

ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ ΣΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

SPARQL

Ι. Χατζηλυγερούδης

Ερωτήματα SPARQL

- Η υποβολή ερωτημάτων σε έγγραφο RDF με χρήση γλωσσών ερωτημάτων βασισμένων σε XML (π.χ. XPath) είναι προβληματική, λόγω πολλών παραλλαγών αναπαράστασης περιγραφών.
- Χρησιμοποιείται η SPARQL ως το υπό υιοθέτηση πρότυπο από το W3C.

Ερωτήματα SPARQL

■ Βασικά ερωτήματα

- Βασίζονται στην ταύτιση υποδειγμάτων/προτύπων γράφων
- Απλούστερο υπόδειγμα γράφου: υπόδειγμα/πρότυπο τριάδας
- Μοιάζει με τριάδα RDF, αλλά επιτρέπονται μεταβλητές στις θέσεις υποκειμένου, κατηγορήματος ή αντικειμένου
- Ο συνδυασμός υποδειγμάτων τριάδων παράγει ένα βασικό υπόδειγμα γράφου
- Απαιτείται ακριβής ταύτιση με κάποιο γράφο

Ερωτήματα SPARQL

- Βασικά ερωτήματα (συν.)

- Απλό παράδειγμα:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/22-rdf-schema#>
```

```
SELECT ?c
```

```
WHERE
```

```
{
```

```
    ?c rdf:type rdfs:Class .
```

```
}
```

(τα υποδείγματα τριάδων, όπου `rdf:type` είναι η ιδιότητα και `rdfs:Class` το αντικείμενο: ανάκτηση όλων των κλάσεων)

Ερωτήματα SPARQL

- Βασικά ερωτήματα (συν.)

- άλλο παράδειγμα:

```
PREFIX uni: <http://www.mydomain.org/uni-ns# >
```

```
SELECT ?i
```

```
WHERE
```

```
{
```

```
    ?i rdf:type rdfs:course .
```

```
}
```

(ανάκτηση όλων των στιγμιοτύπων της κλάσης course)

Ερωτήματα SPARQL

- Δομή select-from-where

- **SELECT**: καθορίζει τον αριθμό και τη σειρά των προς ανάκτηση δεδομένων
- **FROM**: προσδιορίζει την πηγή των δεδομένων (προαιρετικό)
- **WHERE**: επιβάλλει περιορισμούς στις δυνατές απαντήσεις

Π.χ.

```
SELECT ?x ?y
```

```
WHERE
```

```
{
```

```
    ?x uni:phone ?y .
```

```
}
```

(ανάκτηση όλων των τηλεφώνων των μελών του προσωπικού)

Ερωτήματα SPARQL

```
SELECT ?x ?y
WHERE
{
  ?x rdf:type uni:Lecturer ;
    uni:phone ?y .
}
```



```
SELECT ?x ?y
WHERE
{
  ?x rdf:type uni:Lecturer .
  ?x uni:phone ?y .
}
```

(ανάκτηση όλων των διδασκόντων και των τηλεφώνων τους)

1. Ο όρος `?x rdf:type uni:Lecturer` συγκεντρώνει όλα τα στιγμιότυπα της κλάσης `Lecturer` και συνδέει το αποτέλεσμα με την μεταβλητή `?x`
2. Το `uni:phone ?y` συγκεντρώνει όλες τις τριάδες με κατηγορημα `phone`
3. Η *έμμεση συνένωση* (implicit join) (λόγω του «;») περιορίζει τις τριάδες αυτές σε κείνες με κοινό υποκείμενο με τις πρώτες (`?x`)

Ερωτήματα SPARQL

```
SELECT ?n
WHERE
{
  ?x rdf:type uni:Course ;
    uni:isTaughtBy :949352 .
  ?c uni:name ?n .
  FILTER (?c = ?x) .
}
```

(ανάκτηση όλων των μαθημάτων που διδάσκονται από τον διδάσκοντα με κωδικό (ID) 949352)

Το FILTER χρησιμοποιείται για υπόδειξη ενός λογικού περιορισμού. Εδώ ο περιορισμός είναι η *άμεση συνένωση* (explicit join) των μεταβλητών ?c και ?x (χρήση τελεστή ισότητας «=»).

Ερωτήματα SPARQL

■ Προαιρετικά υποδείγματα

- Μέχρι τώρα επιστρέφεται απάντηση αν υπάρχει πλήρης ταύτιση υποδείγματος στη βάση γνώσης
- Συχνά όμως απαιτείται μεγαλύτερη ευελιξία

Π.χ.

```
<uni:lecturer rdf:about="949352">  
  <uni:name>Grigoris Antoniou</uni:name>  
</uni:lecturer>
```

```
<uni:lecturer rdf:about="949318">  
  <uni:name>John Hatzis</uni:name>  
  <uni:email>ihatz@cti.gr</uni:email>  
</uni:lecturer>
```

Δεν επιστρέφει το όνομα του άλλου λέκτορα επειδή δεν έχει email.

```
SELECT ?name ?email  
WHERE  
{  
  ?x rdf:type uni:Lecturer ;  
  uni:name ?name ;  
  uni:email ?email .  
}
```



?name	?email
John Hatzis	ihatz@cti.gr

Ερωτήματα SPARQL

```
<uni:lecturer rdf:about="949352">  
  <uni:name>Grigoris Antoniou</uni:name>  
</uni:lecturer>
```

```
<uni:lecturer rdf:about="949318">  
  <uni:name>John Hatzis</uni:name>  
  <uni:email>ihatz@cti.gr</uni:email>  
</uni:lecturer>
```

```
SELECT ?name ?email  
WHERE  
{  
  ?x rdf:type uni:Lecturer ;  
    uni:name ?name ;  
  OPTIONAL {?x uni:email ?email }  
}
```



?name	?email
Grigoris Antoniou	
John Hatzis	ihatz@cti.gr