Συστήματα και Τεχνολογίες Γνώσης

Θέμα

Μετζάκης Ιωάννης

Α.Μ.: 03116202

ΣΗΜΜΥ 8ο

Περιγραφή

Σε αυτό το θέμα, καλούμαστε να δημιουργήσουμε μία σημασιολογική βάση γνώσης που θα περιέχει δεδομένα σχετικά με μετακινήσεις με μέσα μεταφοράς.

Συγκεκριμένα, τα 3 ζητούμενα της εργασίας αυτής, είναι τα εξής:

1. Κατασκευή οντολογίας για τα μέσα μεταφοράς, με επαρκή μοντελοποίηση του συγκεκριμένου πεδίου.
2. Συγκέντρωση δεδομένων, αναπαράστασή τους ως στιγμιότυπα των εννοιών, ρόλων και ιδιοτήτων τύπων δεδομένων της οντολογίας, και εισαγωγή τους σε αποθήκη τριάδων.
3. Επίδειξη των δυνατοτήτων της βάσης γνώσης, μέσω κατασκευής και εκτέλεσης ερωτημάτων SPARQL.

Κατασκευή Οντολογίας

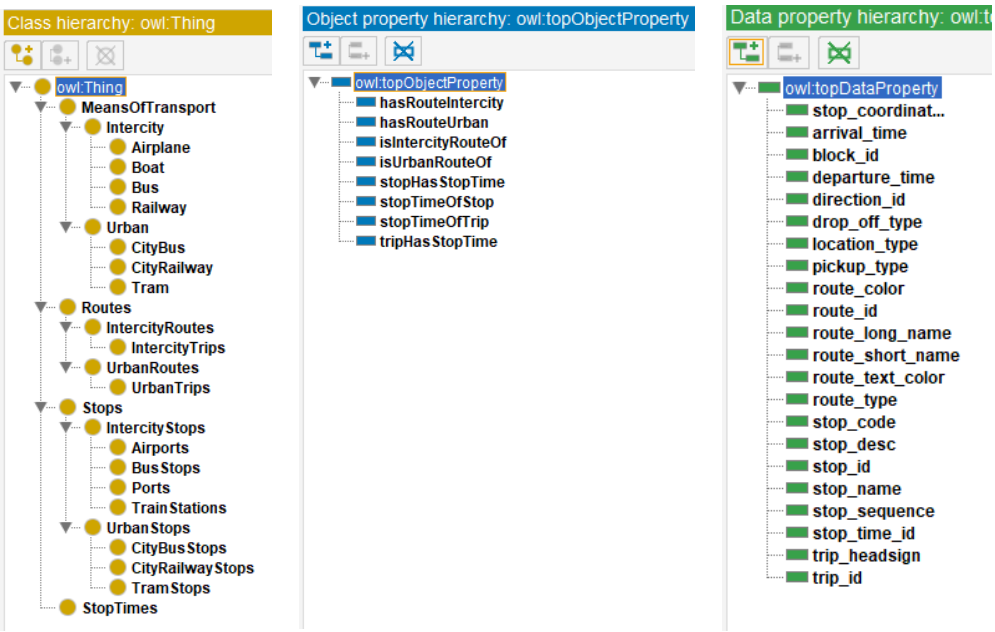
Για το μέρος αυτό, χρησιμοποίησα το γραφικό εργαλείο ανάπτυξης και επεξεργασίας οντολογιών Protege.

Έπειτα από μελέτη των αναγκών για την περιγραφή όλων των μέσων μεταφοράς, αστικών και υπεραστικών, κατέληξα στα παρακάτω συμπεράσματα:

- Για κάθε κλάση, δημιούργησα δύο υποκατηγορίες, ώστε να διαχωρίζονται τα αστικά με τα υπεραστικά δρομολόγια/μέσα/στάσεις.
- Έπειτα, δημιούργησα ξεχωριστή κλάση για κάθε πίνακα των δεδομένων από τα *"Δρομολόγια Αστικών Συγκοινωνιών Αθήνας"*. Συγκεκριμένα:
    - Την κλάση *Trips* την όρισα ως υποκλάση της κλάσης *Routes*, δεδομένου ότι κάθε διαδρομή(πχ ΝΕΟ ΦΑΛΗΡΟ - ΑΣΚΛΗΠΕΙΟ ΒΟΥΛΑ) έχει πολλά δρομολόγια(πχ κάθε 15 λεπτά).
    - Τέλος, όρισα και μία κλάση με ονομασία *StopTimes*, ώστε να μπορεί να κρατηθεί με κάποιο τρόπο ο χρόνος που φτάνει/αναχωρεί ένα μέσο σε μία συγκεκριμένη στάση. Αυτό πραγματοποιήθηκε μέσω των DataProperties *"arrival_time, departure_time",* οι οποίες συνδέονται με την κλάση *StopTimes,* η οποία συνδέεται με τη σειρά της μέσω των ObjectProperties με τις κλάσεις *Trip* και *Stops*.

- Στη συνέχεια, όρισα τα απαραίτητα ObjectProperties για τη συσχέτιση των μέσων με τις διαδρομές (*MeansOfTransport – Routes*), και των χρόνων άφιξης/αναχώρησης με τις στάσεις και τα δρομολόγια (*StopTimes – Stops και StopTimes – Trips,* αντίστοιχα*).*
- Ακολούθως, όρισα και τα DataProperties, τα οποία προέκυψαν κυρίως από τη μελέτη των δεδομένων που θα χρησιμοποιήσω στη συνέχεια.
- Τέλος, διαμόρφωσα κατάλληλα τα annotations και τις περιγραφές κάθε κλάσης (disjointWith, domain, range, κλπ).

Παρακάτω, φαίνεται ένα στιγμιότυπο των κλάσεων, των ObjectProperties και των DataProperties, όπως αυτά είναι και στο owl αρχείο που παραδίδεται μαζί με την παρούσα αναφορά:



Η παραπάνω οντολογία έχει IRI: ***http://www.ex.org/gmetz/ontology/***

Επεξεργασία και Μετατροπή Δεδομένων σε RDF τριάδες

Για το μέρος αυτό, χρησιμοποίησα τα δεδομένα από τα “Δρομολόγια Αστικών Συγκοινωνιών Αθήνας”, και συγκεκριμένα τους πίνακες:

- routes.txt
- stops.txt
- trips.txt
- stop_times.txt

Έπειτα, για ευκολότερη επεξεργασία των παραπάνω δεδομένων μετέτρεψα τα αρχεία αυτά σε csv αρχεία μέσω του python script *txt2csv.ipynb* που συμπεριλαμβάνεται στο παραδοτέο αρχείο, όπως και όλα τα αρχεία κώδικα που θα αναφερθούν στη συνέχεια.

Μέσω του αρχείου *trips2rdf.ipynb* για μετατροπή των δεδομένων του αρχείου trips.txt σε rdf τριάδες, θα δείξω τον τρόπο που έκανα αυτή τη μετατροπή σε κάθε πίνακα.

```
with open(filename, encoding="utf8") as fd:
    data0 = pd.read_csv(r'.\trips.csv')
    data = data0.T

    for row in data:

        trip_node = n['Routes/UrbanRoutes/UrbanTrips/' + str(data[row]['trip_id'])]
        class_node = n['UrbanTrips']

        g.add((trip_node, n.trip_id, Literal(data[row]['trip_id'])))
        g.add((trip_node, n.route_id, Literal(data[row]['route_id'])))
        g.add((trip_node, n.service_id, Literal(data[row]['service_id'])))
        g.add((trip_node, n.trip_headsign, Literal(data[row]['trip_headsign'])))
        g.add((trip_node, n.direction_id, Literal(data[row]['direction_id'])))
        g.add((trip_node, n.block_id, Literal(data[row]['block_id'])))

        g.add((trip_node, RDF.type, class_node))
```

Έπειτα από την εισαγωγή των απαραίτητων δεδομένων κάθε σειράς του csv πια αρχείου ”*trips.csv*” στον γράφο, απαιτείται το σώσιμο των τριάδων σε αρχεία:

```
progress.count()
if progress.is_batch_complete():
    output = 'outs/trips/' + identifier + '{num:0{width}}'.format(num=progress.current_batch, width=4) + '.ttl.n3'
    g.serialize(destination=output, format='turtle')
    g.close()
    g = Graph(identifier=identifier)

if progress.finished():
    break
```

Λόγω του μεγάλου όγκου παραγώμενων αρχείων, τα έκανα merge σε ένα μέσω της εντολής *"printf '%s\0' *.n3 | xargs -0 cat > trips_merged.txt"* και μετέπειτα μετατροπή του txt σε ttl.n3 αρχείο, έτσι ώστε να μπορούν να εισαχθούν πολύ πιο εύκολα στις αποθήκες τριάδων αργότερα.

Παρακάτω βρίσκεται ένα στιγμιότυπο των RDF τριάδων έπειτα από την επεξεργασία των δεδομένων του πίνακα *trips.txt*:

```
@prefix ns1: <http://www.ex.org/gmetz/ontology/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.ex.org/gmetz/ontology/Routes/UrbanRoutes/UrbanTrips/9801501-ΤΗΛΕΜΑ-Τ3-Δευ-Πεμ-02> a ns1:UrbanTrips ;
    ns1:block_id 2.184487e+06 ;
    ns1:direction_id 0 ;
    ns1:route_id "Τ3-20" ;
    ns1:service_id "ΤΗΛΕΜΑ-Τ3-Δευ-Πεμ-02" ;
    ns1:trip_headsign "ΝΕΟ ΦΑΛΗΡΟ - ΑΣΚΛΗΠΕΙΟ ΒΟΥΛΑ" ;
    ns1:trip_id "9801501-ΤΗΛΕΜΑ-Τ3-Δευ-Πεμ-02" .

<http://www.ex.org/gmetz/ontology/Routes/UrbanRoutes/UrbanTrips/9801502-ΤΗΛΕΜΑ-Τ3-Δευ-Πεμ-02> a ns1:UrbanTrips ;
    ns1:block_id 2.18449e+06 ;
    ns1:direction_id 0 ;
    ns1:route_id "Τ3-20" ;
    ns1:service_id "ΤΗΛΕΜΑ-Τ3-Δευ-Πεμ-02" ;
    ns1:trip_headsign "ΝΕΟ ΦΑΛΗΡΟ - ΑΣΚΛΗΠΕΙΟ ΒΟΥΛΑ" ;
    ns1:trip_id "9801502-ΤΗΛΕΜΑ-Τ3-Δευ-Πεμ-02" .
```

Για την κλάση *StopTimes,* δημιούργησα ένα αναγνωριστικό id, όπου δεν υπήρχε στον δωσμένο πίνακα, και προκύπτει μοναδικά από τη συνένωση των αναγνωριστικών id του δρομολογίου και της στάσης που αφορά η κάθε γραμμή του πίνακα *stop_times.txt.*

Όπως ανέφερα και στο πρώτο μέρος, με αυτό το αναγνωριστικό μπορώ να συνδέσω κάθε δρομολόγιο με τις ώρες που 'σταματάει' σε κάθε στάση.

Τέλος, για την γεωγραφική αναπαράσταση των στάσεων, χρησιμοποίησα την προτεινόμενη μεθοδολογία της εκφώνησης, με το λεκτικό *"point(lon lat)"^^ http://www.openlinksw.com/schemas/virtrdf#Geometry*:

```
@prefix ns1: <http://www.ex.org/gmetz/ontology/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.ex.org/gmetz/ontology/Stops/UrbanStops/10001> a ns1:UrbanStops ;
    ns1:location_type 0 ;
    ns1:stop_code 10001 ;
    ns1:stop_coordinates "point(37.99860826413671 23.6649846246733)^^<http://www.openlinksw.com/schemas/virtrdf#Geo
metry>" ;
    ns1:stop_desc "Επί της ΕΛ.ΒΕΝΙΖΕΛΟΥ" ;
    ns1:stop_id 10001 ;
    ns1:stop_name "ΣΤΡΟΦΗ" .

<http://www.ex.org/gmetz/ontology/Stops/UrbanStops/10002> a ns1:UrbanStops ;
    ns1:location_type 0 ;
    ns1:stop_code 10002 ;
    ns1:stop_coordinates "point(37.996719436997 23.6631823242898)^^<http://www.openlinksw.com/schemas/virtrdf#Geome
try>" ;
    ns1:stop_desc "Επί της ΛΕΩΦ.ΕΛ.ΒΕΝΙΖΕΛΟΥ" ;
    ns1:stop_id 10002 ;
```

Όλες τις RDF τριάδες τις αποθηκεύω στη συνέχεια στην αποθήκη τριάδων στο openlink Virtuoso, μέσω του Quad Store Upload.

Κατασκευή και Εκτέλεση ερωτημάτων SPARQL

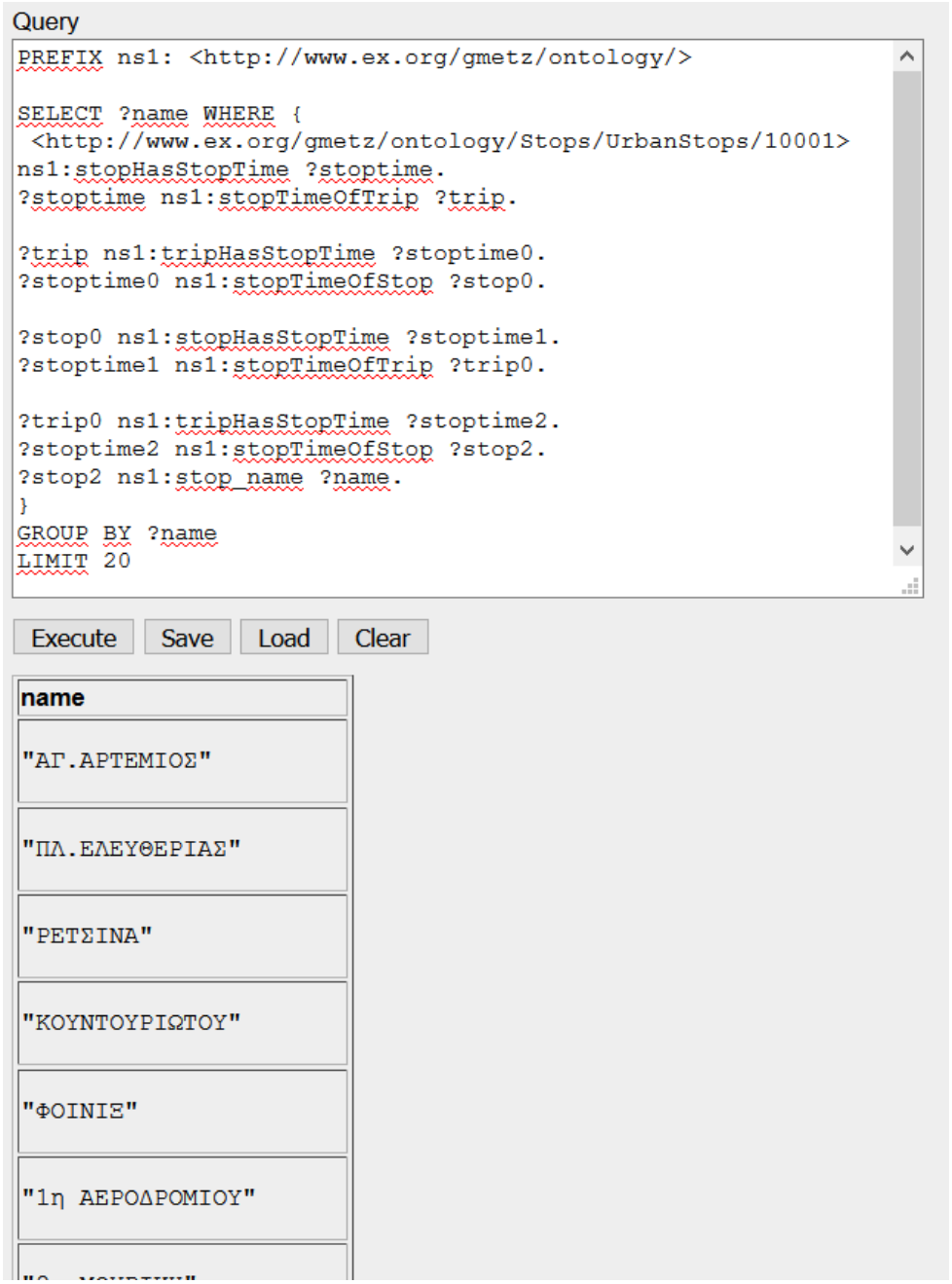
Για το τελευταίο μέρος, καλούμαστε να εξετάσουμε τις δυνατότητες της σημασιολογικής βάσης γνώσης που δημιουργήσαμε, μέσω ερωτημάτων SPARQL.

Αρχικά, για το πρώτο ενδεικτικό ερώτημα, ενώ έκανα αρκετές δοκιμές για την επεξεργασία των γεωγραφικών πληροφοριών, μέσω των queries του "linkedgeodata.org" *(bif:st_intersects (?g, bif:st_point(x, y), 10))*, όπου x,y οι δωσμένες συντεταγμένες, και g, οι ζητούμενες, δεν έβγαζε κανένα αποτέλεσμα.

Δοκίμασα όλες τις παρακάτω αναπαραστάσεις των συντεταγμένων και δεν υπήρξε αποτέλεσμα:

```
"point(37.99860826413671 23.6649846246733)^^<http://www.openlinksw.com/sc
"point(37.99860826413671 23.6649846246733)"
"point(37.99860826413671, 23.6649846246733)"
"bif:st_point(37.99860826413671, 23.6649846246733)"
"POINT(37.99860826413671, 23.6649846246733)"
"POINT(37.99860826413671 23.6649846246733)"
```

Για το δεύτερο ζητούμενο ερώτημα, το παρακάτω query δίνει τη ζητούμενη απάντηση, όταν ψάχνουμε σε ποιες στάσεις μπορούμε να φτάσουμε από μία άλλη στάση, με μία μόνο αλλαγή μέσου:



Για 2 ή 3 αλλαγές, απλά θα συνεχίζαμε την παραπάνω ‘αλυσίδα’.

Στο παραδοτέο zip αρχείο, βρίσκεται το αρχείο της οντολογίας, τα αρχεία με τις RDF τριάδες, (routes.ttl.n3, trips.ttl.n3, stops.ttl.n3, stop_times.ttl.n3), οι κώδικες που χρησιμοποίησα για τη μετατροπή σε τριάδες RDF (σε μορφή .ipynb για jupyter notebook), καθώς και η παρούσα αναφορά.