



ΣΥΣΤΗΜΑ ΑΜΕΣΟΥ ΣΥΜΠΕΡΑΣΜΟΥ ΓΙΑ RDF ΚΑΙ RDFS

Σύστημα άμεσου συμπερασμού για RDF και RDFS (1/3)

- Η αξιωματική σημασιολογία μπορεί να χρησιμοποιηθεί για αυτοματοποιημένη συλλογιστική στις γλώσσες RDF και RDF Schema
- Ωστόσο, απαιτείται ένα σύστημα αποδείξεων λογικής πρώτης τάξης για να επιτευχθεί αυτό
 - Αυτή είναι μία υπερβολική απαίτηση και μπορεί να μην κλιμακώνεται όταν εμπλέκονται εκατομμύρια προτάσεις
- Για το λόγο αυτό, έχει δοθεί και στην RDF μια σημασιολογία (και ένα σύστημα συμπερασμού που είναι ακριβές και πλήρες για τη συγκεκριμένη σημασιολογία) σε απευθείας αντιστοιχία με τριάδες RDF, και όχι με επαναδιατύπωση της RDF με όρους λογικής πρώτης τάξης
- Αυτό το σύστημα συμπερασμού αποτελείται από κανόνες της μορφής

AN το **E** περιέχει συγκεκριμένες τριάδες
TOTE πρόσθεσε στο **E** συγκεκριμένες επιπλέον τριάδες

όπου **E** ένα
αυθαίρετο σύνολο
τριάδων RDF

Σύστημα άμεσου συμπερασμού για RDF και RDFS (2/3)

- Ακολουθούν μερικά βασικά παραδείγματα:

AN το **E** περιέχει την τριάδα $(?x, ?p, ?y)$
TOTE το **E** περιέχει επίσης την τριάδα $(?p, rdfs:type, rdfs:property)$

- Αυτό δηλώνει ότι οποιοσδήποτε πόρος $?p$ που χρησιμοποιείται στη θέση της ιδιότητας μιας τριάδας μπορεί να θεωρηθεί μέλος της κλάσης *rdfs:Property* μέσω συμπερασμού

AN το **E** περιέχει τις τριάδες $(?u, rdfs:subClassOf, ?v)$ και $(?v, rdfs:subclassOf, ?w)$
TOTE το **E** περιέχει επίσης την τριάδα $(?u, rdfs:subClassOf, ?w)$

- Κωδικοποιεί τη μεταβατικότητα της σχέσης της υποκλάσης
- Πολύ σχετικός είναι και ο κανόνας:

AN το **E** περιέχει τις τριάδες $(?x, rdfs:type, ?u)$ and $(?u, rdfs:subClassOf, ?v)$
TOTE το **E** περιέχει επίσης την τριάδα $(?x, rdfs:type, ?v)$

- Πρόκειται για τον ουσιαστικό ορισμό του νοήματος της ιδιότητας *rdfs:subClassOf*

Σύστημα άμεσου συμπερασμού για RDF και RDFS (3/3)

- Ένα τελευταίο παράδειγμα:

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

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

- Ο κανόνας αυτός δηλώνει ότι οποιοσδήποτε πόρος $?y$ που εμφανίζεται ως τιμή της ιδιότητας $?p$ μπορεί να θεωρηθεί μέλος του συνόλου τιμών της $?p$ μέσω συμπερασμού
- Αυτό δείχνει ότι οι ορισμοί του συνόλου τιμών στην RDF Schema δεν χρησιμοποιούνται για τον περιορισμό του συνόλου τιμών μιας ιδιότητας, αλλά μάλλον για το συμπερασμό των μελών του συνόλου τιμών
- Το πλήρες σύνολο αυτών των κανόνων κλειστότητας δεν αριθμεί πάνω από μερικές δεκάδες κανόνες και μπορεί να υλοποιηθεί αποδοτικά χωρίς πολύπλοκες τεχνολογίες απόδειξης θεωρημάτων



ΕΡΩΤΗΜΑΤΑ ΣΤΗ SPARQL

Ερωτήματα στη SPARQL– Εισαγωγή (1/2)

- Γιατί χρειαζόμαστε μία καινούρια γλώσσα ερωτημάτων αντί να χρησιμοποιήσουμε κάποια γλώσσα ερωτημάτων XML?
 - Η απάντηση είναι ότι το επίπεδο αφαίρεσης της XML είναι χαμηλότερο από αυτό της RDF
 - Αυτό θα οδηγήσει σε επιπλοκές αν υποβάλλουμε ερωτήματα σε έγγραφα RDF με μία γλώσσα βασισμένη στην XML
- Υπάρχουν διάφοροι τρόποι συντακτικής αναπαράστασης μιας πρότασης RDF σε XML
 - Π.χ. θέλουμε να ανακτήσουμε όλους τους τίτλους των διδασκόντων
 - Η περιγραφή κάποιου συγκεκριμένου διδάσκοντα μπορεί να είναι:

```
<rdf:Description rdf:about="949318">  
<rdf:type rdf:resource="&uni;lecturer"/>  
<uni:name>David Billington</uni:name>  
<uni:title>Associate Professor</uni:title>  
</rdf:Description>
```

- Ένα κατάλληλο ερώτημα σε γλώσσα Xpath:

```
/rdf:Description[rdf:type="http://www.mydomain.org/uni-ns#lecturer"]/uni:title
```


Ερωτήματα στη SPARQL– Εισαγωγή (2/2)

- Αλλά η ίδια περιγραφή θα μπορούσε να είχε γραφτεί ως εξής:
- Τώρα το προηγούμενο ερώτημα XPath δε λειτουργεί. Πρέπει να γράψουμε:
- Μία 3^η πιθανή αναπαράσταση:
- Για τη συγκεκριμένη συντακτική παραλλαγή, πρέπει να δωθεί ένα ακόμα ερώτημα σε XPath:
- Αφού κάθε περιγραφή ενός διδάσκοντα μπορεί να έχει οποιαδήποτε από αυτές τις ισοδύναμες μορφές, πρέπει να συντάξουμε διαφορετικά ερωτήματα XPath
- Ένας καλύτερος τρόπος θα ήταν η συγγραφή ερωτημάτων σε επίπεδο RDF
 - Μία κατάλληλη γλώσσα ερωτημάτων πρέπει να καταλαβαίνει την RDF (όχι μόνο τη σύνταξη αλλά και το μοντέλο δεδομένων της RDF καθώς και τη σημασιολογία του λεξιλογίου της)

```
<uni:lecturer rdf:about="949318">  
<uni:name>David Billington</uni:name>  
<uni:title>Associate Professor</uni:title>  
</uni:lecturer>
```

```
//uni:lecturer/uni:title
```

```
<uni:lecturer rdf:about="949318"  
uni:name="David Billington"  
uni:title="Associate Professor"/>
```

```
//uni:lecturer/@uni:title
```

Βασικά Ερωτήματα (1/2)

- Η γλώσσα ερωτημάτων **SPARQL** βασίζεται στην ταύτιση υποδειγμάτων γράφων
 - Το απλούστερο υπόδειγμα γράφου είναι το υπόδειγμα της τριάδας, το οποίο μοιάζει με μία τριάδα **RDF**
 - Αλλά υπάρχει η δυνατότητα χρήσης μεταβλητής αντί όρου **RDF** στις θέσεις του υποκειμένου, του κατηγορήματος ή του αντικειμένου
- Ο συνδυασμός υποδειγμάτων τριάδων παράγει ένα βασικό υπόδειγμα γράφου, και απαιτείται ακριβής ταύτιση με κάποιο γράφο προκειμένου ένα υπόδειγμα να θεωρηθεί πλήρες
- Ένα απλό παράδειγμα είναι το ερώτημα:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?c WHERE { ?c rdf:type rdfs:Class . }
```
- Το ερώτημα αυτό ανακτά όλα τα υποδείγματα τριάδων, όπου το *rdf:type* είναι η ιδιότητα και το *rdfs:Class* είναι το αντικείμενο
 - Όταν εκτελεστεί το ερώτημα αυτό, θα ανακτηθούν όλες οι κλάσεις
- Παρόμοια με το μηχανισμό του χώρου ονομάτων που χρησιμοποιείται για τη συγγραφή **RDF** σε **XML**, η **SPARQL** μας επιτρέπει να ορίζουμε προθέματα για τους χώρους ονομάτων και να τα χρησιμοποιούμε στο υπόδειγμα του ερωτήματος

Βασικά Ερωτήματα (2/2)

- Για την ανάκτηση όλων των στιγμιοτύπων μιας συγκεκριμένης κλάσης (π.χ. της *course*) γράφουμε:

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

- Η SPARQL δεν υποστηρίζει ρητά τη σημασιολογία της γλώσσας RDFS
- Επομένως, το αποτέλεσμα του ερωτήματος εξαρτάται τελικά από το αν το σύστημα που αποκρίνεται στο ερώτημα υποστηρίζει τη σημασιολογία της RDFS
 - Αν αυτό ισχύει, τότε το αποτέλεσμα του ερωτήματος θα περιλαμβάνει επίσης όλα τα στιγμιότυπα των υποκλάσεων της κλάσης *course*
 - Διαφορετικά, θα ανακτηθούν μόνο τα στιγμιότυπα με ρητό τύπο *course*

Χρήση της Δομής `select-from-where` (1/3)

- Όπως συμβαίνει στην `SQL`, τα ερωτήματα στην `SPARQL` έχουν δομή του τύπου `SELECT-FROM-WHERE`:
 - `SELECT` καθορίζει την *προβολή* (*projection*): τον αριθμό και τη σειρά των ανακτημένων δεδομένων
 - `FROM` χρησιμοποιείται για τον προσδιορισμό της πηγής, στην οποία γίνεται το ερώτημα
 - Ο όρος είναι προαιρετικός
 - Όταν δεν ορίζεται μπορούμε απλώς να υποθέσουμε ότι υποβάλλουμε ένα ερώτημα στη βάση γνώσης ενός συγκεκριμένου συστήματος
 - `WHERE` επιβάλλει περιορισμούς στις δυνατές λύσεις με τη μορφή προτύπων που περιέχουν υποδείγματα γράφων, καθώς και με τη μορφή λογικών περιορισμών
- Π.χ. για να ανακτήσουμε όλα τα τηλέφωνα των μελών του προσωπικού:

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

- Εδώ τα `?x` και `?y` είναι μεταβλητές, και το `?x uni:phone ?y` αναπαριστά ένα υπόδειγμα τριάδας πόρου-ιδιότητας-τιμής

Χρήση της Δομής **select-from-where** (2/3)

- Μπορούμε να δημιουργήσουμε πιο πολύπλοκα υποδείγματα γράφων για την ανάκτηση περισσότερο σύνθετων πληροφοριών από τα ερωτήματά μας
 - Π.χ. για να ανακτήσουμε όλους τους διδάσκοντες και τα τηλέφωνα τους:

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

- Εδώ ο όρος `?x rdf:type uni:Lecturer` συγκεντρώνει όλα τα στιγμιότυπα της κλάσης `Lecturer` και συνδέει το αποτέλεσμα με τη μεταβλητή `?x`
- Το δεύτερο μέρος συγκεντρώνει όλες τις τριάδες με κατηγορία `phone`
 - Όμως υπάρχει μία *έμμεση συνένωση*, με την έννοια ότι περιορίζουμε το δεύτερο υπόδειγμα μόνο στις τριάδες εκείνες με υποκείμενο που βρίσκεται στη μεταβλητή `?x`
 - Στην περίπτωση αυτή χρησιμοποιούμε μία συντακτική συντόμευση
 - Το ερωτηματικό υποδηλώνει ότι το υπόδειγμα τριάδας που ακολουθεί έχει κοινό υποκείμενο με το προηγούμενο υπόδειγμα
 - Επομένως, το ερώτημα είναι ισοδύναμο με το:

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

Χρήση της Δομής **select-from-where** (3/3)

- Θα δείξουμε τώρα μία άμεση συνένωση μέσω ενός ερωτήματος που ανακτά τα ονόματα όλων των μαθημάτων που διδάσκονται από το διδάσκοντα με ID 949352

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

- Στη SPARQL χρησιμοποιούμε τη συνθήκη FILTER για να υποδείξουμε ένα λογικό περιορισμό
 - Στην περίπτωση αυτή, ο περιορισμός είναι η άμεση συνένωση των μεταβλητών ?c και ?x με τη χρήση ενός τελεστή ισότητας (=)

Προαιρετικά Υποδείγματα (1/2)

- Τα υποδείγματα γράφων που είδαμε μέχρι τώρα είναι υποχρεωτικά
 - Είτε η βάση γνώσης ταυτίζεται με το πλήρες υπόδειγμα, οπότε και επιστρέφεται μια απάντηση
 - Ή δεν ταυτίζεται, οπότε και το ερώτημα δεν παράγει αποτέλεσμα
- Ωστόσο, μπορεί να θέλουμε συχνά περισσότερη ευελιξία
 - Θεωρείστε το ακόλουθο απόσπασμα RDF:

```
<uni:lecturer rdf:about="949352">  
<uni:name>Grigoris Antoniou</uni:name>  
</uni:lecturer>  
<uni:professor rdf:about="949318">  
<uni:name>David Billington</uni:name>  
<uni:email>david@work.example.org</uni:email>  
</uni:professor>
```

- Το τμήμα αυτό περιέχει πληροφορίες για δύο διδάσκοντες
 - Για τον πρώτο, παρέχεται μόνο το όνομα
 - Για τον δεύτερο, παρέχεται και το e-mail του

Προαιρετικά Υποδείγματα (2/2)

- Θα υποβάλουμε τώρα ένα ερώτημα για όλους τους διδάσκοντες και τα e-mail τους:

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

- Το αποτέλεσμα θα ήταν:

?name	?email
David Billington	david@work.example.org
- Παρόλο που ο Grigoris Antoniou αναφέρεται ως διδάσκων, το ερώτημα δεν επιστρέφει το όνομά του
 - Δεν υπάρχει ταύτιση με το υπόδειγμα του ερωτήματος, επειδή το συγκεκριμένο άτομο δεν έχει e-mail
- Η λύση είναι να προσαρμόσουμε το ερώτημα με τη χρήση ενός προαιρετικού υποδείγματος:

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

?name	?email
Grigoris Antoniou	
David Billington	david@work.example.org

Το νόημα είναι περίπου ότι το ερώτημα θα έπρεπε να «επιστρέψει όλα τα ονόματα των διδασκόντων καθώς και τα e-mail τους αν αυτά είναι γνωστά»