

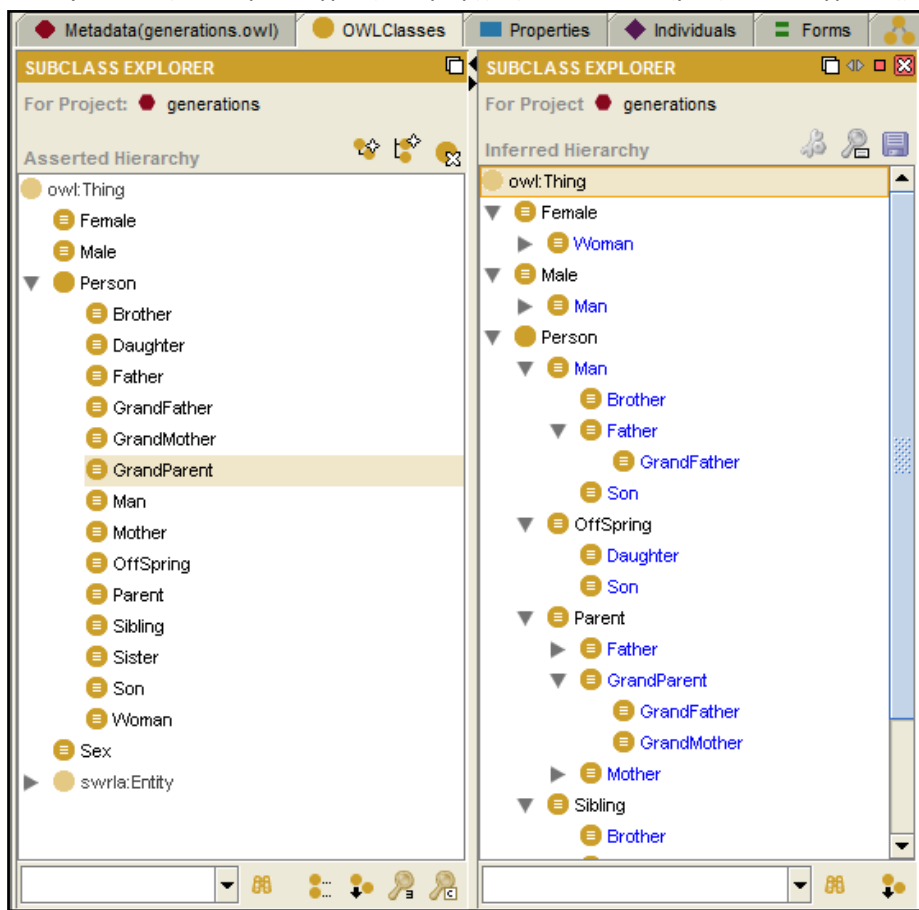
ΕΚΠΟΝΗΣΗ ΕΡΓΑΣΙΑΣ (Συνέχεια)

Απαραίτητο προαπαιτούμενο της άσκησης είναι να έχετε υλοποιήσει την *4η Σειρά Ασκήσεων*, στην οποία θα έχετε σχεδιάσει και αναπτύξει μια οντολογία της επιλογής σας, πάνω στην οποία βασίζονται τα παρακάτω ερωτήματα. Μπορείτε να κάνετε αλλαγές στην οντολογία της *4ης Σειράς Ασκήσεων* και να προσθέσετε όσες κλάσεις, ιδιότητες και αντικείμενα χρειάζεται, αν σας βοηθάει στο να απαντήσετε τα ερωτήματα.

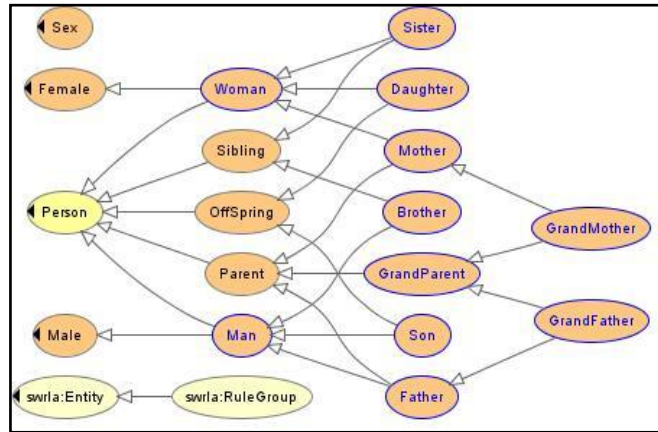
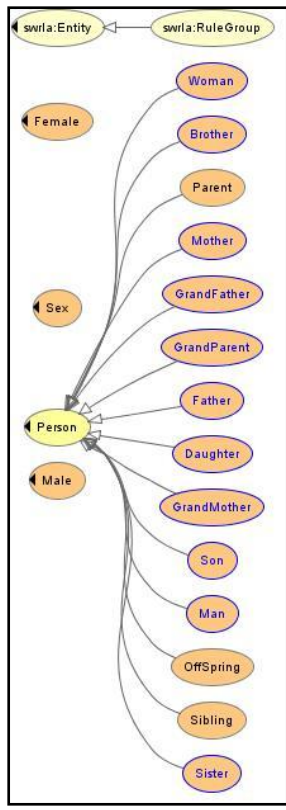
Ερώτημα 1

Από το μενού επιλέξτε πρώτα έναν Reasoner (**Reasoning->Pellet**) και στην συνέχεια (**Reasoning->ClassifyTaxonomy**).

- a. Από την καρτέλα **Classes**, εξάγετε δυο screenshots, ένα με την *“asserted class hierarchy”* και ένα με την *“inferred class hierarchy”*, όπως το παράδειγμα (οι ιεραρχίες να είναι πλήρως αναπτυγμένες):



- b. Από την καρτέλα **OWL Viz**, επιλέξτε την οντότητα *Thing*, και την επιλογή , (show all classes) και στην συνέχεια με την επιλογή  (export to image) εξάγετε εικόνες από Asserted και Inferred Model, όπως το παράδειγμα: (Δείτε το παράρτημα με οδηγίες για την εγκατάσταση του OWLViz)



c. Καταγράψτε τις διαφορές που ενδεχομένως υπάρχουν μεταξύ των ιεραρχιών των κλάσεων και των αντίστοιχων γράφων που εξάγατε από τα παραπάνω ερωτήματα και προσπαθήστε να τις ερμηνεύσετε.

Ερώτημα 2

Από το μενού επιλέξτε πρώτα έναν Reasoner **(Reasoning->Pellet)** και στην συνέχεια: **(Reasoning->Classify Taxonomy)** και **(Reasoning->Compute Inferred Types)**.

Να επιλέξετε 5 αντικείμενα που ανήκουν σε διαφορετικές κλάσεις μεταξύ τους και να γράψετε για το καθένα (σε μορφή triples και φυσικής γλώσσας) την γνώση:

- Που έχει δοθεί άμεσα (asserted)
- Που προκύπτει από την οντολογία μετά από reasoning(inferred)

Για παράδειγμα στην οντολογία *generations.owl*, για την δήλωση του στιγμιότυπου William

```
<Person rdf:about="#William">
  <hasSexrdf:resource="#MaleSex"/>
  <hasChildrdf:resource="#Peter"/>
</Person>
```

Έχουμε:

Asserted			Inferred
William	is-a	Person	William is-a Grandfather
William	hasSex	MaleSex	
William	hasChild	Peter	

Προσπαθήστε με την επιλογή σας να αναδείξετε την χρησιμότητα της οντολογίας σας και την δυνατότητα να παράγει νέα πληροφορία μετά από Reasoning.

Ερώτημα 3

Επιλέξτε πέντε έννοιες (γνώση) που έχετε αναπαραστήσει στην οντολογία σας οι οποίες μπορούν να αξιοποιηθούν από την Μηχανή Συμπερασμού για να παράγει νέα γνώση ή να διαπιστώσει ασυνέπειες. Μερικά παραδείγματα τέτοιων εννοιών είναι η δήλωση κάποιας κλάσης με περιορισμούς (Necessary ή Necessary & Sufficient) ή λογική σύνθεση, ή η δήλωση κάποιας ιδιότητας ως functional, transitive κτλ.

Για κάθε μια από τις πέντε έννοιες:

- i) Γράψτε την γνώση που αναπαριστά σε φυσική γλώσσα, σε κώδικα OWL και σε μορφή DL (Description Logic).
- ii) Αναφέρετε ένα παράδειγμα αξιοποίησης της γνώσης από την Μηχανή Συμπερασμού.

Το κάθε παράδειγμα μπορεί να δείχνει είτε την παραγωγή νέας γνώσης, ή την εμφάνιση ασυνέπειας. Μπορείτε να προσθέσετε αντικείμενα και να τους δώσετε τιμές σε ιδιότητες, για την επεξήγηση σας. Σε κάθε περίπτωση να παραθέσετε σε μορφή τριπλετών την άμεση γνώση (asserted) και την γνώση που εξάγεται από την μηχανή συμπερασμού (inferred) όπως φαίνεται στο Ερώτημα 2 της 3ης Σειράς Ασκήσεων.

Ερώτημα 4

Από το μενού επιλέξτε

- Reasoning -> Open SPARQL Query panel
- Reasoning -> Open SWRL Tab

- a. Σχηματίστε 6 queries στην γλώσσα SPARQL ώστε να ανακτήσετε πληροφορία από την οντολογία σας. Παραθέστε το ερώτημα σε φυσική γλώσσα και σε SPARQL και τα αποτελέσματα που επέστρεψε το **SPARQL Query Panel**.

Χρησιμοποιήστε τουλάχιστον δύο φορές τα παρακάτω: **ORDER BY, FILTER, OPTIONAL**

Χρησιμοποιήστε τουλάχιστον μια φορά τα παρακάτω: **UNION, MAX, DISTINCT**

Ελέγξτε τα αποτελέσματα που επέστρεψε το σύστημα και σχολιάστε αν το SPARQL Query panel αξιοποιεί τον μηχανισμό συμπερασμού. Αν δεν τον αξιοποιεί συμπληρώστε τα αποτελέσματα που θα έπρεπε να επιστρέψει κάθε ερώτημα αν χρησιμοποιούσε μηχανισμό συμπερασμού.

Σχηματίστε τα queries με τρόπο ώστε να αναδεικνύουν την δυνατότητα Reasoning της οντολογίας σας και ελέγξτε αν έχουν καλυφθεί οι προδιαγραφές της οντολογίας όπως τις καταγράψατε στο πρώτο ερώτημα απαντώντας στις αντίστοιχες ερωτήσεις για τους στόχους.

<http://protege.stanford.edu/doc/sparql/>

<http://www.xml.com/pub/a/2005/11/16/introducing-sparql-querying-semantic-web-tutorial.html?page=1>

- b. Διαβάστε τον παρακάτω οδηγό για την SWRL, και μελετήστε το παράδειγμα οντολογίας που συνδυάζει OWL οντολογία με κανόνες SWRL.

<http://www.scribd.com/doc/23580395/SWRL-Tutorial-01>

<http://protege.cim3.net/file/pub/ontologies/family.swrl.owl/family.swrl.owl>

Στη συνέχεια από το **SWRLTab** προσθέστε 6 κανόνες στην γλώσσα SWRL. Παραθέστε για κάθε κανόνα, τον κώδικα SWRL και μια περιγραφή του σε φυσική γλώσσα.

Προσπαθήστε να συντάξετε κανόνες που περιγράφουν γνώση που δεν καλύπτεται ήδη από την οντολογία σας και που θα ήταν δύσκολο να αναπαρασταθεί με τις δυνατότητες της OWL. Αν χρειάζεται προσθέστε κατάλληλες κλάσεις, σχέσεις ή αντικείμενα.

Ενεργοποιείτε τον Pellet και για κάθε κανόνα δείξτε ένα παράδειγμα που ενεργοποιείται, παραθέτοντας τα δεδομένα (τριπλέτες) που προκάλεσαν την πυροδότηση του κανόνα και τα δεδομένα (τριπλέτες) που δημιούργησε ο κανόνας.

Ερώτημα 5:

Εξηγήστε τι είναι τα:

- *open-world assumption*
- *non-unique-name assumption*

χρησιμοποιώντας για το καθένα ένα παράδειγμα από την οντολογία που φτιάξατε (αν χρειάζεται προσθέστε κατάλληλες κλάσεις ή αντικείμενα).

Ερώτημα 6:

- Αναπτύξτε εφαρμογή σε Java που να αξιοποιεί την οντολογία που σχεδιάσατε προσπαθώντας να αναδείξετε την δυνατότητα επέκτασης της πληροφορίας της μέσω του μηχανισμού Reasoning, και λαμβάνοντας υπόψη τις απαντήσεις σας στο Ερώτημα 1.Β. της 4ης Σειράς Ασκήσεων.

Η εφαρμογή που θα φτιάξετε θα πρέπει:

- Να φορτώνει την οντολογία OWL που δημιουργήσατε και να εφαρμόζει μοντέλο συμπερασμού (Inference)
- Να δίνει δυνατότητα ανάκτησης και παρουσίασης στιγμιότυπων της οντολογίας κατόπιν επιλογής κάποιας κλάσης από τον χρήστη.
- Να δίνει δυνατότητα εισαγωγής νέων στιγμιότυπων.
- Να δίνει την δυνατότητα αναζήτησης στιγμιότυπων με βάση τιμές παραμέτρων που θα καθορίζει ο χρήστης, χρησιμοποιώντας ερωτήματα SPARQL ή RDQL.
- Να δίνει δυνατότητα επιλογής στιγμιότυπου και παρουσίασης της πληροφορίας που σχετίζεται με αυτό.

Η χρήση γραφικού περιβάλλοντος θα μετρήσει θετικά.

Στην αναφορά:

- Αναφέρετε τα εργαλεία και τις βιβλιοθήκες που χρησιμοποιήσατε για την υλοποίηση.
- Περιγράψτε σύντομα τον τρόπο υλοποίησης και παραθέστε κομμάτια κώδικα που υλοποιούν τα ζητούμενα.
- Περιγράψτε με ποιον τρόπο η εφαρμογή που αναπτύξατε εκμεταλλεύεται την οντολογία σας και πως αξιοποιεί την δυνατότητα συμπερασμού (Reasoning).

Προτείνεται ανάπτυξη σε γλώσσα Java με το JENA API. Μπορείτε να χρησιμοποιήσετε οποιαδήποτε γλώσσα και αντίστοιχο API υποστηρίζει διαχείριση OWL Οντολογίας, μηχανισμό συμπερασμού και ερωτήματα SPARQL.

<http://jena.apache.org/documentation/ontology/index.html>

<http://jena.apache.org/documentation/inference/index.html>

<http://opentox.org/data/documents/development/RDF%20files/JavaOnly/query-reasoning-with-jena-and-sparql>

<http://www.ibm.com/developerworks/xml/library/j-jena/>

Bonus Ερώτημα (+1 στον τελικό βαθμό του μαθήματος):

Αξιοποιήστε στην εφαρμογή του ερωτήματος 6, τους κανόνες που δημιουργήσατε στο ερώτημα 5b.

Η Jena δεν μπορεί από μόνη της να αξιοποιήσει κανόνες SWRL. Ωστόσο μπορείτε να χρησιμοποιήσετε σε συνδυασμό με την Jena τον Pellet που υποστηρίζει SWRL.

- <http://clarkparsia.com/pellet/fag/using-pellet-in-jena/>
- <http://allthingssemantic.blogspot.gr/2012/04/configuring-pellet-reasoner-and-jena.html>

Εναλλακτικά μπορείτε να προσπαθήσετε να μετατρέψετε τους κανόνες στην γλώσσα 'Jena Rules' που μπορεί να αξιοποιήσει η Jena από μόνη της με την Rule Engine που περιέχει.

- <http://hydrogen.informatik.tu-cottbus.de/wiki/index.php/JenaRules>
- <http://jena.apache.org/documentation/inference/>
- <http://lyle.smu.edu/~coyle/cse7392.semweb/handouts/s13.Creating%20Rules%20in%20Jena.pdf>

Παραδοτέα:

- Τα αρχεία .owl (της οντολογίας) και .pproj (του Protégé).
- **Αναφορά** με ότι ζητάει το κάθε ερώτημα, παραθέτοντας όπου απαιτείται εικόνες.
 - Για το ερώτημα 6, παραθέστε και σχολιάστε τον κώδικα που υλοποιεί τα ζητούμενα. Επίσης περιγράψτε σύντομα τα εργαλεία και τις βιβλιοθήκες που χρησιμοποιήσατε στην ανάπτυξη.
- **Πηγαίο** (κώδικας και project αρχεία αν υπάρχουν) και **εκτελέσιμο** κώδικα για το ερώτημα 6.

Το συμπιεσμένο αρχείο που θα περιέχει τα παραπάνω πρέπει να έχει όνομα : «**KRWEB_AM_PROJECT.rar**» (όπου AM ο αριθμός μητρώου σας) και η παράδοση γίνεται από το site του μαθήματος στο eclass.

Βαθμολογία:

Η εργασία είναι ατομική με ποσοστό 55% στον τελικό βαθμό του μαθήματος. Η βαθμολογία θα στηριχθεί σε βαθμολόγηση των παραδοτέων και σε προφορική εξέταση που θα οριστεί.

Ερώτημα 1:	10%
Ερώτημα 2:	15%
Ερώτημα 3:	20%
Ερώτημα 4:	25%
Ερώτημα 5:	10%
Ερώτημα 6:	20%

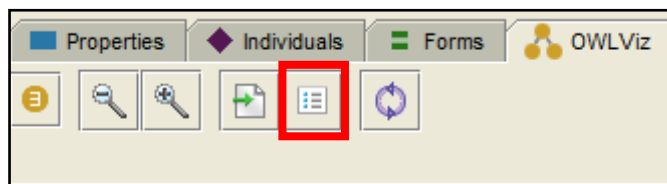
ΠΑΡΑΡΤΗΜΑ: Οδηγίες για εγκατάσταση του OWLViz

Κατεβάστε και εγκαταστήστε το GraphViz.

Προτείνεται να κατεβάσετε παλαιότερη έκδοση (πχ 2.24) απο εδώ:

<http://www.graphviz.org/pub/graphviz/stable/windows/>

Στο Protégé από το μενού **OWL->Preferences** επιλέξτε την καρτέλα **Tab**s και τσεκάρετε το **OWLViz Tab**. Στη συνέχεια στην καρτέλα **OWLViz** πατήστε το κουμπί Options (βλ εικόνα).



Στο **Dot Application Path** πατήστε **Browse** και πηγαίνετε στον φάκελο εγκατάσταση του GraphViz και στη συνέχεια στο bin\dot.exe

