

SPARQL – Entailment Regimes

Dedukčné režimy

Spracované podľa prednášok

- <http://www.inf.tu-dresden.de/content/institutes/ki/cl/study/summer14/fswt/slides/FSWT2014-L17-SPARQL-Entailment.pdf>

a dokumentov W3C

- <http://www.w3.org/TR/sparql11-entailment/>

Dalšie zdroje

Motivačný príklad

Data

ex:JozefM

ex:prednasa

ex:RZZ.

ex:prednasa

rdfs:domain

ex:Docent .

ex:Docent

rdfs:subClassOf

ex:Osoba .

Dotaz

PREFIX ex: <http://example.org/>

SELECT ?x WHERE { ?x a ex:Osoba }

Odpoveď

Žiadne riešenia

Prečo? *Keďže riešenia zobrazujú vzor na podgrafy G , tento spôsob vyhodnocovania dotazu nazýva sa **subgraph matching**. Z iterpolačnej lemy vyplýva, že riešenia sú teda **len jednoduché dôsledky** dotazovaného grafu odpovedajúce vzoru dotazu.*

Fuseki pracuje v **režime jednoduchej dedukcie** (*pokiaľ nie je inak špeciálne nakonfigurovaný*).

Motivačný príklad – 2. pokus

Fuseki server umožňuje nakonfigurovať servis, ktorý pracuje so svojou dátovou sadou vo zvolenom inferenčnom režime.

Pozn. Konfiguračný súbor pre dátovú sadu je potrebné vložiť do podadresára.

Detaily: <https://jena.apache.org/documentation/fuseki2/fuseki-configuration.html>

Po načítaní dát do takejto sady

```
ex:JozefM      ex:prednasa      ex:RZZ.
ex:prednasa    rdfs:domain      ex:Docent .
ex:Docent      rdfs:subClassOf  ex:Osoba .
```

dotaz

```
PREFIX ex: <http://eg.org/>
SELECT ?x WHERE { ?x a ex:Osoba }
```

vráti odpoveď

```
-----
| x           |
=====
| ex:JozefM  |
-----
```

Entailment relation – relácia dôsledku

Obecne môžeme definovať pojem dedukčný režim

SPARQL entailment regime (dedukčný režim) definuje:

1. Množinu prípustných RDF grafov (well-formed graphs) pre daný režim.
2. Binárnu **reláciu dôsledku** na množine prípustných grafov, ktorá stanovuje či/kedy je jeden graf dôsledkom druhého grafu v tomto dedukčnom režime.

Príklady.

Dedukčný režim pre jednoduchú interpretáciu by sme mohli teda definovať nasledovne:

1. Všetky RDF grafy sú prípustné.
2. Grafy E je v relácii s grafom G práve vtedy keď E je **jednoduchým dôsledkom** G.

Podobne **dedukčný režim pre RDFS interpretáciu** by sme mohli definovať:

1. Všetky RDF grafy sú prípustné.
2. Grafy E je v relácii s grafom G práve vtedy keď E je **RDFS dôsledkom** G.

Tento prístup má však ešte viaceré nedostatky ako ilustrujú nasledujúce príklady.

Logický dôsledok - problémy

Príklad1.

Uvažujme graf

G: ex:o ex:p ex:s

a SPQRQL dotaz (v jednoduchom režime)

ex:o ex:p ?x

Jednoduché dôsledky grafu G sú, okrem neho samotného (triviálny dôsledok) aj ďalšie grafy:

D1: ex:o ex:p _:b1

D2: ex:o ex:p _:b2

a.t.d'

Riešenia D1, D2 sú **logické dôsledky** riešenia G. Dotaz by mal vrátiť len riešenie G, jeho dôsledky netreba vracat' ako iné riešenia.

Ďalšia prirodzená požiadavka je, aby množina riešení bola **konečná**.

Príklad2.

Uvažujme v RDFS interpretácii dotaz

?x rdf:type rdf:Property

Keďže v RDFS interpretácii pre každé celé číslo n platí RDFS-axióma:

rdf:_n rdf:type rdf:Property

má uvedený dotaz nekonečne veľa riešení pre akýkoľvek dotazovaný graf.

SPARQL entailment regime

Okrem definície relácie dôsledku musí dedukčný režim stanoviť ďalšie podmienky, ktoré zabezpečia, že množina riešení je konečná, že neobsahuje zbytočné – redundantné riešenia. Navyše by mal stanoviť aj správanie v ďalších výnimočných situáciách.

SPARQL dedukčný režim definuje:

- **Meno a IRI režimu.** Toto IRI môže vrátiť service description ako hodnotu property [sd:defaultEntailmentRegime](#), ktorá udáva použitý režim. Okrem toho môže mať každý pomenovaný graf vlastný režim daný property [sd:entailmentRegime](#).
- **Prípustné grafy** pre daný režim .
- **Prípustné dotazy** pre daný režim .
- **Reláciu dôsledku:** Binárnu **reláciu** na množine prípustných grafov, ktorá stanovuje či/kedy je jeden graf dôsledkom druhého grafu v danom dedukčnom režime.
- **Pravidlá pre odpovede na dotazy:** Teda pravidlá jednoznačne určujúce, ktoré dôsledky budú tvoriť množinu riešení pre daný dotaz. Tieto pravidlá sú potrebné napr. na vylúčenie redundantných riešení či zabezpečenie konečnosti množiny riešení.
- **Error handling:** Správanie v prípade chyby, neprípustného dotazu, dotazu na graf, ktorý je neprípustný alebo nekonzistentný v danom režime.

Keďže exaktná definícia konkrétnych režimov, najmä pravidiel pre odpovede, je pomerne komplexná, nebudeme ju uvádzať. Detailnejší popis pre RDFS režim nájdete v dokumentoch TUD a kompletnú definíciu podávajú štandardy.

Implementácia dedukčných režimov

Nech máme daný dedukčný režim R , graf G a jednoduchý vzor P .

Ako nájsť všetky dôsledky grafu G v dedukčnom režime R , ktoré odpovedajú vzoru P ?

- materializácia / forward-chaining
- spätné zreťazenie / backward-chaining
- hybridné stratégie

Materializácia

Data

```
ex:JozefM      ex:prednasa      ex:RZZ.  
ex:prednasa   rdfs:domain     ex:Docent .  
ex:Docent     rdfs:subClassOf ex:Osoba .
```

Dotaz

```
SELECT ?x WHERE { ?x a ex:Osoba }
```

Dotazovaný graf postupne dopĺňame s využitím logických pravidiel o trojice, ktoré sú jeho logickými dôsledkami v danom dedukčnom režime.

Materializácia

Data

```
ex:JozefM      ex:prednasa      ex:RZZ .
ex:prednasa    rdfs:domain      ex:Docent .
ex:Docent      rdfs:subClassOf  ex:Osoba .
ex:JozefM      rdf:type      ex:Docent .
```

Dotaz

```
SELECT ?x WHERE { ?x a ex:Osoba }
```

Dedukčné pravidlo

```
rdfs2 ppp rdfs:domain xxx .
      uuu ppp vvv .
-----
      uuu rdf:type xxx .
```

aplikujeme na

ppp → ex:prednasa

xxx → ex:Docent

uuu → ex:JozefM

a dostaneme dôsledok:

```
ex:JozefM rdf:type ex:Docent .
```

Materializácia

Data

```
ex:JozefM      ex:prednasa      ex:RZZ.  
ex:prednasa    rdfs:domain      ex:Docent .  
ex:Docent     rdfs:subClassOf ex:Osoba .  
ex:JozefM     rdf:type        ex:Docent .  
ex:JozefM     rdf:type        ex:Osoba .
```

Dotaz

```
SELECT ?x WHERE { ?x a ex:Osoba }
```

Dedukčné pravidlo

```
rdfs9 xxx rdfs:subClassOf yyy .  
zzz rdf:type xxx .
```

```
zzz rdf:type yyy .
```

aplikujeme na

```
xxx → ex:Docent
```

```
yyy → ex:Osoba
```

```
zzz → ex:JozefM
```

a dostaneme dôsledok:

```
ex:JozefM rdf:type ex:Osoba .
```

Materializácia

Data

```
ex:JozefM      ex:prednasa      ex:RZZ.  
ex:prednasa   rdfs:domain      ex:Docent .  
ex:Docent     rdfs:subClassOf  ex:Osoba .  
ex:JozefM     rdf:type          ex:Docent .  
ex:JozefM    rdf:type        ex:Osoba .
```

Dotaz

```
SELECT ?x WHERE { ?x a ex:Osoba }
```

Riešenie

x → **ex:JozefM**

Nedostatky materializácie

- veľkosť dotazovaného grafu veľmi rastie - veľká pamäťová náročnosť
- po každej zmene dotazovaného grafu treba urobiť znovu materializáciu

Spätné reťazenie

Data

```
ex:JozefM      ex:prednasa      ex:RZZ.  
ex:prednasa   rdfs:domain     ex:Docent .  
ex:Docent     rdfs:subClassOf ex:Osoba .
```

Dotaz

```
SELECT ?x WHERE { ?x a ex:Osoba }
```

Namiesto rozširovania dotazovaného grafu modifikujeme dotaz pridávaním nových alternatív bez toho aby sme zmenili množinu riešení. Postupujeme teda odzadu:

Spätné reťazenie

Data

```
ex:JozefM      ex:prednasa      ex:RZZ.  
ex:prednasa   rdfs:domain     ex:Docent .  
ex:Docent     rdfs:subClassOf  ex:Osoba .
```

Dotaz

```
SELECT ?x WHERE { ?x a ex:Osoba }
```

Dedukčné pravidlo

```
rdfs9 xxx rdfs:subClassOf yyy .
```

```
zzz rdf:type xxx .
```

```
zzz rdf:type yyy .
```

aplikované

```
ex:Docent     rdfs:subClassOf  ex:Osoba .
```

```
?x rdf:type ex:Docent .
```

```
?x rdf:type ex:Osoba .
```

Umožní rozšíriť dotaz na:

```
SELECT ?x WHERE { ?x a ex:Osoba } UNION  
                { ?x a ex:Docent }
```

Spätne reťazenie

Data

```
ex:JozefM      ex:prednasa      ex:RZZ.  
ex:prednasa  rdfs:domain  ex:Docent .  
ex:Docent     rdfs:subClassOf  ex:Osoba .
```

Dotaz

```
SELECT ?x WHERE { ?x a ex:Osoba } UNION  
                { ?x a ex:Docent }
```

Dedukčné pravidlo

```
rdfs2 ppp rdfs:domain xxx .  
uuu ppp vvv .  
-----  
uuu rdf:type xxx .
```

aplikujeme

```
ex:prednasa rdfs:domain ex:Docent .  
?x ex:prednasa _:b .  
-----  
?x rdf:type ex:Docent.
```

Umožní rozšíriť dotaz na:

```
SELECT ?x WHERE { ?x a ex:Osoba } UNION  
                { ?x a ex:Docent } UNION  
                { ?x ex:prednasa _:b }
```

Spätne reťazenie

Data

```
ex:JozefM      ex:prednasa      ex:RZZ.  
ex:prednasa    rdfs:domain      ex:Docent .  
ex:Docent      rdfs:subClassOf  ex:Osoba .
```

Dotaz

```
SELECT ?x WHERE {  
  { ?x a ex:Osoba } UNION  
  { ?x a ex:Docent } UNION  
  { ?x ex:prednasa _:b }
```

Riešenie

3. Alternetiva { ?x ex:prednasa _:b }

poskytne riešenie: **x** → **ex:JozefM**

Nedostatky spätneho reťazenia

- veľmi obtiažne resp. nemožné nájsť všetky riešenia
- spätne reťazenie je potrebné robiť pre každý dotaz – väčšia časová náročnosť