



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Αναπαράσταση Γνώσης στον Παγκόσμιο Ιστό

Ενότητα 3: RDF

Περιγραφή Πόρων Ιστού, Μέρος 3^ο – Σημασιολογία & SPARQL

Ιωάννης Χατζηλυγερούδης

Πολυτεχνική Σχολή

Τμήμα Μηχ/κών Η/Υπολογιστών & Πληροφορικής

Περιεχόμενα ενότητας

Μέρος 3^ο – RDF Schema - SPARQL

1. Σημασιολογία
2. Συλογισμός
3. Ερωτήματα SPARQL
4. Σύνοψη Ενότητας



RDF και RDFS - Σημασιολογία

Σημασιολογία RDF και RDFS

■ Μέσω κατηγορηματικής λογικής με ισότητα

■ Προσέγγιση:

○ Θεμελιώδη στοιχεία \rightarrow σταθερές

Resource, Class, Property, subclassOf κλπ

○ Χρήση θεωρίας λιστών

*nil, cons(x,l), first(l),
rest(l), item(x,l), list(l)*

○ Συνήθης μορφή αξιωμάτων σημασιολογίας:

Type(subClassOf, Property) (Η subclassOf είναι ιδιότητα)

○ Κλ με ισότητα, οι μεταβλητές ξεκινούν με **?**, έμμεση καθολική ποσοτικοποίηση (\forall)



Βασικά κατηγορήματα

- **PropVal** (P, R, V)

(αναπαράσταση πρότασης *RDF*)

- **Type** (R, T)

(αντί του *PropVal* ($type, R, T$): ο πόρος R είναι τύπου T)

Δηλ. $Type(?r, ?t) \leftarrow \rightarrow PropVal(type, ?r, ?t)$



Κλάσεις

- Όλες οι κλάσεις είναι στιγμιότυπα της Class:

Type(Class, Class), Type(Resource, Class)

Type(Property, Class), Type(Literal, Class)

- Η Resource είναι η γενικότερη κλάση → όλες οι κλάσεις και οι ιδιότητες είναι πόροι:

Type(?p, Property) → Type(?p, Resource)

Type(?c, Class) → Type(?c, Resource)

- Το κατηγορημα σε μια πρόταση είναι ιδιότητα

PropVal(?p, ?r, ?c) → Type(?p, Property)



Ιδιότητα *type*

■ Ιδιότητα *type*

ο $Type(type, Property) \equiv PropVal(type, type, Property)$

(Το *type* είναι ιδιότητα)

ο $Type(?r, ?c) \rightarrow (Type(?r, Resource) \wedge Type(?c, Class))$

(Η *type* μπορεί να ισχύει για πόρους και έχει μια κλάση ως τιμή της)



Ιδιότητα *FuncProp*

■ Βοηθητική ιδιότητα *FuncProp*

- Συναρτησιακή ιδιότητα=ιδιότητα που αποτελεί συνάρτηση, δηλ. συσχετίζει ένα πόρο με μια τιμή το πολύ
- Οι συναρτησιακές ιδιότητες δεν αποτελούν έννοια της RDF, αλλά χρησιμοποιούνται για αξιωματοποίηση άλλων θεμελιωδών στοιχείων
- Η σταθερά *FuncProp*: αναπαριστά την κλάση όλων των συναρτησιακών ιδιοτήτων.

$$\text{Type} (?p, \text{FuncProp}) \leftrightarrow (\text{Type} (?p, \text{Property}) \wedge \forall ?r \forall ?v1 \forall ?v2 \\ (\text{PropVal} (?p, ?r, ?v1) \wedge \text{PropVal} (?p, ?r, ?v2) \rightarrow ?v1 \equiv ?v2))$$

Η p είναι μια συναρτησιακή ιδιότητα, αν και μόνο αν είναι ιδιότητα και δεν υπάρχουν $r, v1, v2$ τέτοια ώστε να ισχύουν οι σχέσεις $p(r, v1)$ και $p(r, v2)$ όταν $v1 \neq v2$.



Υποστασιοποιημένες προτάσεις

- Όλες είναι πόροι, η Statement είναι στιγμιότυπο της Class:

```
Type(?s, Statement) → Type(?s, Resource)
Type(Statement, Class)
```

- Μια πρόταση μπορεί να αναλυθεί στα μέρη της τριάδας RDF:

```
Type(?st, Statement) →
∃?p ∃?r ∃?v(PropVal(Predicate, ?st, ?p) ∧
PropVal(Subject, ?st, ?r) ∧ PropVal(Object, ?st, ?v))
```

- Οι ιδιότητες Subject, Predicate και Object είναι συναρτησιακές ιδιότητες (: κάθε πρόταση έχει ακριβώς ένα υποκείμενο, ένα κατηγορημα, ένα αντικείμενο):

```
Type(Subject, FuncProp)
Type(Predicate, FuncProp)
Type(Object, FuncProp)
```



Υποστασιοποιημένες προτάσεις

- Πληροφορίες τυποποίησης:

$\text{PropVal}(\text{Subject}, ?st, ?r) \rightarrow$
 $(\text{Type} (?st, \text{Statement}) \wedge \text{Type} (?r, \text{Resource}))$

$\text{PropVal}(\text{Predicate}, ?st, ?p) \rightarrow$
 $(\text{Type} (?st, \text{Statement}) \wedge \text{Type} (?p, \text{Property}))$

$\text{PropVal}(\text{Object}, ?st, ?v) \rightarrow$
 $(\text{Type} (?st, \text{Statement}) \wedge (\text{Type} (?v, \text{Resource}) \vee$
 $\text{Type} (?v, \text{Literal})))$

(Αν το Object εμφανιστεί ως ιδιότητα σε μια πρόταση RDF, τότε πρέπει να ισχύει για κάποια υποστασιοποιημένη πρόταση και να παίρνει ως τιμή είτε πόρο είτε λεκτικό).



Υποδοχείς

- Όλοι είναι πόροι:

$$\text{Type} (?c, \text{Container}) \rightarrow \text{Type} (?c, \text{Resource})$$

- Είναι λίστες:

$$\text{Type} (?c, \text{Container}) \rightarrow \text{list} (?c)$$

- Οι υποδοχείς είναι πολυσύνολα (bags), ακολουθίες (sequences) ή εναλλακτικά (alternatives):

$$\text{Type} (?c, \text{Container}) \leftrightarrow (\text{Type} (?c, \text{Bag}) \vee \text{Type} (?c, \text{Seq}) \vee \text{Type} (?c, \text{Alt}))$$

- Τα πολυσύνολα και οι ακολουθίες (sequences) είναι ξένα μεταξύ τους:

$$\neg (\text{Type} (?c, \text{Bag}) \wedge \text{Type} (?c, \text{Seq}))$$


Υποκλάσεις και υποϊδιότητες

- Η `subClassOf` είναι ιδιότητα:

`Type(subClassOf, Property)`

- Αν μια κλάση `C` είναι υποκλάση της `C'` τότε όλα τα στιγμιότυπα της `C` είναι και στιγμιότυπα της `C'`:

$$\text{PropVal}(\text{subClassOf}, ?c, ?c') \leftrightarrow$$
$$(\text{Type}(?c, \text{Class}) \wedge \text{Type}(?c', \text{Class}) \wedge$$
$$\forall ?x (\text{Type}(?x, ?c) \rightarrow \text{Type}(?x, ?c')))$$

- Τα ίδια και για την `subPropertyOf`:

`Type(subPropertyOf, Property)`

$$\text{PropVal}(\text{subPropertyOf}, ?p, ?p') \leftrightarrow$$
$$(\text{Type}(?p, \text{Property}) \wedge \text{Type}(?p', \text{Property}) \wedge$$
$$\forall ?r \forall ?v (\text{PropVal}(?p, ?r, ?v) \rightarrow \text{PropVal}(?p', ?r, ?v)))$$


Περιορισμοί

- Κάθε πόρος περιορισμού (constraint resource) είναι πόρος:

```
PropVal(subClassOf, ConstraintResource, Resource)
```

- Οι ιδιότητες περιορισμού (constraint properties) είναι ιδιότητες, που είναι και πόροι περιορισμού:

```
Type(?cp, ConstraintProperty) ↔  
(Type(?cp, ConstraintResource) ∧ (Type(?cp, Property)))
```

- Τα domain και range είναι ιδιότητες περιορισμού:

```
Type(domain, ConstraintProperty)
```

```
Type(range, ConstraintProperty)
```



Περιορισμοί

- Τα domain και range ορίζουν το πεδίο ορισμού και το σύνολο τιμών μιας ιδιότητας αντίστοιχα.
- Πεδίο ορισμού μιας ιδιότητας P: το σύνολο αντικειμένων για τα οποία ισχύει η P. Αν D είναι το πεδίο ορισμού, τότε για κάθε $P(x, y)$, $x \in D$.

$$\text{PropVal}(\text{domain}, ?p, ?d) \rightarrow \forall ?x \forall ?y (\text{PropVal} (?p, ?x, ?y) \rightarrow \text{Type} (?x, ?d))$$

- Πεδίο τιμών μιας ιδιότητας P: το σύνολο όλων των τιμών που μπορεί να πάρει η P. Αν R είναι το σύνολο τιμών, τότε για κάθε $P(x, y)$, $y \in R$.

$$\text{PropVal}(\text{range}, ?p, ?r) \rightarrow \forall ?x \forall ?y (\text{PropVal} (?p, ?x, ?y) \rightarrow \text{Type} (?x, ?r))$$


Περιορισμοί

- Οι παρακάτω τύποι εξάγονται μέσω συμπερασμού από τους προηγούμενους.

`PropVal(domain, range, Property)`

`PropVal(range, range, Property)`

`PropVal(domain, domain, Property)`

`PropVal(range, range, Property)`

Με βάση την παραπάνω τυποποίηση της σημασιολογίας μπορούν να εξαχθούν συμπεράσματα.

Π.χ. αν **domain(acadStaffMem, teaches)**
και **SubClassOf(acadStaffMem, staffMem)**
και ισχύει **teaches(DB, Jhatzis)**
τότε εξάγεται ότι **acadStaff(Jhatzis)** .



Συλλογισμός/Συμπερασμός σε RDF και RDFS

Συλλογισμός/Συμπερασμός

- Με βάση το προηγούμενο αξιωματικό σύστημα ορισμού σημασιολογίας για RDF και RDFS, θα μπορούσε να οριστεί ένα σύστημα αυτόματου συλλογισμού από προτάσεις RDF και RDFS.
Μειονέκτημα: υψηλή πολυπλοκότητα, χαμηλή αποδοτικότητα (λόγω ΚΛΠΤ).
- Υπάρχει ένα σύστημα συλλογισμού που αντιστοιχεί σε μια σημασιολογία RDF, και είναι ακριβές και πλήρες γι' αυτήν, και που λειτουργεί απ' ευθείας σε προτάσεις/τριάδες RDF, χωρίς διαμεσολάβηση ενδιάμεσης μετατροπής σε ΚΛΠΤ.
- Το σύστημα αυτό διαθέτει κανόνες της μορφής:
ΑΝ το E περιέχει συγκεκριμένες τριάδες
ΤΟΤΕ πρόσθεσε στο E συγκεκριμένες επί πλέον τριάδες
(το E είναι ένα σύνολο τριάδων RDF)



Συλλογισμός/Συμπερασμός

■ Βασικά παραδείγματα κανόνων συμπερασμού:

AN το E περιέχει την τριάδα ($?x, ?p, ?y$)

TOTE το E επίσης περιέχει την τριάδα ($?p, \text{rdf:type}, \text{rdf:Property}$)

(οποιοσδήποτε πόρος $?p$ που χρησιμοποιείται στη θέση της ιδιότητας μιας τριάδας μπορεί να θεωρηθεί μέλος της κλάσης `rdf:Property` μέσω συμπερασμού)

AN το E περιέχει τις τριάδες ($?u, \text{rdfs:subClassOf}, ?v$)

και ($?v, \text{rdfs:subClassOf}, ?w$)

TOTE το E επίσης περιέχει την τριάδα ($?u, \text{rdfs:subClassOf}, ?w$)

(μεταβατικότητα της σχέσης της υποκλάσης)



Συλλογισμός/Συμπερασμός

■ Βασικά παραδείγματα κανόνων συμπερασμού:

ΑΝ το E περιέχει τις τριάδες ($?x$, `rdf:type`, $?u$)

και ($?u$, `rdfs:subClassOf`, $?v$)

ΤΟΤΕ το E επίσης περιέχει την τριάδα ($?x$, `rdf:type`, $?v$)

(νόημα της ιδιότητας `rdfs:subClassOf`)

ΑΝ το E περιέχει τις τριάδες ($?x$, $?p$, $?y$) και ($?p$, `rdfs:range`, $?u$)

ΤΟΤΕ το E επίσης περιέχει την τριάδα ($?y$, `rdf:type`, $?u$)

(οποιοσδήποτε πόρος $?y$ που εμφανίζεται ως τιμή της ιδιότητας $?p$ μπορεί να θεωρηθεί μέλος του συνόλου τιμών της $?p$ μέσω συμπερασμού-χρήση ορισμών συνόλου τιμών όχι για περιορισμό, αλλά συμπερασμό των μελών του συνόλου)



Ερωτήματα SPARQL

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 υπουιοθέτηση πρότυπο από το W3C.



Βασικά ερωτήματα SPARQL

- 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



Βασικά ερωτήματα SPARQL

- 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 .
}
```

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



Βασικά ερωτήματα SPARQL

- άλλο παράδειγμα:

```
PREFIX uni: <http://www.mydomain.org/uni-ns#>
SELECT ?i
WHERE {
    ?i rdf:type rdfs:course .
}
```

Ερμηνεία: ανάκτηση όλων των στιγμιοτύπων της κλάσης course



Δομή select-from-where

■ 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 .
}
```

Ερμηνεία: ανάκτηση όλων των τηλεφώνων των μελών του προσωπικού



Δομή select-from-where

```
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 .
}
```

Ερμηνεία: ανάκτηση όλων των τηλεφώνων των μελών του προσωπικού

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`)

Χρήση FILTER

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

Ερμηνεία: 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 «=»).



Προαιρετικά υποδείγματα

- Μέχρι τώρα επιστρέφεται απάντηση αν υπάρχει πλήρης ταύτιση υποδείματος στη βάση γνώσης
- Συχνά όμως απαιτείται μεγαλύτερη ευελιξία
 - Για παράδειγμα:

```
<uni:lecturer rdf:about="949352">  
  <uni:name>Grig. 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



Προαιρετικά υποδείγματα

- Χρήση του **OPTIONAL**

```
<uni:lecturer rdf:about="949352">
  <uni:name>Grig. 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
John Hatzis	ihatz@cti.gr
Grigoris Antoniou	



Σύνοψη Ενότητας RDF

Σύνοψη Ενότητας

- Το RDF παρέχει μια βάση για την αναπαράσταση και την επεξεργασία μεταδεδομένων
- Το RDF διαθέτει ένα μοντέλο δεδομένων που βασίζεται σε γράφους. Οι βασικές έννοιές του είναι: *πόρος*, *ιδιότητα* και *πρόταση*. Κάθε πρόταση είναι μια τριάδα πόρος-ιδιότητα-τιμή.
- Για την υποστήριξη της συντακτικής διαλειτουργικότητας, το RDF έχει μια σύνταξη που βασίζεται στην XML. Η XML και το RDF αλληλοσυμπληρώνονται, επειδή το RDF υποστηρίζει τη συντακτική διαλειτουργικότητα.
- Η φιλοσοφία του RDF είναι αποκεντρωμένη και επιτρέπει το σταδιακό χτίσιμο της γνώσης, καθώς και το διαμοιρασμό και την επαναχρησιμοποίησή της.
- Το RDF είναι ανεξάρτητο από το πεδίο εφαρμογής. Το RDF Schema παρέχει ένα μηχανισμό για την περιγραφή συγκεκριμένων πεδίων.



Σύνοψη Ενότητας

- Το RDF Schema είναι μια στοιχειώδης γλώσσα οντολογιών. Παρέχει συγκεκριμένα θεμελιώδη στοιχεία μοντελοποίησης με σταθερό νόημα. Οι έννοιες-κλειδιά του RDF Schema είναι οι κλάσεις, οι σχέσεις υποκλάσης, οι ιδιότητες, οι σχέσεις υποϊδιότητας και οι περιορισμοί στο πεδίο ορισμού και στο σύνολο τιμών.
- Υπάρχουν γλώσσες ερωτημάτων για τα RDF και RDFS, στις οποίες περιλαμβάνεται η SPARQL.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.



Σημείωμα Αναφοράς

Copyright: Πανεπιστήμιον Πατρών, Ιωάννης Χατζηλυγερούδης, 2015.
«Αναπαράσταση Γνώσης στον Παγκόσμιο Ιστό. RDF». Έκδοση: 1.0. Πάτρα
2015. Διαθέσιμο από τη δικτυακή διεύθυνση:
<https://eclass.upatras.gr/courses/CEID1098/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



Σημείωμα Χρήσης Έργων Τρίτων

- Οι διαφάνειες είναι κατά μεγάλο μέρος βασισμένες στο βιβλίο "Εισαγωγή στον Σημασιολογικό Ιστό", των Grigoris Antoniou και Frank van Harmelen, Β' Έκδοση, 2009 (Ελληνική Έκδοση).

