

# Organizační úvod

*Poznámka* (Organizační úvod)

Nahrávky budou. (Z minulého roku anglicky, z letoška česky.)

## 1 Úvod

### Definice 1.1 (Strojově čitelný soubor)

Strojově čitelný soubor je vlastnost konkrétního souboru, ne formátu (jelikož do formátu můžu nacist data v jiném formátu).

Strojová čitelnost se špatně definuje.

### Definice 1.2 (Binární soubor)

Binární soubor je takový, kde je struktura popsána na úrovni bitů (bit po bitu). Není čitelný textovými editory.

TODO!!!

## 2 RDF

### Definice 2.1 (RDF – resource description framework)

RDF je formát popisu grafu, kde se každé tvrzení (tedy trojice) má tvar „subjekt predikát objekt“, tj. „kdo co s-čím“. Vše se identifikuje pomocí IRI odkazující na definici (nebo v případě některých objektů (často Stringů/čísel/datumů) – literálem).

*Poznámka*

Uri budeme často zkracovat (takové zkrácení se zapisuje jako např. `@prefix dcterms: https://...`).  
Obecné zkratky lze najít na `prefix.cc`.

### Definice 2.2 (Literál)

Literál má dvě části – text odpovídající formátu a uri na ?XML schéma toho typu. Nebo je tvaru `"text"@jazyk`.

*Například*

Nejčastější predikát je `rdf:type` – „je typu“.

### Definice 2.3 (Blank node)

Existují i nepojmenované uzly.

### Definice 2.4 (RDF serializace)

(Jak zapsat RDF do textu.)

- RDF 1.1 N-Triples = každá trojice se zapíše jako `<uri> <uri> <uri> . # comment.`
- RDF 1.1 Turtle = použijí se prefixy, středníky na shodný subjekt a čárku na shodný subjekt i predikát + se používají relativní IRI (base se definuje pomocí `@base` IRI, implicitní je URL dokumentu) + multiline stringy a odescapevané znaky + `rdf:type` má zkratku `a` + blank nody se píší pomocí hranatých závorek + běžné literály nemusí mít typ.
- RDF 1.1 N-Quads = místo trojice se kóduje i pojmenování grafu.
- RDF Trig = Turtle + pojmenované grafy (jsou reprezentovány jako bloky).

### Definice 2.5 (Reifikace)

Pokud chci něco říct o naší trojici, můžu to udělat tak, že si definuji (zase pomocí trojic) objekt, který jako subjekt bude mít subjekt, atd. a navíc bude mít doplňující informace. Tato metoda se nazývá reifikace.

### Definice 2.6 (Pojmenovaný graf, dataset)

Vztahy lze seskupit do tzv. pojmenovaného grafu.

Pojmenované grafy + defaultní graf se nazývá dataset.

### Definice 2.7 (RDFS)

Nadstavba RDF, které umožňuje definovat třídy a dědičnost. `rdfs:Class`, `rdfs:subClassOf`, `rdf:Property`, `rdfs:range`, `rdfs:domain`, `rdfs:subPropertyOf`.

Oproti OOP není třeba definovat třídy, lze definovat property jako takové.

Také umožňuje label, comment, seeAlso: `rdfs:label`, `rdfs:comment`, `rdfs:seeAlso`, `rdfs:isDefinedBy`?

### Definice 2.8 (rdf:List a jiné kolekce)

Ve specifikaci RDF je přímo definován spojový seznam (`rdf:List` + anonymní prvky + `rdf:nil`).

`rdf:_i`, kde *i* je libovolné číslo jsou predikáty náležení do kolekce (`rdf:TODO`).

### Definice 2.9 (Open World Assumption (OWA))

Tvrzení může být pravdivé, i když to nevíme. (Tj. máme i odpověď nevím.)

TODO!!!

### Definice 2.10 (Otevřená data 5 hvězdičková klasifikace dat)

První hvězdička je za uvedenou licenci, druhá je za strojovou čitelnost, třetí je za otevřený formát, čtvrtá za URI odkazy, pátá za připojení do systému LOD.

## 3 SPARQL

### Definice 3.1 (SPARQL)

SPARQL je dotazovací jazyk nad daty v RDF. SPARQL endpoint je HTTP služba pro dotazování v SPARQL na daných open datech.

*Poznámka*

Doporučovaný user formulář je yasgui.

Funguje tak, že se píše RDF trojice s ?nazevproměnné v místě, kde chceme něco doplnit (a zjistit, co to je). To jsou tzv. datové vzory.

Výsledkem je pak tabulka řešení, kde je v každém řádku jeden match a v každém sloupci jedna proměnná, v políčkách je tam pak doplněno.

Do dotazu lze připsat `OPTIONAL` a výsledek pak bude matchovat, i když tato část bude chybět a v tabulce pak bude `NOT BOUND`. Také lze přidat `FILTER` pro podmínky s proměnnými.

Oproti SQL máme ještě RDF operátory: `bound`, `isIri`, `isBlank`, `isLiteral` a přístup k literálu: `str`, `language`, `typeof`?

Taktéž fungují / jako v cestě k souboru, která se navíc zadává Regexem.

Jedním dotazem se můžeme ptát na více SPARQL endpointů, což uděláme pomocí příkazu `SERVICE`.

TODO!!!

## 4 Nejčastější slovníky RDF

#### **Definice 4.1** (Dublin Core metadata)

Jeden z prvních slovníků, vznikl na popis knih (a dalších děl). Jsou to pojmy se zkratkou dcterms (Dublin Core Metadata Initiative).

#### **Definice 4.2** (skos)

Konceptuální slovník. Důležité jsou např. `skos:prefLabel`, `skos:altLabel`, `skos:hiddenLabel`. Dále třeba `notation` a různé typy `skos:semanticRelation`.

#### **Definice 4.3** (GoodRelations)

Slovník pro e-komerci.

#### **Definice 4.4** (Schema.org)

Založen firmami Google, Microsofte, Yahoo a Yandex. Integruje existující slovníky. Určeno pro jednoduchou anotaci webových stránek, ne k dobré strukturalizaci.

#### **Definice 4.5** (Wikidata)

Komunitní RDF data. Má k sobě také slovník. Běží na softwaru Wikibase.

TODO!!!

## 5 Hierarchické datové formáty

#### **Definice 5.1** (Dokumentově orientované XML)

Dokument, do kterého se vloží značky (tj. bez značek je stále čitelný).

#### **Definice 5.2** (Datově orientované XML)

To jsou pouze data se značkami (tj. bez značek je „nečitelný“).

#### **Definice 5.3** (XML 1.0 a XML 1.1)

1.0 má list povolených znaků, 1.1 zakázaných. Aplikace ale zamrzly u 1.0.

TODO syntaxe XML

TODO!!!