` reprezentuje seznam; hlava seznamu se adresuje pomocí vlastnosti `rdf:first` a tělo se adresuje pomocí vlastnosti `rdf:rest`. Uzavření seznamu lze provést pomocí speciálního seznamu `rdf:nil`.



Powered by yFiles

RDFS

RDFS je jednoduchý ontologický jazyk (rdfs je zkratka pro <http://www.w3.org/2000/01/rdf-schema>), umožňující:

RDFS

RDFS je jednoduchý ontologický jazyk (rdfs je zkratka pro <http://www.w3.org/2000/01/rdf-schema>), umožňující:

- definovat třídy (třída `rdfs:Class`) :
`<rdfs:Class rdf:ID="Person"/>`

RDFS

RDFS je jednoduchý ontologický jazyk (rdfs je zkratka pro <http://www.w3.org/2000/01/rdf-schema>), umožňující:

- definovat třídy (třída rdfs:Class) :

```
<rdfs:Class rdf:ID="Person"/>
```

- vytvářet hierarchie tříd (vlastnost rdfs:subClassOf) :

```
<rdfs:Class rdf:ID="LivingPerson">
```

```
  <rdfs:subClassOf rdf:resource="#Person">
```

```
</rdfs:Class>
```

RDFS

RDFS je jednoduchý ontologický jazyk (rdfs je zkratka pro <http://www.w3.org/2000/01/rdf-schema>), umožňující:

- definovat třídy (třída rdfs:Class) :

```
<rdfs:Class rdf:ID="Person"/>
```

- vytvářet hierarchie tříd (vlastnost rdfs:subClassOf) :

```
<rdfs:Class rdf:ID="LivingPerson">
  <rdfs:subClassOf rdf:resource="#Person">
</rdfs:Class>
```

- vícenásobnou dědičnost :

```
<rdfs:Class rdf:ID="LivingPerson">
  <rdfs:subClassOf rdf:resource="#Person">
  <rdfs:subClassOf rdf:resource="#LivingEntity">
</rdfs:Class>
```

RDFS (2)

RDFS je jednoduchý ontologický jazyk (rdfs je zkratka pro <http://www.w3.org/2000/01/rdf-schema>), umožňující:

RDFS (2)

RDFS je jednoduchý ontologický jazyk (rdfs je zkratka pro <http://www.w3.org/2000/01/rdf-schema>), umožňující:

- definovat vlastnosti (resource `rdf:Property`) :

```
<rdf:Property rdf:ID="hasParent"/>
```

RDFS (2)

RDFS je jednoduchý ontologický jazyk (rdfs je zkratka pro <http://www.w3.org/2000/01/rdf-schema>), umožňující:

- definovat vlastnosti (resource `rdf:Property`) :
`<rdf:Property rdf:ID="hasParent"/>`
- vytvářet hierarchie vlastností (vlastnost `rdfs:subPropertyOf`) :
`<rdfs:Property rdf:ID="hasMother">`
`<rdfs:subPropertyOf rdf:resource="#hasParent">`
`</rdfs:Property>`

RDFS (2)

RDFS je jednoduchý ontologický jazyk (rdfs je zkratka pro <http://www.w3.org/2000/01/rdf-schema>), umožňující:

- definovat vlastnosti (resource `rdf:Property`) :
`<rdf:Property rdf:ID="hasParent"/>`
- vytvářet hierarchie vlastností (vlastnost `rdfs:subPropertyOf`) :
`<rdfs:Property rdf:ID="hasMother">`
`<rdfs:subPropertyOf rdf:resource="#hasParent">`
`</rdfs:Property>`
- vícenásobná dědičnost

RDFS (2)

RDFS je jednoduchý ontologický jazyk (rdfs je zkratka pro <http://www.w3.org/2000/01/rdf-schema>), umožňující:

- definovat vlastnosti (resource rdf:Property) :
- vytvářet hierarchie vlastností (vlastnost rdfs:subPropertyOf) :

```
<rdf:Property rdf:ID="hasParent"/>
```

```
<rdfs:Property rdf:ID="hasMother">
```

```
  <rdfs:subPropertyOf rdf:resource="#hasParent">
```

```
</rdfs:Property>
```

- vícenásobná dědičnost
- definovat doménu a obor hodnot dané vlastnosti :

```
<rdfs:Property rdf:ID="hasMother">
```

```
  <rdfs:subPropertyOf rdf:resource="#hasParent">
```

```
  <rdfs:domain rdf:resource="#Person">
```

```
  <rdfs:range rdf:resource="#Woman">
```

```
</rdfs:Property>
```

RDF(S) vs. Topic Maps

- (převzato z <http://www.kosek.cz/xml/2006znalosti/foil40.html>)

-

	TM	SW
<i>Autor</i>	ISO	W3C
<i>Vznik</i>	2000	1999
<i>Základ</i>	rejstříky, tezaury, ...	formální logika, teorie grafů
<i>Určeno především</i>	pro lidi	pro stroje
<i>Řešený problém</i>	zlepšení zpracování informací aplikováním technologií pro repre	

- Proč existují dva standardy?
 - když se o sobě komunity dozvěděly, bylo už příliš pozdě na sjednocení přístupu
 - v mnohém jsou si obě technologie podobné, nicméně některé technické aspekty znesnadňují jejich úplnou integraci
 - ve většině případů lze provádět obousměrnou konverzi dat mezi mapami témat a RDF
 - v opravdu hodně dlouhé perspektivě může dojít ke sloučení technologií

Shrnutí RDF/RDFS

- RDF/XML je syntakticky redundantní – jeden RDF graf lze vyjádřit mnoha ekvivalentními způsoby v RDF/XML.

Shrnutí RDF/RDFS

- RDF/XML je syntakticky redundantní – jeden RDF graf lze vyjádřit mnoha ekvivalentními způsoby v RDF/XML.
- RDF umožňuje typované literály – datové typy XML schématu

Shrnutí RDF/RDFS

- RDF/XML je syntakticky redundantní – jeden RDF graf lze vyjádřit mnoha ekvivalentními způsoby v RDF/XML.
- RDF umožňuje typované literály – datové typy XML schématu
- RDF poskytuje vlastně správu grafů znalostí s využitím prostorů jmen.

Shrnutí RDF/RDFS

- RDF/XML je syntakticky redundantní – jeden RDF graf lze vyjádřit mnoha ekvivalentními způsoby v RDF/XML.
- RDF umožňuje typované literály – datové typy XML schématu
- RDF poskytuje vlastně správu grafů znalostí s využitím prostorů jmen.
- RDFS poskytuje možnosti pro tvorbu anotací (`rdfs:seeAlso`, `rdfs:label`, atd.) a definice datových typů

Shrnutí RDF/RDFS

- RDF/XML je syntakticky redundantní – jeden RDF graf lze vyjádřit mnoha ekvivalentními způsoby v RDF/XML.
- RDF umožňuje typované literály – datové typy XML schématu
- RDF poskytuje vlastně správu grafů znalostí s využitím prostorů jmen.
- RDFS poskytuje možnosti pro tvorbu anotací (`rdfs:seeAlso`, `rdfs:label`, atd.) a definice datových typů
- RDF + RDFS = RDF(S), což představuje jednoduchou ontologickou platformu pro sémantický web.

Shrnutí RDF/RDFS

- RDF/XML je syntakticky redundantní – jeden RDF graf lze vyjádřit mnoha ekvivalentními způsoby v RDF/XML.
- RDF umožňuje typované literály – datové typy XML schématu
- RDF poskytuje vlastně správu grafů znalostí s využitím prostorů jmen.
- RDFS poskytuje možnosti pro tvorbu anotací (`rdfs:seeAlso`, `rdfs:label`, atd.) a definice datových typů
- RDF + RDFS = RDF(S), což představuje jednoduchou ontologickou platformu pro sémantický web.
- RDFS není podmnožinou FOL (např. `rdfs:Class rdfs:type rdfs:Class`).

Shrnutí RDF/RDFS

- RDF/XML je syntakticky redundantní – jeden RDF graf lze vyjádřit mnoha ekvivalentními způsoby v RDF/XML.
- RDF umožňuje typované literály – datové typy XML schématu
- RDF poskytuje vlastně správu grafů znalostí s využitím prostorů jmen.
- RDFS poskytuje možnosti pro tvorbu anotací (`rdfs:seeAlso`, `rdfs:label`, atd.) a definice datových typů
- RDF + RDFS = RDF(S), což představuje jednoduchou ontologickou platformu pro sémantický web.
- RDFS není podmnožinou FOL (např. `rdfs:Class rdfs:type rdfs:Class`).
- RSS (RDF site summary) je známou aplikací RDF schématu, které se využívá pro přehled novinových zpráv, vyhledávacích výsledků, apod.

OWL



Powered by yFiles

OWL (2)

- **OWL-Lite** je syntaktickou variantou $\mathcal{ALC}_{trans}\mathcal{HIF}(\mathcal{D})$

OWL (2)

- **OWL-Lite** je syntaktickou variantou $\mathcal{ALC}_{trans}\mathcal{HIF}(\mathcal{D})$
- **OWL-DL** je syntaktickou variantou $\mathcal{ALC}_{trans}\mathcal{HOIN}(\mathcal{D})$

OWL (2)

- **OWL-Lite** je syntaktickou variantou $\mathcal{ALC}_{trans}\mathcal{HIF}(\mathcal{D})$
- **OWL-DL** je syntaktickou variantou $\mathcal{ALC}_{trans}\mathcal{HOIN}(\mathcal{D})$
- **OWL-Full** umožňuje konstrukty vyšších řádů a nelze jej tudíž přeložit do FOL.

OWL (2)

- **OWL-Lite** je syntaktickou variantou $\mathcal{ALC}_{trans}\mathcal{HIF}(\mathcal{D})$
- **OWL-DL** je syntaktickou variantou $\mathcal{ALC}_{trans}\mathcal{HOIN}(\mathcal{D})$
- **OWL-Full** umožňuje konstrukty vyšších řádů a nelze jej tudíž přeložit do FOL.
- slovník OWL je definován na <http://www.w3.org/2002/07/owl> (budeme zkracovat na "owl").

OWL (2)

- **OWL-Lite** je syntaktickou variantou $\mathcal{ALC}_{trans}\mathcal{HIF}(\mathcal{D})$
- **OWL-DL** je syntaktickou variantou $\mathcal{ALC}_{trans}\mathcal{HOIN}(\mathcal{D})$
- **OWL-Full** umožňuje konstrukty vyšších řádů a nelze jej tudíž přeložit do FOL.
- slovník OWL je definován na <http://www.w3.org/2002/07/owl> (budeme zkracovat na "owl").
- všechny varianty OWL jsou vybudovány nad RDF(S), využívající dědičnost, anotace, aj.

OWL (3)

- Každá OWL ontologie začíná s běžnou RDF hlavičkou, na kterou navazuje specifikace ontologie :

```
<owl:Ontology rdf:about="">
  <owl:imports
    rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
  <owl:versionInfo
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    version 1.2
  </owl:versionInfo>
  <rdfs:comment xml:lang="en">Some comment</rdfs:comment>
</owl:Ontology>
```


OWL (3)

- Každá OWL ontologie začíná s běžnou RDF hlavičkou, na kterou navazuje specifikace ontologie :

```

<owl:Ontology rdf:about="">
  <owl:imports
    rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
  <owl:versionInfo
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    version 1.2
  </owl:versionInfo>
  <rdfs:comment xml:lang="en">Some comment</rdfs:comment>
</owl:Ontology>

```

- `owl:imports` umožňuje importovat jinou ontologii.

OWL (4)

- OWL představuje vlastní definici třídy `owl:Class`, která je podtřídou `rdfs:Class`.

OWL (4)

- OWL představuje vlastní definici třídy owl:Class, která je podtřídou rdfs:Class.
- většina konstruktů tříd/vlastností je analogická odpovídajícím konstruktům deskripční logiky, např. axiom $Mother \equiv Woman \sqcap \exists hasChild.Person$ lze zapsat jako :

```

<owl:Class rdf:about="Mother">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Woman"/>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#hasChild"/>
          <owl:someValuesFrom>
            <owl:Class rdf:about="#Person"/>
          </owl:someValuesFrom>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```