



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αποδοτική Αναγνώριση Υπονοούμενων Ιεραρχικών Σχέσεων σε OWL Οντολογίες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΟΠΑΝΟΥ ΒΕΛΙΣΑΡΙΟΥ

Επιβλέπων: Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2011

.....

ΚΟΠΑΝΟΣ ΒΕΛΙΣΑΡΙΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

© 2011 - All rights reserved

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας ήταν ο σχεδιασμός και η ανάπτυξη ενός συστήματος υπολογισμού όλων των σχέσεων ιεραρχίας που υπονοούνται μεταξύ των εννοιών (κλάσεων) μιας οντολογίας με εκφραστικότητα περιορισμένη στο τμήμα (profile) EL της γλώσσας αναπαράστασης OWL με χρήση μόνο ενός σχεσιακού Συστήματος Διαχείρισης Βάσεων Δεδομένων (DBMS), παρακάμπτοντας δηλαδή τη Μηχανή Συλλογιστικής Ανάλυσης (Reasoner), και χρησιμοποιώντας έναν περισσότερο αποδοτικό αλγόριθμο στον οποίο το μέγεθος της μνήμης που παραχωρείται μπορεί να παραμετροποιηθεί από τον χρήστη.

Η λειτουργία του υποσυστήματος που προστέθηκε στο DBRS βασίζεται στην μέθοδο της Δομικής Υπαγωγής (Structural Subsumption), μια τεχνική εξαγωγής υπονοούμενων σχέσεων βασισμένη σε κανόνες (rules). Η διαδικασία της Δομικής Υπαγωγής διαιρείται σε δύο διακριτά στάδια. Αρχικά, απαιτείται μια προεπεξεργασία της οντολογίας μέσω της λεγόμενης διαδικασίας **κανονικοποίησης (normalization)** των αξιωμάτων της. Σε αυτό το στάδιο τα αξιώματα της οντολογίας «αποσυντίθεται» σε ένα σύνολο κανονικοποιημένων μορφών έτσι ώστε να αποτελέσουν τις αρχικές σχέσεις βάσει των οποίων θα υπολογιστεί η πλήρης ιεραρχία των κλάσεων στο επόμενο στάδιο, γνωστό και ως **ταξινόμηση (classification)**. Το τελευταίο περιλαμβάνει την εξαντλητική (fix-point) εφαρμογή ενός αριθμού επαγωγικών κανόνων, οι οποίοι εξάγουν τις υπονοούμενες σχέσεις βασιζόμενοι στη σημασιολογία των αξιωμάτων της οντολογίας. Στη δική μας περίπτωση, τόσο η αποθήκευση και διαχείριση της κανονικοποιημένης οντολογίας όσο και η εφαρμογή των επαγωγικών κανόνων γίνεται εξολοκλήρου από το DBMS.

Λέξεις κλειδιά: Μεταδεδομένα, Οντολογία, Περιγραφική Λογική, Βάση Γνώσης, OWL, Συλλογιστική Ανάλυση, Δομική Υπαγωγή, Σχεσιακό σύστημα Διαχείρισης Βάσεων Δεδομένων, ταξινόμηση, κανονικοποίηση

Abstract

The purpose of this thesis was to design and develop a system for calculating all the hierarchy relations implied between the concepts (classes) of an ontology with limited expressiveness in EL profile of OWL knowledge representation language using only a relational Database Management System (DBMS), bypassing the Machine Reasoning Analysis (Reasoner), and using a more efficient algorithm in which the size of memory allocated can be customized by the user.

The subsystem added to the existent DBRS is based on method of Structural Subsumption, a technique of exporting implied relationships based on rules. The process of Structural Subsumption divided into two distinct stages. Initially, there is a pretreatment of the ontology through the so-called normalization procedure of the axioms. At this stage the axioms of the ontology "decompose" in a set of normalized forms in order to form initial relationships on which to compute the full hierarchy of classes to the next stage, known as classification. The latter includes an exhaustive (fix-point) implementation of a number of inductive rules, which derive implicit relationships based on the semantics of the axioms of the ontology. In our case, both the storage and management of the normalized ontology and the application of inductive rules is entirely covered by the DBMS.

Keywords: Metadata, Ontology, Description Logic, Knowledge Base, OWL, Reasoning Analysis, Structural Subsumption, Relational Database Management System, classification, normalization

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων (ΕΣΒΓΔ) του Εθνικού Μετσόβιου Πολυτεχνείου και αποτέλεσε μια πολύ καλή αφορμή για να ασχοληθώ με ορισμένα εξαιρετικά ενδιαφέροντα προβλήματα που θέτει η προσπάθεια ανάπτυξης του Σημασιολογικού Ιστού. Στο σημείο αυτό, θα ήθελα αρχικά να ευχαριστήσω τους καθηγητές Ι. Βασιλείου και Τ. Σελλή που μου έδωσαν την ευκαιρία να ασχοληθώ με τα συγκεκριμένα θέματα, ενώ ιδιαίτερος ευχαριστώ τον υποψήφιο διδάκτορα Γιάννη Λιαγούρη για το ενδιαφέρον που έδειξε, τις πολύτιμες παρατηρήσεις του και την γενικώς πολύ καλή συνεργασία που είχα καθόλη τη διάρκεια της διπλωματικής.

Πίνακας περιεχομένων

| | | |
|----------|--|-----------|
| 1 | Εισαγωγή | 1 |
| 1.1 | Αντικείμενο διπλωματικής εργασίας | 3 |
| 1.1.1 | Συνεισφορά..... | 5 |
| 1.2 | Οργάνωση κειμένου..... | 5 |
| 2 | Θεωρητικό υπόβαθρο | 7 |
| 2.1 | Η έννοια της οντολογίας..... | 7 |
| 2.2 | Περιγραφική Λογική (Description Logic - DL)..... | 10 |
| 2.3 | Συλλογιστική ανάλυση στην Περιγραφική Λογική | 22 |
| 2.3.1 | Η μέθοδος της προσυμπλήρωσης | 25 |
| 2.3.2 | Tableau αλγόριθμοι | 28 |
| 2.3.3 | Βασισμένοι σε datalog αλγόριθμοι..... | 33 |
| 2.3.4 | Δομική Υπαγωγή | 37 |
| 2.4 | Η γλώσσα OWL..... | 59 |
| 2.4.1 | Γενικά χαρακτηριστικά | 62 |
| 2.4.2 | OWL 2 EL..... | 65 |
| 2.4.3 | OWL 2 QL..... | 76 |
| 2.4.4 | OWL 2 RL..... | 82 |
| 2.3.2 | Συκεντρωτικό Υλικό..... | 90 |
| 3 | Ανάλυση Συστήματος | 94 |
| 3.1 | Αρχιτεκτονική - Διαχωρισμός υποσυστημάτων | 94 |
| 3.2 | Περιγραφή υποσυστημάτων | 97 |
| 3.2.1 | Υποσύστημα γραφικής διαπροσωπίας χρήστη | 97 |
| 3.2.2 | Υποσύστημα φόρτωσης οντολογίας..... | 97 |
| 3.2.3 | Υποσύστημα κατασκευής Σχήματος Ετικετών | 101 |
| 3.2.4 | Υποσύστημα διαχείρισης οντολογίας..... | 102 |
| 3.2.5 | Υποσύστημα αποτίμησης ερωτημάτων | 102 |

| | | |
|----------|---|------------|
| 4.2.6 | Υποσύστημα διαχείρισης Βάσης Δεδομένων | 102 |
| 4.2.7 | Υποσύστημα διαχείρισης Μηχανής Συλλογιστικής Ανάλυσης | 102 |
| 4 | Σχεδίαση Συστήματος | 103 |
| 4.1 | Υποσύστημα κανονικοποίησης..... | 104 |
| 4.1.1 | Εφαρμογή διαχείρισης αξιωμάτων..... | 104 |
| 4.1.2 | Εφαρμογή προσωρινής αποθήκευσης κανονικοποιημένων αξιωμάτων | 105 |
| 4.1.3 | Εφαρμογή απόδοσης μοναδικού αναγνωριστικού | 105 |
| 4.1.4 | Εφαρμογή χειρισμού εκφράσεων OWL κλάσεων | 106 |
| 4.1.5 | Εφαρμογή αποθήκευσης οντολογίας στο DBMS | 108 |
| 4.2 | Υποσύστημα διαχείρισης ισοδυναμιών | 108 |
| 4.2.1 | Εφαρμογή ομαδοποίησης ισοδυναμιών..... | 109 |
| 4.2.2 | Εφαρμογή ενημέρωσης σχήματος βάσεως | 109 |
| 4.3 | Υποσύστημα ταξινόμησης..... | 109 |
| 4.3.1 | Εφαρμογή υπολογισμού μεταβατικού κλεισίματος ιδιοτήτων | 112 |
| 4.3.2 | Εφαρμογή απαλοιγής επαγωγικού κανόνα 5 | 112 |
| 4.3.3 | Εφαρμογή δημιουργίας τρέχοντος cluster | 112 |
| 4.3.4 | Εφαρμογή υπολογισμού μεταβατικού κλεισίματος κλάσεων..... | 113 |
| 4.3.5 | Εφαρμογή υπολογισμού σχέσεων ιεραρχίας λόγω υπαρξιακού | 114 |
| 4.3.6 | Εφαρμογή εύρεσης εννοιών που πρέπει να επανεξεταστούν..... | 115 |
| 4 | Βιβλιογραφία..... | 116 |

1 *Εισαγωγή*

Ο όρος Σημασιολογικός Ιστός (Semantic Web) πρωτοεμφανίστηκε τη δεκαετία του 1990 αναφερόμενος στο όραμα εμπλουτισμού του (υπάρχοντος) Συντακτικού Ιστού (Syntactic Web) με σημασιολογική πληροφορία (semantics), δηλαδή με πληροφορία που αναφέρεται στα ίδια τα δεδομένα που υπάρχουν, οργανώνονται στο διαδίκτυο, παρουσιάζονται και μεταφέρονται μέσω αυτού. Η σημασιολογική πληροφορία, ή αλλιώς μεταδεδομένα (metadata), στοχεύει στο να καταστήσει τους πόρους του διαδικτύου (web resources) προσπελάσιμους από αυτοματοποιημένες διαδικασίες, δηλαδή από αλγοριθμικές διαδικασίες (software) που δεν απαιτούν καθόλου ή τουλάχιστον περιορίζουν σε ένα βαθμό την «ανθρώπινη παρέμβαση». Με απλά λόγια, ο Σημασιολογικός Ιστός (ΣΙ) είναι ένας «πιο έξυπνος» ιστός όπου οι υπολογιστές θα μπορούν να «καταλαβαίνουν» την πληροφορία, να την οργανώνουν καλύτερα και συνεπώς να επιτελούν περισσότερα και πολυπλοκότερα καθήκοντα.

Η προσπάθεια προσθήκης των μεταδεδομένων στον Παγκόσμιο (συντακτικό) Ιστό (World Wide Web - WWW) γεννά μια σειρά από προκλήσεις που τα τελευταία χρόνια απασχολούν ιδιαίτερος την επιστημονική έρευνα. Τα ζητήματα που ανακύπτουν είναι πολλά. Σε πρώτο βήμα, τα σημασιολογικά σχήματα πρέπει να περιγραφούν αυστηρώς με χρήση κατάλληλων τυπικών φορμαλισμών (να μετατραπούν δηλαδή σε γνώση), δεδομένου ότι η αυστηρότητα αυτή είναι που θα εξασφαλίσει και τη δυνατότητα αυτόματης ανάλυσής τους από υπολογιστές. Στη συνέχεια, οι συγκεκριμένοι τυπικοί φορμαλισμοί πρέπει, με τη σειρά τους, να συνοδεύονται κι από τυπικές γλώσσες που

αφενός θα μπορούν να «διαβαστούν» από υπολογιστές, αφετέρου θα διαθέτουν σύνταξη συμβατή με τα ισχύοντα πρότυπα του Παγκόσμιου Ιστού (π.χ. XML), έτσι ώστε να μπορούν εύκολα να ενσωματωθούν σε αυτά ή απλώς να συνδυαστούν μαζί τους.

Σε άμεση σχέση με τα προηγούμενα, ένα εξίσου κομβικό ζήτημα (και βασική προϋπόθεση για την περαιτέρω ανάπτυξη του ΣΙ) είναι η εύρεση αλγορίθμων που θα μπορούν να αναλύσουν¹ σημασιολογική πληροφορία πλούσιας εκφραστικότητας με όσο το δυνατόν αποδοτικότερο τρόπο. Ο λόγος για τον οποίο η σχετική έρευνα επικεντρώνεται στην αυξημένη απόδοση είναι προφανής. Οι διαδικασίες ανάλυσης πρέπει, αν λάβουμε υπόψη μας και την χρονική καθυστέρηση της μεταφοράς των δεδομένων πάνω στα πρωτόκολλα του διαδικτύου, να αποκρίνονται σε επιτρεπούς για τον χρήστη χρόνους. Από την άλλη, ερχόμενοι στο δεύτερο σκέλος, οι συλλογιστικοί αλγόριθμοι πρέπει να μπορούν να αναλύουν πλήρως (sound & complete) γνώση μεγάλης εκφραστικότητας, έτσι ώστε να είναι σε θέση να χειριστούν όλα τα χαρακτηριστικά και τις (ενδεχομένως περίπλοκες) σχέσεις μεταξύ των δεδομένων του Παγκόσμιου Ιστού, όπως αυτές υφίστανται τώρα.

Ένα επίσης σημαντικό πρόβλημα που αναδύεται από την εμπειρία των πρώτων πραγματικών εφαρμογών (real-world applications) του ΣΙ είναι η ανάγκη ολοκληρωμένης διαχείρισης μεταδεδομένων πολύ μεγάλου όγκου. Στις πραγματικές εφαρμογές, ο όγκος (και η πολυπλοκότητα) των μεταδεδομένων τείνει να αυξάνεται, με αποτέλεσμα την αδυναμία συλλογιστικής ανάλυσής τους, λόγω αυξημένου κόστους σε κατανάλωση κύριας μνήμης (main memory)², ακόμα όμως και την αδυναμία, σε ορισμένες περιπτώσεις, απλής φόρτωσής τους (loading) σε αυτήν. Αν μάλιστα στο τελευταίο προσθέσουμε και το γεγονός ότι οι σημασιολογικές σχέσεις είναι μέχρι στιγμής πολύ δύσκολο να καταταμηθούν και να αναλυθούν επιμέρους, αντιλαμβανόμαστε εύκολα το λόγο για τον οποίο η επιστημονική έρευνα προσανατολίζεται ήδη στην ανάπτυξη τεχνικών που θα εκμεταλλεύονται μηχανισμούς

¹ Η λειτουργία των αλγορίθμων συλλογιστικής ανάλυσης (όπως ονομάζονται) είναι διττή. Από τη μία, στοχεύουν στον έλεγχο της συνέπειας (consistency) των σημασιολογικών σχέσεων, ενώ από την άλλη, στην εξαγωγή νέας πληροφορίας, δηλαδή νέων σχέσεων και χαρακτηριστικών, που δε δηλώνεται ρητά στο αρχικό σύνολο γνώσης, αλλά προκύπτει από αυτό μέσω μιας λογικής επαγωγής (logical inference).

² Οι συλλογιστικοί αλγόριθμοι λειτουργούν αποκλειστικώς στην κύρια μνήμη.

δευτερεύουσας μνήμης (second storage mechanisms), έτσι ώστε να ξεπεραστούν τα όποια απαγορευτικά όρια παρουσιάζονται.

1.1 Αντικείμενο διπλωματικής εργασίας

Σκοπός της παρούσας διπλωματικής εργασίας ήταν ο σχεδιασμός και η ανάπτυξη ενός συστήματος υπολογισμού όλων των σχέσεων ιεραρχίας που υπονοούνται μεταξύ των εννοιών (κλάσεων) μιας οντολογίας με εκφραστικότητα περιορισμένη στο τμήμα (profile) EL [1] της γλώσσας αναπαράστασης OWL[2]. Η υλοποίηση έγινε στα πλαίσια επέκτασης του συστήματος DBRS³ που έχει αναπτυχθεί στο εργαστήριο και το οποίο συνδυάζει ένα σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων (PostgreSQL) με μια Μηχανή Συλλογιστικής Ανάλυσης (Pellet Reasoner) προκειμένου να διαχειρίζεται αποδοτικά οντολογίες μεγάλου όγκου εκφρασμένες σε OWL. Μετά την προσθήκη της νέας λειτουργικότητας που περιγράφουμε στη συνέχεια, το DBRS μπορεί πλέον να υπολογίζει το πλήρες σύνολο των έγκυρων σχέσεων ιεραρχίας μεταξύ των κλάσεων μιας OWL EL οντολογίας με χρήση μόνο του DBMS, παρακάμπτοντας δηλαδή τη Μηχανή Συλλογιστικής Ανάλυσης (Reasoner), και χρησιμοποιώντας έναν περισσότερο αποδοτικό αλγόριθμο στον οποίο το μέγεθος της μνήμης που παραχωρείται μπορεί να παραμετροποιηθεί από τον χρήστη.

Ο λόγος για τον οποίο περιορίσαμε την ανάλυσή μας στο υποσύνολο EL της γλώσσας OWL είναι διττός. Πρώτον, στο συγκεκριμένο τμήμα εκφραστικότητας οι διαδικασίες συλλογιστικής ανάλυσης απαιτούν *πολυωνυμικό χρόνο* σε σχέση με τον αριθμό των αρχικών (explicit) αξιωμάτων της οντολογίας, εξασφαλίζοντας έτσι την ευκολία (tractability) που απαιτείται για τη διαχείριση δεδομένων μεγάλου όγκου. Δεύτερον, η πλειοψηφία των μεγάλων σε όγκο ιατρικών οντολογιών στις οποίες ο υπολογισμός των σχέσεων ιεραρχίας είναι ένα υπαρκτό πρόβλημα, όπως π.χ. Snomed CT, GALEN, GO, και CheBi μπορούν να εκφραστούν (εξ'ολοκλήρου ή τουλάχιστον σε μεγάλο βαθμό) με το συγκεκριμένο υποσύνολο της γλώσσας OWL.

³ <http://www.dbnet.ece.ntua.gr/pubs/details.php?id=1523&clang=0>

Η λειτουργία του υποσυστήματος που προστέθηκε στο DBRS βασίζεται στην μέθοδο της Δομικής Υπαγωγής (Structural Subsumption), μια τεχνική εξαγωγής υπονοούμενων σχέσεων βασισμένη σε κανόνες (rules). Εν συντομία, η διαδικασία της Δομικής Υπαγωγής διαιρείται σε δύο διακριτά στάδια. Αρχικά, απαιτείται μια προεπεξεργασία της οντολογίας μέσω της λεγόμενης διαδικασίας **κανονικοποίησης (normalization)** των αξιωμάτων της. Σε αυτό το στάδιο τα αξιώματα της οντολογίας «αποσυντίθεται» σε ένα σύνολο κανονικοποιημένων μορφών έτσι ώστε να αποτελέσουν τις αρχικές σχέσεις βάσει των οποίων θα υπολογιστεί η πλήρης ιεραρχία των κλάσεων στο επόμενο στάδιο, γνωστό και ως **ταξινόμηση (classification)**. Το τελευταίο περιλαμβάνει την εξαντλητική (fix-point) εφαρμογή ενός αριθμού επαγωγικών κανόνων, οι οποίοι εξαγωγή τις υπονοούμενες σχέσεις βασιζόμενοι στη σημασιολογία των αξιωμάτων της οντολογίας. Στη δική μας περίπτωση, τόσο η αποθήκευση και διαχείριση της κανονικοποιημένης οντολογίας όσο και η εφαρμογή των επαγωγικών κανόνων γίνεται εξολοκλήρου από το DBMS.

Η τεχνική της Δομικής Υπαγωγής έχει αποδειχθεί ότι είναι ένας ορθός και πλήρης αλγόριθμος ταξινόμησης μιας οντολογίας για μια σειρά από τμήματα της Περιγραφικής Λογικής. Διαιρείται σε δύο βασικά στάδια:

- **Κανονικοποίηση (Normalization)** όπου τα αρχικά αξιώματα της οντολογίας μετατρέπονται σε ένα σύνολο κανονικοποιημένων μορφών βάσει των οποίων θα υπολογιστεί η πλήρης ιεραρχία των κλάσεων
- **Ταξινόμηση (Classification)** στην διάρκεια της οποίας γίνεται μία εξαντλητική εφαρμογή ενός συνόλου επαγωγικών κανόνων (fix-point) στην κανονικοποιημένη οντολογία με σκοπό της εξαγωγή όλων των υπονοούμενων σχέσεων ιεραρχίας.

Αυτό που θέλουμε να υπολογίσουμε είναι οι πιθανές σχέσεις ιεραρχίας μεταξύ των *ονοματισμένων εννοιών (named classes)*. Λόγω της φύσης των αξιωμάτων η παραπάνω διαδικασία δεν περιορίζεται απλώς στον υπολογισμό του μεταβατικού κλεισίματος, αλλά είναι υπολογιστικά πιο δύσκολη.

1.1.1 Συνεισφορά

Η συνεισφορά της διπλωματικής μας εργασίας θα μπορούσε να συνοψιστεί στην προσαρμογή μιας τεχνικής συλλογιστικής ανάλυσης, όπως είναι η Δομική Υπαγωγή, σε ένα σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS), καθιστώντας έτσι δυνατή την ταξινόμηση του Tbox οντολογιών χωρίς τη μεσολάβηση reasoner. Την λειτουργία αυτή καθιστά πολύ σημαντική το γεγονός ότι :

- Οι υπάρχουσες μεγάλο όγκου οντολογίες που χρησιμοποιούνται αυτή τη στιγμή, κυρίως ιατρικού περιεχομένου, έχουν πολύ μεγάλα και πολύπλοκα Tboxes.
- Μέχρι σήμερα η πλειονότητα των συστημάτων συλλογιστικής ανάλυσης καταφεύγει στην χρήση εξωτερικών reasoners που λειτουργούν αποκλειστικά στην κύρια μνήμη.
- Παρότι τα συστήματα αυτά έχουν καλή απόδοση, δεν επιδεικνύουν καλή κλιμάκωση (scaling) όσο ο όγκος του Tbox αυξάνει, ενώ επιπλέον χρειάζονται και καταναλώνουν πολύ κύρια μνήμη.
- Το σύστημα που δημιουργήσαμε, βασισμένο στην μέθοδο της Δομικής Υπαγωγής και εκμεταλευόμενο τις δυνατότητες των DBMS αποφεύγει τα παραπάνω προβλήματα επιτελώντας την ίδια λειτουργία, αλλά σε μικρότερους χρόνους και με καλύτερη χρήση της κύριας μνήμης.

1.2 Οργάνωση κειμένου

Η διπλωματική εργασία οργανώνεται στα παρακάτω κεφάλαια:

- Στο Κεφάλαιο 2 παρουσιάζεται το γενικό θεωρητικό υπόβαθρο μαζί με τις βασικές έννοιες που είναι απαραίτητες για την κατανόηση της εργασίας.
- Στο Κεφάλαιο 3 αναλύονται οι βασικές απαιτήσεις που ικανοποιεί το σύστημά μας. Η μεθοδολογία παρουσίασης περιλαμβάνει το χωρισμό του DBRS σε υποσυστήματα και τη συνοπτική περιγραφή των λειτουργιών που το καθένα από

αυτά επιτελεί. Στο τέλος του κεφαλαίου, δίνεται το μοντέλο Οντοτήτων-Συσχετίσεων της βάσης δεδομένων που χρησιμοποιεί εσωτερικά το DBRS.

- Στο Κεφάλαιο 4 αναφέρονται οι εφαρμογές που υλοποιούν τις λειτουργίες που περιγράφηκαν στο προηγούμενο κεφάλαιο. Στο τέλος, παρουσιάζεται το σχεσιακό διάγραμμα της βάσης δεδομένων του DBRS, το οποίο προέκυψε από την επεξεργασία του μοντέλου Οντοτήτων-Συσχετίσεων.
- Στο Κεφάλαιο 5 παρατίθεται η σχετική βιβλιογραφία.

2 *Θεωρητικό υπόβαθρο*

Στο κεφάλαιο αυτό δίνεται η περιγραφή εννοιών που αποτελούν το θεωρητικό υπόβαθρο της παρούσας εργασίας και που απαιτούνται για την κατανόησή της. Ήδη, στην εισαγωγή, έχουμε αναφερθεί στους όρους Σημασιολογικός Ιστός (ΣΙ) και μεταδεδομένα. Εδώ θα ασχοληθούμε με την έννοια της οντολογίας (ontology) και με μια οικογένεια λογικών φορμαλισμών που ονομάζεται Περιγραφική Λογική (Description Logic - DL). Θα εστιάσουμε το ενδιαφέρον μας σε μεθόδους συλλογιστικής ανάλυσης (reasoning) των μεταδεδομένων, δίνοντας έμφαση στον επικρατέστερο Tableau αλγόριθμο, καθώς και σε τεχνικές συνδυασμού τους με DBMSs. Οι δύο τελευταίες ενότητες αφορούν στις τυπικές γλώσσες (RDF και OWL) που χρησιμοποιούνται για την περιγραφή οντολογιών. Ο εξοικειωμένος αναγνώστης μπορεί να παραβλέψει αυτό το κεφάλαιο.

2.1 *Η έννοια της οντολογίας*

Η προσθήκη των μεταδεδομένων στο υπάρχον δικτυακό οικοδόμημα προϋποθέτει την οργάνωσή τους με τέτοιο τρόπο που να καθιστά τη διαχείρισή τους αποτελεσματική. Κλειδί στην επίτευξη αυτού του εγχειρήματος αποτελεί η χρήση των οντολογιών. Σαφής ορισμός για την γενικότερη έννοια της οντολογίας δεν υπάρχει και, μάλιστα, ο όρος διαφοροποιείται αρκετά ανάλογα με τον επιστημονικό τομέα στον οποίο χρησιμοποιείται. Ωστόσο, ένας ορισμός ο οποίος μπορεί να συνοψίσει όλη την

λειτουργικότητα της έννοιας της οντολογίας και έχει γίνει ευρέως αποδεκτός είναι του Tom Gruber⁴.

Μια οντολογία είναι μια τυπική (formal), κατηγορηματική (explicit) προδιαγραφή μιας διαμοιρασμένης (shared) εννοιολογικής αναπαράστασης (conceptualization).

Στον παραπάνω ορισμό ο όρος εννοιολογική αναπαράσταση (conceptualization) αναφέρεται σε ένα αφηρημένο μοντέλο φαινομένων του κόσμου στο οποίο έχουν προσδιοριστεί οι έννοιες που σχετίζονται με τα φαινόμενα αυτά, ο όρος κατηγορηματική (explicit) σημαίνει ότι το είδος των εννοιών που χρησιμοποιούνται και οι περιορισμοί που αφορούν την χρήση αυτών των εννοιών είναι προσδιορισμένα με σαφήνεια, ο όρος αυστηρή (formal) αναφέρεται στο ότι η οντολογία πρέπει να είναι μηχανικά αναγνώσιμη και ο όρος διαμοιρασμένη (shared) αναφέρεται στο ότι η οντολογία πρέπει να αποτυπώνει γνώση κοινής αποδοχής στα πλαίσια μιας κοινότητας.

Αν προσπαθήσουμε να συμπυκνώσουμε τον τρόπο με τον οποίο αντιλαμβανόμαστε τις οντολογίες στα πλαίσια του ΣΙ, τότε καταλήγουμε στο εξής: Τυπική περιγραφή ενός συνόλου πληροφοριών και των συσχετίσεων μεταξύ τους. Κάθε οντολογία μπορεί να θεωρηθεί ως σύνθεση δύο βασικών μερών:

- Ένα λεξιλόγιο (intensional knowledge) που αποτελείται από ονόματα εννοιών (concepts) και σχέσεων (relationships). Χρησιμοποιείται για να περιγράψει την πληροφορία ή αλλιώς τον «κόσμο» που μοντελοποιούμε και διαφοροποιείται ανάλογα με αυτόν⁵. Στην ορολογία της Περιγραφικής Λογικής, όπως θα δούμε παρακάτω, το τμήμα αυτό ονομάζεται TBox (Terminology Box).

- Ένα σύνολο επιπλέον γνώσης (extensional knowledge) σχετικά με τον «κόσμο», που περιλαμβάνει δηλώσεις/ισχυρισμούς (assertions). Οι δηλώσεις αντιστοιχίζουν

⁴ Αμερικανός πρωτοπόρος ερευνητής στα πεδία της αναπαράστασης γνώσης και της μηχανικής οντολογιών. http://en.wikipedia.org/wiki/Tom_Gruber

⁵ Για παράδειγμα, η μοντελοποίηση μιας ανθρώπινης οικογένειας απαιτεί διαφορετικές έννοιες και σχέσεις απ' ότι η μοντελοποίηση της γνώσης μας σχετικά με τις υπάρχουσες ποικιλίες κρασιών. Υπό αυτή την έννοια, κάθε οντολογία έχει το δικό της λεξιλόγιο.

άτομα (individuals) σε έννοιες και ζεύγη ατόμων ή ζεύγη ατόμου-σταθεράς (literal) σε σχέσεις. Αναφέρονται συχνά στη βιβλιογραφία και ως στιγμιότυπα (instances). Στην Περιγραφική Λογική, το τμήμα αυτό ονομάζεται ABox (Assertion Box) και σημειώνουμε ότι σε μια οντολογία μπορεί να απουσιάζει τελείως.

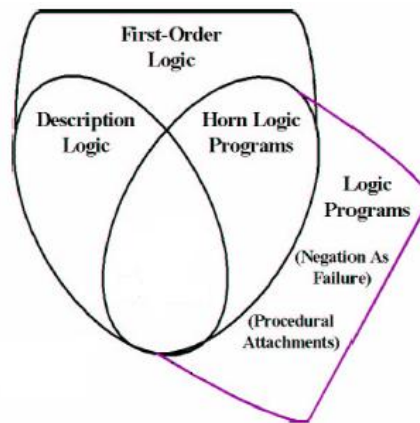
Τα άτομα της οντολογίας αποτελούν, στην ουσία, τα αντικείμενα που θέλουμε να διαχειριστούμε και συχνά είναι αναγνωριστικά (URIs) πόρων του διαδικτύου. Οι έννοιες του λεξιλογίου ισοδυναμούν με σύνολα ατόμων που μοιράζονται ένα τουλάχιστον κοινό χαρακτηριστικό και αναφέρονται συχνά ως κλάσεις (classes). Οι σχέσεις αντιστοιχούν σε σύνολα από ζεύγη ατόμων (ή ζεύγη ατόμου-σταθεράς) και ονομάζονται ιδιότητες (properties), επειδή ακριβώς προσδίδουν ιδιότητες στα άτομα συνδέοντάς τα μεταξύ τους (ή με κάποια σταθερά). Στην Περιγραφική Λογική, ονομάζονται ρόλοι (roles).

Οι οντολογίες χαρακτηρίζονται ως ο τυπικός (formal) προσδιορισμός των μεταδεδομένων ο οποίος εξασφαλίζει μια «κοινή αντίληψη» της περιοχής που μας ενδιαφέρει και παρέχει, δια της «φύσης» του, τη δυνατότητα συλλογιστικής ανάλυσης (reasoning), τόσο για την εξαγωγή νέων (υπονοούμενων) σχέσεων όσο και για τον έλεγχο της ισχύος ήδη υπάρχουσών (εύρεση αντιφάσεων). Η έννοια της τυπικότητας, όσον αφορά στον προσδιορισμό της μεταπληροφορίας, είναι καθοριστική για τη δυνατότητα των υπολογιστών να την αναλύσουν και άρα να διαχειριστούν αυτόματα και την ίδια την πληροφορία.

Η κατασκευή οντολογιών που μπορούν να «διαβαστούν» από υπολογιστές απαιτεί προφανώς την ύπαρξη τυπικών γλωσσών που θα χρησιμοποιούνται αποκλειστικά γι' αυτό το σκοπό και θα ικανοποιούν μια σειρά κριτηρίων. Ανάμεσα σ' αυτά είναι η ευκολία στη χρήση, η επαρκής εκφραστικότητα (expressivity), η δυνατότητα για συλλογιστική ανάλυση και φυσικά η συμβατότητα με τις ήδη υπάρχουσες τεχνολογίες του παγκόσμιου ιστού (π.χ. HTML, XML). Θα επανέλθουμε στα ζητήματα αυτά στις επόμενες ενότητες.

2.2 Περιγραφική Λογική (Description Logic - DL)

Η Περιγραφική Λογική (Description Logic - DL)⁶ είναι μια από τις θεωρίες αναπαράστασης γνώσης. Η εκφραστικότητά της, το τι δηλαδή μπορεί να περιγραφεί, περιορίζεται σε ένα τμήμα μιας άλλης θεωρίας που ονομάζεται Λογική Πρώτης Τάξης (First Order Logic - FOL). Μια τρίτη θεωρία αναπαράστασης γνώσης είναι ο Λογικός Προγραμματισμός (Logic Programs - LP) ο οποίος επίσης «μοιράζεται ένα κοινό κομμάτι» με τη Λογική Πρώτης Τάξης, γνωστό ως Λογική Horn (Horn Logic). Η έκταση της εκφραστικότητας των διαφόρων θεωριών αναπαράστασης γνώσης φαίνεται στο παρακάτω σχήμα.



Σχήμα 2.1: Γραφική αναπαράσταση εκφραστικότητας λογικών φορμαλισμών

Κάθε μια από τις προαναφερθείσες θεωρίες έχει και διαφορετική «φιλοσοφία» που αποτυπώνεται και στη σύνταξή της. Στη FOL η αναπαράσταση γίνεται με κατηγορήματα (predicates), μεταβλητές (variables) και σταθερές (constants), ενώ στον LP έχουμε κανόνες αιτίου-αιτιατού. Από την άλλη, η «φιλοσοφία» της DL είναι περισσότερο ανθρωποκεντρική (human-centred) και μοιάζει με το αντικειμενοστρεφές μοντέλο προγραμματισμού (object oriented programming - OOP). Η μοντελοποίηση

⁶ Συχνά ο όρος συναντάται και στον πληθυντικό (Description Logics) επειδή ακριβώς μπορεί να θεωρηθεί ως σύνολο λογικών φορμαλισμών (γλωσσών αναπαράστασης) που εντάσσονται στη θεωρία της Περιγραφικής Λογικής, αλλά έχουν διαφορετικές δυνατότητες ως προς την εκφραστικότητα της γνώσης που μπορούν να μοντελοποιήσουν.

στην DL γίνεται με έννοιες, ρόλους, άτομα και σταθερές (literals), όπως αυτά παρουσιάστηκαν στην προηγούμενη ενότητα.

Η DL, όπως κάθε θεωρία αναπαράστασης γνώσης, διαθέτει τρία βασικά στοιχεία:

- *Λεξιλόγιο (Vocabulary)*

Απαρτίζεται από ονόματα εννοιών (κλάσεων) και ρόλων. Οι έννοιες χρησιμοποιούνται για να ομαδοποιήσουν άτομα με κοινά χαρακτηριστικά και οι ρόλοι για να εκφράσουν συσχετίσεις μεταξύ τους ή μεταξύ αυτών και σταθερών. Όπως αναφέρθηκε και στην Ενότητα 2.1, το λεξιλόγιο διαφοροποιείται ανάλογα με τη γνώση που κάθε φορά μοντελοποιείται.

- *Συντακτικό (Syntax)*

Βασίζεται στους κατασκευαστές (constructors) και σε σύμβολα που εκφράζουν σχέσεις (π.χ. ιεραρχικές) μεταξύ στοιχείων του λεξιλογίου ή/και ατόμων. Οι κατασκευαστές δηλώνουν μια λειτουργία (operation) μεταξύ δύο ή περισσότερων εννοιών ή ρόλων (π.χ. ένωση). Ο συνδυασμός τους με τα σύμβολα του συντακτικού και τα στοιχεία του λεξιλογίου οδηγεί στην κατασκευή των αξιωμάτων (axioms). Η γνώση που μοντελοποιούμε με DL είναι, στην ουσία, ένα σύνολο αξιωμάτων.

- *Σημασιολογία (Semantics)*

Αφορά στην ερμηνεία των στοιχείων του λεξιλογίου, των κατασκευαστών και των συμβόλων του συντακτικού. Η σημασιολογία στην DL είναι συνολοθεωρητική: Μια έννοια ερμηνεύεται ως σύνολο από αντικείμενα, ένας ρόλος ως σύνολο από ζεύγη αντικειμένων και ένα άτομο ως αντικείμενο.

Ο όρος μοντελοποιημένη γνώση, που χρησιμοποιήσαμε στα προηγούμενα, αναφέρεται στις θεωρίες αναπαράστασης γνώσης ως Βάση Γνώσης (Knowledge Base - KB) και αποτελεί θεμελιώδη έννοια την οποία θα χρησιμοποιούμε στο εξής.

Βάση Γνώσης (KB) ονομάζουμε ένα σύνολο γνώσης που περιγράφεται (μοντελοποιείται) με χρήση ενός τυπικού φορμαλισμού ή αλλιώς μιας γλώσσα αναπαράστασης (description language). Τα βασικά δομικά στοιχεία μιας KB είναι οι ατομικές έννοιες (atomic concepts), οι ατομικοί ρόλοι (atomic roles) και τα άτομα (individuals). Σύνθετες εκφράσεις (complex descriptions) μπορούν να κατασκευαστούν από αυτά χρησιμοποιώντας τους κατασκευαστές (constructors) που προσφέρει η εκάστοτε γλώσσα αναπαράστασης.

Η βασική γλώσσα αναπαράστασης, την οποία επεκτείνουν όλες οι άλλες, είναι η \mathcal{AL} (Attributive Language) που περιγράφουμε στη συνέχεια. Συμβολίζουμε με A, B ατομικές έννοιες, με r, r_i ($i=1,2,\dots$) ατομικούς ρόλους, με C, C_i ($i=1,2,\dots$), D σύνθετες εκφράσεις και με a, b, c άτομα. Οι σύνθετες εκφράσεις στην \mathcal{AL} σχηματίζονται βάσει του παρακάτω συντακτικού κανόνα:

$C, D \rightarrow A \mid \top \mid \perp \mid \neg A \mid C \ D \mid \forall r.C \mid \exists r.T$, όπου

- \top *Καθολική έννοια* (Universal concept). Περιλαμβάνει όλα τα άτομα της KB
- \perp *Κενή έννοια* (Bottom concept). Δεν περιλαμβάνει κανένα άτομο της KB.
- $\neg A$ *Ατομική άρνηση* (Atomic negation). Η έννοια που περιλαμβάνει όλα εκείνα τα άτομα της KB που δεν ανήκουν στην ατομική έννοια A και μόνο αυτά.
- $C \ D$ *Τομή* (Intersection) δύο σύνθετων εννοιών. Η έννοια που περιλαμβάνει μόνο τα άτομα εκείνα της KB που ανήκουν και στις δύο αρχικές έννοιες C και D .
- $\forall r.C$ *Περιορισμός τιμής* (Value restriction). Η έννοια που περιλαμβάνει όλα εκείνα τα άτομα της KB που συνδέονται μέσω του ατομικού ρόλου r μόνο με άτομα που ανήκουν στην σύνθετη έννοια C .
- $\exists r.T$ *Περιορισμένη υπαρξιακή ποσοτικοποίηση* (Limited existential quantification). Η έννοια που περιλαμβάνει όλα εκείνα τα άτομα της KB που συνδέονται μέσω του ατομικού ρόλου r με ένα τουλάχιστον άτομο της καθολικής έννοιας, δηλαδή της KB.

Οι επεξηγήσεις που δώσαμε για κάθε ένα από τα στοιχεία-έννοιες που συναντάμε στο συντακτικό κανόνα της \mathcal{AL} αποτελούν στην ουσία μια ερμηνεία της σημασίας τους, δηλαδή μια απόδοση της σημασιολογίας τους με φυσική γλώσσα. Προκειμένου όμως να ορίσουμε αυστηρά αυτή τη σημασιολογία, εισάγουμε την έννοια των *διερμηνειών* (interpretations). Μια διερμηνεία I αποτελείται από μια δομή (Δ^I, \cdot^I) , όπου Δ^I το πεδίο της διερμηνείας (domain of interpretation) και \cdot^I μια συνάρτηση αντιστοίχισης⁷. Η συνάρτηση αυτή αντιστοιχίζει σε κάθε ατομική έννοια A ένα σύνολο $A^I \subseteq \Delta^I$ και σε κάθε ατομικό ρόλο r μια δυαδική σχέση $r^I \subseteq \Delta^I \times \Delta^I$. Επεκτείνεται στις σύνθετες εκφράσεις της \mathcal{AL} βάσει των παρακάτω:

$$\top^I = \Delta^I$$

$$\perp^I = \emptyset$$

$$(\neg A)^I = \Delta^I \setminus A^I$$

$$(C \sqcap D)^I = C^I \cap D^I$$

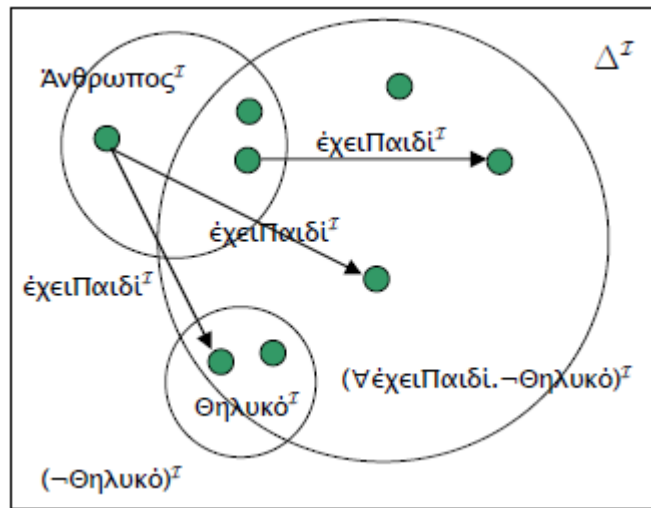
$$(\forall r.C)^I = \{a \in \Delta^I \mid \forall b. (a, b) \in r^I \rightarrow b \in C^I\}$$

$$(\exists r.C)^I = \{a \in \Delta^I \mid \exists b. (a, b) \in r^I \wedge b \in C^I\}$$

Όπως γίνεται αντιληπτό κάθε \mathcal{AL} έννοια ερμηνεύεται ως ένα υποσύνολο του Δ^I . Για παράδειγμα η έννοια \top ερμηνεύεται ως το σύνολο το οποίο περιέχει όλα τα αντικείμενα του χώρου ερμηνείας, ενώ η έννοια \perp ερμηνεύεται ως το κενό σύνολο, το οποίο και δικαιολογεί την ονομασία που τους έχουμε προσδώσει. Εν συνεχεία η έννοια $C \sqcap D$ ερμηνεύεται ως το σύνολο το οποίο προκύπτει από την τομή των ερμηνειών των εννοιών C και D . Επιπρόσθετα, η ερμηνεία της έννοιας $\forall r.C$ περιέχει το σύνολο των αντικειμένων του Δ^I τα οποία αν συμμετέχουν στο ρόλο r^I με κάποιο άλλο αντικείμενο, τότε το αντικείμενο αυτό ανήκει στην ερμηνεία της έννοιας C δηλαδή στο σύνολο C^I . Είναι πολύ σημαντικό στο σημείο αυτό να τονίσουμε ότι ένα αντικείμενο ανήκει στην ερμηνεία της έννοιας $\forall r.C$ ακόμα και αν δε σχετίζεται μέσω της σχέσης r^I με κανένα άλλο αντικείμενο. Έτσι λοιπόν, στο σύννηθες παράδειγμα με τις έννοιες μιας οικογένειας

⁷ Αυτός ο τυπικός ορισμός της σημασιολογίας είναι γνωστός στη βιβλιογραφία ως Tarski-style.

έχουμε ότι και η ερμηνεία της έννοιας $\text{Άνθρωπος} \sqcap \forall \text{έχειΠαιδί}. \neg \text{Θηλυκό}$ αποτελεί ένα σύνολο του Δ^I . Το σύνολο αυτό περιέχει τα αντικείμενα του Δ^I τα οποία ανήκουν ταυτόχρονα στην ερμηνεία της έννοιας Άνθρωπος , δηλαδή στο σύνολο Άνθρωπος^I , και αν συμμετέχουν στο ρόλο έχειΠαιδί^I με κάποιο άλλο αντικείμενο, τότε το αντικείμενο αυτό δεν ανήκει στο σύνολο αυτό που εμείς έχουμε αποδώσει ως ερμηνεία της έννοιας Θηλυκό . Στο παρακάτω σχήμα φαίνεται διαισθητικά η ερμηνεία της παραπάνω έννοιας.



Σχήμα 2.2 Γραφική αναπαράσταση της ερμηνείας της έννοιας $\text{Άνθρωπος} \sqcap \forall \text{έχειΠαιδί}. \neg \text{Θηλυκό}$

Κάθε KB που βασίζεται σε DL διαιρείται σε δύο κύρια μέρη: TBox και ABox⁸. Οι όροι μας είναι ήδη γνωστοί από την προηγούμενη ενότητα που αφορούσε στις οντολογίες. Εδώ θα τους περιγράψουμε ως σύνολα αξιωμάτων από την οπτική γωνία της Περιγραφικής Λογικής.

⁸ Η συγγένεια του όρου Βάση Γνώσης με αυτόν της Οντολογίας είναι κάτι παραπάνω από προφανές. Εκείνο που θέλουμε να σημειώσουμε εδώ είναι ότι, όταν (στα πλαίσια του ΣΙ) αναφερόμαστε σε οντολογίες, εννοούμε Βάσεις Γνώσεις (KBs) που έχουν «γραφτεί» με χρήση μιας τυπικής γλώσσας όπως οι RDF και OWL που θα δούμε στις Ενότητες 2.3 και 2.4.

- *TBox* (Terminology Box): Είναι εκείνο το σύνολο των αξιωμάτων της KB που αφορούν στην ορολογία, δηλαδή στο λεξιλόγιο που χρησιμοποιείται. Σε περίπτωση εκφραστικότητας AL^9 , το TBox περιλαμβάνει δύο ειδών αξιώματα:

- *Αξιώματα υπαγωγής εννοιών* (Concept inclusion axioms)

Όπως υποδηλώνει και το όνομά τους, εκφράζουν σχέσεις ιεραρχίας μεταξύ κλάσεων της KB και έχουν τη μορφή $C \sqsubseteq D$. Ένα αξίωμα υπαγωγής $C \sqsubseteq D$ δηλώνει ότι τα άτομα που ανήκουν στην έννοια-υπαγωγέας (subsumee) C ανήκουν, επίσης, και στην έννοια-υπαγόμενος (subsumer) D , αλλά όχι το αντίστροφο. Η αυστηρή σημασιολογία των αξιωμάτων αυτού του είδους ορίζεται ως εξής:

Μια διερμηνεία I ικανοποιεί το αξίωμα $C \sqsubseteq D$ ανν $C^I \subseteq D^I$

- *Αξιώματα Ισοδυναμίας εννοιών* (Concept equality axioms)¹⁰

Εκφράζουν σχέσεις ισοδυναμίας μεταξύ κλάσεων της KB και έχουν τη μορφή $C \equiv D$. Ένα αξίωμα ισοδυναμίας $C \equiv D$ δηλώνει ότι τα άτομα που ανήκουν στην έννοια C ανήκουν επίσης και στην έννοια D και αντιστρόφως. Η αυστηρή σημασιολογία των αξιωμάτων αυτού του είδους ορίζεται ως εξής:

Μια διερμηνεία I ικανοποιεί το αξίωμα $C \equiv D$ ανν $C^I \equiv D^I$

Αυτομάτως καταλαβαίνουμε από τη σημασιολογία τους ότι τα αξιώματα ισοδυναμίας μπορούν να εκφραστούν και ως αξιώματα υπαγωγής. Στην ουσία, κάθε αξίωμα ισοδυναμίας εκφράζεται με δύο αξιώματα υπαγωγής ως εξής¹¹:

$$C \equiv D \Leftrightarrow C \sqsubseteq D \text{ και } D \sqsubseteq C$$

⁹ Το εύρος της γνώσης μιας KB δηλώνεται με το όνομα της γλώσσας αναπαράστασης που χρησιμοποιείται.

¹⁰ Τα αξιώματα αυτά, όταν έχουν τη μορφή Όνομα έννοιας \equiv Σύνθετη έκφραση, αναφέρονται και ως αξιώματα ορισμού (definition axioms), θεωρώντας ότι ο ορισμός της έννοιας του πρώτου μέλους είναι η σύνθετη έκφραση του δεύτερου.

¹¹ Αυτό ισχύει μόνο αν στο TBox δεν έχουμε κύκλους (cycles), δηλαδή το όνομα μιας έννοιας δεν υπάρχει στον ορισμό της όπως π.χ. στο αξίωμα: Άνθρωπος \equiv Ζώο Π ΞΈχει_Πατέρα.Άνθρωπος.

- *ABox* (Assertion Box): Είναι εκείνο το σύνολο των αξιωμάτων που αφορούν στις δηλώσεις σχετικά με τα άτομα της KB. Σε περίπτωση εκφραστικότητας \mathcal{AL} , το *ABox* περιλαμβάνει τα εξής είδη αξιωμάτων

- *Αξιώματα δηλώσεων εννοιών* (Concept assertion axioms)

Εκφράζουν αντιστοιχήσεις ατόμων της KB σε έννοιες και έχουν τη μορφή $a : C$. Ένα αξίωμα $a : C$ δηλώνει ότι το άτομο a είναι τύπου C . Η αυστηρή σημασιολογία των αξιωμάτων αυτού του είδους ορίζεται ως εξής:

Μια διερμηνεία I ικανοποιεί το αξίωμα $a : C$ αν $a^I \in C^I$

- *Αξιώματα δηλώσεων ρόλων* (Role assertion axioms)

Εκφράζουν αντιστοιχήσεις ζευγών ατόμων σε ατομικούς ρόλους και έχουν τη μορφή $(a, b) : r$, όπου a υποκείμενο (subject) και b αντικείμενο (object) της δήλωσης. Ένα αξίωμα $(a, b) : r$ δηλώνει ότι τα άτομα a και b συνδέονται μεταξύ τους μέσω της σχέσης r . Η αυστηρή σημασιολογία των αξιωμάτων αυτού του είδους ορίζεται ως εξής:

Μια διερμηνεία I ικανοποιεί το αξίωμα $(a, b) : r$ αν $(a^I, b^I) \in r^I$

Συνοψίζοντας, μια διερμηνεία I που ικανοποιεί όλα τα αξιώματα του TBox, με τον τρόπο που ορίσαμε παραπάνω, λέμε ότι ικανοποιεί το TBox της KB και αποτελεί ένα μοντέλο (model) του. Αν επίσης ικανοποιεί κατ' αντιστοιχία και το ABox, τότε η I ονομάζεται μοντέλο της Βάσης Γνώσης.

Όπως αναφέραμε στα προηγούμενα, η εκφραστικότητα μιας γλώσσας αναπαράστασης εξαρτάται από τους κατασκευαστές που αυτή παρέχει. Προσθέτοντας λοιπόν νέους κατασκευαστές στην βασική γλώσσα AL , παίρνουμε γλώσσες μεγαλύτερης εκφραστικότητας. Παραδείγματα τέτοιων κατασκευαστών είναι:

- *Ένωση εννοιών* (Concept union)

Δηλώνεται με το γράμμα \cup . Η ένωση δύο σύνθετων εννοιών συμβολίζεται με $C \sqcup D$ και αποτελεί την έννοια εκείνη η οποία περιλαμβάνει όλα τα άτομα που ανήκουν είτε στην C είτε στην D είτε και στις δύο έννοιες μαζί. Η σημασιολογία της ορίζεται ως εξής:

$$(C \sqcup D)^I \Leftrightarrow C^I \cup D^I$$

- *Πλήρης υπαρξιακή ποσοτικοποίηση* (Full existential quantification)

Δηλώνεται με το γράμμα \exists . Η πλήρης υπαρξιακή ποσοτικοποίηση συμβολίζεται με $\exists r.C$ και αποτελεί την έννοια εκείνη η οποία περιλαμβάνει όλα εκείνα τα άτομα που συνδέονται, μέσω του ατομικού ρόλου r , με ένα τουλάχιστον άτομο της σύνθετης έννοιας C . Η σημασιολογία της ορίζεται ως εξής:

$$(\exists r.C)^I = \{a \in \Delta^I \mid \exists b. (a, b) \in r^I \wedge b \in C^I\}$$

- *Περιορισμός αριθμού* (Number restriction)¹²

Δηλώνεται με το γράμμα \mathcal{N} . Συμβολίζεται με $\geq nr$ και με $\leq nr$. Ο περιορισμός $\geq nr$ αποτελεί την έννοια που περιλαμβάνει όλα εκείνα τα άτομα που μετέχουν το λιγότερο (at least) n φορές στις δηλώσεις του ατομικού ρόλου r . Κατ' αντιστοιχία, ο περιορισμός $\leq nr$ αποτελεί την έννοια που περιλαμβάνει όλα εκείνα τα άτομα που μετέχουν το πολύ (at most) n φορές στις δηλώσεις του ατομικού ρόλου r . Η σημασιολογία τους ορίζεται ως εξής:

$$(\geq nr)^I = \{a \in \Delta^I \mid \#\{b \mid (a, b) \in r^I\} \geq n\}$$

$$(\leq nr)^I = \{a \in \Delta^I \mid \#\{b \mid (a, b) \in r^I\} \leq n\}$$

όπου ο συμβολισμός $\#\{ \quad \}$ δηλώνει την πληθικότητα των μελών ενός συνόλου (cardinality of set). Ένας ρόλος για τον οποίο ισχύει μέγιστος περιορισμός

¹² Αναφέρεται συχνά και ως περιορισμός μεγέθους συνόλου (cardinality restriction)

αριθμού $\leq nr$ με $n=1$ ονομάζεται *λειτουργικός ρόλος* (functional role) και η ύπαρξή του συμβολίζεται με \mathcal{F} .

- *Σύνθετη άρνηση* (Complex negation)¹³

Δηλώνεται με το γράμμα C και συμβολίζεται με $\neg C$. Η σύνθετη άρνηση $\neg C$ είναι η σύνθετη εκείνη έννοια που περιλαμβάνει όλα τα άτομα της KB που δεν ανήκουν στην έννοια C . Η σημασιολογία της ορίζεται ως εξής:

$$(\neg C)^I = \Delta^I \setminus C^I$$

Επεκτείνοντας την \mathcal{AL} με οποιοδήποτε υποσύνολο των παραπάνω κατασκευαστών, παίρνουμε μια νέα γλώσσα αναπαράστασης (μεγαλύτερης εκφραστικότητας από την αρχική) που ανήκει στη λεγόμενη οικογένεια \mathcal{AL} -γλωσσών. Για παράδειγμα, επεκτείνοντας την \mathcal{AL} με περιορισμό αριθμού, παίρνουμε τη γλώσσα \mathcal{ALN} . Εκείνο που αξίζει να σημειώσουμε εδώ είναι ότι ο συνδυασμός δύο κατασκευαστών μπορεί να προσφέρει εκφραστικότητα ίδια με αυτή ενός άλλου κατασκευαστή. Χαρακτηριστική περίπτωση αποτελεί η σύνθετη άρνηση που, όταν προστεθεί στη γλώσσα \mathcal{AL} , την εξοπλίζει με ένωση εννοιών, αφού όπως γνωρίζουμε από τη θεωρία συνόλων $\neg(C^I \cap D^I) = (\neg C)^I \cup (\neg D)^I$ επομένως $\neg(C \sqcap D) = (\neg C) \sqcup (\neg D)$. Έτσι, γράφουμε \mathcal{ALC} και όχι $\mathcal{AL\mathcal{E}C}$.

Άλλοι κατασκευαστές που προσδίδουν επιπλέον εκφραστικότητα είναι οι:

- *Ποιοτικός περιορισμός αριθμού* (Qualified number restriction)¹⁴

Δηλώνεται με το γράμμα Q και συμβολίζεται με $\geq nr.C$ και $\leq nr.C$. Ο περιορισμός $\geq nr.C$ αποτελεί την έννοια που περιλαμβάνει όλα εκείνα τα άτομα που μετέχουν ως υποκείμενα το λιγότερο (at least) n φορές στις δηλώσεις του ατομικού ρόλου r , έχοντας ως αντικείμενο άτομο της έννοιας C . Κατ' αντιστοιχία, ο περιορισμός $\leq nr.C$ αποτελεί την έννοια που περιλαμβάνει όλα

¹³ Αναφέρεται συχνά και ως πλήρης άρνηση (full negation)

¹⁴ Αναφέρεται συχνά και ως ποιοτικός περιορισμός μεγέθους συνόλου (qualified cardinality restriction)

εκείνα τα άτομα που μετέχουν ως υποκείμενα το πολύ (at most) n φορές στις δηλώσεις του ατομικού ρόλου r , έχοντας ως αντικείμενο άτομο της έννοιας C . Η σημασιολογία τους ορίζεται ως εξής:

$$(\geq nr.C)^I = \{a \in \Delta^I \mid \#\{b \mid (a, b) \in r^I \wedge b \in C^I\} \geq n\}$$

$$(\leq nr.C)^I = \{a \in \Delta^I \mid \#\{b \mid (a, b) \in r^I \wedge b \in C^I\} \leq n\}$$

- *Μεταβατικός ρόλος (Transitive role)*

Συμβολίζεται με $Tra(r)$ και, συνήθως, στη βιβλιογραφία η ύπαρξή του υποδηλώνεται με το γράμμα S που αποτελεί σύντμηση των \mathcal{AL} , C και μεταβατικών ρόλων. Ο $Tra(r)$ είναι εκείνος ο ρόλος του οποίου οι δηλώσεις έχουν μεταβατική ιδιότητα, δηλαδή μπορεί να σχηματίζουν μια «αλυσίδα» ή, όπως συχνά ονομάζεται, ένα μεταβατικό κλείσιμο (Transitive Closure - TC). Η σημασιολογία του μεταβατικού ρόλου ορίζεται ως εξής:

$$(Tra(r))^I = \{a, b, c \in \Delta^I \mid \{(a, b), (b, c)\} \subseteq r^I \rightarrow (a, c) \in r^I\}$$

- *Αντίστροφος ρόλος (Inverse role)*

Δηλώνεται με το γράμμα I και συμβολίζεται με r^- . Ο r^- είναι εκείνος ο ρόλος που περιλαμβάνει ακριβώς τα αντίστροφα ζεύγη ατόμων σε σχέση με τον ατομικό ρόλο r . Η σημασιολογία του r^- ορίζεται ως εξής:

$$(r^-)^I = \{a, b \in \Delta^I \mid (a, b) \in r^I \leftrightarrow (b, a) \in r^I\}$$

Ένας ρόλος r που είναι αντίστροφος με τον εαυτό του ονομάζεται συμμετρικός ρόλος (symmetric role).

- *Απαριθμητικά (Nominals)*

Δηλώνονται με το γράμμα O και συμβολίζονται με $\{a, b, \dots\}$, όπου a, b, \dots άτομα της KB. Χρησιμεύουν στον ορισμό απαριθμημένων εννοιών (enumerated concepts), δηλαδή εννοιών που ορίζονται ως ένα σύνολο συγκεκριμένων ατόμων της KB. Η σημασιολογία τους ορίζεται ως εξής: $(\{a\})^I = \{a^I\}$.

Σε εκφραστικές γλώσσες αναπαράστασης συναντάμε και άλλους κατασκευαστές όπως ένωση ρόλων (role union), τομή ρόλων (role intersection), ανακλαστικό ρόλο (reflexive role), μη ανακλαστικό ρόλο (irreflexive role) κ.α. Δεν έχουμε σκοπό να αναλύσουμε εξαντλητικά όλες τις περιπτώσεις. Εκείνο, όμως, που αξίζει να σημειώσουμε εδώ είναι ένα ενδιαφέρον είδος αξιωμάτων που δεν αναφέραμε μέχρι τώρα καθότι δεν υπάρχει στην *AL*. Ο λόγος για τα αξιώματα υπαγωγής ρόλων (role inclusion axioms), η ύπαρξη των οποίων σε μια *KB* δηλώνεται με το γράμμα *H*. Τα αξιώματα αυτά έχουν τη μορφή $r_1 \sqsubseteq r_2$, όπου r_1 και r_2 ατομικοί ρόλοι. Ένα αξίωμα $r_1 \sqsubseteq r_2$ δηλώνει ότι τα ζεύγη ατόμων του ρόλου r_1 είναι υποσύνολο των ζευγών του ρόλου r_2 , αλλά όχι το αντίστροφο. Η σημασιολογία των αξιωμάτων αυτού του είδους ορίζεται ως εξής:

Μια διερμηνεία I ικανοποιεί το αξίωμα $r_1 \sqsubseteq r_2$ αν $r_1^I \subseteq r_2^I$

Κατ' αντιστοιχία, ορίζονται και αξιώματα ισοδυναμίας ρόλων (role equivalence) με τον τρόπο που περιγράψαμε και για τα αξιώματα ισοδυναμίας των εννοιών στην *AL*.

Πολλά συστήματα διαχείρισης οντολογιών, όπως και το *DBRS*, χειρίζονται γλώσσες αναπαράστασης που υποστηρίζουν και *τύπους δεδομένων* (datatypes), όπως αυτοί ορίζονται αυστηρά με χρήση της *XML*. Η ύπαρξη τύπων δεδομένων και *τιμών δεδομένων* (literals) δηλώνεται με το γράμμα ^(D) και συνοδεύεται από ρόλους τύπων δεδομένων (datatype roles), οι οποίοι συνδέουν άτομα με σταθερές εκφράζοντας μια σχέση μεταξύ τω δύο.

Στον επόμενο πίνακα συνοψίζονται οι διάφοροι συμβολισμοί και οι αντίστοιχες εκφραστικότητες που συναντάμε στην Περιγραφική Λογική.

Συμβολισμός

Εκφραστικότητα - Κατασκευαστές

| | |
|------------------|--|
| \mathcal{A} | Ατομική άρνηση Τομή εννοιών Περιορισμός τιμής Περιορισμένη υπαρξιακή ποσοτικοποίηση |
| \mathcal{FL} | \mathcal{A} χωρίς ατομική άρνηση |
| \mathcal{FL}_o | \mathcal{FL} χωρίς περιορ. υπαρξιακή ποσοτικοποίηση |
| \mathcal{U} | Ένωση εννοιών |
| \mathcal{E} | Πλήρης υπαρξιακή ποσοτικοποίηση |
| \mathcal{N} | Περιορισμοί αριθμών |
| \mathcal{C} | Σύνθετη άρνηση |
| \mathcal{Q} | Ποιοτικοί περιορισμοί αριθμών |
| \mathcal{S} | \mathcal{A} , \mathcal{C} και μεταβατικοί ρόλοι |
| \mathcal{I} | Αντίστροφοι ρόλοι |
| \mathcal{O} | Απαριθμητικά |
| \mathcal{H} | Ιεραρχία ατομικών ρόλων |
| \mathcal{R} | Ανακλαστικοί ρόλοι Μη ανακλαστικοί ρόλοι Ασύμβατοι ρόλοι Αξιώματα υπαγωγής σύνθετων ρόλων |
| \mathcal{F} | Λειτουργικοί ρόλοι |
| \mathcal{D} | Τιμές δεδομένων Τύποι δεδομένων Ρόλοι τύπου δεδομένων |

Πίνακας 2.1: Σύμβολα εκφραστικότητας στην DL

2.3 Συλλογιστική ανάλυση στην Περιγραφική Λογική

Όπως καταστήσαμε σαφές και στα προηγούμενα, η Περιγραφική Λογική δεν αποσκοπεί μόνο στη μοντελοποίηση της γνώσης, αλλά και στην εύκολη ανάλυσή της. Η ανάλυση αυτή περιλαμβάνει δύο σκέλη. Το πρώτο αφορά στον έλεγχο της ισχύος των αξιωμάτων της KB, ενώ το δεύτερο στην εξαγωγή νέων αξιωμάτων που προκύπτουν από τα υπάρχοντα μέσω μιας αλγοριθμικής διαδικασίας.

Οι επαγωγικές διαδικασίες (inference procedures) σε μια KB εκφραστικότητας \mathcal{AL} διακρίνονται σε:

- *TBox επαγωγικές διαδικασίες*

Πρόκειται για επαγωγικές διαδικασίες που σχετίζονται με την ορολογία της KB και διακρίνονται στις ακόλουθες:

- *Ικανοποιησιμότητα (Satisfiability)*

Μια έννοια είναι ικανοποιήσιμη αν υπάρχει ένα μη κενό σύνολο από άτομα της KB τα οποία ανήκουν σε αυτή. Πιο αυστηρά, μια έννοια C είναι ικανοποιήσιμη, δεδομένου ενός TBox T , αν υπάρχει ένα μοντέλο I του T τέτοιο ώστε $C^I \neq \emptyset$. Σε αυτή την περίπτωση, η διερμηνεία I ονομάζεται και μοντέλο της C . Η ικανοποιησιμότητα μιας έννοιας C μπορεί να αναχθεί σε εύρεση της σχέσης υπαγωγής (βλ. αμέσως μετά): $\perp \sqsubseteq C$.

- *Υπαγωγή (Subsumption)*

Μια έννοια C υπάγεται σε μια έννοια D αν όλα τα άτομα που ανήκουν στη C ανήκουν και στη D , αλλά όχι το αντίστροφο. Πιο αυστηρά, μια έννοια C υπάγεται μιας έννοιας D αν, δεδομένου ενός TBox T , υπάρχει ένα μοντέλο I για το οποίο ισχύει $C^I \subseteq D^I$. Αν η υπαγωγή των εννοιών C και D δηλώνεται ρητά στο T ή μπορεί να εξαχθεί από αυτό με συλλογιστική, τότε γράφουμε: $T \models C \sqsubseteq D$.

○ *Ισοδυναμία* (Equivalence)

Μια έννοια C είναι ισοδύναμη με μια έννοια D αν όλα τα άτομα που ανήκουν στη C ανήκουν και στη D και αντίστροφα. Πιο αυστηρά, μια έννοια C είναι ισοδύναμη με μια έννοια D αν, δεδομένου ενός TBox T , υπάρχει ένα μοντέλο I για το οποίο ισχύει $C^I \equiv D^I$. Βάσει των όσων έχουμε πει μέχρι τώρα, η εύρεση σχέσης ισοδυναμίας μπορεί να αναχθεί σε εύρεση σχέσεων υπαγωγής. Αν η ισοδυναμία των εννοιών C και D δηλώνεται ρητά στο T ή μπορεί να εξαχθεί από αυτό με συλλογιστική, τότε γράφουμε: $T \models C \equiv D$.

○ *Ασυμβατότητα* (Disjointness)

Μια έννοια C είναι ασύμβατη με μια έννοια D αν δεν υπάρχουν άτομα που να ανήκουν και στις δύο μαζί. Πιο αυστηρά, μια έννοια C είναι ασύμβατη με μια έννοια D αν, δεδομένου ενός TBox T , υπάρχει ένα μοντέλο I για το οποίο ισχύει $C^I \cap D^I = \emptyset$. Επομένως, είναι φανερό ότι η εύρεση σχέσης ασυμβατότητας μπορεί να αναχθεί σε εύρεση της σχέσης υπαγωγής: $(C \sqcap D) \sqsubseteq \perp$.

Έως τώρα, είδαμε ότι οι επαγωγικές διαδικασίες του TBox μπορούν να αναχθούν όλες σε εύρεση σχέσεων υπαγωγής. Όμως, ένα ενδιαφέρον στοιχείο της Περιγραφικής Λογικής είναι ότι όλες μπορούν επίσης να αναχθούν σε εύρεση ικανοποιησιμότητας, δεδομένου ότι επιτρέπεται σύνθετη άρνηση (\neg). Υπό αυτές τις συνθήκες, δεδομένου ενός TBox T , ισχύουν:

Η έννοια C υπάγεται στην έννοια D αν $C \sqcap \neg D$ μη ικανοποιήσιμη

Οι έννοιες C και D είναι ισοδύναμες αν $C \sqcap \neg D$ και $\neg C \sqcap D$ μη ικανοποιήσιμες

Οι έννοιες C και D είναι ασύμβατες αν $C \sqcap D$ μη ικανοποιήσιμη

- *ABox επαγωγικές διαδικασίες*

Πρόκειται για επαγωγικές διαδικασίες που σχετίζονται με τις δηλώσεις της KB και διακρίνονται στις ακόλουθες:

- *Συνέπεια (Consistency)*

Ένα ABox είναι συνεπές αν, δεδομένου ενός TBox, δε διαθέτει αξιώματα που προκαλούν αντιφάσεις (contradictions). Πιο αυστηρά, ένα ABox A είναι συνεπές αν, δεδομένου ενός TBox T , υπάρχει ένα μοντέλο I που ικανοποιεί όλα τα αξιώματα του A .

- *Συνεπαγωγή (Entailment)*

Ένα ABox A συνεπάγεται ένα αξίωμα αν κάθε διερμηνεία I που ικανοποιεί το A , ικανοποιεί και το αξίωμα αυτό. Στην περίπτωση αυτή, για παράδειγμα σε μια δήλωση έννοιας $a : C$, γράφουμε $A \models a : C$.

Κατ' αντιστοιχία με τις αναγωγές στο TBox, η συνεπαγωγή σε ένα ABox μπορεί να αναχθεί σε συνέπεια του ABox ως εξής:

$$A \models a : C \text{ αν } A \cup \{a : \neg C\} \text{ μη συνεπές}$$

Από το τελευταίο, αντιλαμβανόμαστε ότι όλες οι επαγωγικές διαδικασίες του TBox μπορούν να αναχθούν εύκολα σε συνέπεια ABox. Για παράδειγμα, δυο σύνθετες έννοιες C και D είναι ασύμβατες αν, δεδομένου ενός TBox T και ενός υποτιθέμενου¹⁵ ατόμου a , για κάθε διερμηνεία I του ABox A , το $A \cup \{a : C \sqcap D\}$ είναι μη συνεπές. Την ιδέα αυτή, δηλαδή την αναγωγή όλων των επαγωγικών διαδικασιών σε εύρεση συνέπειας ABox, χρησιμοποιεί ο ευρύτατα διαδεδομένος Tableau αλγόριθμος που θα δούμε στην Παράγραφο .

¹⁵ Υποτιθέμενο με την έννοια ότι δεν υπάρχει εξαρχής στο ABox της KB, αλλά εισάγεται προσωρινά προκειμένου να αποδειχθεί το ζητούμενο.

2.3.1 Η μέθοδος της Προσυμπλήρωσης

Η βασική ιδέα της μεθόδου της προσυμπλήρωσης (precompletion) είναι η εξαγωγή όλης της γνώσης που προκύπτει από τις δηλώσεις ρόλων του ABox σε μορφή δηλώσεων εννοιών. Με άλλα λόγια, σκοπός της μεθόδου είναι η κατασκευή ενός, ισοδύναμου με το αρχικό, «ελεύθερου από ρόλους» (role-free) ABox.

Με τη διαδικασία απαλοιφής των δηλώσεων ρόλων, κάθε άτομο της KB απομονώνεται από τα υπόλοιπα, καθότι δε συνδέεται πλέον (μέσω κάποιου ρόλου) με κανένα άλλο. Οι δηλώσεις ρόλων σε μια KB αυξάνουν την πολυπλοκότητα της γνώσης και, κατά συνέπεια, δυσκολεύουν το έργο της ανάλυσής της. Στην περίπτωση που δεν εξάγουμε εξαρχής όλη την πληροφορία που προκύπτει από τις δηλώσεις ρόλων, χρειαζόμαστε πιο πολύπλοκους (και ίσως λιγότερο αποδοτικούς) αλγόριθμους συλλογιστικής ανάλυσης προκειμένου να χειριστούμε και τις σχέσεις των ατόμων. Αυτό το πρόβλημα ήταν που γέννησε την ιδέα της προσυμπλήρωσης.

Για να γίνει πιο κατανοητό, δίνουμε μια εκδοχή του αλγορίθμου σε γνώση εκφραστικότητας *SHF*. Ο αλγόριθμος βασίζεται στην εφαρμογή κανόνων (rules) στα αξιώματα της KB έτσι ώστε να απαλειφθούν, εφόσον δε θα χρειάζονται μετά το πέρας της διαδικασίας, οι δηλώσεις ρόλων του ABox. Παρακάτω ισχύουν οι συμβολισμοί που ακολουθήσαμε μέχρι τώρα για τις σύνθετες έννοιες και τους ατομικούς ρόλους. Οι κανόνες για εκφραστικότητα *SHF* είναι:

- $A \rightarrow \{a : C_2\} \cup A$ αν $a : C_1$ περιέχεται στο ABox A , $C_1 \sqsubseteq C_2$ περιέχεται στο TBox και $a : C_2$ δεν περιέχεται στο A . Ο κανόνας αυτός είναι γενικός.
- $A \rightarrow \{a : D\} \cup A$ αν $a : C_1 \sqcup C_2$ περιέχεται στο A , $D = C_1$ ή $D = C_2$ και καμία από τις δηλώσεις $a : C_1$, $a : C_2$ δεν περιέχεται στο A . Αυτός ο κανόνας χειρίζεται την ένωση εννοιών.
- $A \rightarrow \{b : C\} \cup A$ αν $a : \exists r.C$ περιέχεται στο A , $(a, b) : r_1$ υπάρχει στο A , r_1 απλός ρόλος αντικειμένου, $r \sim_A r_1$ και $b : C$ δεν περιέχεται στο A . Αυτός ο κανόνας χειρίζεται την πλήρη υπαρξιακή ποσοτικοποίηση.

- $A \rightarrow \{b : \forall r_2.C\} \cup A$ αν $a : \forall r_3.C$ περιέχεται στο A , $(a, b) : r_1$ περιέχεται στο A , R_2 μεταβατικός ρόλος, $r_1 \sqsubseteq r_2 \sqsubseteq r_3$ περιέχεται στο TBox και $b : \forall r_2.C$ δεν περιέχεται στο A . Ο κανόνας αυτός χειρίζεται περιορισμούς τιμών για μεταβατικούς ρόλους.
- $A \rightarrow \{a : C_1, a : C_2\} \cup A$ αν $a : C_1 \sqcap C_2$ περιέχεται στο A και κανένα από τα $a : C_1$, $a : C_2$ δεν περιέχεται στο A . Αυτός ο κανόνας χειρίζεται την τομή εννοιών.
- $A \rightarrow \{b : C\} \cup A$ αν $a : \forall r.C$ περιέχεται στο A , $(a, b) : r_1$ περιέχεται στο A , r_1 απλός ρόλος αντικειμένου, $r_2 \sqsubseteq r$ περιέχεται στο TBox, $r_1 \sim_A r_2$ και $b : C$ δεν περιέχεται στο A . Ο κανόνας αυτός χειρίζεται τους περιορισμούς τιμών.
- $A \rightarrow \{b : C\} \cup A$ αν $a : \forall r.C$ περιέχεται στο A , $(a, b) : r_1$ περιέχεται στο A , r_1 απλός ρόλος αντικειμένου, $r_1 \sqsubseteq r$ περιέχεται στο TBox και $b : C$ δεν περιέχεται στο A . Ο κανόνας αυτός χειρίζεται τους περιορισμούς τιμών.

Εισάγαμε το σύμβολο \sim_A προκειμένου να εκφράσουμε με συντομία μια ιδιαίτερη σχέση μεταξύ δύο ατομικών ρόλων. Το $r_1 \sim_A r_2$ δηλώνει ότι, αν $(a, b) : r_1$ και $(a, c) : r_2$, τότε b ταυτόσημο του c . Με άλλα λόγια, πρόκειται για το ίδιο άτομο με διαφορετικό όμως όνομα. Όπου στα προηγούμενα χρησιμοποιούμε τον συγκεκριμένο συμβολισμό, εννοούμε ακριβώς αυτό.

Η λογική ορθότητα των κανόνων που παραθέσαμε είναι προφανής. Θα προσπαθήσουμε να εξηγήσουμε με απλά λόγια έναν από αυτούς, τον κανόνα που χειρίζεται περιορισμούς τιμών για μεταβατικούς ρόλους. Υποθέτουμε ότι έχουμε μια KB και ισχύουν τα εξής:

r_2 ατομικός μεταβατικός ρόλος

$(a, b) : r_1$

$r_1 \sqsubseteq r_2$ και $r_2 \sqsubseteq r_3$

$a : \forall r_3.C$

Βάσει της σημασιολογίας που έχουμε ορίσει, η τελευταία σχέση σημαίνει ότι το άτομο a ανήκει στην έννοια που περιλαμβάνει όλα εκείνα τα άτομα της KB που

συνδέονται μέσω του ρόλου r_3 μόνο με άτομα που ανήκουν στην έννοια C . Επομένως, το άτομο a συνδέεται μέσω του r_3 μόνο με άτομο της έννοιας C . Επειδή όμως το άτομο a συνδέεται με το b μέσω του r_1 και ισχύει ότι $r_1 \sqsubseteq r_2$, αυτό σημαίνει ότι το άτομο a συνδέεται με το b μέσω και του ρόλου r_2 . Όμως, ο ρόλος r_2 είναι μεταβατικός πράγμα που σημαίνει ότι, αν ισχύουν όλα τα προηγούμενα και υποθέσουμε ότι το b συνδέεται μέσω του r_2 με ένα άτομο c , τότε και το a συνδέεται μέσω του r_2 με το c . Φτάσαμε λοιπόν σε ένα σημείο που για να διατηρείται η συνέπεια της KB πρέπει να ισχύει $b: \forall r_2.C$, δηλαδή το b να συνδέεται μέσω του r_2 μόνο με άτομο της έννοιας C . Διαφορετικά θα προέκυπτε αντίφαση, αφού το άτομο a θα συνδεόταν μέσω του r_3 ($r_2 \sqsubseteq r_3$) με άτομο που δεν ανήκει στην έννοια C .

Έχει αποδείχθηκε ότι, όποια κι αν είναι οι σειρά με την οποία εφαρμόζονται οι κανόνες, ο αλγόριθμος θα σταματήσει σε πεπερασμένο χρόνο έχοντας υπολογίσει την πλήρη πληροφορία. Ωστόσο, μια προσεκτική στρατηγική στην σειρά εφαρμογής των κανόνων μπορεί να βελτιώσει το συνολικό χρόνο της διαδικασίας.

Όταν ο αλγόριθμος σταματήσει, τα αξιώματα δηλώσεων ρόλων του $ABox$ είναι περιττά και μπορούν να απαλειφθούν, καθότι η πληροφορία που συνεπάγονται είναι πλέον ρητά δηλωμένη στη KB με τη μορφή αξιωμάτων δηλώσεων εννοιών. Έτσι, για παράδειγμα, οι τύποι κάθε ατόμου μπορούν να εξαχθούν εύκολα με συνδυασμό των ήδη υπολογισμένων δηλώσεων εννοιών για το συγκεκριμένο άτομο και μιας $TBox$ επαγωγικής διαδικασίας η οποία θα αφορά στην εύρεση σχέσεων υπαγωγής για τις έννοιες που εμφανίζονται σε αυτές τις δηλώσεις. Υπό αυτήν την έννοια, η προσυμπλήρωση μας απαλλάσσει από όλες τις $ABox$ επαγωγικές διαδικασίες. Ο λόγος για τον οποίο δεν επικράτησε είναι επειδή χαρακτηρίζεται από εκθετική πολυπλοκότητα $O(\exp(n))$ εξαιτίας των μη ντετερμινιστικών κανόνων που διαθέτει (π.χ. κανόνας που χειρίζεται την ένωση εννοιών).

2.3.2 Tableau αλγόριθμοι

Γενικά, οι Tableau αλγόριθμοι για την Περιγραφική Λογική (DL) προσπαθούν να αποδείξουν την ικανοποιησιμότητα μιας έννοιας C παράγοντας ένα μοντέλο αυτής, δηλαδή μια ερμηνεία I στην οποία το σύνολο που αντιστοιχεί στην έννοια C δεν είναι κενό ($C^I \neq \emptyset$). Tableau είναι ένας γράφος που αντιπροσωπεύει ένα μοντέλο όπου οι κόμβοι, που αναπαριστούνται με x , αντιστοιχούν σε άτομα και οι ακμές, που αναπαριστούνται από το όνομα του ρόλου (π.χ. r) αντιστοιχούν σε ατομικούς ρόλους. Κάθε κόμβος του tableau ανατίθεται σε ένα σύνολο εννοιών που αναπαριστούνται με $L(x)$. Αυτές είναι οι έννοιες στις οποίες ανήκει ο κόμβος στο συγκεκριμένο μοντέλο, ή με άλλα λόγια, στη συγκεκριμένη ολοκλήρωση (completion), και ανανεώνονται σταδιακά κατά την κατασκευή του μοντέλου.

Μια ενδιαφέρουσα ιδιότητα των κατασκευασμένων σε DL μοντέλων είναι το ότι είναι δέντρα (και όχι γράφοι). Με άλλα λόγια, εάν μια DL έννοια είναι ικανοποιήσιμη, κάτι το οποίο ισχύει για κάθε έννοια μιας \mathcal{ELH} οντολογίας, τότε όλα τα δυνατά της μοντέλα είναι της μορφής δέντρου. Αυτή είναι η καλούμενη ιδιότητα των μοντέλων-δέντρων (tree-model property) είναι ο κύριος λόγος που τα βασισμένα σε DL συστήματα συμπεριφέρονται καλά σε πρακτικές εφαρμογές.

Η δημιουργία των προαναφερθέντων μοντέλων σε σχήμα δέντρου επιτυγχάνεται μέσω μιας επαναληπτικής διαδικασίας όπου ένα σύνολο συμπερασματικών κανόνων (inference rules), οι οποίοι διαφέρουν ανάλογα με την εκφραστικότητα της οντολογίας, δηλαδή των κατασκευαστών που υποστηρίζονται, εφαρμόζονται ο ένας μετά τον άλλον σε όλα τα αξιώματα της οντολογίας, συμπεριλαμβάνοντας και εκείνα που παράχθηκαν σε όλα τα προηγούμενα βήματα του αλγορίθμου, μέχρις ότου (i) κανένας κανόνας δεν μπορεί να εφαρμοστεί πλέον ή (ii) βρεθεί μία αντίφαση στο μοντέλο.

Στην γενική περίπτωση, και αν παραλείψουμε τις διάφορες τεχνικές βελτιστοποίησης, η ιεραρχία της οντολογίας δημιουργείται αφού ελεγχθεί η σχέση υπαγωγής μεταξύ κάθε ζευγαριού από ονοματισμένες κλάσεις ορισμένες στο TBox, που σημαίνει ότι εάν έχουμε n ονόματα κλάσεων, τότε χρειαζόμαστε $O(n^2)$ συνολικούς ελέγχους. Στις βασισμένες σε Tableau διαδικασίες, κάθε έλεγχος υπαγωγής της μορφής

$C_1 \rightarrow C_2$ μεταφράζεται σε δύο ελέγχους ικανοποιησιμότητας για τις περιγραφές εννοιών (i) $C_1 \rightarrow C_2$ και (ii) $C_2 \rightarrow C_1$ αντίστοιχα.

Στο πλαίσιο αυτό, αν η πρώτη περιγραφή δεν είναι ικανοποιήσιμη για κάθε δυνατό μοντέλο της οντολογίας, τότε το $C_1 \rightarrow C_2$ ισχύει. Με το ίδιο σκεπτικό, αν η δεύτερη περιγραφή δεν είναι ικανοποιήσιμη σε κάθε δυνατό μοντέλο, τότε το $C_2 \rightarrow C_1$ ισχύει. Τέλος, στην περίπτωση που και οι δύο περιγραφές δεν είναι ικανοποιήσιμες σε κάθε δυνατό μοντέλο της οντολογίας, τότε οι δύο κλάσεις είναι ισοδύναμες, δηλαδή $C_1 \equiv C_2$.

Για να μπορέσουμε να δώσουμε έμφαση στον τρόπο με τον οποίο οι βασισμένοι σε Tableau αλγόριθμοι πραγματοποιούν την διαδικασία ταξινόμησης, πρέπει πρώτα να εισαγάγουμε τους ακόλουθους συμπερασματικούς κανόνες:

1. - κανόνας: αν η έννοια $C_1 \rightarrow C_2 \in L(x) \& \{C_1, C_2\} \in L(x)$, τότε $L(x) = L(x) \cup \{C_1, C_2\}$
2. - κανόνας: αν η έννοια $C_1 \rightarrow C_2 \in L(x) \& \{C_1, C_2\} \in L(x)$, τότε $L(x) = L(x) \cup \{C_1\}$ ή $L(x) = L(x) \cup \{C_2\}$
3. - κανόνας: αν η έννοια $r.C \in L(x) \& (y, r_1 \rightarrow r)$ τέτοια ώστε $r_1(x, y) \in C \in L(y)$, τότε δημιουργείστε καινούργιο κόμβο y με $L(y) = \{C\}$ και προσθέστε τον ισχυρισμό ρόλου (role assertion) $r(x, y)$ στο μοντέλο
4. - κανόνας: αν η έννοια $r.C \in L(x) \& (y, r_1 \rightarrow r)$ τέτοια ώστε $r_1(x, y) \in C \in L(y)$, τότε $L(y) = L(y) \cup \{C\}$

Προφανώς οι κανόνες αυτοί εφαρμόζονται στο σύνολο εννοιών που έχουν ανατεθεί σε κάθε άτομο (κόμβο) του Tableau.

Εκτός από το γεγονός ότι οι βασισμένοι σε Tableau αλγόριθμοι έχουν πολυπλοκότητα εκθετική στην χειρότερη περίπτωση (είναι διαδικασίες διάψευσης-refutation procedures και έτσι παράγουν μη ντετερμινιστικά συμπεράσματα ακόμα και για αρκετά απλά TBoxes), το κύριο μειονέκτημά τους είναι ότι καταναλώνουν πολύ μνήμη. Ακόμα και αν το μέγεθος του αρχικού TBox χωράει στην κύρια μνήμη, το μέγεθος του κατασκευασμένου μοντέλου μπορεί γίνει απαγορευτικά μεγάλο (κυρίως λόγω των “φρέσκων” ατόμων που εισάγονται από τον \rightarrow -κανόνα) και να αποτραπεί έτσι

η εφαρμογή των συμπερασματικών κανόνων από το να φτάσει σε ένα σταθερό σημείο(fix-point) ή από το να αναγνωρίσει μία αντίφαση.

Οι Tableau βασισμένες υλοποιήσεις έχουν σαν στόχο εκφραστικές OWL οντολογίες (ειδικά τα μη βατά κομμάτια της γλώσσας) όπου μπορούν να προκύψουν ασυνέπειες στο μοντέλο.

2.3.2.1 Παράδειγμα εκτέλεσης

Έστω ότι έχουμε μια \mathcal{ELH} οντολογία με το ακόλουθο TBox που αποτελείται από δύο αξιώματα κλάσης και ένα αξίωμα ιδιότητας.

$$MuscularOrgan \equiv Organ \sqcap \exists isPartOf.MuscularSystem \quad (1)$$

$$Heart \sqsubseteq Organ \sqcap \exists belongsTo.(MuscularSystem \sqcap CirculatorySystem) \quad (2)$$

$$belongsTo \sqsubseteq isPartOf \quad (3)$$

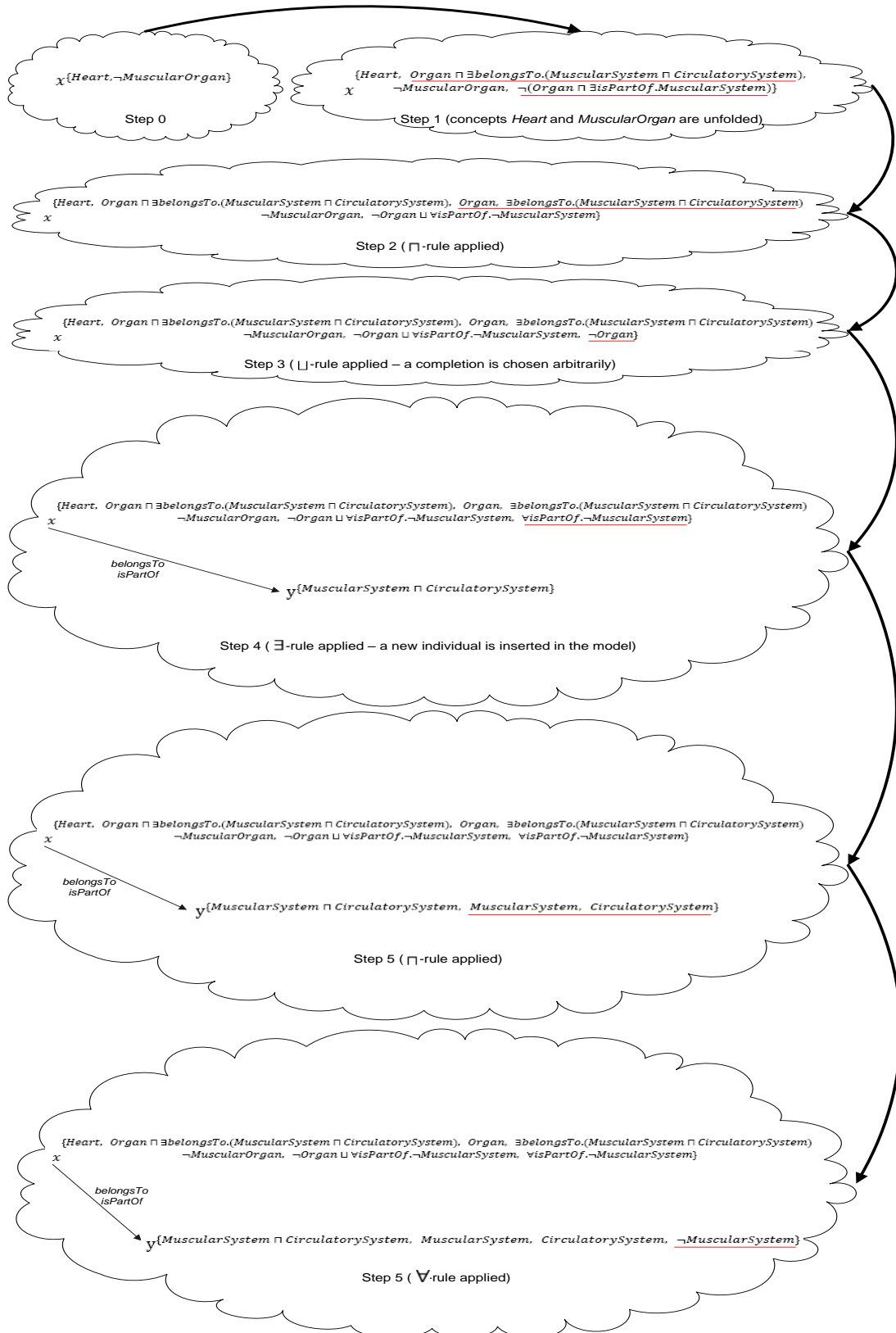
Στη φυσικά γλώσσα, τα παραπάνω τρία αξιώματα σημαίνουν τα εξής:

1. Το πρώτο αξίωμα δείχνει ότι “το MuscularOrgan είναι ένα Organ που είναι επίσης μέρος του MuscularSystem”.
2. Το δεύτερο αξίωμα δείχνει ότι “η Heart είναι ένα Organ που ανήκει και στο MuscularSystem και στο CirculatorySystem”.
3. Το τρίτο αξίωμα δείχνει ότι “αν δύο άτομα συνδέονται μέσω της ιδιότητας belongsTo τότε συνδέονται επίσης με την ιδιότητα isPartOf”.

Για την κατανόηση του αλγορίθμου tableau ας υποθέσουμε ότι πρέπει να ελέγξουμε την σχέση υπαγωγής $Heart \sqsubseteq MuscularOrgan$. Τα βήματα του βασισμένου σε Tableau αλγορίθμου απεικονίζονται στο Σχήμα 2.3. Οι έννοιες που προστέθηκαν σε κάθε βήμα της ολοκλήρωσης είναι οι υπογραμμισμένες με κόκκινο χρώμα.

Στο Step 3, ο μη ντετερμινιστικός \sqsubseteq -κανόνας διαλέγει μία δυνατή ολοκλήρωση όπου ο κόμβος x ανήκει τόσο στην κλάση $Organ$ όσο και στο συμπλήρωμά της. Αυτό είναι προφανώς μία αντίφαση, οπότε ο αλγόριθμος γυρίζει πίσω(backtracks) και συνεχίζει με μία εναλλακτική ολοκλήρωση που δεν παράγει αντίφαση (τουλάχιστον όχι σε αυτό το

βήμα). Τέλος, στο βήμα 5, ο \neg -κανόνας αναγκάζει τον κόμβο y να γίνει μέλος τόσο της κλάσης *MuscularOrgan* όσο και του συμπληρώματός της, οπότε αντιμετωπίζεται και πάλι αντίφαση και εφόσον ο αλγόριθμος δεν μπορεί να γυρίσει αυτή τη φορά πίσω σε ένα εναλλακτικό μοντέλο (δεν υπάρχουν άλλα δυνατά μοντέλα), η διαδικασία τερματίζει και η περιγραφή έννοιας $C_1 \neg C_2$ αποδεικνύεται ότι είναι μη ικανοποιήσιμη σε κάθε μοντέλο του δοσμένου TBox. Ας σημειωθεί ότι αυτό δεν σημαίνει ότι το $C_1 \neg C_2$ ισχύει. Δεν είμαστε σίγουροι ότι το C_1 υπάγεται στο C_2 εκτός αν ελέγξουμε επίσης για το $C_2 \neg C_1$ κατά τον ίδιο τρόπο. Ο έλεγχος αυτός παραλείπεται εδώ λόγω έλλειψης χώρου.



Σχήμα 2.3 Ο αλγόριθμος Tableau για το παράδειγμα εκτέλεσης

2.3.3 Βασισμένοι σε Datalog Αλγόριθμοι

Μία εναλλακτική λύση στο πρόβλημα ταξινόμησης μιας οντολογίας είναι η μετάφραση των αξιωμάτων της οντολογίας σε Datalog πρόγραμμα και η εκτέλεση του συλλογισμού χρησιμοποιώντας μία Datalog μηχανή. Ακολουθώντας αυτή την προσέγγιση, και σε περίπτωση που η οντολογία που θέλουμε να ταξινομήσουμε είναι μεγάλη σε μέγεθος, που σημαίνει ότι μπορεί να είναι δύσκολη η διαχείριση του TBox αποκλειστικά στην κύρια μνήμη, μπορεί να χρησιμοποιήσουμε επίσης τεχνικές προσανατολισμένες στον δίσκο (disk-oriented techniques) και βελτιστοποιήσεις στον χώρο των Παραγωγικών Συστημάτων Βάσεων Δεδομένων (Deductive Database Systems) έτσι ώστε να παρέχουμε πιο εξελικτικούς (scalable) αλγορίθμους συλλογισμού.

Για τους σκοπούς του παραδείγματός μας, οι αντιστοιχίες μεταξύ μίας οντολογίας (εκφρασμένης σε DL) και των αντίστοιχων datalog κανόνων απεικονίζονται στο Πίνακα 2.2.

| DL Syntax | Datalog Syntax | Meaning |
|--|-----------------------|---|
| Ατομικές Έννοιες (Atomic Concepts) A | $P(X)$ | μια ατομική έννοια αντιστοιχεί σε ένα κατηγορημα P . Άτομα τύπου A αντιστοιχούνται σε σταθερές που ικανοποιούν το κατηγορημα P . |
| Ατομικοί Ρόλοι (Atomic Roles) r | $R(X, Y)$ | ένας ατομικός ρόλος αντιστοιχεί σε ένα κατηγορημα R δύο μεταβλητών. Άτομα που συνδέονται μέσω του ρόλου r αντιστοιχούνται σε σταθερές που ικανοποιούν το κατηγορημα R . |
| Τομή Εννοιών (Concept Intersection) $C_1 \quad C_2$ | $P_1(X) \quad P_2(X)$ | άτομα που ανήκουν σε δύο ή παραπάνω έννοιες αντιστοιχούνται σε σταθερές που ικανοποιούν όλα τα αντίστοιχα κατηγορήματα |

| | | |
|---|--|--|
| <p>Υπαρξιακός Περιορισμός (Existential Restriction)</p> <p>$r.C$</p> | <p>$R(X,Y) \quad P_c(Y)$</p> | <p>άτομα της έννοιας $r.C$ αντιστοιχούνται σε σταθερές που ικανοποιούν το κατηγορημα R μαζί με σταθερές που ικανοποιούν επίσης το κατηγορημα P_c. Το κατηγορημα P_c αντιστοιχεί στην έννοια C, ενώ το κατηγορημα R αντιστοιχεί στον ρόλο r.</p> |
| <p>Υπαγωγή Κλάσης (Class Subsumption)</p> <p>$C_1 \quad C_2$</p> | <p>$P_2(X) \quad P_1(X)$</p> | <p>τετριμμένο</p> |
| <p>Υπαγωγή Ρόλου (Role Subsumption)</p> <p>$r_1 \quad r_2$</p> | <p>$R_2(X,Y)$ $R_1(X,Y)$</p> | <p>τετριμμένο</p> |
| <p>Ισχυρισμός Κλάσης (Class Assertion)</p> <p>$C(a)$</p> | <p>$P(a)$</p> | <p>τετριμμένο</p> |
| <p>Ισχυρισμός Ρόλου (Role Assertion)</p> <p>$r(a,b)$</p> | <p>$R(a,b)$</p> | <p>τετριμμένο</p> |

Πίνακας 2.2 Αντιστοιχίες από DL σε Datalog για το κομμάτι της

Έχοντας μεταφράσει μία οντολογία (όποτε δυνατόν – δείτε παρακάτω) σε ένα datalog πρόγραμμα, μπορούμε εύκολα να αποφασίσουμε για την σχέση υπαγωγής μεταξύ δύο εννοιών¹⁶, έστω C_1 και C_2 , με τον ακόλουθο τρόπο. Έστω P_1 και P_2 τα κατηγορήματα που αντιστοιχούν στις έννοιες C_1 και C_2 αντίστοιχα. Θέλουμε να ελέγξουμε αν η C_1 υπάγεται στην C_2 . Τα βήματα της διαδικασίας είναι:

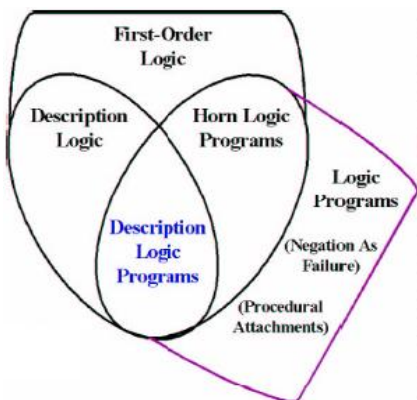
¹⁶ Η σχέση υπαγωγής μεταξύ δύο ρόλων μπορεί εύκολα να αναγνωριστεί μέσω μιας απλής επανάληψης (recursion).

1. Ορίζουμε μια σταθερά a η οποία ικανοποιεί το κατηγορημα P_1 , δηλαδή εισάγουμε ένα νέο γεγονός $P_1(a)$
2. Αξιολογούμε το ερώτημα $P_2(a)$?
3. Ορίζουμε μια σταθερά b που ικανοποιεί το κατηγορημα P_2 , δηλαδή εισάγουμε ένα νέο γεγονός $P_2(b)$
4. Αξιολογούμε το ερώτημα $P_1(b)$?

Προφανώς, σε περίπτωση που το ερώτημα $P_2(a)$? δώσει θετική απάντηση και το ερώτημα $P_1(b)$? δώσει αρνητική, τότε η έννοια C_1 υπάγεται στην έννοια C_2 , δηλαδή το $C_1 \subseteq C_2$ ισχύει.

Η Datalog βασίζεται στην Horn Λογική (Horn Logic) η οποία, ακριβώς όπως η Περιγραφική Λογική, είναι ένα αποφασίσιμο (decidable) υποσύνολο της Λογικής Πρώτης Τάξης (First-Order Logic). Όπως φαίνεται Σχήμα 2.4, τα αντίστοιχα κομμάτια έχουν ένα “κοινό μέρος”, που ονομάζεται Προγράμματα Περιγραφικής Λογικής (Description Logic Programs), στα οποία μπορεί να οριστεί μια αντιστοιχία ένα-προς-ένα μεταξύ των δύο φορμαλισμών. Δυστυχώς, έξω από αυτήν την “περιοχή”, η προαναφερθείσα αντιστοιχία δεν είναι απλή (αν και εφικτή).

Πίσω στο παράδειγμα εκτέλεσης που ορίστηκε στην ενότητα 2.3.2.1, μπορούμε εύκολα να διαπιστώσουμε ότι το αξίωμα $\text{MuscularOrgan} \quad \text{Organ} \quad \text{isPartOf.MuscularSystem}$ μπορεί να μεταφραστεί στον κανόνα $\text{MuscularOrgan}(X) \quad \text{Organ}(X) \quad \text{PartOf}(X,Y) \quad \text{MuscularSystem}(Y)$, αλλά το αντίστροφο, δηλαδή το αξίωμα $\text{MuscularOrgan} \quad \text{Organ} \quad \text{isPartOf.MuscularSystem}$, δεν μπορεί να μεταφραστεί σε έναν ασφαλή κανόνα horn (safe horn rule), και ως εκ τούτου δεν μπορεί να εκφραστεί στη datalog. Με άλλα λόγια, το πρώτο TBox αξίωμα του παραδείγματος εκτέλεσης (και το δεύτερο επίσης) δεν μπορούν να μεταφραστούν σε έναν έγκυρο κανόνα για μια datalog μηχανή.



Σχήμα 2.4 Λογικοί Φορμαλισμοί

Μια πρόσφατη δουλειά εισάγει μια επέκταση της datalog, που ονομάζεται $\text{datalog}^{\text{neg}}$, η οποία καλύπτει το DL^{neg} κομμάτι της DL και έτσι μπορεί να χρησιμοποιηθεί για την ταξινόμηση μιας DL^{neg} οντολογίας. Παρόλα αυτά, η προσέγγιση που παρουσιάζεται δεν έχει υλοποιηθεί ως ένα συγκεκριμένο σύστημα.

Ένα πλεονέκτημα της βασισμένης σε datalog προσέγγισης σε σχέση με τους βασισμένους σε Tableau αλγορίθμους είναι ότι δεν περιέχουν μη ντετερμινιστικά συμπεράσματα και έτσι έχουν πολυωνμική στην χειρότερη περίπτωση πολυπλοκότητα σε σχέση με το μέγεθος των δεδομένων.

Επειδή έχουν σχεδιαστεί για την επίτευξη πολυωνμικής στην χειρότερη περίπτωση πολυπλοκότητας, οι βασισμένοι σε datalog αλγόριθμοι μπορούν να εφαρμοστούν μόνο σε ορισμένα περιορισμένα κομμάτια της Περιγραφικής Λογικής. Πέρα από αυτά τα κομμάτια, η δημιουργία προγραμμάτων datalog που να είναι ισοδύναμα με τις οντολογίες που δίνονται ως είσοδο απαιτεί περίπλοκες λογικές μετατροπές. Για παράδειγμα, πιο εκφραστικές OWL οντολογίες, π.χ. αυτές που υποστηρίζουν την ένωση (\cup) εννοιών, μπορούν να αντιστοιχιστούν μόνο σε μία διαζευκτική επέκταση της datalog, δηλαδή σε προγράμματα datalog που περιέχουν κανόνες με μία ή παραπάνω προτάσεις ένωσης (union clauses) στις κεφαλές τους.

2.3.4 Αλγόριθμοι βασιμμένοι στην Δομική Υπαγωγή

Μία εναλλακτική πρόταση για τη λύση του προβλήματος της ταξινόμησης είναι η εφαρμογή αλγορίθμων βασιμμένων στην τεχνική της Δομικής Υπαγωγής (structural subsumption). Με απλά λόγια, οι αλγόριθμοι που ανήκουν σε αυτή την οικογένεια εκτελούν την ταξινόμηση της οντολογίας ταυτόχρονα εφαρμόζοντας επαναληπτικά, μέχρι να φτάσουν σε ένα συγκεκριμένο σημείο, ένα σύνολο από ορθούς και πλήρεις συμπερασματικούς κανόνες (inference rules) που καθορίζονται ανάλογα με τη δομή των αξιωμάτων που υπάρχουν στο Tbox. Γενικά, οι αλγόριθμοι δομικής υπαγωγής δεν εφαρμόζονται στο αρχικό TBox αλλά σε μία κανονικοποιημένη εκδοχή του, η οποία προκύπτει μετά από μία διαδικασία κανονικοποίησης (normalization).

Στα επόμενα υποκεφάλαια θα παρουσιαστούν μία σειρά από αλγορίθμους ταξινόμησης βασιμμένους στην τεχνική της Δομικής Υπαγωγής για διάφορες γλώσσες αναπαράστασης.

2.3.4.1 Δομική Υπαγωγή στη οικογένεια

Σε πρώτη φάση θα μελετήσουμε έναν αλγόριθμο υπαγωγής βασιμμένο στην τεχνική της Δομικής Υπαγωγής για την γλώσσα αναπαράστασης FL^- . Δεδομένου ότι με N_C συμβολίζεται το σύνολο των ατομικών εννοιών, με N_R το σύνολο των ατομικών ρόλων η σύνταξη και C, D είναι σύνθετες έννοιες, η σύνταξη και η σημασιολογία της γλώσσας αναπαράστασης FL^- συνοψίζεται στον παρακάτω πίνακα.

| Κατασκευαστής | Σύνταξη | Σημασιολογία |
|--|---------|---|
| TOP – καθολική έννοια | T | Δ^I |
| Ατομική έννοια $P \in N_C$ | P | $P^I \subseteq \Delta^I$ |
| Τομή | $C \ D$ | $C^I \ D^I$ |
| Περιορισμένη υπαρξιακή ποσοτικοποίηση, $r \in N_R$ | $. T$ | $\{ a \in \Delta^I \mid b \in (a, b) \in r^I \}$ |
| Περιορισμός τιμής, $r \in N_R$ | $r.C$ | $\{ a \in \Delta^I \mid b \in (a, b) \in r^I \ \& \ y \in C^I \}$ |

Πίνακας 2.3 Σύνταξη και σημασιολογία της FL^-

Ο αλγόριθμος λειτουργεί σε 2 φάσεις. Τα αρχικά αξιώματα μετατρέπονται στα λογικά ισοδύναμα αυτών σε κανονική μορφή ακολουθώντας τους παρακάτω κανόνες:

- Όλες οι φωλιασμένες τομές ενώνονται σε γραμμικό χρόνο

$$A (B C) > A B C$$

- Όλες οι τομές περιορισμών τιμής ενοποιούνται

$$r.C \quad r.D > r.(C D)$$

Μετά την κανονικοποίηση όλες οι έννοιες θα έχουν παρασταθεί ως αλυσίδα τομών. Έστω ότι $C = C_1 \dots C_n$ και $D = D_1 \dots D_m$ έννοιες όπως έχουν προκύψει από την κανονικοποίηση. Ο αλγόριθμος βασισμένος στην τεχνική της Δομικής Υπαγωγής $\text{SUBS?}[C,D]$ (δηλαδή $D \sqsubseteq C$) επιστρέφει TRUE αν και μόνο αν για κάθε C_i ισχύουν:

1. Αν το C_i είναι ατομική έννοια ή περιορισμένη υπαρξιακή ποσοτικοποίηση $r.T$, τότε υπάρχει ένα D_j τέτοιο ώστε $C_i = D_j$.
2. Αν το C_i είναι μία σύνθετη έννοια της μορφής $r.C'$, τότε υπάρχει ένα $r.D'$ (με ίδια ατομική έννοια r) τέτοιο ώστε $\text{SUBS?}[C',D']$ να είναι TRUE

Έχει αποδειχθεί ότι ο παραπάνω αλγόριθμος είναι tractable (σε πολυωνυμικό χρόνο) και συγκεκριμένα μπορεί να ολοκληρωθεί σε $O(n^2)$ όπου n είναι το μήκος της μεγαλύτερης παραμέτρου.

Για την κατανόηση του παραπάνω αλγορίθμου θα χρησιμοποιήσουμε ένα παράδειγμα εκτέλεσης. Ας θεωρήσουμε την παρακάτω οντολογία FL⁻ η οποία στην φυσική γλώσσα θα μπορούσε να περιγραφεί ως εξής:

- “Πατέρας ενήλικων παιδιών είναι ο άνδρας που έχει τουλάχιστον ένα παιδί και όλα του τα παιδιά είναι ενήλικα”
- “Πατέρας ενήλικων αγοριών είναι ο άνδρας που έχει τουλάχιστον ένα παιδί, όλα του τα παιδιά είναι ενήλικα και όλα τα παιδιά του είναι αγόρια”

Με βάση τον φορμαλισμό της FL η παραπάνω οντολογία θα μπορούσε να περιγραφεί ως εξής:

$$Fatherwithadultchildren = Male \quad hasChild.T \quad hasChild.Adult \quad (1)$$

$$Fatherwithadultboys = Male \quad hasChild.T \\ hasChild.Adult \quad hasChild.Male \quad (2)$$

Μετά την εφαρμογή των κανόνων της κανονικοποίησης το αξίωμα (2) έχει μετατραπεί στο ακόλουθο.

$$Fatherwithadultboys = Male \quad hasChild.T \quad hasChild.(Adult \quad ale) \quad (3)$$

Ο αλγόριθμος της υπαγωγής που περιγράφεται παραπάνω επιστρέφει TRUE στο ερώτημα $SUBS?[C (= Fatherwithadultchildren), D (= Fatherwithadultboys)]$ εφ'όσον

$$C_1 = D_1 (=Male) \text{ και } C_2 = D_2 (= hasChild.T) \\ \text{και } SUBS?[C'_3 (=Adult), D'_3 (=Adult \quad ale)] = TRUE$$

Ένας παρόμοιος αλγόριθμος έχει ισχύ και για την πιο εκτεταμένη περιγραφική λογική , η οποία περιέχει επιπλέον εκφραστικότητα εξαιτίας της ύπαρξης του κατασκευαστή της πλήρους υπαρξιακής ποσοτικοποίησης ($r.C - \{ x \quad \Delta^I / \quad y: (x,y) \quad r^I \quad y \quad C^I \}$) αλλά δεν είναι πλέον tractable. Έχει αποδειχθεί ότι το πρόβλημα της υπαγωγής στην είναι co-NP-complete.

2.3.4.2 Δομική Υπαγωγή στην οικογένεια .

Μέσα στο επόμενο κεφάλαιο θα περιγράψουμε την τεχνική της Δομικής Υπαγωγής στην γλώσσα αναπαράστασης , και κατ'επέκταση και στις υπογλώσσες αυτής και . Κύριο χαρακτηριστικό της γλώσσας αναπαράστασης είναι η ταυτόχρονη ύπαρξη τόσο των κατασκευαστών πλήρους υπαρξιακής ποσοτικοποίησης και περιορισμού αριθμού όσο και των κατασκευαστών της τομής, της ατομικής άρνησης και του περιορισμού τιμής. Δεδομένου ότι με N_C συμβολίζεται το σύνολο των ατομικών

εννοιών, με N_R το σύνολο των ατομικών ρόλων η σύνταξη και C, D είναι σύνθετες έννοιες, η σύνταξη και η σημασιολογία της γλώσσας αναπαράστασης (και των υπογλωσσών) συνοψίζεται στους παρακάτω πίνακες.

| Κατασκευαστής | Σύνταξη | | | |
|---|--------------|---|---|---|
| TOP – καθολική έννοια | \top | x | x | |
| BOTTOM – κενή έννοια | \perp | x | x | x |
| Ατομική έννοια $P \in N_C$ | P | x | x | x |
| Ατομική άρνηση, $P \in N_C$ | $\neg P$ | x | x | x |
| Τομή | $C \sqcap D$ | x | x | x |
| Πλήρης υπαρξιακή ποσοτικοποίηση, $r \in N_R$ | $r.C$ | x | x | |
| Περιορισμός τιμής, $r \in N_R$ | $r.C$ | x | x | |
| Περιορισμός αριθμού, $r \in N_R, n$ | (nr) | x | | x |
| Περιορισμός αριθμού, $r \in N_R, n$ | (nr) | x | | x |

Πίνακας 2.4 Σύνταξη της περιγραφής εννοιών

| Σύνταξη | Σημασιολογία |
|----------------------|---|
| \top | Δ^I |
| \perp | |
| P | $P^I \subseteq \Delta^I$ |
| $\neg P$ | $\Delta^I \setminus P^I$ |
| $C \sqcap D$ | $C^I \sqcap D^I$ |
| $r.C$ | $\{a \in \Delta^I \mid b: (a,b) \in r^I \wedge b \in C^I\}$ |
| $r.C$ | $\{a \in \Delta^I \mid b: (a,b) \in r^I \wedge b \in C^I\}$ |
| $(nr), r \in N_R, n$ | $\{a \in \Delta^I \mid \#\{b: (a,b) \in r^I\} = n\}$ |
| $(nr), r \in N_R, n$ | $\{a \in \Delta^I \mid \#\{b: (a,b) \in r^I\} = n\}$ |

Πίνακας 2.5 Η σημασιολογία της περιγραφής εννοιών

Όπως έχει ήδη επισημανθεί, για την εφαρμογή ενός αλγορίθμου βασισμένου στην τεχνική της Δομικής Υπαγωγής, είναι συνήθως αναγκαίο η μετατροπή των αρχικών αξιωμάτων της οντολογίας σε μία ισοδύναμη κανονική μορφή. Και στην περίπτωση της γλώσσας αναπαράστασης χρειάζεται να προηγηθεί μία διαδικασία κανονικοποίησης. Η περιγραφή μιας – σύνθετης έννοιας C είναι σε κανονική μορφή όταν όλες οι τομές $C_I \quad C_n$ που εμφανίζονται στο C πληρούν τις ακόλουθες προϋποθέσεις:

1. Περιέχουν το πολύ ένα περιορισμό αριθμού της μορφής (nr)
2. Περιέχουν το πολύ ένα περιορισμό αριθμού της μορφής (nr)
3. Περιέχουν το πολύ ένα περιορισμό τιμής της μορφής $r.C$

Για να επιτευχθεί η κανονική μορφή εφαρμόζονται επαναληπτικά και εξαντλητικά σε γραμμικό χρόνο οι παρακάτω κανόνες:

- $(mr) \quad (nr) \quad (nr)$ εάν $n = m$,
- $(mr) \quad (nr) \quad (nr) \quad m$, και
- $r.C \quad r.C \quad r.(C \quad D)$

Σε αυτό το σημείο είναι χρήσιμη η εισαγωγή μιας σειράς από συντομεύσεις που θα διευκολύνουν την κωδικοποίηση του αλγορίθμου της Δομικής Υπαγωγής. Με την προϋπόθεση ότι η σύνθετη έννοια C είναι σε κανονική μορφή μπορούμε να αναπαραστήσουμε ως:

- $\text{prim}(C)$ το σύνολο των ατομικών εννοιών που εμφανίζονται στο υψηλότερο επίπεδο του C
- $\text{min}_r(C) = \max\{k \mid C \sqsubseteq (kr)\}$ (Σημείωση: είναι πάντα πεπερασμένο)¹⁷
- $\text{max}_r(C) = \min\{k \mid C \sqsubseteq (kr)\}$ (Σημείωση: αν δεν υπάρχει τέτοιο k τότε $\text{max}_r(C) = \infty$)
- $\text{val}_r(C) = C'$ εάν υπάρχει περιορισμός τιμής της μορφής $r.C'$ στο υψηλότερο επίπεδο του C , αλλιώς $\text{val}_R(C) = \top$

¹⁷ Το $\text{min}_r(C)$ θα πρέπει να τονιστεί ότι δεν προκύπτει αποκλειστικά από τους περιορισμούς αριθμών (mr) απλώς η ύπαρξη αυτού διευκολύνει τη διαδικασία και μειώνει την πολυπλοκότητα του υπολογισμού (στη χειρότερη περίπτωση υπολογίζονται σε χρόνο πολυωνυμικό ως προς το μέγεθος του C). Αν υπάρχει (mr) στο C για τον υπολογισμό του $\text{min}_r(C)$ θα εξεταστούν τα k για τα οποία ισχύει $m = k \cdot \text{max}\{m, |\text{ex}_r(C)|\}$ σε διαφορετική περίπτωση για τα k όπου $0 < k < |\text{ex}_r(C)|$. Π.χ. για το $C = r.(A \quad r.(\neg A \quad \text{έχουμε ότι } \text{min}_r(C)=2$.

- $\text{exr}_r(C) = \{ C' \mid \text{εάν υπάρχουν } C \sqsubseteq C' \text{ στο υψηλότερο επίπεδο του } C \}$
- π ένα υπαρξιακό διαχώρισμα (existential partition) του συνόλου $\text{exr}_r(C) = \{C_1, \dots, C_m\}$ τέτοιο ώστε να ισχύει α . Η πληθικότητα του π είναι $n = \min\{\max_r(C), |\text{exr}_r(C)|\}$ και β . $\bigvee_{C' \in S} C' \sqsubseteq \text{val}_r(C) \neq \perp$ για όλα τα $S \in \pi$.¹⁸
- $\Gamma_r(C)$ το σύνολο των υπαρξιακών διαχωρισμάτων του C
- $\text{exr}_r(C)^\pi = \{ \bigvee_{C' \in S} C' \mid S \in \pi \}$
- $\kappa_r(C) = \min_r(\bigvee_{C' \in \text{exr}_r(C)} r.C')$
- $\text{exr}_r(C)^* = \bigvee_{\pi \in \Gamma_r(C)} \text{exr}_r(C)^\pi$

Με την βοήθεια των παραπάνω συμβολισμών πλέον είναι εφικτή η κωδικοποίηση ενός θεωρήματος βασισμένου πάνω στην τεχνική της Δομικής Υπαγωγής, ο οποίος δίνει την απάντηση για το πρόβλημα υπαγωγής 2 σύνθετων – εννοιών.

Έστω C, D οι περιγραφές δύο – σύνθετων εννοιών με $\text{exr}_r(C) = \{C_1, \dots, C_m\}$. Τότε $C \sqsubseteq D$ αν και μόνο αν $C \equiv \perp$ και $D \equiv \top$, ή ισχύουν τα ακόλουθα:

1. $\text{prim}(D) \subseteq \text{prim}(C)$
2. $\max_r(C) \leq \max_r(D)$
3. $\min_r(C) \geq \min_r(D)$
4. για κάθε $D' \in \text{exr}_r(D)$ ισχύει ότι
 - (α) $\text{exr}_r(C) = \emptyset$, $\min_r(C) \geq 1$ και $\text{val}_r(C) \sqsubseteq D'$ ή
 - (β) $\text{exr}_r(C) \neq \emptyset$ και για κάθε $\pi \in \Gamma_r(C)$ υπάρχει $C' \in \text{exr}_r(C)^\pi$ τέτοιο ώστε $C' \sqsubseteq \text{val}_r(C) \sqsubseteq D'$
5. εάν $\text{val}_r(D) \neq \top$, τότε
 - (α) $\max_r(C) = 0$ ή
 - (β) $\kappa_r(C) < \max_r(C)$ και $\text{val}_r(C) \sqsubseteq \text{val}_r(D)$ ή
 - (γ) $0 < \kappa_r(C) = \max_r(C) \text{val}_r(C) \sqsubseteq \text{val}_r(D)$ για όλα τα $C' \in \text{exr}_r(C)^*$

Έχει αποδειχτεί ότι το παραπάνω θεώρημα δίνει ορθές και πλήρεις απαντήσεις στο πρόβλημα της υπαγωγής δύο σύνθετων – εννοιών και θα μπορούσε να αποτελέσει την βάση για την δημιουργία ενός πρακτικού αλγορίθμου δομικής υπαγωγής.

¹⁸ Διαχώρισμα ενός συνόλου S είναι ένα σύνολο $\{S_1, \dots, S_n\}$ από μη κενά υποσύνολα $S_i \subseteq S$ του S τέτοια ώστε τα S_i είναι ανα ζεύγη disjoint και $S = S_1 \sqcup \dots \sqcup S_n$.

2.3.4.3 Δομική Υπαγωγή στην οικογένεια

Πρόσφατα, έχει αποδειχθεί ότι μία μικρή σχετικά περιγραφική λογική η \mathcal{EL} , η οποία υποστηρίζει την τομή και την πλήρη υπαρξιακή ποσοτικοποίηση, έχει αξιοσημείωτες αλγοριθμικές επιδόσεις στην ταξινόμηση εννοιών. Η περιγραφική λογική \mathcal{EL} συνδέεται με το profile OWL 2 EL της OWL 2, στο οποίο τα σημαντικότερα προβλήματα συλλογιστικής αποφασίζονται σε πολυωνυμικό χρόνο (tractable). Επομένως το πρόβλημα της υπαγωγής στην \mathcal{EL} παραμένει tractable τόσο με κυκλικό όσο και με ακυκλικό TBox όσο και με την παρουσία GCIs (general concept inclusion - γενικευμένες υπαγωγές εννοιών) σε αντίθεση με άλλες γλώσσες αναπαράστασης όπως η \mathcal{FLo} .

Στη συνέχεια θα εξετάσουμε διάφορους αλγορίθμους βασισμένους στην τεχνική της Δομικής Υπαγωγής για διάφορες επεκτάσεις της \mathcal{EL} ξεκινώντας από τις απλές (\mathcal{ELH}) και καταλήγοντας σε πιο σύνθετες ($\mathcal{EL}++$).

2.3.4.3.1 Δομική Υπαγωγή στην

Η σύνταξη και η σημασιολογία της \mathcal{ELH} δεδομένου της ύπαρξης ενός συνόλου ατομικών εννοιών N_C και ενός συνόλου ατομικών ρόλων N_R φαίνονται στους παρακάτω πίνακες τόσο για τις έννοιες όσο και για τα αξιώματα.

| Έννοιες | Σύνταξη | Σημασιολογία |
|--|--------------|---------------------------------------|
| TOP – καθολική έννοια | \top | Δ^I |
| Ατομική έννοια $P \in N_C$ | P | $P^I \subseteq \Delta^I$ |
| Τομή | $C \sqcap D$ | $C^I \cap D^I$ |
| Πλήρης υπαρξιακή ποσοτικοποίηση, $\exists \in N_R$ | $r.C$ | $\{ a \in \Delta^I \mid b \in C^I \}$ |

Πίνακας 2.6 Σύνταξη και σημασιολογία των εννοιών της \mathcal{ELH}

| Αξιώματα | Σύνταξη | Σημασιολογία |
|--|-------------------|---------------------|
| Υπαγωγή εννοιών (concept inclusion) | $C \sqsubseteq D$ | $C^I \subseteq D^I$ |
| Υπαγωγή ρόλων (role inclusion) | $r \sqsubseteq s$ | $r^I \subseteq s^I$ |

Πίνακας 2.7 Σύνταξη και σημασιολογία των αξιωμάτων της \mathcal{ELH}

Και στην περίπτωση των \mathcal{EL} οντολογιών είναι απαραίτητη μια διαδικασία κανονικοποίησης καθώς ο αλγόριθμος βασισμένος στην Δομική Υπαγωγή εφαρμόζεται σε μία κανονικοποιημένη μορφή του αρχικού TBox, η διαδικασία κανονικοποίησης *αναθέτει νέα ονόματα στις σύνθετες περιγραφές εννοιών* που εφαρμόζονται στα αξιώματα, και έχει ως αποτέλεσμα ένα καινούργιο TBox T όπου κάθε αξίωμα έχει μία από τις ακόλουθες μορφές:

- $A \sqsubseteq B$
- $A_1 \sqcap A_2 \sqsubseteq B$
- $A \sqsubseteq \exists r.B$
- $\exists r.A \sqsubseteq B$
- $r \sqsubseteq s$

Ας σημειωθεί ότι το βήμα κανονικοποίησης απαιτεί πολυωνιμικό χρόνο σε σχέση με το μέγεθος του αρχικού TBox. Πέρνοντας υπόψην αυτούς τους πέντε τύπους κανονικοποιημένων αξιωμάτων, οι συμπερασματικοί κανόνες για το \mathcal{ELH} κομμάτι της Περιγραφικής Λογικής είναι οι ακόλουθοι:

1. Αν $A \in T$ τότε $T = T \cup \{A \sqsubseteq A, A \sqsubseteq \top\}$
2. Αν $A \sqsubseteq B \in T$ και $B \sqsubseteq C \in T$, τότε $T = T \cup \{A \sqsubseteq C\}$
3. Αν $A \sqsubseteq B \in T$ και $A \sqsubseteq C \in T$ και $B \sqcap C \sqsubseteq D \in T$, τότε $T = T \cup \{A \sqsubseteq D\}$
4. Αν $A \sqsubseteq B \in T$ και $B \sqsubseteq \exists r.C \in T$, τότε $T = T \cup \{A \sqsubseteq \exists r.C\}$
5. Αν $A \sqsubseteq \exists r.B \in T$ και $r \sqsubseteq s \in T$, τότε $T = T \cup \{A \sqsubseteq \exists s.B\}$
6. Αν $A \sqsubseteq \exists r.B \in T$ και $B \sqsubseteq C \in T$ και $\exists r.C \sqsubseteq D \in T$, τότε $T = T \cup \{A \sqsubseteq D\}$

Ο κανόνας 1 εφαρμόζεται μόνο μία φορά, στην αρχή κάθε διαδικασίας, ενώ οι εναπομείναντες κανόνες εφαρμόζονται επαναληπτικά σε όλα τα αξιώματα, συμπεριλαμβανομένων αυτών που παράχθηκαν σε προηγούμενα βήματα του αλγορίθμου TBox T , δηλαδή μέχρι το T να μην αλλάζει μετά από μία πλήρη εφαρμογή κάθε συμπερασματικού κανόνα σε όλα τα αξιώματα που περιλαμβάνει στο τρέχον βήμα (fix-point).

Για να γίνει πιο κατανοητό το πως δουλεύει ο αλγόριθμος, θα χρησιμοποιήσουμε το παράδειγμα εκτέλεσης που ορίστηκε στην ενότητα 2.3.2.1. Αρχικά, η διαδικασία κανονικοποίησης αλλάζει το αρχικό TBox στο ακόλουθο:

| | |
|--|--------------------------------|
| $MuscularOrgan \sqsubseteq Organ$ | (από το 1 ^ο αξίωμα) |
| $MuscularOrgan \sqsubseteq \exists isPartOf.MuscularSystem$ | (από το 1 ^ο αξίωμα) |
| $Organ \sqcap \exists isPartOf.MuscularSystem \sqsubseteq MuscularOrgan$ | (από το 1 ^ο αξίωμα) |
| $Heart \sqsubseteq Organ$ | (από το 2 ^ο αξίωμα) |
| $Heart \sqsubseteq \exists belongsTo.MCSystem$ | (από το 2 ^ο αξίωμα) |
| $MuscularSystem \sqcap CirculatorySystem \sqsubseteq MCSystem$ | (από το 2 ^ο αξίωμα) |
| $MCSystem \sqsubseteq MuscularSystem$ | (από το 2 ^ο αξίωμα) |
| $MCSystem \sqsubseteq CirculatorySystem$ | (από το 2 ^ο αξίωμα) |
| $belongsTo \sqsubseteq isPartOf$ | (3 ^ο αξίωμα) |

Στη συνέχεια, οι συμπερασματικοί κανόνες εφαρμόζονται στο T το οποίο σταδιακά αυξάνεται με τα ακόλουθα αξιώματα (αποτελέσματα του πρώτου κανόνα είναι):

| | |
|--|---|
| $Heart \sqsubseteq \exists belongsTo.MuscularSystem$ | (παράχθηκε από τον 4 ^ο κανόνα) |
| $Heart \sqsubseteq \exists isPartOf.MuscularSystem$ | (παράχθηκε από τον 5 ^ο κανόνα) |
| $Heart \sqsubseteq MuscularOrgan$ | (παράχθηκε από τον 3 ^ο κανόνα) |

Τα παραπάνω αξιώματα προστίθενται στο T το ένα μετά το άλλο. Συγκεκριμένα, το πρώτο αξίωμα πάει σαν είσοδος στον 5^ο κανόνα για να παραχθεί το δεύτερο αξίωμα, ενώ το δεύτερο αξίωμα πάει σαν είσοδος στον 3^ο κανόνα για να παραχθεί το τρίτο. Ας σημειωθεί, ότι σε αυτή την περίπτωση, μόνο μία από τις τρεις σχέσεις υπαγωγής που πέρνουμε σαν έξοδο παράγεται κατά την δεύτερη φάση ($Heart \sqsubseteq MuscularOrgan$). Τα άλλα δύο αξιώματα ($MuscularOrgan \sqsubseteq Organ$, $Heart \sqsubseteq Organ$) παράγονται κατά την κανονικοποίηση του TBox.

2.3.4.3.2 Δομική Υπαγωγή στην \mathcal{EL}^+

Η περιγραφική λογική \mathcal{EL}^+ αποτελεί μία επέκταση της \mathcal{EL} με την προσθήκη αξιωμάτων της μορφής γενικευμένων υπαγωγών εννοιών (general concept inclusion – GCIs) και επιπλέον εκφραστικότητα στα αξιώματα υπαγωγής ρόλων. Η σύνταξη και η σημασιολογία τόσο των εννοιών όσο και των αξιωμάτων της \mathcal{EL}^+ φαίνεται στους παρακάτω πίνακες.

| Έννοιες | Σύνταξη | Σημασιολογία |
|--|--------------|---|
| TOP – καθολική έννοια | \top | Δ^I |
| Ατομική έννοια $P \sqsubseteq N_C$ | P | $P^I \subseteq \Delta^I$ |
| Τομή | $C \sqcap D$ | $C^I \cap D^I$ |
| Πλήρης υπαρξιακή ποσοτικοποίηση, $r \sqsubseteq N_R$ | $r.C$ | $\{x \in \Delta^I \mid \exists y: (x,y) \in r^I \wedge y \in C^I\}$ |

Πίνακας 2.8 Σύνταξη και σημασιολογία των εννοιών της \mathcal{EL}^+

| Αξιώματα | Σύνταξη | Σημασιολογία |
|--|---|---|
| Γενικευμένη υπαγωγή εννοιών (general concept inclusion- GCI) ¹⁹ | $C \sqsubseteq D$ | $C^I \subseteq D^I$ |
| Υπαγωγή ρόλων (role inclusion - RI) ²⁰ | $r_1 \circ \dots \circ r_n \sqsubseteq s$ | $r_1^I \circ \dots \circ r_n^I \subseteq s^I$ |

Πίνακας 2.9 Σύνταξη και σημασιολογία των αξιωμάτων της \mathcal{EL}^+

Μία \mathcal{EL}^+ οντολογία αποτελείται από ένα πεπερασμένο σύνολο αξιωμάτων γενικευμένης υπαγωγής εννοιών και αξιωμάτων υπαγωγής ρόλων. Το πρόβλημα που επιζητά λύση είναι το πρόβλημα της ταξινόμησης (classification), δηλαδή ο υπολογισμός όλης της ιεραρχίας των σχέσεων υπαγωγής των ατομικών εννοιών που περιλαμβάνονται σε μία οντολογία. Τη λύση σε αυτό το πρόβλημα το δίνει ο αλγόριθμος που ακολουθεί βασισμένος στην τεχνική της Δομικής Υπαγωγής.

Όπως και στις προηγούμενες περιπτώσεις το πρώτο στάδιο του αλγορίθμου περιλαμβάνει την μετατροπή του TBox σε κανονική μορφή μέσω μιας διαδικασίας κανονικοποίησης που γίνεται σε χρόνο γραμμικό ως προς το μέγεθος του TBox. Στο καινούργιο TBox T κάθε αξίωμα έχει μία από τις ακόλουθες μορφές:

- $A \sqsubseteq B$

¹⁹ Η κύρια χρήση των GCIs στην \mathcal{EL}^+ είναι ο ορισμός ατομικών εννοιών με όρους περιγραφής σύνθετων εννοιών. Για αυτό το λόγο εισάγονται μία σειρά από όροι: (α) Ο ορισμός έννοιας (concept definition) $A \sqsubseteq C$ ως συνδυασμό των GCIs $A \sqsubseteq C$ και $C \sqsubseteq A$ με το A να αντιστοιχεί σε ατομική έννοια. Το C περιγράφει τις αναγκαίες και ικανές συνθήκες για να είναι κάτι στιγμιότυπο του A . (β) Ο ορισμός πρωταρχικής έννοιας (primitive concept definition) $A \sqsubseteq C$ με A ατομική έννοια, ο οποίος αναφέρεται στις αναγκαίες αλλά όχι ικανές συνθήκες για να είναι κάτι στιγμιότυπο του A .

²⁰ Υπαγωγές ρόλων της μορφής $r \sqsubseteq s$ αποκαλούνται ιεραρχίες ρόλων (role hierarchies). Η μεταβατικότητα (transitivity) των ρόλων μπορεί να εκφραστεί μέσω του $\text{rol} \sqsubseteq r$. Με τα RIs μπορούν επίσης να εκφραστούν κανόνες right - identity στη μορφή $\text{rol} \sqsubseteq r$.

- $A_1 A_2 B$
- $A .B$
- $.A B$
- r
- $r o s$

Αντίστοιχα, οι συμπερασματικοί κανόνες για το EL^+ κομμάτι της Περιγραφικής Λογικής οι οποίοι επαναλαμβάνονται επαναληπτικά (με εξαίρεση τον κανόνα 1) και εξαντλητικά έως ότου δεν προκύπτει καμία αλλαγή στο TBox T είναι οι ακόλουθοι:

1. Αν $A \in T$, τότε $T = T \cup \{A \sqsubseteq A, A \sqsubseteq \perp\}$
2. Αν $A \sqsubseteq B \in T$ και $B \sqsubseteq C \in T$, τότε $T = T \cup \{A \sqsubseteq C\}$
3. Αν $A \sqsubseteq B \in T$ και $A \sqsubseteq C \in T$ και $B \sqsubseteq C \sqsubseteq D \in T$, τότε $T = T \cup \{A \sqsubseteq D\}$
4. Αν $A \sqsubseteq B \in T$ και $B \sqsubseteq .C \in T$, τότε $T = T \cup \{A \sqsubseteq .C\}$
5. Αν $A \sqsubseteq .B \in T$ και $r \sqsubseteq s \in T$, τότε $T = T \cup \{A \sqsubseteq .B\}$
6. Αν $A \sqsubseteq .B \in T$ και $B \sqsubseteq C \in T$ και $.C \sqsubseteq D \in T$, τότε $T = T \cup \{A \sqsubseteq D\}$
7. Αν $A \sqsubseteq .B \in T$ και $B \sqsubseteq .C \in T$ και $r o s \sqsubseteq t \in T$, τότε $T = T \cup \{A \sqsubseteq .C\}$

2.3.4.3.3 Δομική Υπαγωγή στην EL^{++}

Η περιγραφική λογική EL^{++} αποτελεί μια περισσότερη εκφραστική επέκταση της EL στην οποία έχουν προστεθεί οι κατασκευαστές της κενής έννοιας (BOTTOM – concept), των ονομαστικών εννοιών (nominals), μία περιορισμένη εκδοχή των συμπαγών πεδίων ορισμού (concrete domain) και μία αυστηρή μορφή σύνθεσης ρόλων. Η σύνταξη και η σημασιολογία τόσο των εννοιών όσο και των αξιωμάτων της EL^{++} , δεδομένου της ύπαρξης δεδομένου της ύπαρξης ενός συνόλου ατομικών εννοιών N_C , ενός συνόλου ατομικών ρόλων N_R και πιθανόν ενός συνόλου ατομικών ονομάτων (individuals) N_I δίνεται στους ακόλουθους πίνακες.

| Έννοιες | Σύνταξη | Σημασιολογία |
|---|---|--|
| TOP – καθολική έννοια | \top | Δ^I |
| BOTTOM – κενή έννοια ²¹ | \perp | |
| Ονομαστικές έννοιες Nominals ²² | $\{a\}$ | $\{a^I\}$ |
| Ατομική έννοια $P \quad N_C$ | P | $P^I \subseteq \Delta^I$ |
| Τομή | $C \quad D$ | $C^I \quad D^I$ |
| Πλήρης υπαρξιακή ποσοτικοποίηση, $r \quad N_R$ | $r.C$ | $\{ a \in \Delta^I \mid b: (a,b) \in r^I \quad b \in C^I \}$ |
| Συμπαγές πεδίο ορισμού Concrete domain ²³ | $p(f_1, \dots, f_k)$ για $f \in P^{D_j}$ | $\{ a \in \Delta^I \mid b_1, \dots, b_k \in \Delta^{D_j} : f_i^I(a) = b_i \text{ για } 1 \leq i \leq k \quad (b_1, \dots, b_k) \in p^{D_j} \}$ |

Πίνακας 2.10 Σύνταξη και σημασιολογία των εννοιών της ⁺⁺

| Αξιώματα | Σύνταξη | Σημασιολογία |
|---|---|---|
| Γενικευμένη υπαγωγή εννοιών (general concept inclusion- GCI) | $C \sqsubseteq D$ | $C^I \subseteq D^I$ |
| Υπαγωγή ρόλων (role inclusion - RI) ²⁴ | $r_1 \circ \dots \circ r_n \sqsubseteq s$ | $r_1^I \circ \dots \circ r_n^I \subseteq s^I$ |

Πίνακας 2.11 Σύνταξη και σημασιολογία των αξιωμάτων της ⁺

Πριν από την παρουσίαση του αλγορίθμου (τόσο του αρχικού σταδίου της κανονικοποίησης όσο και του σταδίου της εφαρμογής των κανόνων συμπερασμού) είναι

²¹ Το bottom concept σε συνδυασμό με GSIs μπορεί να χρησιμοποιηθεί για να εκφράσει disjointness σύνθετων περιγραφών εννοιών π.χ. $C \sqsubseteq D \sqsubseteq \perp$ δηλώνει ότι τα C και D είναι ξένα μεταξύ τους (disjoint).

²² Η αξίωση μοναδικού ονόματος για τα individual names μπορεί να δηλωθεί γράφοντας $\{a\} \sqsubseteq \{b\} \sqsubseteq \perp$ για όλα τα individuals a, b .

²³ Ο constructor αυτός παρέχει μία διεπαφή (interface), που επιτρέπει την αναφορά σε strings και ακεραίους. Τυπικά ένα concrete domain D είναι ένα ζεύγος $(\Delta D, PD)$ όπου PD είναι ένα σύνολο από ονόματα κατηγορημάτων (predicate names). Κάθε $p \in P$ είναι συνδεδεμένο με ένα arity $n > 0$ (arity: ο αριθμός των παραμέτρων μιας συνάρτησης ή ενός τελεστή π.χ. πρόσθεση $n=2$, απόλυτη τιμή $n=1$) και με ένα extension $pD \sqsubseteq (\Delta D)^n$. Για την παροχή ενός συνδέσμου μεταξύ της περιγραφικής λογικής και του concrete domain, εισάγουμε ένα σύνολο χαρακτηριστικών ονομάτων (feature names) N_F . Στον παραπάνω πίνακα το p δηλώνει ένα predicate μερικών concrete domain D και τα f_1, \dots, f_k είναι feature names. Μία περιγραφική λογική EL++ μπορεί να περιέχει έναν αριθμό από concrete domain D_1, \dots, D_n τέτοια ώστε $\Delta D_i \cap \Delta D_j = \emptyset$ για $1 \leq i < j \leq n$. Για να γίνει εμφανής η χρήση συγκεκριμένων concrete domain D_1, \dots, D_n συμβολίζουμε $EL++(D_1, \dots, D_n)$ αντί για $EL++$.

²⁴ Ισχύει ότι και στη παραπομπή 17

σκόπιμο να γίνουν μια σειρά από παρατηρήσεις και ορισμοί συμβολισμών που είναι αναγκαίοι για την κατανόηση όσων ακολουθούν.

- Αντί για την χρήση TBox γίνεται χρήση ενός παρόμοιου λεξιλογίου που ονομάζεται CBox (constraint box – σύνολο περιορισμών) και αποτελείται από ένα πεπερασμένο σύνολο αξιωμάτων γενικευμένης υπαγωγής εννοιών (GCIs) και υπαγωγής ρόλων (RIs).
- Δεδομένου ενός CBox C , χρησιμοποιούμε τον συμβολισμό BC_C για να δηλώσουμε το μικρότερο δυνατό σύνολο περιγραφής εννοιών που περιέχει τα εξής: (α) την έννοια top T . (β) όλες τις ατομικές έννοιες που χρησιμοποιούνται στο C . (γ) όλες τις περιγραφές εννοιών της μορφής $\{a\}$ ή $p(f_1, \dots, f_k)$ που εμφανίζονται στο C .
- Ορίζουμε R_C το σύνολο όλων των ρόλων που εμφανίζονται στο C .
- Ορίζεται ο συμβολισμός $\sim_{\mathcal{R}} \subseteq BC_C \times BC_C$ ως εξής:

Ισχύει $C \sim_{\mathcal{R}} D$ αν υπάρχουν $C_1, \dots, C_k \in BC_C$ τέτοια ώστε

(α) $C_1 = C$ ή $C_1 = \{b\}$ για κάποιο ατομικό όνομα (individual) b

(β) Υπάρχει $r_j(C_j, C_{j+1})$ για κάποιο $r_j \in R_C$

(γ) $C_k = D$

- Ορίζεται ο συμβολισμός $\text{conj}_j(\Gamma)$ για ένα σύνολο Γ από περιγραφές εννοιών $++(D_1, \dots, D_n)$ και $1 \leq j \leq n$ ως εξής:

$$\text{conj}_j(\Gamma) := \bigwedge_{p(f_1, \dots, f_k) \in \Gamma \text{ with } p \in \mathcal{P}^{D_j}} p(f_1, \dots, f_k).$$

Το πρώτο στάδιο του αλγορίθμου περιλαμβάνει την μετατροπή του CBox σε κανονική μορφή μέσω μιας διαδικασίας κανονικοποίησης που γίνεται σε χρόνο γραμμικό ως προς το μέγεθος του CBox. Ένα CBox C είναι σε κανονική μορφή αν ισχύουν τα παρακάτω:

- Όλα τα GCIs έχουν μία από τις παρακάτω μορφές, όπου $C_1, C_2 \in \text{BC}_C$ και $D \in \text{BC}_C \setminus \{\perp\}$

- $C_1 \supseteq D$
- $C_1 \supseteq C_2 \supseteq D$
- $C_1 \supseteq \cdot C_2$
- $\cdot C_1 \supseteq D$

- Όλα τα RIs είναι της μορφής $r \supseteq s$ ή $r_1 \circ r_2 \supseteq s$

Για να μετατραπεί το CBox σε κανονική μορφή εφαρμόζονται επαναληπτικά και εξαντλητικά σε γραμμικό χρόνο οι παρακάτω κανόνες κανονικοποίησης όπου $C', D' \in \text{BC}_C$ και u, A νέες ατομικές έννοιες και ρόλοι που εισάγονται στο BC_C .

1. $r_1 \circ \dots \circ r_k \supseteq s \rightarrow \{r_1 \circ \dots \circ r_{k-1} \supseteq u, u \circ r_k \supseteq s\}$
2. $C \sqcap D' \supseteq E \rightarrow \{D' \supseteq A, C \sqcap A \supseteq E\}$
3. $\exists r. C' \supseteq D \rightarrow \{C' \supseteq A, \exists r. A \supseteq D\}$
4. $\perp \supseteq D \rightarrow \emptyset$
5. $C' \supseteq D' \rightarrow \{C' \supseteq A, A \supseteq D'\}$
6. $B \supseteq \exists r. C' \rightarrow \{B \supseteq \exists r. A, A \supseteq C'\}$
7. $B \supseteq C \sqcap D \rightarrow \{B \supseteq C, B \supseteq D\}$

Έχει αποδειχθεί ότι για να μην χαθεί η γραμμικότητα της κανονικοποίησης είναι απαραίτητο οι παραπάνω κανόνες να εφαρμόζονται τμηματικά εξαντλητικά, δηλαδή αρχικά εξαντλητική εφαρμογή των κανόνων 1 έως 4 και στη συνέχεια 5 έως 7.

Σε δεύτερη φάση, οι συμπερασματικοί κανόνες για το $^{++}$ κομμάτι της Περιγραφικής Λογικής οι οποίοι επαναλαμβάνονται επαναληπτικά (με εξαίρεση τον κανόνα 1) και εξαντλητικά έως ότου δεν προκύπτει καμία αλλαγή στο CBox C είναι οι ακόλουθοι:

1. Αν $A \vdash B$, τότε $C = C \{A \vdash B\}$
2. Αν $A \vdash B$ και $B \vdash D$, τότε $C = C \{A \vdash D\}$
3. Αν $A \vdash B$ και $A \vdash D$ και $B \vdash D \vdash E$, τότε $C = C \{A \vdash E\}$
4. Αν $A \vdash B$ και $B \vdash .D$, τότε $C = C \{A \vdash .D\}$
5. Αν $A \vdash .B$ και $B \vdash D$ και $.D \vdash E$, τότε $C = C \{A \vdash E\}$
6. Αν $A \vdash .B$ και $B \vdash \perp$, τότε $C = C \{A \vdash \perp\}$
7. Αν $\{a\} \vdash A$ και $\{a\} \vdash B$ και $A \sim_{\mathcal{R}} B$, τότε $C = C \{A \vdash B\}$
8. Αν το $\text{con}_j(\mathcal{S}(A))^{25}$ είναι μη ικανοποιήσιμο²⁶ D_j , τότε $C = C \{A \vdash \perp\}$
9. Αν από τον συμπερασμό²⁷ του $\text{con}_j(\mathcal{S}(A))$ στο D_j συνεπάγεται το $p(f_l, .f_k) \vdash C$, τότε $C = C \{A \vdash p(f_l, .f_k)\}$
10. Αν $A \vdash p(f_l, .f_k)$ και $A \vdash p'(f'_l, .f'_k)$ και $p \vdash P^{D_j}$ και $p' \vdash P^{D_l}$ και $j \neq l$ και $f_s = f'_s$ για κάποιο s, t , τότε $C = C \{A \vdash \perp\}$
11. Αν $A \vdash .B$ και $r \vdash .B$, τότε $C = C \{A \vdash .B\}$
12. Αν $A \vdash .B$ και $B \vdash .C$ και $r \vdash s$, τότε $T = T \{A \vdash .C\}$

Ο αλγόριθμος ταξινόμησης που έχει βασιστεί στους παραπάνω συμπερασματικούς κανόνες είναι πολυωνυμικός καθώς έχει αποδειχθεί ότι σε ένα κανονικοποιημένο CBox οι κανόνες μπορούν να εφαρμοστούν μόνο ένα πολυωνυμικό αριθμό φορών και η εφαρμογή κάθε κανόνα γίνεται σε πολυωνυμικό χρόνο.

²⁵ Το $\mathcal{S}(A)$ είναι το σύνολο με όλες τις υπερκλάσεις του A . π.χ. Αν $A \vdash B$, $A \vdash D$ τότε $\mathcal{S}(A) = \{B, C\}$

2.3.4.4 Δομική Υπαγωγή στην Horn-Shiq

Το τελευταίο υποτιμήμα της περιγραφικής λογικής για το οποίο θα μελετήσουμε την τεχνική της Δομικής Υπαγωγής είναι ένα tractable υποτιμήμα της περιγραφικής λογικής SHIQ γνωστό ως Η σύνταξη και η σημασιολογία τόσο των εννοιών όσο και των αξιωμάτων της SHIQ, δεδομένου της ύπαρξης δεδομένου της ύπαρξης ενός συνόλου ατομικών εννοιών N_C , ενός συνόλου ατομικών ρόλων N_R και πιθανόν ενός συνόλου ατομικών ονομάτων (individuals) N_I δίνεται στους ακόλουθους πίνακες.

| Έννοιες | Σύνταξη | Σημασιολογία |
|--|---------------|--|
| TOP – καθολική έννοια | \top | Δ^I |
| BOTTOM – κενή έννοια | \perp | |
| Ατομική έννοια $P \in N_C$ | P | $P^I \subseteq \Delta^I$ |
| Σύνθετη άρνηση, $C \in N_C$ | $\neg C$ | $\Delta^I \setminus C^I$ |
| Τομή εννοιών | $C \sqcap D$ | $C^I \cap D^I$ |
| Ένωση εννοιών | $C \sqcup D$ | $C^I \cup D^I$ |
| Πλήρης υπαρξιακή ποσοτικοποίηση, $r \in N_R$ | $r.C$ | $\{ a \in \Delta^I \mid b: (a,b) \in r^I \wedge b \in C^I \}$ |
| Περιορισμός τιμής, $r \in N_R$ | $r.C$ | $\{ a \in \Delta^I \mid b: (a,b) \in r^I \wedge b \in C^I \}$ |
| Ποιοτικός περιορισμός αριθμού – ελάχιστη πληθικότητα, $r \in N_R, n$ | $(\geq n)r.C$ | $\{ a \in \Delta^I \mid \#\{ b: (a,b) \in r^I \wedge b \in C^I \} \geq n \}$ |
| Ποιοτικός περιορισμός αριθμού – μέγιστη πληθικότητα, $r \in N_R, n$ | $(\leq n)r.C$ | $\{ a \in \Delta^I \mid \#\{ b: (a,b) \in r^I \wedge b \in C^I \} \leq n \}$ |

Πίνακας 2.12 Σύνταξη και σημασιολογία των εννοιών της SHIQ

| Αξιώματα | Σύνταξη | Σημασιολογία |
|---|-------------------|---|
| Γενικευμένη υπαγωγή εννοιών (general concept inclusion- GCI) | $C \sqsubseteq D$ | $C' \subseteq D'$ |
| Υπαγωγή ρόλων (role inclusion - RI) | $r \sqsubseteq s$ | $r' \subseteq s'$ |
| Μεταβατικός ρόλος (role transitivity) | $\text{Tra}(r)$ | $r' \circ r' \subseteq r'$ |
| Αντίστροφος ρόλος (inverse role) | r^{-} | $\{a, b \in \Delta' \mid (a, b) \in r' \leftrightarrow (b, a) \in r'^{-}\}$ |

Πίνακας 2.13 Σύνταξη και σημασιολογία των αξιωμάτων της *SHIQ*

Για να ορίσουμε τον Horn υποτήμα της *SHIQ* θα πρέπει να αναφερθούμε συνοπτικά στην απόδοση πολικότητας μιας *SHIQ* έννοιας. Οι θετικές και αρνητικές πολικότητες των *SHIQ* εννοιών ορίζονται ως εξής :

- Το C εμφανίζεται θετικά στο C
- Το C εμφανίζεται θετικά (αρνητικά) στο $\neg C$, C_+ D_+ , $C_+ \sqcup D_+$, $r.C_+$, $r.C_+$, $nr.C_+$, $mr.C$ και $C \sqsubseteq D_+$, εάν το C εμφανίζεται θετικά (αρνητικά) στο C_+ ή στο D_+ , ή εμφανίζεται αρνητικά (θετικά) στο C .

Μία έννοια C εμφανίζεται θετικά (αρνητικά) σε μια οντολογία O , εάν το C εμφανίζεται θετικά (αρνητικά) σε ένα αξίωμα της O . Είναι επομένως δυνατό για ένα concept να εμφανίζεται και θετικά και αρνητικά σε ένα αξίωμα ή μια οντολογία.

Μία *SHIQ* οντολογία O είναι Horn, εάν :

- Δεν εμφανίζεται θετικά στην O κανένα concept της μορφής $C \sqcup D$ ή $mr.C$ με $m > 1$.
- Δεν εμφανίζεται αρνητικά στην O κανένα concept της μορφής $\neg C$, $r.C$, $nr.C$ με $n > 1$, ή $mr.C$.

Πριν από την εφαρμογή του αλγορίθμου ταξινόμησης βασισμένο στην τεχνική της Δομικής Υπαγωγής είναι απαραίτητη η πραγματοποίηση μιας προεργασίας πάνω στην *SHIQ* οντολογία αντίστοιχη με την διαδικασία της κανονικοποίησης που έχουμε συναντήσει και σε άλλες περιγραφικές λογικές.

Το πρώτο στάδιο αυτής της προεργασίας ονομάζεται δομικός μετασχηματισμός (structural transformation) και χρησιμοποιείται για την απλοποίηση των αξιωμάτων της οντολογίας, διατηρώντας όμως την δομή της. Δεδομένης μια *SHIQ* οντολογία O , για κάθε υποέννοια C στην O εισάγουμε μία νέα ατομική έννοια A_C και ορίζουμε μια συνάρτηση $st(C)$ ως :

- $st(A) = A$, $st(T) = T$, $st(\perp) = \perp$
- $st(\neg C) = \neg A_C$
- $st(C \sqcap D) = st(A_C \sqcap A_D)$, $st(C \sqcup D) = st(A_C \sqcup A_D)$
- $st(r.C) = r.A_C$, $st(\bar{r}.C) = \bar{r}.A_C$
- $st(nr.C) = nr.A_C$, $st(mr.C) = mr.A_C$

Το αποτέλεσμα της εφαρμογής του δομικού μετασχηματισμού στην O είναι η δημιουργία σε πολυωνυμικό χρόνο μιας νέας οντολογία O' , η οποία είναι λογικά αντίστοιχη της αρχικής και περιέχει όλα τα role inclusions και role transivity axioms της O , με την προσθήκη των παρακάτω αξιωμάτων υπαγωγής εννοιών:

- $A_C \sqsubseteq st(C_+)$ για κάθε C που εμφανίζεται θετικά στην O
- $st(C_-) \sqsubseteq A_C$ για κάθε C που εμφανίζεται αρνητικά στην O
- $A_C \sqsubseteq A_D$ για κάθε concept inclusion $C \sqsubseteq D \in O$

Το δεύτερο στάδιο της απαραίτητης προεργασίας είναι μία μορφή κανονικοποίησης της οντολογίας που έχει προκύψει από την εφαρμογή του δομικού μετασχηματισμού ώστε να περιέχει αποκλειστικά αξιώματα της παρακάτω μορφής.

- $\prod A_i \sqsubseteq C$
- $r_1 \sqsubseteq r_2$
- $\text{Tra}(r)$

όπου $\prod A_i$ μία τομή ατομικών εννοιών (πιθανώς και κενή) και C μια απλή έννοια (simple concept)²⁸

Για να επιτευχθεί η κανονική μορφή που επιθυμούμε πρέπει να εφαρμόσουμε μια σειρά από κανόνες κανονικοποίησης στην οντολογία O' που προκύπτει από το στάδιο του δομικού μετασχηματισμού. Από τη στιγμή που η αρχική οντολογία O ήταν \mathcal{SHIQ} , το C_+ , το οποίο εμπλέκεται μέσω της συνάρτησης $\text{st}()$ με τα νέα αξιώματα της O' , μπορεί να είναι μόνο της μορφής $\top, \perp, A, \neg C, C \sqcap D, r.C, r.C, ns.C$ και $1r.C$, και το C αντίστοιχα μπορεί να είναι μόνο της μορφής $\top, \perp, A, C \sqcap D, C \sqcup D, r.C$ και $1s.C$.

Επομένως, τα αξιώματα υπαγωγής της οντολογίας O' της μορφής $A \sqsubseteq \text{st}(C_+)$ τα οποία δεν πληρούν τις προϋποθέσεις κανονικοποίησης, μαζί με τον αντίστοιχο κανόνα που θα πρέπει να εφαρμοστεί ώστε να έρθουν σε κανονική μορφή είναι τα εξής:

- $A \sqsubseteq \text{st}(\neg C) = \neg A_C \Rightarrow A \sqcap A_C \sqsubseteq \perp$
- $A \sqsubseteq \text{st}(ns.C) = ns.A_C \Rightarrow A \sqsubseteq \bigwedge_{1 \leq i \leq n} B_i, B_i \sqsubseteq A_C, 1 \leq i \leq n, B_i \sqcap B_j \sqsubseteq \perp, 1 \leq i < j \leq n$, όπου τα B_i είναι καινούργια atomic concepts.

²⁸ Μία έννοια C ονομάζεται απλή (simple) εάν είναι της μορφής $\perp, A, r.A, r.A$ και $1s.A$.

Αντίστοιχα, αξιώματα υπαγωγής της οντολογίας O' της μορφής $\text{st}(C.) \sqsubseteq A$ τα οποία δεν πληρούν τις προϋποθέσεις κανονικοποίησης, μαζί με τον αντίστοιχο κανόνα που θα πρέπει να εφαρμοστεί ώστε να έρθουν σε κανονική μορφή είναι τα εξής:

- $\text{st}(C \sqcup D) = A_C \sqcup A_D \sqsubseteq A \Rightarrow A_C \sqsubseteq A, A_D \sqsubseteq A$
- $\text{st}(r.C) = r.A_C \sqsubseteq A \Rightarrow A_C \sqsubseteq r.A$
- $\text{st}(Is.C) = Is.A_C \sqsubseteq A \Rightarrow A_C \sqsubseteq s.A$

Το τελευταίο στάδιο της διαδικασίας της προεργασίας αποτελείται από μία τεχνική για την απαλοιφή των αξιωμάτων μεταβατικών ρόλων. Τα μεταβατικά αξιώματα που υπάρχουν στην κανονικοποιημένη οντολογία μπορούν να αλληλεπιδράσουν μόνο με αξιώματα της μορφής $\prod A_i \sqsubseteq r.B$ μέσω απλών αξιωμάτων υπαγωγής ρόλων. Επομένως αυτό το τελευταίο στάδιο της προεργασίας εισάγει μια νέα τριπλέτα αξιωμάτων για κάθε αξίωμα της μορφής $\prod A_i \sqsubseteq r.B$ για το οποίο υπάρχει και ένας μεταβατικό ρόλος t ($\text{Tra}(t)$) ο οποίος είναι υπορόλος του r . Η τριπλέτα αυτή όπου B^t είναι μία νέα ατομική έννοια είναι η εξής:

- $\prod A_i \sqsubseteq t.B^t$
- $B^t \sqsubseteq t.B$
- $B^t \sqsubseteq B$

Μετά την ολοκλήρωση της απαραίτητης προεργασίας μιας SHIQ οντολογίας μπορούμε να εφαρμόσουμε εξαντλητικά και επαναληπτικά τους συμπερασματικούς κανόνες βασισμένους στην τεχνική της Δομικής Υπαγωγής που δίνονται παρακάτω. Στους παρακάτω κανόνες τα M και N αντιστοιχούν σε τομές ατομικών εννοιών, τα A, A_i, B σε ατομικές έννοιες και το C σε απλές (simple) έννοιες.

1. Αν $M \sqsubseteq O$, τότε $O = O \sqcup \{M\}$
2. Αν $M \sqsubseteq A \sqsubseteq O$, τότε $O = O \sqcup \{M \sqsubseteq A \sqsubseteq A\}$
3. Αν $M \sqsubseteq A_i \sqsubseteq O$ και $\prod A_i \sqsubseteq C \sqsubseteq O$, τότε $O = O \sqcup \{M \sqsubseteq C\}$
4. Αν $M \sqsubseteq r.N \sqsubseteq O$ και $N \sqsubseteq \perp \sqsubseteq O$, τότε $O = O \sqcup \{M \sqsubseteq \perp\}$

5. Αν $M \xrightarrow{r_1} N \in O$, $M \xrightarrow{r_2} A \in O$ και $r_1 \neq r_2 \in O$, τότε $O = O \cup \{ M \xrightarrow{r_1} (N \cup A) \}$
6. Αν $M \xrightarrow{r_1} N \in O$, $N \xrightarrow{r_2} A \in O$ και $r_1 \neq r_2 \in O$, τότε $O = O \cup \{ M \xrightarrow{r_1} A \}$
7. Αν $M \xrightarrow{r_1} N_1 \in O$, $N_1 \in B \in O$, $r_1 \in S \in O$, $M \xrightarrow{r_2} N_2 \in O$, $N_2 \in B \in O$, $r_2 \in S \in O$ και $M \xrightarrow{1s} B$, τότε $O = O \cup \{ M \xrightarrow{r_1} (N_1 \cup N_2) \}$
8. Αν $M \xrightarrow{r_1} N_1 \in O$, $N_1 \xrightarrow{r_2} (N_2 \cup A) \in O$, $r_1 \in S \in O$, $M \in B \in O$, $N_2 \in A \in B \in O$, $r_2 \in S \in O$ και $N_1 \xrightarrow{1s} B$, τότε $O = O \cup \{ M \xrightarrow{r_1} A, M \xrightarrow{r_2} N_1 \}$

Η τεχνική της Δομικής Υπαγωγής στο *SHIQ* τμήμα της περιγραφικής λογικής δεν είναι πολωνυμικός, όπως συμβαίνει με το προηγούμενα υποτιμήματα που μελετήσαμε, αλλά έχει αποδειχθεί ότι ο αλγόριθμος της απόφασης για την υπαγωγή εννοιών είναι ExpTime-complete.

2.4 Η γλώσσα OWL 2

Οι έρευνες που οδήγησαν στην κατασκευή της αρχικής γλώσσας OWL (Web Ontology Language) προήλθαν από μια αδιαμφισβήτητη διαπίστωση. Τα υπάρχοντα μοντέλα μέχρι εκείνη την στιγμή, όπως το RDFS, είχαν περιορισμένη εκφραστικότητα. Ομάδες εργασίας του World Wide Web Consortium (W3C) αναγνώρισαν μια σειρά από περιπτώσεις όπου δεν επιδείκνυαν επάρκεια και έπρεπε να επεκταθούν. Οι σημαντικότερες από αυτές είναι:

- Λογικός συνδυασμός κλάσεων (Boolean combination of classes)

Δε μπορούμε να ορίσουμε κλάσεις ως συνδυασμό ήδη υπαρχόντων κλάσεων.

(π.χ. Άνθρωπος \equiv Άντρας \sqcup Γυναίκα)

- Ασυμβατότητα κλάσεων (Disjointness of classes)

Δε μπορούμε να πούμε ότι οι κλάσεις Άντρας και Γυναίκα είναι ασύμβατες.

- Ειδικά χαρακτηριστικά ιδιοτήτων (special characteristics of properties)

Δε μπορούμε να δηλώσουμε μεταβατικές ιδιότητες (π.χ. Μεγαλύτερος_από), αντίστροφες ιδιότητες (π.χ. Είναι_μέρος_του και Έχει_μέρος) και λειτουργικές ιδιότητες (π.χ. Έχει_μητέρα).

- Τοπικό εύρος ιδιοτήτων (Local scope of properties)

Δε μπορούμε να πούμε ότι το εύρος (range) της ιδιότητας Έχει_παιδί είναι Άτομο, όταν αναφέρεται σε ανθρώπους, ή Ελέφαντας, όταν αναφέρεται σε ελέφαντες.

- Περιορισμοί μεγέθους συνόλων (Cardinality restrictions)

Δε μπορούμε να εκφράσουμε περιορισμούς όσον αφορά στον αριθμό (μέγιστο ή ελάχιστο) των διακριτών ατόμων που μπορούν να συσχετιστούν μέσω μιας ιδιότητας (π.χ. ένα κατάστημα μπορεί να έχει το πολύ 30 υπαλλήλους).

- Ταυτότητα/Διαφορετικότητα ατόμων (Equality/Inequality of individuals)

Δε μπορούμε να πούμε ότι δύο άτομα με διαφορετικά ονόματα είναι τα ίδια ή να δηλώσουμε ρητά ότι είναι διαφορετικά.

Η υπέρβαση των παραπάνω μειονεκτημάτων και η ταυτόχρονη προσπάθεια για εξασφάλιση αποδοτικής συλλογιστικής ανάλυσης οδήγησαν σε μια πιο «δυνατή» γλώσσα κατασκευής οντολογιών. Αρχικά και για να «κληροδοτηθούν» τα πλεονεκτήματα του RDF/S, προτάθηκαν οι DAML-ONT και OIL. Από αυτές προέκυψε η DAML+OIL [DAML] που με τη σειρά της αποτέλεσε τη βάση για την υλοποίηση της OWL.

Η OWL χρησιμοποιεί σε μεγάλο βαθμό το RDF/S και μπορούμε να πούμε ότι κατά κάποιο τρόπο το επεκτείνει. Αυτό γιατί η OWL, αν και έχει XML-like σύνταξη όπως ο «πρόγονός» της, αυτή είναι σαφώς βελτιωμένη και συνοδεύεται από ένα γραφικό μέρος που υιοθετεί συμβάσεις της UML (Unified Modeling Language) [UML] γλώσσας. Παρέχει ένα σύνολο κατασκευαστών (constructors) όπως τομή (conjunction), ένωση (disjunction), άρνηση (negation), υπαρξιακή ποσοτικοποίηση (existential quantification), περιορισμούς αριθμών (cardinality restrictions) κ.α. Η παραπομπή στην ορολογία της Περιγραφικής Λογικής είναι προφανής, γι' αυτό και στη βιβλιογραφία η OWL χαρακτηρίζεται ως DL-based.

Η OWL διαιρούταν σε τρεις γλώσσες

1. OWL Full, η οποία αναφερόταν σ' ολόκληρη τη γλώσσα.
2. OWL DL, η οποία ήταν το μέρος της OWL που ήταν αποφασίσιμο (decidable in finite time) και πρόσφερε πλήρη (sound & complete) συλλογιστική ανάλυση αντιστοιχώντας στην εκφραστικότητα $SHOIN^{(D)}$ της Περιγραφικής Λογικής
3. OWL Lite, η οποία ήταν το τμήμα της OWL με τη χαμηλότερη εκφραστικότητα και κατά συνέπεια την ευκολότερη συλλογιστική ανάλυση αντιστοιχώντας στο τμήμα $SHIF^{(D)}$ της Περιγραφικής Λογικής.

Μετά την πρώτη παρουσίαση της OWL το 2004, το 2009 προτάθηκε από το W3C η επέκταση της γλώσσας OWL με ονομασία OWL 2 στην οποία προστέθηκαν νέα χαρακτηριστικά όπως η δυνατότητα με το ίδιο URI να ορίζονται τόσο κλάσεις όσο και άτομα και ρόλοι, η διερμηνεία να βασίζεται στο περιεχόμενο και η εισαγωγή μιας απλής

μορφής meta –modelling. Αντίστοιχα με τις γλώσσες της OWL, στην OWL 2 ορίζονται οι γλώσσες OWL 2 Full και OWL 2 DL, ενώ δεν υπάρχει αντίστοιχη της Lite.

Όπως γίνεται εύκολα αντιληπτό από όσα προηγήθηκαν, η γλώσσα OWL 2 χαρακτηρίζεται από υψηλή εκφραστικότητα , τόσο υπολογιστική όσο και για τους χρήστες, καθιστώντας δύσκολη την χρήση της για την υλοποίηση σύνθετων εφαρμογών. Για την αντιμετώπιση αυτού του χαρακτηριστικού σχεδιάστηκαν τα επιπλέον profiles αυτής της γλώσσας ως προσεγγιστικά υποτιμήματα της OWL 2 κατάλληλα για μια ποικιλία εφαρμογών πέρα από τις ήδη υπάρχουσες γλώσσες της OWL 2. Στα OWL 2 Profiles θυσιάζεται ένα κομμάτι της εκφραστικότητας της OWL 2 με αντάλλαγμα την αύξηση της απόδοσης της συλλογιστικής διαδικασίας και την βελτίωση της υπολογιστικής δυναμικότητας.

Υπάρχουν πολλά διαφορετικά υποτιμήματα της OWL 2 που έχουν καλές υπολογιστικές ικανότητες αλλά ως OWL 2 Profiles έχουν επιλεγεί μόνο τρία από αυτά τα οποία είχαν ήδη μία πρωταρχική κοινότητα χρηστών. Τα τρία αυτά profiles είναι ανεξάρτητα μεταξύ τους, δηλαδή κανένα δεν είναι υποτιμήμα κάποιου άλλου. Τα κύριο χαρακτηριστικό και των 3 profiles είναι ότι είναι tractable, δηλαδή μπορούν να αποφανθούν για συλλογιστικά ερωτήματα σε πολυωνυμικό χρόνο, και ντετερμινιστικά, δηλαδή δεν απαιτούν επιλογές πρόβλεψης (guessing) ή πισωγυρισμάτων (backtracking) στην εκτέλεση των ερωτημάτων.

Συνοπτικά τα τρία OWL 2 profiles θα εξεταστούν στα επόμενα υποκεφάλαια. Πριν συμβεί αυτό όμως, θα ασχοληθούμε με την κοινή σύνταξη των οντοτήτων (entities), των σταθερών (literals – συνήθως συμβολοσειρές και ακέραιοι) και των ατόμων (individuals) και για τα 3 profiles της OWL 2. Σε κάθε profile θα αναφερόμαστε μόνο στα τμήματα της OWL 2 τα οποία τα οποία διαφοροποιούνται σε σχέση με το OWL 2 Specification²⁹

²⁹ OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax

2.4.1 Γενικά χαρακτηριστικά της OWL 2

Οι οντότητες (entities) είναι τα βασικά δομικά στοιχεία μιας OWL 2 οντολογίας και ορίζουν το λεξιλόγιο (vocabulary) της οντολογίας. Κάθε οντότητα είναι συνδεδεμένη μοναδικά με ένα IRI ενώ διαχωρίζονται στις παρακάτω κατηγορίες ανάλογα με τον σκοπό και τον τρόπο που έχουν οριστεί.

- *Κλάσεις (Classes)*

Οι κλάσεις αντιπροσωπεύουν ένα σύνολο από άτομα (individuals). Για παράδειγμα η κλάση `a:Child` αντιπροσωπεύει το σύνολο όλων των παιδιών. Στην OWL 2 υπάρχουν δύο προκαθορισμένες κλάσεις με IRIs `owl:Thing`³⁰ και `owl:Nothing` που αντιπροσωπεύουν το σύνολο όλων των ατόμων και το άδειο σύνολο αντίστοιχα.

- *Τύποι Δεδομένων (Datatypes)*

Τα datatypes αντιπροσωπεύουν ένα σύνολο από τιμές δεδομένων (data types) όπως είναι οι συμβολοσειρές και οι ακέραιοι. Για παράδειγμα το datatype `xsd:integer` υποδηλώνει το σύνολο των ακεραίων, το `xsd:dateTime` το σύνολο των ημερομηνιών και το `owl:real` το σύνολο των πραγματικών. Αντίστοιχα datatypes υπάρχουν για μια σειρά από τιμές δεδομένων. Συγκεντρωτικά τα πιο συνηθισμένα datatypes εμφανίζονται στον παρακάτω πίνακα.

³⁰ Το πρόθεμα συμβολίζει το αντίστοιχο IRI που έχει γίνει ο ορισμός. Τα πιο συνηθισμένα είναι τα εξής:

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

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

xsd: <http://www.w3.org/2001/XMLSchema#>

owl: <http://www.w3.org/2002/07/owl#>

Πραγματικοί, δεκαδικοί και ακέραιοι αριθμοί

| | | | |
|-------------------------------|-----------------------------|--------------------------|--------------------------|
| <i>owl:real</i> | <i>owl:rational</i> | <i>xsd:decimal</i> | <i>xsd:integer</i> |
| <i>xsd:nonNegativeInteger</i> | <i>xsd:negativeInteger</i> | <i>xsd:long</i> | <i>xsd:int</i> |
| <i>xsd:nonPositiveInteger</i> | <i>xsd:positiveInteger</i> | <i>xsd:short</i> | <i>xsd:byte</i> |
| <i>xsd:unsignedLong</i> | <i>xsd:unsignedInt</i> | <i>xsd:unsignedShort</i> | <i>xsd:unsignedByte</i> |
| Float | | <i>xsd:double</i> | <i>xsd:float</i> |
| Συμβολοσειρές | | | |
| <i>xsd:string</i> | <i>xsd:normalizedString</i> | <i>xsd:token</i> | <i>xsd:language</i> |
| <i>xsd:Name</i> | <i>xsd:NCName</i> | <i>xsd:NMTOKEN</i> | |
| Boolean | | <i>xsd:boolean</i> | |
| Binary Data | | <i>xsd:hexBinary</i> | <i>xsd:base64Binary</i> |
| IRIs | | <i>xsd:anyURI</i> | |
| Χρονικά στιγμιότυπα | | <i>xsd:dateTime</i> | <i>xsd:dateTimeStamp</i> |

Πίνακας 2. Συνηθισμένα Datatypes στην OWL 2

- *Ιδιότητες αντικειμένων (Object Properties)*

Τα object properties συνδέουν ζεύγη ατόμων (individuals). Για παράδειγμα το object property *a:parentOf* μπορεί να συνδέσει 2 άτομα μεταξύ τους. Στην OWL 2 υπάρχουν δύο προκαθορισμένα object properties με IRIs *owl:topObjectProperty* και *owl:bottomObjectProperty* που συνδέει όλα τα πιθανά ζεύγη ατόμων και δεν συνδέει κανένα ζεύγος ατόμων αντίστοιχα.

- *Ιδιότητες δεδομένων (Data Properties)*

Τα data properties συνδέουν ένα άτομο (individual) με μία σταθερά (literal). Για παράδειγμα το data property *a:hasName* μπορεί να συνδέσει ένα άτομο (*a:John*) με ένα string (π.χ. "John Papadopoulos"). Στην OWL 2 υπάρχουν δύο προκαθορισμένα data properties με IRIs *owl:topDataProperty* και *owl:bottomDataProperty* που συνδέει όλα τα πιθανά άτομα με όλες τις σταθερές και δεν συνδέει κανένα άτομο με σταθερά αντίστοιχα.

- *Ιδιότητες σχολίων (Annotation Properties)*

Τα annotation properties μπορούν να χρησιμοποιηθούν για να προσθέσουν ένα σχόλιο σε μία οντολογία, σε ένα IRI ή σε ένα αξίωμα. Για παράδειγμα το annotation property `rdfs:comment` μπορεί να προσθέσει ένα σχόλιο σε ένα συγκεκριμένο άτομο. Στην OWL 2 υπάρχουν μια σειρά από προκαθορισμένα annotation properties από σχετικά απλά όπως το `rdfs:comment` που προσθέτει ένα σχόλιο σε ένα IRI και το `rdfs:label` που προσθέτει ένα αντιληπτό από τον άνθρωπο όνομα σε ένα IRI έως πιο σύνθετα όπως το `owl:incompatibleWith` που καθορίζει το IRI από μία προηγούμενη έκδοση της οντολογίας που δεν είναι συμβατή με την τρέχουσα έκδοση αυτής.

Εκτός από οντότητες στην OWL 2 υπάρχουν και άτομα (individuals). Υπάρχουν δύο τύποι individuals στην OWL 2, τα ονοματισμένα άτομα (Named Individuals) και τα ανώνυμα (Anonymous Individuals). Ένα παράδειγμα named individual είναι το `a:John` ενώ ένα παράδειγμα ανώνυμου ατόμου με τη αντίστοιχη σύνταξη είναι το `_:a1`.

Επιπλέον, στην OWL 2 υπάρχουν και οι σταθερές (literals) μέσω των οποίων γίνεται η αναπαράσταση τιμών δεδομένων όπως ένα συγκεκριμένο string ή οι ακέραιοι. Κάθε σταθερά αποτελείται από ένα lexical form, που είναι ένα string, και ένα datatype και η σύνταξη του είναι `lexicalForm^^datatypeIRI`. Για παράδειγμα η σταθερά-literal `"1"^^xsd:integer` αντιπροσωπεύει τον ακέραιο 1. Στην περίπτωση που το datatype είναι τύπου `rdf:PlainLiteral` τότε οι σταθερές μπορούν να χρησιμοποιηθούν ως συντομογραφίες, δηλαδή το `"abc"` μπορεί να χρησιμοποιηθεί μέσα στην οντολογία ως συντομογραφία του ορισμού σταθεράς `"abc"^^rdf:PlainLiteral`.

Συνήθως σε μία OWL 2 οντολογία είναι απαραίτητο κάθε οντότητα να ορίζεται και αυτό είναι εφικτό μέσα από έναν τύπο αξιώματος γνωστό ως Declaration. Η σύνταξη του ορισμού μιας οντότητας είναι η εξής:

```

Declaration := 'Declaration' '(' axiomAnnotations Entity ')'
Entity :=
  'Class' '(' Class ')' |
  'Datatype' '(' Datatype ')' |
  'ObjectProperty' '(' ObjectProperty ')' |
  'DataProperty' '(' DataProperty ')' |
  'AnnotationProperty' '(' AnnotationProperty ')' |
  'NamedIndividual' '(' NamedIndividual ')'

```

Για παράδειγμα η κλάση άνθρωπος ορίζεται ως Declaration (Class (a:Person)) ενώ ένα άτομο ορίζεται ως Declaration (NamedIndividuals (a:John)).

Τέλος, στην OWL 2 υπάρχουν μια σειρά από αξιώματα ισχυρισμών που αφορούν τα άτομα (individual). Το πιο σημαντικό από αυτά τα αξιώματα είναι το αξίωμα ισχυρισμού κλάσης (ClassAssertion) μέσω του οποίου συνδέεται ένα άτομο με μία κλάση (π.χ. ClassAssertion (a:Person a:John)) η σύνταξη του οποίου είναι η εξής:

```

ClassAssertion := 'ClassAssertion' '(' ClassExpression Individual ')'

```

2.4.2 OWL 2 EL

Το profile αυτό, βασισμένο στην περιγραφική λογική $\mathcal{EL}++$ από την οποία πήρε και το όνομα του, είναι κατάλληλο για εφαρμογές που χρησιμοποιούν οντολογίες με μεγάλο αριθμό ιδιοτήτων και κλάσεων, όπως η πολύ μεγάλη βιοιατρική οντολογία SNOMED-CT. Η OWL 2 EL μπορεί να αιχμαλωτίσει την εκφραστική δύναμη που χρησιμοποιείται από τέτοιας κλίμακας οντολογίες και να αποφασίσει για τα βασικά προβλήματα συλλογιστική (συνέπεια οντολογίας - ontology consistency, υπαγωγής εκφράσεων κλάσεων - class expression subsumption, έλεγχος στιγμιοτύπου - instance checking) σε χρόνο πολυωνυμικό ως προς το μέγεθος της οντολογίας. Η συλλογιστική ανάλυση (reasoning) στην OWL 2 EL είναι PTIME – complete.

Στην OWL 2 EL οι οντότητες (entities) και τα οι τύποι δεδομένων (datatypes) ορίζονται όπως έχουν παρουσιαστεί στο προηγούμενο κεφάλαιο με ένα επιπλέον

περιορισμό. Για να επιτυγχάνονται οι επιθυμητές υπολογιστικές ιδιότητες τα υποστηριζόμενα datatypes από την OWL 2 EL θα πρέπει να είναι τέτοια ώστε η τομή του πεδίου τιμών δύο από αυτών να είναι είτε κενή είτε πεπερασμένη. Επομένως στην OWL 2 EL δεν επιτρέπεται για προφανείς λόγους η χρήση των ακόλουθων datatypes (xsd:double, xsd:float, xsd:nonPositiveInteger, xsd:positiveInteger, xsd:negativeInteger, xsd:long, xsd:int, xsd:short, xsd:byte, xsd:unsignedLong, xsd:unsignedInt, xsd:unsignedByte, xsd:unsignedShort, xsd:language και xsd:boolean). Επίσης, η OWL 2 EL δεν υποστηρίζει ανώνυμα άτομα (anonymous individuals).

Για να είναι αποδοτική το reasoning στο profile OWL 2 EL το σύνολο των υποστηριζόμενων εκφράσεων που σχετίζονται με τις κλάσεις (ClassExpression) είναι το ακόλουθο και το καθένα από αυτά περιγράφεται αναλυτικά στη συνέχεια :

```
ClassExpression :=
  Class | ObjectIntersectionOf | ObjectOneOf |
  ObjectSomeValuesFrom | ObjectHasValue | ObjectHasSelf |
  DataSomeValuesFrom | DataHasValue
```

- **Class** (παραπομπή στο 2.4.1)
- **ObjectIntersectionOf** := 'ObjectIntersectionOf' '(' **ClassExpression** **ClassExpression** { **ClassExpression** } ')'

Αντιπροσωπεύει την τομή δύο ή περισσότερων εκφράσεων κλάσεων και περιέχει όλα τα άτομα που είναι στιγμιότυπα όλων των εκφράσεων κλάσεων που περιέχονται στα ορίσματα. Στην περιγραφική λογική DL αντιστοιχεί με το $A \sqcap B \sqcap C$

Για παράδειγμα έχουμε την κλάση a:Man με στιγμιότυπα a:John και a:Alex [“Ο Γιάννης είναι άνδρας” “Ο Αλέξης είναι άνδρας”]³¹ και την κλάση a:Married που περιέχει επίσης το στιγμιότυπο a:John [“Ο Γιάννης είναι παντρεμένος”] τότε το

³¹ Με τον συμβολισμό [] θα αναφέρουμε το αντίστοιχο παράδειγμα στην φυσική γλώσσα.

ObjectIntersectionOf (a:Man a:Married) περιέχει το στιγμιότυπο a:John [“Ο Γιάννης είναι ένα άτομο το οποίο είναι άντρας και παντρεμένος”].

- **ObjectOneOf** := 'ObjectOneOf' (' **Individual** ')

Γενικά στην OWL 2 η έκφραση ObjectOneOf χρησιμοποιείται για την απαρίθμηση ατόμων³². Συγκεκριμένα όμως στο profile OWL 2 EL επιτρέπεται μόνο ένα άτομο. Για παράδειγμα το ObjectOneOf (a:John) είναι μία απαρίθμηση με μόνο ένα άτομο.

- **ObjectSomeValuesFrom** := 'ObjectSomeValuesFrom' (' **ObjectPropertyExpression ClassExpression** ')

Αντιπροσωπεύει την πλήρη υπαρξιακή ποσοτικοποίηση της περιγραφικής λογικής ($r.C$). Αποτελείται από δύο ορίσματα, μία έκφραση ιδιοτήτων αντικειμένων και μία έκφραση κλάσης, και περιέχει όλα τα άτομα που συνδέονται μέσω της ιδιότητας αντικειμένων με ένα άτομο, το οποίο είναι στιγμιότυπο της έκφρασης κλάσης που δίνεται ως όρισμα.

Για παράδειγμα έχουμε την ιδιότητα αντικειμένων a:fatherOf που συνδέει τα άτομα a:John και a:Alex [“Ο Γιάννης είναι ο πατέρας του Αλέξη”] και έχουμε ότι το άτομο a:Alex είναι στιγμιότυπο της κλάσης a:Man [“Ο Αλέξης είναι άνδρας”] τότε το ObjectSomeValuesFrom(a:fatherOf a:Man) περιέχει το στιγμιότυπο a:John [“Ο Γιάννης είναι ένα άτομο για το οποίο ισχύει ότι είναι πατέρας κάποιου, ο οποίος είναι άνδρας”].

- **ObjectHasValue** := 'ObjectHasValue' (' **ObjectPropertyExpression Individual** ')

Η συγκεκριμένη έκφραση αποτελεί μια συντακτική συντόμευση για την έκφραση κλάσης ObjectSomeValuesFrom (ObjectPropertyExpression ObjectOneOf (a)), δηλαδή περιέχει όλα τα άτομα που συνδεόνται μέσω της ιδιότητας αντικειμένων με ένα άτομο.

Για παράδειγμα έχουμε την ιδιότητα αντικειμένων a:fatherOf που συνδέει τα άτομα a:John και a:Alex [“Ο Γιάννης είναι ο πατέρας του Αλέξη”] τότε το ObjectHasValue(

³² **ObjectOneOf** := 'ObjectOneOf' (' **Individual {Individual}** ') (π.χ. ObjectOneOf (a:John a:Alex))

a:fatherOf a:Alex) περιέχει το άτομο a:John [“Ο Γιάννης είναι ένα άτομο για το οποίο ισχύει ότι είναι πατέρας κάποιου, ο οποίος είναι ο Αλέξης”].

- **ObjectHasSelf** := 'ObjectHasSelf' (' **ObjectPropertyExpression** ')

Η έκφραση αυτή περιέχει όλα τα άτομα που είναι συνδεδεμένα μέσω μιας συγκεκριμένης ιδιότητας αντικειμένων, η οποία δίνεται ως όρισμα, με τον εαυτό τους.

Για παράδειγμα έχουμε την ιδιότητα αντικειμένων a:likes που συνδέει τα άτομα a:John με a:Alex και a:John με a:John [“Ο Γιάννης συμπαθεί τον Αλέξη” “Ο Γιάννης συμπαθεί τον εαυτό του”] τότε το ObjectHasSelf(a:likes) περιέχει το άτομο a:John [“Ο Γιάννης είναι ένα άτομο για το οποίο ισχύει ότι συμπαθεί τον εαυτό του”].

- **DataSomeValuesFrom** := 'DataSomeValuesFrom' (' **DataPropertyExpression** **DataRange** ')

Η συγκεκριμένη υπαρξιακή έκφραση κλάσης δέχεται δύο ορίσματα, μία έκφραση ιδιοτήτων δεδομένων και μια έκφραση εύρους δεδομένων. Τα επιτρεπόμενα εύρη δεδομένων για την OWL 2 EL θα οριστούν στην συνέχεια αλλά το συνηθέστερο εύρος δεδομένων που χρησιμοποιείται είναι ένα απλό datatype. Η έκφραση αυτή περιλαμβάνει όλα τα άτομα που συνδέονται μέσω μιας ιδιότητας δεδομένων με ένα επιτρεπόμενο εύρος δεδομένων (συνήθως ένα datatype) που δίνεται ως όρισμα.

Για παράδειγμα έχουμε την ιδιότητα δεδομένων a:hasName που συνδέει το άτομο a:John με το string “John Papadopoulos” [“Το άτομο Γιάννης έχει όνομα Γιάννης Παπαδόπουλος”] τότε η έκφραση DataSomeValuesFrom(a:hasName “John Papadopoulos”) περιέχει το άτομο a:John [“Ο Γιάννης είναι ένα άτομο για το οποίο ισχύει ότι έχει το όνομα Γιάννης Παπαδόπουλος”]

- **DataHasValue** := 'DataHasValue' (' **DataPropertyExpression** **Literal** ')

Η συγκεκριμένη έκφραση κλάσης μπορεί να θεωρηθεί ως μια συντόμευση της έκφρασης DataSomeValuesFrom στην περίπτωση που το δεύτερο όρισμα είναι μία σταθερά. Επομένως, η έκφραση αυτή περιλαμβάνει όλα τα άτομα που συνδέονται μέσω μιας ιδιότητας δεδομένων με μία σταθερά που δίνεται ως όρισμα.

Για παράδειγμα έχουμε την ιδιότητα δεδομένων a:hasAge που συνδέει το άτομο a:John με το literal “23”^^xsd:integer [“Το άτομο Γιάννης είναι 23 ετών”] τότε η

έκφραση `DataHasValue(a:hasAge "23"^^xsd:integer)` περιέχει το άτομο `a:John` [“Ο Γιάννης είναι ένα άτομο για το οποίο ισχύει ότι έχει ηλικία ίση με 23”]

Εκτός από τις υποστηριζόμενες εκφράσεις κλάσεων το profile OWL 2 EL υποστηρίζει μια σειρά από εκφράσεις εύρους δεδομένων (Data Ranges) οι οποίες αναλύονται ακολούθως.

DataRange := Datatype | DataIntersectionOf | DataOneOf

- **Datatype** (παραπομπή στο 2.4.1)
- **DataIntersectionOf** := 'DataIntersectionOf' (' DataRange DataRange { DataRange })'
Αντιπροσωπεύει την τομή δύο ή περισσότερων εκφράσεων εύρους δεδομένων και επιτρέπει όλα τα δεδομένα που είναι επιτρεπτά σε όλα τα εύρη δεδομένων που περιέχονται στα ορίσματα.
Για παράδειγμα το `DataIntersectionOf (xsd:nonNegativeInteger xsd:nonPositiveInteger)`³³ περιέχει το 0.
- **DataOneOf** := 'DataOneOf' (' Literal)'
Γενικά στην OWL 2 η έκφραση `DataOneOf` χρησιμοποιείται για την απαρίθμηση σταθερών³⁴. Συγκεκριμένα όμως στο profile OWL 2 EL επιτρέπεται μόνο μία σταθερά. Για παράδειγμα το `DataOneOf ("23"^^xsd:integer)` είναι μία απαρίθμηση με μόνο μία σταθερά - literal.

Στη συνέχεια θα εξετάσουμε τους επιτρεπόμενους τύπους αξιωμάτων στο profile OWL 2 EL. Τα αξιώματα διακρίνονται σε τρεις κατηγορίες, ανάλογα με τον τύπο των εκφράσεων που εμπλέκονται σε αυτά, τα αξιώματα κλάσεων (`ClassAxiom`), τα αξιώματα ιδιοτήτων αντικειμένων (`ObjectPropertyAxiom`) και τα αξιώματα ιδιοτήτων δεδομένων (`DataPropertyAxiom`).

Τα επιτρεπόμενα αξιώματα κλάσεων συνοψίζονται στο παρακάτω σχήμα και το καθένα από αυτά αναλύεται στη συνέχεια.

³³ Το παράδειγμα είναι εποπτικό καθώς τα συγκεκριμένα datatypes δεν είναι επιτρεπτά στην OWL 2 EL.

³⁴ **DataOneOf** := 'DataOneOf' (' Literal {Literal})' (π.χ. `DataOneOf ("John" "23"^^xsd:integer)`)

ClassAxiom := SubClassOf | EquivalentClasses | DisjointClasses

- **SubClassOf := 'SubClassOf' '(' subClassExpression superClassExpression ')'**
subClassExpression := ClassExpression
superClassExpression := ClassExpression

Το συγκεκριμένο αξίωμα κλάσης είναι το βασικότερο και συνηθέστερο αξίωμα στην περιγραφή μιας οντολογίας με τον φορμαλισμό OWL, είναι αντίστοιχο με την σχέση υπαγωγής ($A \sqsubseteq B$) της περιγραφικής λογικής και χρησιμοποιείται για την δημιουργία μιας ιεραρχίας κλάσεων. Στην περιγραφική λογική θα μπορούσαμε να γράψουμε $\text{subClassExpression} \sqsubseteq \text{superClassExpression}$. Σημασιολογικά το αξίωμα αυτό δηλώνει ότι όλα τα στιγμιότυπα της υποκλάσης είναι και στιγμιότυπα της υπερκλάσης.

Για παράδειγμα το αξίωμα `SubClassOf (a:Baby a:Child)` στην φυσική γλώσσα μπορεί να μεταφραστεί ως [“ Όποιο άτομο είναι μωρό είναι και παιδί”] ενώ το αξίωμα `SubClassOf (a:Parent ObjectSomeValuesFrom (a:hasChild a:Child))` αντιστοιχεί στο [“ Όποιο άτομο είναι γονιός έχει και την ιδιότητα ότι έχει τουλάχιστον ένα παιδί”].

- **EquivalentClasses := 'EquivalentClasses' '(' ClassExpression ClassExpression {ClassExpression} ')'**

Το συγκεκριμένο αξίωμα υλοποιεί τη σχέση ισοδυναμίας της περιγραφικής λογικής ($A \equiv B$), θέτει δύο ή περισσότερες εκφράσεις κλάσεων ισοδύναμες και σημασιολογικά δηλώνει ότι τα άτομα που είναι στιγμιότυπα της μιας έκφρασης κλάσης είναι σίγουρα στιγμιότυπα και των άλλων.

Για παράδειγμα το αξίωμα `EquivalentClasses(a:Boy ObjectIntersectionOf (a:Child a:Man))` αντιστοιχεί στο [“Το αγόρι είναι ένα αρσενικό παιδί”]

- **DisjointClasses := 'DisjointClasses' '(' ClassExpression ClassExpression {ClassExpression} ')'**

Με το συγκεκριμένο αξίωμα δύο ή περισσότερες εκφράσεις κλάσεων τίθενται ξένες μεταξύ τους, δηλαδή σημασιολογικά δηλώνει ότι κανένα άτομο δεν μπορεί να είναι ταυτόχρονα στιγμιότυπο σε δύο από τις εκφράσεις κλάσεων που δίνονται ως ορίσματα.

Για παράδειγμα το αξίωμα `DisjointClasses(a:Boy a:Girl)` αντιστοιχεί στο [“ Ένα άτομο δεν μπορεί να είναι ταυτόχρονα και αγόρι και κορίτσι”]

Τα επιτρεπόμενα αξιώματα ιδιοτήτων αντικειμένων συνοψίζονται στο παρακάτω σχήμα και αναλύονται στη συνέχεια.

```
ObjectPropertyAxiom :=  
EquivalentObjectProperties | SubObjectPropertyOf |  
ObjectPropertyDomain | ObjectPropertyRange |  
ReflexiveObjectProperty | TransitiveObjectProperty
```

- **EquivalentObjectProperties := 'EquivalentObjectProperties' '('**

ObjectPropertyExpression ObjectPropertyExpression {ObjectPropertyExpression} ')'

Το αξίωμα αυτό δηλώνει ότι δύο ή περισσότερες εκφράσεις ιδιότητας αντικειμένων είναι σημασιολογικά ισοδύναμες ή αλλιώς ότι αν δύο άτομα συνδέονται μεταξύ τους με μια ιδιότητα αντικειμένων τότε θα συνδέονται και με τις υπόλοιπες. Το αξίωμα αυτό είναι μια συντακτική συντόμευση των αξιωμάτων `SubObjectPropertyOf (OPE1 OPE2)` και `SubObjectPropertyOf (OPE2 OPE1)`.

Για παράδειγμα το αξίωμα `EquivalentObjectProperties(a:hasBoy a:hasMaleChild)` αντιστοιχεί στο [“Το να έχει κάποιος ένα αγόρι είναι το ίδιο με το να έχει ένα αρσενικό παιδί”]

- **SubObjectPropertyOf := 'SubObjectPropertyOf' '(' subObjectPropertyExpression**
superObjectPropertyExpression ')'

subObjectPropertyExpression := ObjectPropertyExpression | propertyExpressionChain

propertyExpressionChain := 'ObjectPropertyChain ' '(' ObjectPropertyExpression

ObjectPropertyExpression {ObjectPropertyExpression} ')'

superObjectPropertyExpression := ObjectPropertyExpression

Το συγκεκριμένο αξίωμα είναι αντίστοιχο με το αξίωμα υπαγωγής ρόλων της περιγραφικής λογικής ($r \sqsubseteq s$) και υποδηλώνει ότι η μια έκφραση ιδιότητας αντικειμένων είναι υποέκφραση ιδιότητας της άλλης. Σημασιολογικά το αξίωμα αυτό δηλώνει ότι αν ένα άτομο x συνδέεται μέσω της υποέκφρασης ιδιότητας αντικειμένων με ένα άτομο y τότε συνδέεται με το y και μέσω της υπερέκφρασης ιδιότητας αντικειμένων. Μία ειδική περίπτωση είναι όταν η υποέκφραση ιδιότητας δεν είναι μια απλή έκφραση ιδιότητας αντικειμένων αλλά είναι μία αλυσίδα

ιδιοτήτων. Σε αυτή την περίπτωση σημασιολογικά το αξίωμα δηλώνει ότι αν ένα άτομο x συνδέεται μέσω της αλυσίδας με το άτομο y τότε συνδέεται και μέσω της υπερέκφρασης ιδιότητας.

Για παράδειγμα το αξίωμα `SubObjectPropertyOf (a:hasDog a:hasPet)` αντιστοιχεί στο [“ Όποιο άτομο έχει σκυλί συνεπάγεται ότι έχει και κατοικίδιο”] ενώ το αξίωμα `SubObjectPropertyOf (ObjectPropertyChain (a:hasMother a:hasSister) a:hasAunt)` αντιστοιχεί στο [“ Όποιο άτομο έχει μητέρα η οποία έχει αδελφή συνεπάγεται ότι αυτό το άτομο έχει θεία”]

- **ObjectPropertyDomain := 'ObjectPropertyDomain' '(' ObjectPropertyExpression ClassExpression ')'**

Το συγκεκριμένο αξίωμα ορίζει το πεδίο ορισμού (domain) μιας έκφρασης ιδιότητας αντικειμένων, δηλαδή θέτει έναν περιορισμό (constraint) στις επιτρεπόμενες εκφράσεις κλάσεων που μπορούν να συνδεθούν μέσω μιας έκφρασης ιδιότητας αντικειμένων με άλλες εκφράσεις κλάσεων. Επομένως, σημασιολογικά εάν ένα άτομο x συνδέεται μέσω της έκφρασης ιδιότητας αντικειμένων που δίνεται ως όρισμα με ένα οποιοδήποτε άλλο άτομο τότε αυτό το άτομο x είναι στιγμιότυπο της δεδομένης έκφρασης κλάσεως. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubClassOf (ObjectSomeValuesFrom (OPE owl:Thing) CE)`.

Για παράδειγμα το αξίωμα `ObjectPropertyDomain (a:hasDog a:Person)` αντιστοιχεί στο ισοδύναμο στη φυσική γλώσσα [“ Όποιο άτομο έχει σκυλί πρέπει να είναι και άνθρωπος”].

- **ObjectPropertyRange := 'ObjectPropertyRange' '(' ObjectPropertyExpression ClassExpression ')'**

Το αξίωμα αυτό είναι αντίστοιχο με το προηγούμενο με την διαφορά ότι ορίζεται το πεδίο εμβέλειας (range) μιας έκφρασης ιδιότητας αντικειμένων, δηλαδή θέτει έναν περιορισμό (constraint) στις επιτρεπόμενες εκφράσεις κλάσεων, οι οποίες μέσω μιας έκφρασης ιδιότητας αντικειμένων μπορούν να συνδεθούν από άλλες εκφράσεις κλάσεων. Επομένως, σημασιολογικά εάν ένα οποιοδήποτε άτομο συνδέεται μέσω της έκφρασης ιδιότητας αντικειμένων που δίνεται ως όρισμα με ένα άτομο x τότε αυτό το

άτομο x είναι στιγμιότυπο της δεδομένης έκφρασης κλάσεως. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubClassOf (owl:Thing ObjectAllValuesFrom (OPE CE))`.

Για παράδειγμα το αξίωμα `ObjectPropertyRange (a:hasDog a:Dog)` αντιστοιχεί στο ισοδύναμο στη φυσική γλώσσα [“ Κάποιος μπορεί να έχει σκυλί μόνο κάτι που είναι σκυλί”].

▪ **ReflexiveObjectProperty** := 'ReflexiveObjectProperty' (' **ObjectPropertyExpression** ')

Το συγκεκριμένο αξίωμα υποδηλώνει ότι μια έκφραση ιδιότητας αντικειμένων είναι ανακλαστική (reflexive), δηλαδή σημασιολογικά ότι όλα τα άτομα συνδέονται με τον εαυτό τους μέσω της συγκεκριμένης έκφρασης ιδιότητας αντικειμένων. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubClassOf (owl:Thing ObjectHasSelf (OPE))`.

Για παράδειγμα το αξίωμα `ReflexiveObjectProperty (a:knows)` αντιστοιχεί στο [“ Κάθε άτομο γνωρίζει τον εαυτό του”].

▪ **TransitiveObjectProperty** := 'TransitiveObjectProperty' (' **ObjectPropertyExpression** ')

Το συγκεκριμένο αξίωμα υποδηλώνει ότι μια έκφραση ιδιότητας αντικειμένων είναι μεταβατική (transitive), δηλαδή σημασιολογικά ότι εάν ένα άτομο x συνδέεται μέσω της έκφρασης ιδιότητας αντικειμένων με το άτομο y και το άτομο y συνδέεται με το άτομο z μέσω της ίδιας έκφρασης ιδιότητας αντικειμένων τότε και το άτομο x θα συνδέεται με το άτομο z μέσω της συγκεκριμένης έκφρασης ιδιότητας αντικειμένων. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubObjectPropertyOf (ObjectPropertyChain (OPE OPE) OPE)`.

Για παράδειγμα το αξίωμα `TransitiveObjectProperty (a:ancestorOf)` αντιστοιχεί στο [“ Κάθε άτομο γνωρίζει τον εαυτό του”].

Τέλος, τα επιτρεπόμενα αξιώματα ιδιοτήτων δεδομένων (data property axiom) συνοψίζονται παρακάτω και αναλύονται στη συνέχεια.

DataPropertyAxiom :=
SubDataPropertyOf | **EquivalentDataProperties** |
DataPropertyDomain | **DataPropertyRange** | **FunctionalDataProperty**

- **SubDataPropertyOf** := 'SubDataPropertyOf' (' subDataPropertyExpression superDataPropertyExpression ')'

subDataPropertyExpression := **DataPropertyExpression**

superDataPropertyExpression := **DataPropertyExpression**

Το συγκεκριμένο αξίωμα υποδηλώνει ότι η μια έκφραση ιδιότητας δεδομένων είναι υποέκφραση ιδιότητας της άλλης. Σημασιολογικά το αξίωμα αυτό δηλώνει ότι αν ένα άτομο x συνδέεται μέσω της υποέκφρασης ιδιότητας δεδομένων με μία σταθερά (literal) y τότε συνδέεται με το y και μέσω της υπερέκφρασης ιδιότητας δεδομένων.

Για παράδειγμα το αξίωμα `SubDataPropertyOf (a:hasLastName a:hasName)` αντιστοιχεί στο [“ Όποιο άτομο έχει ένα επίθετο συνεπάγεται ότι έχει και ένα όνομα”].

- **EquivalentDataProperties** := 'EquivalentDataProperties' (' DataPropertyExpression DataPropertyExpression {DataPropertyExpression} ')'

Το αξίωμα αυτο δηλώνει ότι δύο ή περισσότερες εκφράσεις ιδιότητας δεδομένων είναι σηματολογικά ισοδύναμες ή αλλιώς ότι αν ένα άτομο συνδέεται με μία σταθερά (literal) y μέσω μιας έκφρασης ιδιότητας δεδομένων τότε θα συνδέεται και μέσω των υπολοίπων. Το αξίωμα αυτό είναι μια συντακτική συντόμευση των αξιωμάτων `SubDataPropertyOf (DPE1 DPE2)` και `SubDataPropertyOf (DPE2 DPE1)`.

Για παράδειγμα το αξίωμα `EquivalentDataProperties(a:hasName a:hatName)` δηλώνει ότι είναι ισοδύναμο ένα άτομο να συνδεθεί με το ονομά του μέσω μια ιδιότητας δεδομένων είτε εκφρασμένη στα Αγγλικά είτε στα Γερμανικά

- **DataPropertyDomain** := 'DataPropertyDomain' ('DataPropertyExpression ClassExpression')

Το συγκεκριμένο αξίωμα ορίζει το πεδίο ορισμού (domain) μιας έκφρασης ιδιότητας δεδομένων, δηλαδή θέτει έναν περιορισμό (constraint) στις επιτρεπόμενες εκφράσεις κλάσεων που μπορούν να συνδεθούν μέσω μιας έκφρασης ιδιότητας δεδομένων με οποιαδήποτε σταθερά (literal) . Επομένως, σηματολογικά εάν ένα άτομο x συνδέεται μέσω της έκφρασης ιδιότητας δεδομένων που δίνεται ως όρισμα με ένα οποιοδήποτε literal τότε αυτό το άτομο x είναι στιγμιότυπο της δεδομένης

έκφρασης κλάσεως. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubClassOf (DataSomeValuesFrom (DPE rdfs:Literal) CE)`.

Για παράδειγμα το αξίωμα `DataPropertyDomain (a:hasName a:Person)` αντιστοιχεί στο ισοδύναμο στη φυσική γλώσσα [“ Όποιο άτομο έχει όνομα πρέπει να είναι και άνθρωπος”].

- **DataPropertyRange := 'DataPropertyRange' (' DataPropertyExpression DataRange ')**

Το αξίωμα αυτό είναι αντίστοιχο με το προηγούμενο με την διαφορά ότι ορίζεται το πεδίο εμβέλειας (range) μιας έκφρασης ιδιότητας δεδομένων, δηλαδή θέτει έναν περιορισμό (constraint) στα επιτρεπόμενα εύρη δεδομένων, τα οποία μέσω μιας έκφρασης ιδιότητας δεδομένων μπορούν να συνδεθούν με κάποιες εκφράσεις κλάσεων. Επομένως, σημασιολογικά εάν ένα οποιοδήποτε άτομο συνδέεται μέσω της έκφρασης ιδιότητας δεδομένων που δίνεται ως όρισμα με μία σταθερά (literal) x τότε αυτό η σταθερά x ανήκει στο δεδομένο εύρος δεδομένων. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubClassOf (owl:Thing DataAllValuesFrom (DPE DR)`).

Για παράδειγμα το αξίωμα `DataPropertyRange (a:hasName xsd:string)` αντιστοιχεί στο ισοδύναμο στη φυσική γλώσσα [“ Κάποιος μπορεί να έχει όνομα μόνο κάτι που είναι τύπου συμβολοσειράς”].

- **FunctionalDataProperty := 'FunctionalDataProperty' (' DataPropertyExpression ')**

Το συγκεκριμένο αξίωμα υποδηλώνει ότι μια έκφραση ιδιότητας δεδομένων είναι λειτουργική (functional), δηλαδή σημασιολογικά ότι για όλα τα άτομα μπορεί να υπάρξει μέχρι μία το πολύ σταθερά (literal) με την οποία συνδεόνται μέσω της έκφρασης ιδιότητας δεδομένων. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubClassOf (owl:Thing DataMaxCardinality (1 DPE))`.

Για παράδειγμα το αξίωμα `FunctionalDataProperty (a:hasAge)` αντιστοιχεί στο [“ Κάθε άτομο έχει το πολύ μία ηλικία”].

2.4.3 OWL 2 QL

Το profile OWL 2 QL απευθύνεται σε εφαρμογές που χρησιμοποιούν πολύ μεγάλους όγκους δεδομένων και η απάντηση ερωτημάτων (query answering) είναι το σημαντικότερο πρόβλημα συλλογιστικής. Το ακρωνύμιο QL αντανακλά το γεγονός ότι η απάντηση ερωτημάτων σε αυτό το profile μπορεί να πραγματοποιηθεί ξαναγράφοντας ερωτήματα σε μία τυπική σχεσιακή γλώσσα ερωτημάτων (Query Language).

Στο profile OWL 2 QL, η συνδυαστική απάντηση ερωτήματος (conjunctive query answering) μπορεί να εφαρμοστούν με συμβατικά συστήματα σχεσιακών βάσεων δεδομένων. Χρησιμοποιώντας μια κατάλληλη τεχνική συλλογιστικής, μια ορθή και πλήρης διαδικασία απάντησης ερωτημάτων μπορεί να εκτελεστεί ακόμα και σε LOGSPACE χρόνο με σεβασμό στο μέγεθος των δεδομένων (ισχυρισμούς). Όπως και στην OWL 2 EL, αλγόριθμοι πολυωνυμικού χρόνου αλγόριθμοι μπορούν να χρησιμοποιηθούν για τα λοιπά προβλήματα συλλογιστικής όπως η συνέπεια οντολογία (ontology consistency) και η υπαγωγή εκφράσεων κλάσεων (class expression subsumption).

Επιπλέον, βασίζεται στην *DL-Lite* οικογένεια της περιγραφικής λογικής και η εκφραστική δύναμη του profile είναι αναγκαστικά αρκετά περιορισμένη, αν και περιλαμβάνει τα περισσότερα από τα κύρια χαρακτηριστικά των εννοιολογικών μοντέλων (conceptual model), όπως τα UML διαγράμματα κλάσεων και τα ER διαγράμματα.

Στο OWL 2 QL οι οντότητες (entities) και οι τύποι δεδομένων (datatypes) ορίζονται όπως έχουν παρουσιαστεί στην ενότητα 2.4.2 για το OWL 2 EL με τους ίδιους περιορισμούς σχετικά με τα υποστηριζόμενα datatypes και τη μη υποστήριξη ανώνυμων ατόμων (anonymous individuals).

Κύριο χαρακτηριστικό του OWL 2 QL και σημαντική διαφοροποίηση σε σχέση με το OWL 2 EL είναι η ύπαρξη περιορισμών σχετικά με τους τύπους των υποστηριζόμενων εκφράσεων που σχετίζονται με τις κλάσεις (ClassExpression) ανάλογα με το που εμπλέκονται αυτές οι εκφράσεις κλάσεων στα αξιώματα υπαγωγής. Για αυτό το λόγο

ορίζουμε τις εκφράσεις υποκλάσεων (*subClassExpression*) και τις εκφράσεις υπερκλάσεων (*superClassExpression*) αντίστοιχα.

Για τις εκφράσεις υποκλάσεων (*subClassExpression*) οι υποστηριζόμενες εκφράσεις συνοψίζονται στο ακόλουθο σχήμα και περιγράφονται αναλυτικά στη συνέχεια.

```
subClassExpression :=  
Class |  
subObjectSomeValuesFrom | DataSomeValuesFrom
```

- **Class** (βλ. 2.4.1)
- **DataSomeValuesFrom** := 'DataSomeValuesFrom' (' **DataPropertyExpression** **DataRange** ')
(βλ. **DataSomeValuesFrom** στην 2.4.2)
- **subObjectSomeValuesFrom** := 'ObjectSomeValuesFrom' (' **ObjectPropertyExpression** *owl:Thing* ')

Αντίστοιχο με το **ObjectSomeValuesFrom** που ορίστηκε στην ενότητα 2.4.2 με τη διαφορά ότι το όρισμα της έκφρασης κλάσης περιορίζεται στην παράμετρο *owl:Thing*. Επομένως αντιπροσωπεύει την υπαρξιακή ποσοτικοποίηση της περιγραφικής λογικής ($\exists r.T$) και περιέχει όλα τα άτομα που συνδέονται μέσω της ιδιότητας αντικειμένων με ένα τουλάχιστον άτομο.

Για παράδειγμα έχουμε την ιδιότητα αντικειμένων *a:fatherOf* που συνδέει τα άτομα *a:John* και *a:Alex* ["Ο Γιάννης είναι ο πατέρας του Αλέξη"] και έχουμε ότι το άτομο *a:Alex* είναι στιγμιότυπο της κλάσης *a:Man* ["Ο Αλέξης είναι άνδρας"] τότε το **ObjectSomeValuesFrom**(*a:fatherOf owl:Thing*) περιέχει το στιγμιότυπο *a:John* ["Ο Γιάννης είναι ένα άτομο για το οποίο ισχύει ότι είναι πατέρας κάποιου"].

Για τις εκφράσεις υπερκλάσεων (*superClassExpression*) οι υποστηριζόμενες εκφράσεις συνοψίζονται στο ακόλουθο σχήμα και περιγράφονται αναλυτικά στη συνέχεια.

```
superClassExpression :=  
Class |  
superObjectIntersectionOf | superObjectComplementOf |  
superObjectSomeValuesFrom | DataSomeValuesFrom
```

- **Class** (βλ. 2.4.1)

▪ **superObjectIntersectionOf** := 'ObjectIntersectionOf' '(' **superClassExpression** **superClassExpression** { **superClassExpression** })' (βλ. **ObjectIntersectionOf** στην 2.4.2)

▪ **superObjectComplementOf** := 'ObjectComplementOf' '(' **ClassExpression**)'

Η συγκεκριμένη έκφραση κλάσης περιέχει όλα τα άτομα (individual) που δεν είναι στιγμιότυπα της έκφρασης κλάσης που δίνεται ως όρισμα, δηλαδή αντιστοιχεί στη έννοια του συμπληρώματος.

Για παράδειγμα έχουμε τις κλάσεις a:Man και a:Woman, οι οποίες είναι ξένες μεταξύ τους. Αν το a:John είναι στιγμιότυπο της κλάσης a:Man [“Ο Γιάννης είναι άνδρας”] τότε το ObjectComplementOf (a:Woman) περιέχει το στιγμιότυπο a:John. [“Ο Γιάννης είναι ένα άτομο το οποίο δεν μπορεί να είναι γυναίκα”].

▪ **superObjectSomeValuesFrom** := 'ObjectSomeValuesFrom' '(' **ObjectPropertyExpression** **superClassExpression**)' (βλ. **ObjectSomeValuesFrom** στην 2.4.2)

▪ **DataSomeValuesFrom** := 'DataSomeValuesFrom' '(' **DataPropertyExpression** **DataRange**)' (βλ. **DataSomeValuesFrom** στην 2.4.2)

Εκτός από τις υποστηριζόμενες εκφράσεις κλάσεων το profile OWL 2 QL υποστηρίζει μια σειρά από εκφράσεις εύρους δεδομένων (Data Ranges) οι οποίες αναλύονται ακολούθως.

DataRange := **Datatype** | **DataIntersectionOf**

▪ **Datatype** (βλ. στην 2.4.1)

▪ **DataIntersectionOf** := 'DataIntersectionOf' '(' **DataRange** **DataRange** { **DataRange** })' (βλ. **DataIntersectionOf** στην 2.4.2)

Στη συνέχεια θα εξετάσουμε τους επιτρεπόμενους τύπους αξιωμάτων στο profile OWL 2 QL. Τα αξιώματα διακρίνονται σε τρεις κατηγορίες, ανάλογα με τον τύπο των εκφράσεων που εμπλέκονται σε αυτά, τα αξιώματα κλάσεων (**ClassAxiom**), τα αξιώματα ιδιοτήτων αντικειμένων (**ObjectPropertyAxiom**) και τα αξιώματα ιδιοτήτων δεδομένων (**DataPropertyAxiom**).

Τα επιτρεπόμενα αξιώματα κλάσεων συνοψίζονται στο παρακάτω σχήμα και το καθένα από αυτά αναλύεται στη συνέχεια.

ClassAxiom := **SubClassOf** | **EquivalentClasses** | **DisjointClasses**

- **SubClassOf** := 'SubClassOf' '(' **subClassExpression** **superClassExpression** ')'

(βλ. **SubClassOf** στην 2.4.2)

- **EquivalentClasses** := 'EquivalentClasses' '(' **subClassExpression** **subClassExpression** {**subClassExpression**} ')'

Αντίστοιχα με τα αξιώματα **EquivalentClasses** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στα αξιώματα ισοδυναμίας στην OWL 2 QL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υποκλάσεις.

- **DisjointClasses** := 'DisjointClasses' '(' **subClassExpression** **subClassExpression** {**subClassExpression**} ')'

Αντίστοιχα με τα αξιώματα **DisjointClasses** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στην OWL 2 QL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υποκλάσεις.

Τα επιτρεπόμενα αξιώματα ιδιοτήτων αντικειμένων συνοψίζονται στο παρακάτω σχήμα και αναλύονται στη συνέχεια.

```

ObjectPropertyAxiom :=
  SubObjectPropertyOf | EquivalentObjectProperties |
  DisjointObjectProperties | InverseObjectProperties |
  ObjectPropertyDomain | ObjectPropertyRange |
  ReflexiveObjectProperty |
  SymmetricObjectProperty | AsymmetricObjectProperty

```

- **EquivalentObjectProperties** := 'EquivalentObjectProperties' '('

ObjectPropertyExpression **ObjectPropertyExpression** {**ObjectPropertyExpression**} ')'

(βλ. **EquivalentObjectProperties** στην 2.4.2)

- **SubObjectPropertyOf** := 'SubObjectPropertyOf' '(' **ObjectPropertyExpression** **ObjectPropertyExpression** ')'

Αντίστοιχα με τα αξιώματα **SubObjectPropertyOf** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στα αξιώματα υπαγωγής ιδιοτήτων αντικειμένων στην OWL 2 QL δεν μπορούν να χρησιμοποιηθούν αλυσίδες αντικειμένων παρά μόνο εκφράσεις ιδιοτήτων αντικειμένων.

- **DisjointObjectProperties** := 'DisjointObjectProperties' '(' **ObjectPropertyExpression** **ObjectPropertyExpression** {**ObjectPropertyExpression**} ')'

Το αξίωμα αυτο δηλώνει ότι δύο ή περισσότερες εκφράσεις ιδιότητας αντικειμένων είναι σημασιολογικά ξένες ή αλλιώς ότι αν δύο άτομα συνδεόνται μεταξύ τους με μια ιδιότητα αντικειμένων τότε δεν θα μπορούν να συνδεόνται και με κάποια από τις υπόλοιπες.

Για παράδειγμα το αξίωμα `DisjointObjectProperties(a:hasFather a:hasMother)` αντιστοιχεί στο [“Δεν μπορεί ένα άτομο να είναι ταυτόχρονα και πατέρας και μητέρα ενός άλλου ατόμου”]

- **InverseObjectProperties** := 'InverseObjectProperties' '(' **ObjectPropertyExpression** **ObjectPropertyExpression** ')'

Το αξίωμα αυτο δηλώνει ότι οι εκφράσεις ιδιότητας αντικειμένων που δίνονται ως ορίσματα είναι σημασιολογικά αντίστροφες ή αλλιώς ότι αν ένα άτομο x συνδέεται μέσω την μίας ιδιότητας με ένα άτομο y , τότε το y συνδέεται μέσω τις άλλης ιδιότητας με το άτομο x.

Για παράδειγμα το αξίωμα `InverseObjectProperties(a:hasFather a:fatherOf)` αντιστοιχεί στο [“Αν ένα άτομο x έχει πατέρα ένα άτομο y τότε το y είναι πατέρας του ατόμου x”]

- **ObjectPropertyDomain** := 'ObjectPropertyDomain' '(' **ObjectPropertyExpression** **superClassExpression** ')'

Αντίστοιχα με τα αξιώματα **ObjectPropertyDomain** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στην OWL 2 QL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υπερκλάσεις (`superClassExpression`).

- **ObjectPropertyRange** := 'ObjectPropertyRange' '(' **ObjectPropertyExpression** **superClassExpression** ')'

Αντίστοιχα με τα αξιώματα **ObjectPropertyRange** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στην OWL 2 QL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υπερκλάσεις (`superClassExpression`).

- **ReflexiveObjectProperty** := 'ReflexiveObjectProperty' '(' **ObjectPropertyExpression** ')'
(βλ. **ReflexiveObjectProperty** στην 2.4.2)

- SymmetricObjectProperty** := 'SymmetricObjectProperty' (' **ObjectPropertyExpression** ')

Το συγκεκριμένο αξίωμα υποδηλώνει ότι μια έκφραση ιδιότητας αντικειμένων είναι συμμετρική (symmetric), δηλαδή σημασιολογικά ότι εάν ένα άτομο x συνδέεται μέσω της έκφρασης ιδιότητας αντικειμένων με το άτομο y τότε και το άτομο x θα συνδέεται με το άτομο y μέσω της συγκεκριμένης έκφρασης ιδιότητας αντικειμένων. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος SubObjectPropertyOf (OPE ObjectInverseOf (OPE)).

Για παράδειγμα το αξίωμα SymmetricObjectProperty (a:hasFriend) αντιστοιχεί στο [“ Αν το άτομο x έχει φίλο το άτομο y, τότε και το άτομο y έχει φίλο το άτομο x”].
- AsymmetricObjectProperty** :='AsymmetricObjectProperty' ('**ObjectPropertyExpression**')

Το συγκεκριμένο αξίωμα υποδηλώνει ότι μια έκφραση ιδιότητας αντικειμένων είναι μη συμμετρική (asymmetric), δηλαδή σημασιολογικά ότι εάν ένα άτομο x συνδέεται μέσω της έκφρασης ιδιότητας αντικειμένων με το άτομο y τότε το άτομο x δεν μπορεί να συνδέεται με το άτομο y μέσω της συγκεκριμένης έκφρασης ιδιότητας αντικειμένων.

Για παράδειγμα το αξίωμα AsymmetricObjectProperty (a:parentOf) αντιστοιχεί στο [“ Αν το άτομο x είναι γονέας του ατόμου y, τότε το άτομο y δεν μπορεί να είναι γονέας του ατόμου x”].

Τέλος, τα επιτρεπόμενα αξιώματα ιδιοτήτων δεδομένων (data property axiom) συνοψίζονται παρακάτω και αναλύονται στη συνέχεια.

DataPropertyAxiom :=
SubDataPropertyOf | **EquivalentDataProperties** | **DisjointDataProperties** |
DataPropertyDomain | **DataPropertyRange**

- SubDataPropertyOf** := 'SubDataPropertyOf' (' **subDataPropertyExpression** **superDataPropertyExpression** ')

subDataPropertyExpression := **DataPropertyExpression**

superDataPropertyExpression := DataPropertyExpression

(βλ. **SubDataPropertyOf** στην 2.4.2)

- **EquivalentDataProperties := 'EquivalentDataProperties' '(' DataPropertyExpression DataPropertyExpression {DataPropertyExpression} ')'**

(βλ. **SubDataPropertyOf** στην 2.4.2)

- **DisjointDataProperties := 'DisjointDataProperties' '(' DataPropertyExpression DataPropertyExpression {DataPropertyExpression} ')'**

Το αξίωμα αυτο δηλώνει ότι δύο ή περισσότερες εκφράσεις ιδιότητας δεδομένων είναι σημασιολογικά ξένες ή αλλιώς ότι αν ένα άτομο συνδέεται με μία σταθερά (literal) y μέσω μιας έκφρασης ιδιότητας δεδομένων τότε δεν θα μπορεί να συνδέεται και μέσω των υπολοίπων.

Για παράδειγμα το αξίωμα **DisjointDataProperties(a:hasName a:hasAge)** δηλώνει ότι ένα άτομο δεν μπορεί να έχει το ίδιο όνομα και την ίδια ηλικία.

- **DataPropertyDomain := 'DataPropertyDomain' '('DataPropertyExpression superClassExpression')**

Αντίστοιχα με τα αξιώματα **DataPropertyDomain** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στην OWL 2 QL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υπερκλάσεις (superClassExpression).

- **DataPropertyRange := 'DataPropertyRange' '(' DataPropertyExpression DataRange ')'**

(βλ. **SubDataPropertyOf** στην 2.4.2)

2.4.4 OWL 2 RL

Το τρίτο και τελευταίο profile που θα μελετήσουμε είναι το OWL 2 RL, το οποίο απευθύνεται σε εφαρμογές που απαιτούν κλιμακούμενη συλλογιστική ανάλυση (scalable reasoning) χωρίς να θυσιάζουν αρκετή εκφραστική δύναμη. Το ακρωνύμιο RL αντανακλά το γεγονός ότι η συλλογιστική ανάλυση σε αυτό το

προφίλ μπορεί να γίνει με τη χρήση μια τυπικής γλώσσας κανόνων (Rule Language) μέσω συλλογιστικών μηχανών βασισμένων σε κανόνες.

Τα σημαντικότερα προβλήματα συλλογιστικής μπορούν να επιλυθούν σε πολυωνυμικό χρόνο με σεβασμό στο μέγεθος της οντολογίας, για παράδειγμα η τυπική συλλογιστική ανάλυση σε OWL 2 RL είναι PTIME-complete.

Στο OWL 2 RL οι οντότητες (entities) και οι τύποι δεδομένων (datatypes) ορίζονται όπως έχουν παρουσιαστεί στην ενότητα 2.4.1 με εξαίρεση την υποστήριξη των παρακάτω datatypes (*owl:topObjectProperty*, *owl:bottomObjectProperty*, *owl:topDataProperty*, *owl:bottomDataProperty*, *owl:real* και *owl:rational*). Επίσης στα αξιώματα ισχυρισμών (Class Assertions) μπορούν να συμμετέχουν μόνο εκφράσεις κλάσεων με τους περιορισμούς (που θα παρουσιαστούν στη συνέχεια) για τις εκφράσεις υπερκλάσεων.

Όπως και στο OWL 2 QL, έτσι και στο OWL 2 RL υπάρχουν περιορισμοί σχετικά με τους τύπους των υποστηριζόμενων εκφράσεων που σχετίζονται με τις κλάσεις (Class Expression) ανάλογα με το που εμπλέκονται αυτές οι εκφράσεις κλάσεων στα αξιώματα. Για αυτό το λόγο ορίζουμε τις εκφράσεις υποκλάσεων (subClassExpression), τις εκφράσεις υπερκλάσεων (superClassExpression) και τις εκφράσεις κλάσεων ισοδυναμίας (equivalentClassExpression) αντίστοιχα.

Για τις εκφράσεις υποκλάσεων (subClassExpression) οι υποστηριζόμενες εκφράσεις συνοψίζονται στο ακόλουθο σχήμα και περιγράφονται αναλυτικά στη συνέχεια.

```
subClassExpression :=  
  Class other than owl:Thing |  
  subObjectIntersectionOf | subObjectUnionOf | ObjectOneOf |  
  subObjectSomeValuesFrom | ObjectHasValue |  
  DataSomeValuesFrom | DataHasValue
```

- **Class** (βλ. 2.4.1)
- **subObjectIntersectionOf** := 'ObjectIntersectionOf' '(' **subClassExpression** **subClassExpression** { **subClassExpression** } ')'

Αντίστοιχα με τα αξιώματα **ObjectIntersectionOf** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στα αξιώματα ισοδυναμίας στην OWL 2 RL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υποκλάσεις.

- **subObjectUnionOf** := 'ObjectUnionOf' (' **subClassExpression** **subClassExpression** { **subClassExpression** })'

Αντιπροσωπεύει την ένωση δύο ή περισσότερων εκφράσεων υποκλάσεων και περιέχει όλα τα άτομα που είναι στιγμιότυπα τουλάχιστον σε μία έκφραση υποκλάσεων που περιέχονται στα ορίσματα. Στην περιγραφική λογική DL αντιστοιχεί με το $A \sqcup B \sqcup C$

Για παράδειγμα έχουμε την κλάση *a:Man* με στιγμιότυπο *a:John* [“Ο Γιάννης είναι άνδρας”] και την κλάση *a:Woman* που περιέχει επίσης το στιγμιότυπο *a:Mary* [“Η Μαίρη είναι γυναίκα”] τότε το **ObjectUnionOf** (*a:Man* *a:Woman*) περιέχει τα στιγμιότυπα *a:John* και *a:Mary* [“Ο Γιάννης και η Μαίρη είναι άτομα τα οποία είναι είτε άντρας είτε γυναίκα”].

- **ObjectOneOf** := 'ObjectOneOf' (' **Individual** ') (βλ. **ObjectOneOf** στην 2.4.2)
- **subObjectSomeValuesFrom** := 'ObjectSomeValuesFrom' (' **ObjectPropertyExpression** *owl:Thing*)'
subObjectSomeValuesFrom := 'ObjectSomeValuesFrom' (' **ObjectPropertyExpression** **subClassExpression** ') (βλ. **subObjectSomeValuesFrom** στην 2.4.3 και **ObjectSomeValuesFrom** στην 2.4.2 για υποκλάσεις)
- **ObjectHasValue** := 'ObjectHasValue' (' **ObjectPropertyExpression** **Individual** ') (βλ. **ObjectHasValue** στην 2.4.2)
- **DataSomeValuesFrom** := 'DataSomeValuesFrom' (' **DataPropertyExpression** **DataRange** ') (βλ. **DataSomeValuesFrom** στην 2.4.2)
- **DataHasValue** := 'DataHasValue' (' **DataPropertyExpression** **Literal** ') (βλ. **ObjectHasValue** στην 2.4.2)

Για τις εκφράσεις υπερκλάσεων (*superClassExpression*) οι υποστηριζόμενες εκφράσεις συνοψίζονται στο ακόλουθο σχήμα και περιγράφονται αναλυτικά στη συνέχεια.

```

superClassExpression :=
  Class other than owl:Thing |
  superObjectIntersectionOf | superObjectComplementOf |
  superObjectAllValuesFrom | ObjectHasValue | superObjectMaxCardinality |
  DataAllValuesFrom | DataHasValue | superDataMaxCardinality

```

- **Class** (βλ. 2.4.1)
- **superObjectIntersectionOf** := 'ObjectIntersectionOf' '(' **superClassExpression** **superClassExpression** { **superClassExpression** })' (βλ. **ObjectIntersectionOf** στην 2.4.2)
- **superObjectComplementOf** := 'ObjectComplementOf' '(' **superClassExpression**)' (βλ. **superObjectComplementOf** στην 2.4.3)
- **superObjectAllValuesFrom** := 'ObjectAllValuesFrom' '(' **ObjectPropertyExpression** **superClassExpression**)'

Αντιπροσωπεύει τον περιορισμό τιμής της περιγραφικής λογικής ($\forall r.C$). Αποτελείται από δύο ορίσματα, μία έκφραση ιδιοτήτων αντικειμένων και μία έκφραση υπερκλάσης, και περιέχει όλα τα άτομα που συνδέονται μέσω της ιδιότητας αντικειμένων μόνο με άτομα, τα οποία είναι στιγμιότυπα της έκφρασης υπερκλάσης που δίνεται ως όρισμα.

Για παράδειγμα έχουμε την ιδιότητα αντικειμένων `a:hasPet` που συνδέει τα άτομα `a:John` και `a:Jack` και `a:John` και `a:Fox` [“Ο Γιάννης έχει κατοικίδιο τον Jack” , “Ο Γιάννης έχει κατοικίδιο τον Fox”] και έχουμε ότι τα άτομα `a:Jack` και `a:Fox` είναι στιγμιότυπα της κλάσης `a:Dog` [“Ο Jack είναι σκύλος” , “Ο Fox είναι σκύλος”] τότε το `ObjectAllValuesFrom(a:hasPet a:Dog)` περιέχει το στιγμιότυπο `a:John` [“Ο Γιάννης είναι ένα άτομο για το οποίο ισχύει ότι όλα τα κατοικίδια που έχει είναι σκυλιά”].

- **ObjectHasValue** := 'ObjectHasValue' '(' **ObjectPropertyExpression** **Individual**)' (βλ. **ObjectHasValue** στην 2.4.2)

- **superObjectMaxCardinality** := 'ObjectMaxCardinality' '(' **zeroOrOne** **ObjectPropertyExpression** [**subClassExpression**])' | 'ObjectMaxCardinality' '(' **zeroOrOne** **ObjectPropertyExpression** owl:Thing)'

Αντιπροσωπεύει τόσο τον περιορισμό αριθμού ($\leq nr$) όσο και τον ποιοτικό περιορισμού αριθμού ($\leq nr.C$) της περιγραφικής λογικής με $n=0$ ή 1 . Αποτελείται από

τρία ορίσματα, τον πληθικό αριθμό, μία έκφραση ιδιοτήτων αντικειμένων και μία έκφραση υποκλάσης είτε το *owl:Thing*, και περιέχει όλα τα άτομα που συνδέονται μέσω της ιδιότητας αντικειμένων με κανένα ή το πολύ 1 άτομο. Αν υπάρχει και το τρίτο όρισμα θα πρέπει επιπλέον τα άτομα αυτά να είναι στιγμιότυπα της έκφρασης υποκλάσης που δίνεται ως όρισμα.

Για παράδειγμα έχουμε την ιδιότητα αντικειμένων *a:hasPet* που συνδέει τα άτομα *a:John* και *a:Jack* [“Ο Γιάννης έχει κατοικίδιο τον Jack] και έχουμε ότι το άτομο *a:Jack* είναι στιγμιότυπο της κλάσης *a:Dog* [“Ο Jack είναι σκύλος”] τότε το *ObjectMaxCardinality(1 a:hasPet a:Dog)* περιέχει το στιγμιότυπο *a:John* [“Ο Γιάννης είναι ένα άτομο για το οποίο ισχύει ότι έχει το πολύ 1 κατοικίδιο το οποίο είναι σκυλί”].

- **DataAllValuesFrom := 'DataAllValuesFrom' (' DataPropertyExpression DataRange ')**

Η συγκεκριμένη καθολική έκφραση κλάσης δέχεται δύο ορίσματα, μία έκφραση ιδιοτήτων δεδομένων και μια έκφραση εύρους δεδομένων, που συνήθως είναι ένα datatype. Η έκφραση αυτή περιλαμβάνει όλα τα άτομα που συνδέονται μέσω μιας ιδιότητας δεδομένων αποκλειστικά με ένα επιτρεπόμενο εύρος δεδομένων (συνήθως ένα datatype) που δίνεται ως όρισμα.

Για παράδειγμα έχουμε την ιδιότητα δεδομένων *a:hasUserID* που συνδέει ένα άτομο με τα *UserID* που χρησιμοποιεί, τα οποία μπορεί να είναι είτε ένα string είτε ένα integer τότε η έκφραση *DataAllValuesFrom(a:hasUserID xsd:integer)* περιέχει όλα τα άτομα που όλα τους τα *UserID* είναι ακέραιοι.

- **DataHasValue := 'DataHasValue' (' DataPropertyExpression Literal ')**

(βλ. **DataHasValue** στην 2.4.2)

- **superDataMaxCardinality := 'DataMaxCardinality' (' zeroOrOne DataPropertyExpression [DataRange] ')**

Αποτελείται από τρία ορίσματα, τον πληθικό αριθμό, μία έκφραση ιδιοτήτων δεδομένων και μία έκφραση εύρους δεδομένων (*rdfs:Literal* αν απουσιάζει), και

περιέχει όλα τα άτομα που συνδέονται μέσω της ιδιότητας δεδομένων με κανένα ή το πολύ 1 τύπο δεδομένων που ορίζονται στο εύρος.

Για παράδειγμα έχουμε την ιδιότητα αντικειμένων `a:hasName` που συνδέει το άτομο `a:John` με ένα δεδομένο τύπου `string` τότε το `DataMaxCardinality(1 a:hasName)` περιέχει το στιγμιότυπο `a:John` [“Ο Γιάννης είναι ένα άτομο για το οποίο ισχύει ότι έχει το πολύ ένα όνομα”].

Εκτός από τις υποστηριζόμενες εκφράσεις κλάσεων το profile OWL 2 RL υποστηρίζει μια σειρά από εκφράσεις εύρους δεδομένων (Data Ranges) οι οποίες αναλύονται ακολούθως.

DataRange := Datatype | DataIntersectionOf

- **Datatype** (βλ. στην 2.4.1)
- **DataIntersectionOf** := 'DataIntersectionOf' (' DataRange DataRange { DataRange } ')
(βλ. **DataIntersectionOf** στην 2.4.2)

Στη συνέχεια θα εξετάσουμε τους επιτρεπόμενους τύπους αξιωμάτων στο profile OWL 2 RL. Τα επιτρεπόμενα αξιώματα κλάσεων (ClassAxioms) συνοψίζονται στο παρακάτω σχήμα και το καθένα από αυτά αναλύεται στη συνέχεια.

ClassAxiom := SubClassOf | EquivalentClasses | DisjointClasses

- **SubClassOf** := 'SubClassOf' (' subClassExpression superClassExpression ')
(βλ. **SubClassOf** στην 2.4.2)
- **EquivalentClasses** := 'EquivalentClasses' (' subClassExpression subClassExpression {subClassExpression} ')

Αντίστοιχα με τα αξιώματα **EquivalentClasses** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στα αξιώματα ισοδυναμίας στην OWL 2 QL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υποκλάσεις.

- **DisjointClasses** := 'DisjointClasses' (' subClassExpression subClassExpression {subClassExpression} ')

Αντίστοιχα με τα αξιώματα **DisjointClasses** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στην OWL 2 QL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υποκλάσεις.

Τα επιτρεπόμενα αξιώματα ιδιοτήτων (Object Property Axioms) αντικειμένων συνοψίζονται στο παρακάτω σχήμα και αναλύονται στη συνέχεια.

ObjectPropertyAxiom :=
SubObjectPropertyOf | EquivalentObjectProperties |
DisjointObjectProperties | InverseObjectProperties |
ObjectPropertyDomain | ObjectPropertyRange |
FunctionalObjectProperty | InverseFunctionalObjectProperty |
IrreflexiveObjectProperty |
SymmetricObjectProperty | AsymmetricObjectProperty
TransitiveObjectProperty

- **SubObjectPropertyOf** (βλ. **SubObjectPropertyOf** στην 2.4.2)
- **EquivalentObjectProperties** := 'EquivalentObjectProperties' '('
ObjectPropertyExpression **ObjectPropertyExpression** {**ObjectPropertyExpression**} ')'
(βλ. **EquivalentObjectProperties** στην 2.4.2)
- **DisjointObjectProperties** := 'DisjointObjectProperties' '(' **ObjectPropertyExpression**
ObjectPropertyExpression {**ObjectPropertyExpression**} ')'
(βλ. **EquivalentObjectProperties** στην 2.4.3)
- **InverseObjectProperties** := 'InverseObjectProperties' '(' **ObjectPropertyExpression**
ObjectPropertyExpression ')'
(βλ. **InverseObjectProperties** στην 2.4.3)
- **ObjectPropertyDomain** := 'ObjectPropertyDomain' '(' **ObjectPropertyExpression**
superClassExpression ')'
Αντίστοιχα με τα αξιώματα **ObjectPropertyDomain** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στην OWL 2 RL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υπερκλάσεις (**superClassExpression**).
- **ObjectPropertyRange** := 'ObjectPropertyRange' '(' **ObjectPropertyExpression**
superClassExpression ')'
Αντίστοιχα με τα αξιώματα **ObjectPropertyRange** που ορίστηκαν στην 2.4.2 με την διαφορά ότι στην OWL 2 RL μπορούν να εμπλακούν μόνο εκφράσεις κλάσεων με τους περιορισμούς που ορίστηκαν για τις υπερκλάσεις (**superClassExpression**).
- **FunctionalObjectProperty** := 'FunctionalObjectProperty' '(' **ObjectPropertyExpression** ')'

Το συγκεκριμένο αξίωμα υποδηλώνει ότι μια έκφραση ιδιότητας αντικειμένων είναι λειτουργική (functional), δηλαδή σημασιολογικά ότι για κάθε άτομο x μπορεί να υπάρχει το πολύ ένα άτομο y με το οποίο να συνδέεται μέσω της έκφρασης ιδιότητας αντικειμένων. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubClassOf (owl:Thing ObjectMaxCardinality (1 OPE))`.

Για παράδειγμα το αξίωμα `FunctionalObjectProperty (a:hasFather)` αντιστοιχεί στο [“Κάθε άτομο μπορεί να έχει το πολύ έναν πατέρα”].

- **InverseFunctionalObjectProperty** := 'InverseFunctionalObjectProperty' (' **ObjectPropertyExpression** ')

Το συγκεκριμένο αξίωμα υποδηλώνει ότι μια έκφραση ιδιότητας αντικειμένων είναι αντίστροφα λειτουργική (inverse-functional), δηλαδή σημασιολογικά ότι για κάθε άτομο x μπορεί να υπάρχει το πολύ ένα άτομο y το οποίο να συνδέεται μέσω της έκφρασης ιδιότητας αντικειμένων με το x. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubClassOf (owl:Thing ObjectMaxCardinality (1 ObjectInverseOf(OPE)))`.

Για παράδειγμα το αξίωμα `FunctionalObjectProperty (a:fatherOf)`.

- **IrreflexiveObjectProperty** := 'FunctionalObjectProperty' (' **ObjectPropertyExpression** ')

Το συγκεκριμένο αξίωμα υποδηλώνει ότι μια έκφραση ιδιότητας αντικειμένων είναι μη ανακλαστική (irreflexive), δηλαδή σημασιολογικά ότι κανένα άτομο δεν μπορεί να συνδέεται με τον εαυτό τους μέσω της συγκεκριμένης έκφρασης ιδιότητας αντικειμένων. Το αξίωμα αυτό είναι μια συντακτική συντόμευση του αξιώματος `SubClassOf (ObjectHasSelf (OPE) owl:Nothing)`.

Για παράδειγμα το αξίωμα `ReflexiveObjectProperty (a:hasFather)` αντιστοιχεί στο [“ Κανένα άτομο δεν μπορεί να έχει πατέρα τον εαυτό του”].

- **SymmetricObjectProperty** := 'SymmetricObjectProperty' (' **ObjectPropertyExpression** ')
(βλ. **SymmetricObjectProperty** στην 2.4.3)
- **AsymmetricObjectProperty** := 'AsymmetricObjectProperty' ('**ObjectPropertyExpression**')
(βλ. **SymmetricObjectProperty** στην 2.4.3)

- **TransitiveObjectProperty** := 'TransitiveObjectProperty' '(' **ObjectPropertyExpression** ')'
(βλ. **TransitiveObjectProperty** στην 2.4.2)

2.4.5 Συγκεντρωτικό υλικό για την OWL 2

Ολοκληρώνοντας την αναφορά μας στην γλώσσα αναπαράστασης γνώσης OWL 2 μπορούμε να συνοψίσουμε στους παρακάτω πίνακες τη σύνταξη και τη σημασιολογία μέσω της διερμηνείας όλων των κατασκευαστών της γλώσσας που συναντήσαμε στα profiles καθώς και κάποιων κατασκευαστών που δεν συναντήσαμε σε κανένα profile αλλά περιλαμβάνονται στην OWL 2³⁵.

| | |
|---|--|
| ObjectInverseOf(OP) | $\{ (x, y) \mid (y, x) \in (OP)^{OP} \}$ |
| DataIntersectionOf(DR ₁ ... DR _n) | $(DR_1)^{DI} \cap \dots \cap (DR_n)^{DI}$ |
| DataUnionOf(DR ₁ ... DR _n) | $(DR_1)^{DI} \cup \dots \cup (DR_n)^{DI}$ |
| DataComplementOf(DR) | $(\Delta_D)^n \setminus (DR)^{DI}$ where n is the arity of DR |
| DataOneOf(lt ₁ ... lt _n) | $\{ (lt_1)^{LI}, \dots, (lt_n)^{LI} \}$ |
| DatatypeRestriction(DT F ₁ lt ₁ ... F _n lt _n) | $(DT)^{DT} \cap (F_1, lt_1)^{FA} \cap \dots \cap (F_n, lt_n)^{FA}$ |
| ObjectIntersectionOf(CE ₁ ... CE _n) | $(CE_1)^C \cap \dots \cap (CE_n)^C$ |
| ObjectUnionOf(CE ₁ ... CE _n) | $(CE_1)^C \cup \dots \cup (CE_n)^C$ |
| ObjectComplementOf(CE) | $\Delta_I \setminus (CE)^C$ |
| ObjectOneOf(a ₁ ... a _n) | $\{ (a_1)^I, \dots, (a_n)^I \}$ |
| ObjectSomeValuesFrom(OPE CE) | $\{ x \mid \exists y : (x, y) \in (OPE)^{OP} \text{ and } y \in (CE)^C \}$ |
| ObjectAllValuesFrom(OPE CE) | $\{ x \mid \forall y : (x, y) \in (OPE)^{OP} \text{ implies } y \in (CE)^C \}$ |
| ObjectHasValue(OPE a) | $\{ x \mid (x, (a)^I) \in (OPE)^{OP} \}$ |
| ObjectHasSelf(OPE) | $\{ x \mid (x, x) \in (OPE)^{OP} \}$ |
| ObjectMinCardinality(n OPE) | $\{ x \mid \#\{ y \mid (x, y) \in (OPE)^{OP} \} \geq n \}$ |
| ObjectMaxCardinality(n OPE) | $\{ x \mid \#\{ y \mid (x, y) \in (OPE)^{OP} \} \leq n \}$ |
| ObjectExactCardinality(n OPE) | $\{ x \mid \#\{ y \mid (x, y) \in (OPE)^{OP} \} = n \}$ |

³⁵ Δεν κρίνεται σκόπιμη για τις ανάγκες της εργασίας η αναλυτική παρουσίαση αυτών των κατασκευαστών.

| | |
|---|---|
| ObjectMinCardinality(n OPE CE) | $\{ x \mid \#\{ y \mid (x, y) \in (OPE)^{OP} \text{ and } y \in (CE)^C \} \geq n \}$ |
| ObjectMaxCardinality(n OPE CE) | $\{ x \mid \#\{ y \mid (x, y) \in (OPE)^{OP} \text{ and } y \in (CE)^C \} \leq n \}$ |
| ObjectExactCardinality(n OPE CE) | $\{ x \mid \#\{ y \mid (x, y) \in (OPE)^{OP} \text{ and } y \in (CE)^C \} = n \}$ |
| DataSomeValuesFrom(DPE ₁ ... DPE _n DR) | $\{ x \mid \exists y_1, \dots, y_n : (x, y_k) \in (DPE_k)^{DP} \text{ for each } 1 \leq k \leq n \text{ and } (y_1, \dots, y_n) \in (DR)^{DT} \}$ |
| DataAllValuesFrom(DPE ₁ ... DPE _n DR) | $\{ x \mid \forall y_1, \dots, y_n : (x, y_k) \in (DPE_k)^{DP} \text{ for each } 1 \leq k \leq n \text{ imply } (y_1, \dots, y_n) \in (DR)^{DT} \}$ |
| DataHasValue(DPE lt) | $\{ x \mid (x, (lt)^{LT}) \in (DPE)^{DP} \}$ |
| DataMinCardinality(n DPE) | $\{ x \mid \#\{ y \mid (x, y) \in (DPE)^{DP} \} \geq n \}$ |
| DataMaxCardinality(n DPE) | $\{ x \mid \#\{ y \mid (x, y) \in (DPE)^{DP} \} \leq n \}$ |
| DataExactCardinality(n DPE) | $\{ x \mid \#\{ y \mid (x, y) \in (DPE)^{DP} \} = n \}$ |
| DataMinCardinality(n DPE DR) | $\{ x \mid \#\{ y \mid (x, y) \in (DPE)^{DP} \text{ and } y \in (DR)^{DT} \} \geq n \}$ |
| DataMaxCardinality(n DPE DR) | $\{ x \mid \#\{ y \mid (x, y) \in (DPE)^{DP} \text{ and } y \in (DR)^{DT} \} \leq n \}$ |
| DataExactCardinality(n DPE DR) | $\{ x \mid \#\{ y \mid (x, y) \in (DPE)^{DP} \text{ and } y \in (DR)^{DT} \} = n \}$ |
| SubClassOf(CE ₁ CE ₂) | $(CE_1)^C \subseteq (CE_2)^C$ |
| EquivalentClasses(CE ₁ ... CE _n) | $(CE_j)^C = (CE_k)^C \text{ for each } 1 \leq j \leq n \text{ and each } 1 \leq k \leq n$ |
| DisjointClasses(CE ₁ ... CE _n) | $(CE_j)^C \cap (CE_k)^C = \emptyset \text{ for each } 1 \leq j \leq n \text{ and each } 1 \leq k \leq n \text{ such that } j \neq k$ |
| DisjointUnion(C CE ₁ ... CE _n) | $(C)^C = (CE_1)^C \cup \dots \cup (CE_n)^C \text{ and } (CE_j)^C \cap (CE_k)^C = \emptyset \text{ for each } 1 \leq j \leq n \text{ and each } 1 \leq k \leq n \text{ such that } j \neq k$ |
| SubObjectPropertyOf(OPE ₁ OPE ₂) | $(OPE_1)^{OP} \subseteq (OPE_2)^{OP}$ |
| SubObjectPropertyOf(ObjectPropertyChain(OPE ₁ ... OPE _n) OPE) | $\forall y_0, \dots, y_n : (y_0, y_1) \in (OPE_1)^{OP} \text{ and } \dots \text{ and } (y_{n-1}, y_n) \in (OPE_n)^{OP} \text{ imply } (y_0, y_n) \in (OPE)^{OP}$ |
| EquivalentObjectProperties(OPE ₁ ...) | $(OPE_j)^{OP} = (OPE_k)^{OP} \text{ for each } 1 \leq j \leq n \text{ and } 1 \leq k \leq n$ |

| | |
|---|---|
| OPE_n) | each $1 \leq k \leq n$ |
| DisjointObjectProperties($OPE_1 \dots OPE_n$) | $(OPE_j)^{OP} \cap (OPE_k)^{OP} = \emptyset$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$ such that $j \neq k$ |
| ObjectPropertyDomain($OPE CE$) | $\forall x, y: (x, y) \in (OPE)^{OP}$ implies $x \in (CE)^C$ |
| ObjectPropertyRange($OPE CE$) | $\forall x, y: (x, y) \in (OPE)^{OP}$ implies $y \in (CE)^C$ |
| InverseObjectProperties($OPE_1 OPE_2$) | $(OPE_1)^{OP} = \{(x, y) \mid (y, x) \in (OPE_2)^{OP}\}$ |
| FunctionalObjectProperty(OPE) | $\forall x, y_1, y_2: (x, y_1) \in (OPE)^{OP}$ and $(x, y_2) \in (OPE)^{OP}$ imply $y_1 = y_2$ |
| InverseFunctionalObjectProperty(OPE) | $\forall x_1, x_2, y: (x_1, y) \in (OPE)^{OP}$ and $(x_2, y) \in (OPE)^{OP}$ imply $x_1 = x_2$ |
| ReflexiveObjectProperty(OPE) | $\forall x: x \in \Delta_I$ implies $(x, x) \in (OPE)^{OP}$ |
| IrreflexiveObjectProperty(OPE) | $\forall x: x \in \Delta_I$ implies $(x, x) \notin (OPE)^{OP}$ |
| SymmetricObjectProperty(OPE) | $\forall x, y: (x, y) \in (OPE)^{OP}$ implies $(y, x) \in (OPE)^{OP}$ |
| AsymmetricObjectProperty(OPE) | $\forall x, y: (x, y) \in (OPE)^{OP}$ implies $(y, x) \notin (OPE)^{OP}$ |
| TransitiveObjectProperty(OPE) | $\forall x, y, z: (x, y) \in (OPE)^{OP}$ and $(y, z) \in (OPE)^{OP}$ imply $(x, z) \in (OPE)^{OP}$ |
| SubDataPropertyOf($DPE_1 DPE_2$) | $(DPE_1)^{DP} \subseteq (DPE_2)^{DP}$ |
| EquivalentDataProperties($DPE_1 \dots DPE_n$) | $(DPE_j)^{DP} = (DPE_k)^{DP}$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$ |
| DisjointDataProperties($DPE_1 \dots DPE_n$) | $(DPE_j)^{DP} \cap (DPE_k)^{DP} = \emptyset$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$ such that $j \neq k$ |
| DataPropertyDomain($DPE CE$) | $\forall x, y: (x, y) \in (DPE)^{DP}$ implies $x \in (CE)^C$ |
| DataPropertyRange($DPE DR$) | $\forall x, y: (x, y) \in (DPE)^{DP}$ implies $y \in (DR)^{DI}$ |
| FunctionalDataProperty(DPE) | $\forall x, y_1, y_2: (x, y_1) \in (DPE)^{DP}$ and $(x, y_2) \in (DPE)^{DP}$ imply $y_1 = y_2$ |
| SameIndividual($a_1 \dots a_n$) | $(a_j)^I = (a_k)^I$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$ |
| DifferentIndividuals($a_1 \dots a_n$) | $(a_j)^I \neq (a_k)^I$ for each $1 \leq j \leq n$ and each $1 \leq k \leq n$ such that $j \neq k$ |
| ClassAssertion($CE a$) | $(a)^I \in (CE)^C$ |
| ObjectPropertyAssertion($OPE a_1 a_2$) | $((a_1)^I, (a_2)^I) \in (OPE)^{OP}$ |

| | |
|---|---------------------------------------|
| NegativeObjectPropertyAssertion(OPE a ₁ a ₂) | $((a_1)', (a_2)') \notin (OPE)^{OP}$ |
| DataPropertyAssertion(DPE a lt) | $((a)', (lt)^{LT}) \in (DPE)^{DP}$ |
| NegativeDataPropertyAssertion(DPE a lt) | $((a)', (lt)^{LT}) \notin (DPE)^{DP}$ |

3

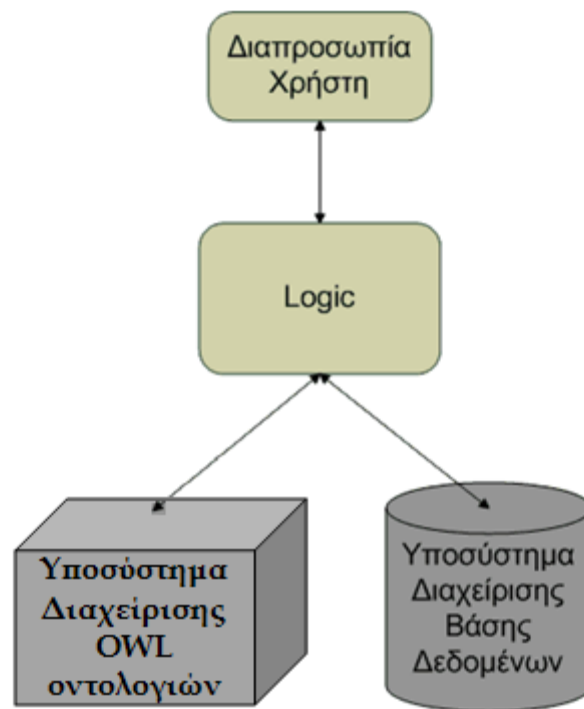
Ανάλυση Συστήματος

Στο κεφάλαιο αυτό αναλύουμε τις απαιτήσεις που ικανοποιεί το DBRS. Η μεθοδολογία παρουσίασης έχει ως βασικό άξονα το διαχωρισμό του συστήματος σε διακριτά υποσυστήματα (Ενότητα 3.1), την συνοπτική περιγραφή των λειτουργιών που αυτά επιτελούν (Ενότητα 3.2) και την ανάδειξη του τρόπου επικοινωνίας μεταξύ τους. Ως υποσύστημα ορίζουμε μια «οικογένεια» από εφαρμογές που είτε επιτελούν παρόμοιες λειτουργίες (και επομένως μπορούν να ομαδοποιηθούν) είτε συμμετέχουν στην εκτέλεση ενός αυτοτελούς καθήκοντος. Ένα υποσύστημα μπορεί να διαιρείται σε επιμέρους υποσυστήματα (υποσύνολα εφαρμογών) τα οποία, με τη σειρά τους, μπορεί επίσης να διαιρούνται σε άλλα κ.ο.κ.

3.1 Αρχιτεκτονική - Διαχωρισμός υποσυστημάτων

Το DBRS αποτελείται από τρία βασικά υποσυστήματα. Το υποσύστημα του χρήστη, το υποσύστημα διαχείρισης της βάσης δεδομένων και το υποσύστημα διαχείρισης της μηχανής συλλογιστικής ανάλυσης. Το πρώτο από αυτά διαιρείται σε τέσσερα υποσυστήματα, ένα εκ των οποίων (το υποσύστημα φόρτωσης της οντολογίας) διαιρείται σε δύο επιπλέον. Αφαιρετικά, θα μπορούσαμε να πούμε ότι το DBRS διαθέτει μια τριεπίπεδη (three-layered) αρχιτεκτονική η οποία φαίνεται στο Σχήμα 3.1. Το πρώτο επίπεδο αφορά στη γραφική διαπροσωπία με την οποία ο χρήστης έρχεται σε επαφή με το σύστημα, ενώ το δεύτερο περιλαμβάνει όλα εκείνα τα υποσυστήματα του DBRS που αποτελούν το επίπεδο Λογικής (Logic) και μεσολαβούν μεταξύ του πρώτου

και του τρίτου επιπέδου, δηλαδή μεταξύ της διαπροσωπίας του χρήστη και των υποσυστημάτων διαχείρισης της βάσης δεδομένων και διαχείρισης OWL οντολογιών. Το επίπεδο Λογικής του DBRS αποτελεί στην ουσία και τον πυρήνα (core) του συστήματος.



Σχήμα 3.1 Τριεπίπεδη Αρχιτεκτονική του συστήματος DBRS

Πιο συγκεκριμένα και όπως φαίνεται στο Σχήμα 3.2, η δομή του DBRS είναι:

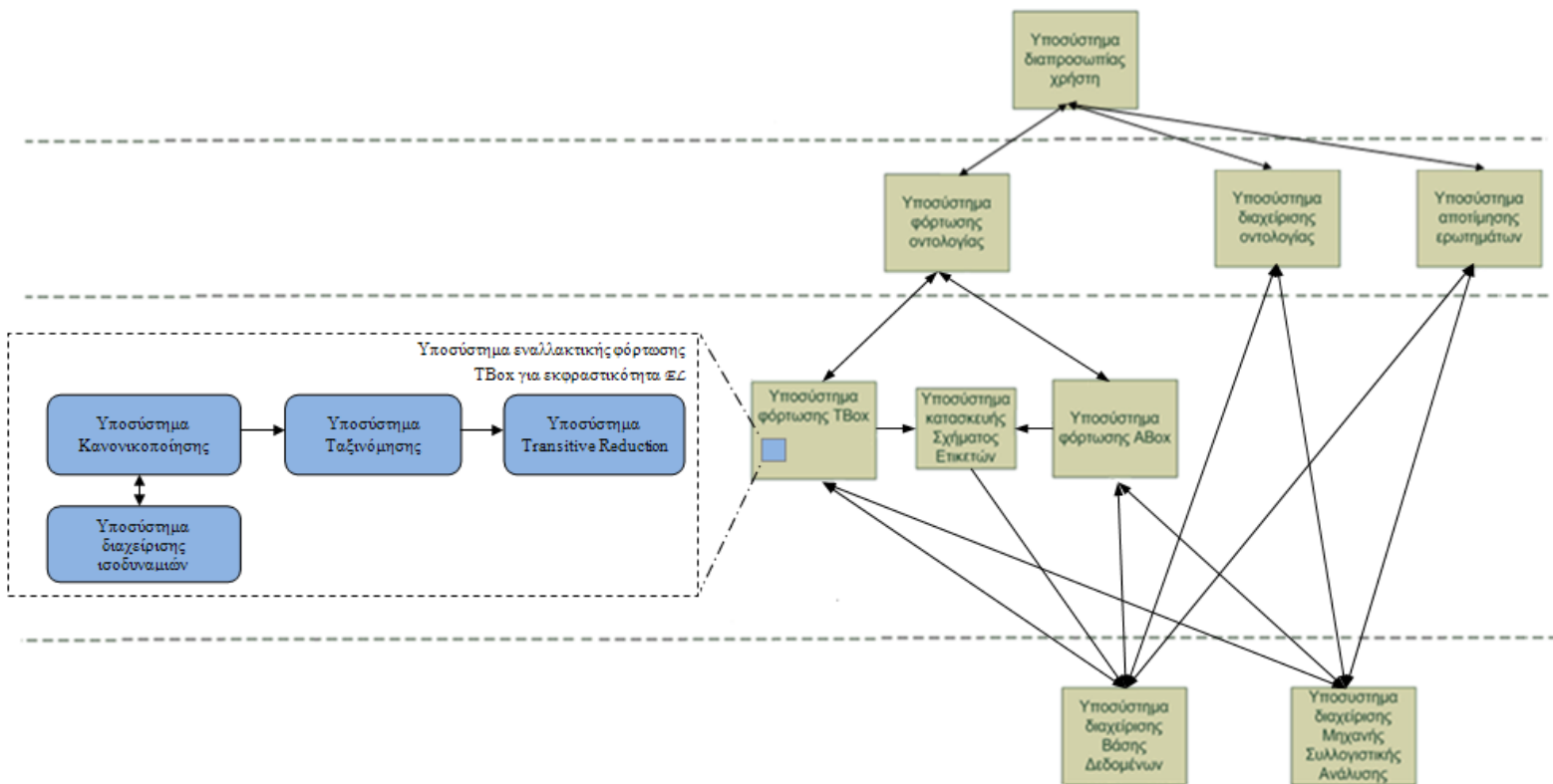
1. Υποσύστημα γραφικής διαπροσωπίας χρήστη
2. Υποσύστημα φόρτωσης οντολογίας
 - 2.1 Υποσύστημα φόρτωσης TBox
 - 2.1.1. Υποσύστημα κανονικοποίησης
 - 2.1.2. Υποσύστημα διαχείρισης ισοδυναμιών
 - 2.1.2. Υποσύστημα ταξινόμησης
 - 2.1.3. Υποσύστημα transitive reduction

2.2 Υποσύστημα φόρτωσης ABox

3. Υποσύστημα κατασκευής σχήματος ετικετών
4. Υποσύστημα διαχείρισης οντολογίας
5. Υποσύστημα αποτίμησης ερωτημάτων
6. Υποσύστημα διαχείρισης Βάσης Δεδομένων
7. Υποσύστημα διαχείρισης Μηχανής Συλλογιστικής Ανάλυσης

Η αντιστοιχία μεταξύ των δύο σχημάτων έχει ως εξής:

Το Υποσύστημα γραφικής διαπροσωπίας χρήστη του Σχήματος 3.2 αντιστοιχεί στη Διαπροσωπία Χρήστη, δηλαδή στο πρώτο επίπεδο του Σχήματος 3.1. Τα υπόλοιπα Υποσυστήματα του Σχήματος 3.2 αντιστοιχούν στο δεύτερο επίπεδο (Logic) του Σχήματος 3.1, με εξαίρεση τα Υποσυστήματα διαχείρισης Βάσης Δεδομένων και διαχείρισης OWL οντολογιών που αντιστοιχούν στο τρίτο και κατώτερο επίπεδο.



Σχήμα 3.2: Αρχιτεκτονική του συστήματος DBRS

3.2 Περιγραφή υποσυστημάτων

Στην ενότητα αυτή περιγράφεται συνοπτικά η λειτουργία κάθε υποσυστήματος που αναφέρθηκε προηγουμένως. Οι εφαρμογές που υλοποιούν αυτές τις λειτουργίες θα παρουσιαστούν λεπτομερώς στο επόμενο κεφάλαιο.

3.2.1 Υποσύστημα γραφικής διαπροσωπίας χρήστη

Το Υποσύστημα γραφικής διαπροσωπίας χρήστη αποτελεί το περιβάλλον με το οποίο ο χρήστης έρχεται σε επαφή με το σύστημα. Μέσω αυτού, του δίνεται η πρόσβαση στις λειτουργίες που προσφέρει το DBRS σχετικά με τη φόρτωση και διαχείριση οντολογιών, αλλά και με τη διεξαγωγή ερωτημάτων.

3.2.2 Υποσύστημα φόρτωσης οντολογίας

Το Υποσύστημα φόρτωσης οντολογίας περιλαμβάνει τις λειτουργίες που επιτελούνται κατά τη φόρτωση της οντολογίας στη βάση δεδομένων του DBRS, παραμετροποιημένες με βάση τις επιλογές του χρήστη. Διαιρείται στο Υποσύστημα φόρτωσης TBox και στο Υποσύστημα φόρτωσης ABox που σχετίζονται με τις λειτουργίες φόρτωσης του TBox και του ABox αντίστοιχα. Ειδική μνεία θα γίνει στα υποσυστήματα που σχετίζονται με την φόρτωση του TBox στην περίπτωση που η εκφραστικότητα της οντολογίας είναι \mathcal{ELH} .

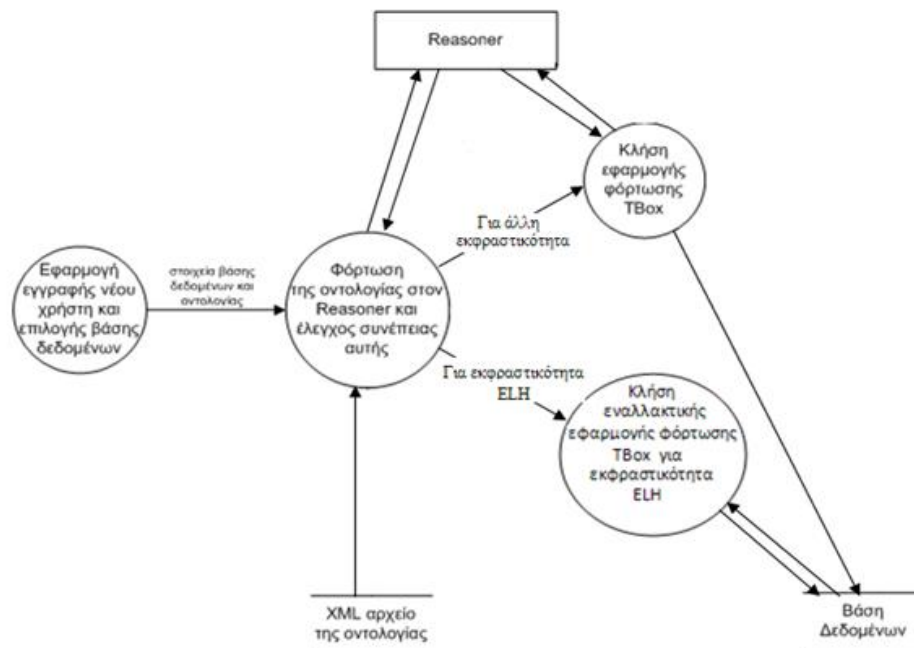
3.2.2.1 Υποσύστημα φόρτωσης TBox

Το Υποσύστημα φόρτωσης TBox περιλαμβάνει τις λειτουργίες που σχετίζονται με τη φόρτωση του TBox της οντολογίας. Αυτές είναι:

- Αποθήκευση Namespaces οντολογίας
- Αποθήκευση κλάσεων και κωδικοποίηση της ιεραρχίας τους μέσω κλήσης του Υποσυστήματος κατασκευής σχήματος ετικετών
- Αποθήκευση πληροφορίας περί ασυμβατότητας κλάσεων
- Αποθήκευση ρόλων και κωδικοποίηση της ιεραρχίας τους μέσω κλήσης του Υποσυστήματος κατασκευής σχήματος ετικετών

- Αποθήκευση πληροφορίας περί αντίστροφων ρόλων
- Αποθήκευση πληροφορίας περί χαρακτηριστικών ρόλων
- Αποθήκευση πληροφορίας σχετικά με το πεδίο (domain) και το εύρος (range) των ρόλων
- Αποθήκευση πληροφορίας περί ασυμβατότητας ρόλων
- Αποθήκευση ορισμών κλάσεων
- Κατασκευή και αποθήκευση του διευρυμένου TBox
- Αποθήκευση ψευδομοντέλων κλάσεων

Στο υποσύστημα φόρτωσης TBox μελετάμε ένα συγκεκριμένο εναλλακτικό υποσύστημα φόρτωσης TBox για την περίπτωση που η εκφραστικότητα της υπό εξέταση οντολογίας διαπιστωθεί ότι είναι *ELH*. Το διάγραμμα ροής για την επιλογή του εναλλακτικού σεναρίου φαίνεται στο Σχήμα 3.3 και τα επιμέρους υποσυστήματα αναλύονται στη συνέχεια.



Σχήμα 3.3 Διάγραμμα ροής δεδομένων για τη διαδικασία φόρτωσης των αξιωμάτων ανάλογα με την εκφραστικότητα της οντολογίας

3.2.2.1.1 Υποσύστημα κανονικοποίησης

Το Υποσύστημα κανονικοποίησης περιλαμβάνει τις λειτουργίες που επιτελούνται κατά την διαδικασία της κανονικοποίησης της οντολογίας και της εισαγωγής των εξαγόμενων δεδομένων στη βάση δεδομένων του DBRS. Πιο συγκεκριμένα επιτελεί τις παρακάτω λειτουργίες.

- Φόρτωση του σώματος της οντολογίας στο DBRS για την περαιτέρω επεξεργασία αυτής.
- Διάκριση του τύπου του κάθε αξιώματος που περιέχεται στην οντολογία
- Επεξεργασία του κάθε τύπου αξιώματος με την επαναληπτική εφαρμογή τυπικών καθορισμένων κανόνων κανονικοποίησης ανάλογα με τον τύπο του αξιώματος
- Ειδική μεταχείριση των αξιωμάτων ισοδυναμίας μέσω κλήσης του υποσυστήματος διαχείρισης ισοδυναμιών.
- Απόδοση ενός μοναδικού αναγνωριστικού σε όλες τις ‘σημασιολογικές ψηφίδες’ που συμμετέχουν στη σύνταξη των αξιωμάτων της οντολογίας μέσω τη χρήσης μιας συνάρτησης κατακερματισμού (hash) η οποία για την ίδια είσοδο επιστρέφει πάντα την ίδια έξοδο. Ο όρος ‘σημασιολογικές ψηφίδες’ μπορεί να αναφέρεται σε ατομικές κλάσεις, σύνθετες κλάσεις³⁶ και ρόλους..
- Αποθήκευση των πληροφοριών που προέκυψαν από την εφαρμογή των κανόνων κανονικοποίησης σε συγκεκριμένη μορφή στη βάση δεδομένων κατάλληλη για την επεξεργασία αυτών στο επόμενο στάδιο.

3.2.2.1.2 Υποσύστημα διαχείρισης ισοδυναμιών

Το Υποσύστημα διαχείρισης ισοδυναμιών περιλαμβάνει όλες τις λειτουργίες που σχετίζονται με τη διαχείριση των αξιωμάτων ισοδυναμίας κλάσεων τα οποία εμφανίζονται στο σώμα της οντολογίας. Το συγκεκριμένο υποσύστημα καλείται να επιτελέσει έναν τριπλό ρόλο

³⁶ Επειδή έχουμε εκφραστικότητα \mathcal{ELH} οι σύνθετες κλάσεις μπορεί να είναι μία τομή ατομικών κλάσεων, μία υπαρξιακή ποσοτικοποίηση, τομή άλλων σύνθετων κλάσεων και συνδυασμός αυτών.

- Να καταχωρεί σε μία συγκεκριμένη δομή όλες τις κλάσεις που είναι ισοδύναμες μεταξύ τους όπως αυτό καθορίζεται στα αξιώματα ισοδυναμίας.
- Μόλις ολοκληρώσει την εισαγωγή στην δομή όλων των ισοδύναμων κλάσεων να δημιουργήσει το βέλτιστο mapping ώστε όλες οι ισοδύναμες κλάσεις να αντιστοιχήσουν σε ένα μοναδικό hashcode (ένα από τα hashcode που έχει αποδοθεί σε κάποια από τις ισοδύναμες κλάσεις) και να αποθήκευσει αυτή τη δομή στη βάση δεδομένων.
- Να ενημερώσει όλους τους πίνακες της βάσης δεδομένων που προκύπτουν από την διαδικασία της κανονικοποίησης με τα αναθεωρημένα hashcode.

3.2.2.1.3 Υποσύστημα ταξινόμησης

Το υποσύστημα ταξινόμησης περιλαμβάνει όλες τις λειτουργίες που σχετίζονται με τον υπολογισμό όλων των υπονοούμενων σχέσεων ιεραρχίας μεταξύ των εννοιών της οντολογία μέσω της εξαντλητικής εφαρμογής ενός συνόλου από επαγωγικούς κανόνες και μπορεί να θεωρηθεί ως τον υπολογισμό ενός εκτεταμένου transitive closure. Επίσης περιλαμβάνει όλες τις λειτουργίες που επιτελούνται κατά την επιλογή του σχεδίου με το οποίο θα εφαρμοστούν οι επαγωγικοί κανόνες καθώς και τις λειτουργίες που σχετίζονται με την εφαρμογή των επαγωγικών κανόνων. Πιο συγκεκριμένα επιτελεί τις παρακάτω λειτουργίες.

- Δημιουργία διακριτών σταδίων εκτέλεσης του αλγορίθμου
- Καθορισμός των συμπερασματικών κανόνων που θα εφαρμοστούν και της σειράς αυτών ανάλογα με το στάδιο που βρίσκεται η διαδικασία
- Προετοιμασία της βάσης δεδομένων με τον καθορισμό συγκεκριμένων παραμέτρων με σκοπό την βελτιστοποίηση της διαδικασίας ανάλογα με το στάδιο που θα ακολουθήσει (π.χ. δημιουργία ευρετηρίων, κατάργηση ευρετηρίων, χρήση προσωρινών πινάκων, χρήση εναλλακτικών πινάκων κτλ.)
- Μετασχηματισμό του κάθε συμπερασματικού κανόνα στην αντίστοιχη αλληλουχία sql statements που απευθύνεται στη βάση δεδομένων και την

δημιουργία των ερωτημάτων για την αποθήκευση των εξαγόμενων από τη διαδικασία δεδομένων σε κατάλληλη μορφή στη βάση δεδομένων και την ενημέρωση των ήδη υπάρχοντων.

3.2.2.1.4 Υποσύστημα transitive reduction

Το υποσύστημα transitive reduction περιλαμβάνει όλες τις λειτουργίες που σχετίζονται με την επεξεργασία των δεδομένων που προέκυψαν από την διαδικασία της ταξινόμησης με σκοπό την δημιουργία μιας ταξινόμησης στην οποία θα περιλαμβάνονται μόνο άμεσες σχέσεις υπαγωγής μεταξύ των ατομικών εννοιών.

3.2.2.2 Υποσύστημα φόρτωσης ABox

Το Υποσύστημα φόρτωσης ABox περιλαμβάνει τις λειτουργίες που σχετίζονται με τη φόρτωση του ABox της οντολογίας. Αυτές είναι:

- Αποθήκευση ατόμων
- Αποθήκευση πληροφορίας περί ταυτότητας ατόμων
- Αποθήκευση πληροφορίας περί διαφορετικών ατόμων
- Αποθήκευση δηλώσεων κλάσεων
- Αποθήκευση δηλώσεων ρόλων αντικειμένου και ρόλων τύπου δεδομένων
- Αποθήκευση δηλώσεων μεταβατικών ρόλων και κωδικοποίησή τους μέσω κλήσης του Υποσυστήματος κατασκευής σχήματος ετικετών
- Αποθήκευση ψευδομοντέλων ατόμων

3.2.3 Υποσύστημα κατασκευής Σχήματος Ετικετών

Το Υποσύστημα κατασκευής Σχήματος Ετικετών αναλαμβάνει την κωδικοποίηση (μέσω ετικετών) των σχέσεων ιεραρχίας της οντολογίας και την αποθήκευσή τους στη βάση δεδομένων. Οι σχέσεις ιεραρχίας μπορεί να αναφέρονται σε κλάσεις, ρόλους, αλλά και δηλώσεις μεταβατικών ρόλων.

3.2.4 Υποσύστημα διαχείρισης οντολογίας

Το Υποσύστημα διαχείρισης οντολογίας περιλαμβάνει όλες τις λειτουργίες που σχετίζονται με τη διαχείριση των οντολογιών οι οποίες είναι αποθηκευμένες σε βάση δεδομένων.

3.2.5 Υποσύστημα αποτίμησης ερωτημάτων

Το Υποσύστημα αποτίμησης ερωτημάτων αναλαμβάνει την αναγνώριση των ερωτημάτων που υποβάλλονται από τον χρήστη μέσω του GUI, τον έλεγχο των παραμέτρων τους και την επιλογή της στρατηγικής αποτίμησής τους

3.2.6 Υποσύστημα διαχείρισης Βάσης Δεδομένων

Το Υποσύστημα διαχείρισης Βάσης Δεδομένων είναι υπεύθυνο για την επικοινωνία των υποσυστημάτων του DBRS με τη βάση δεδομένων και ως εκ τούτου αναλαμβάνει την εκτέλεση όλων εκείνων των λειτουργιών που σχετίζονται με εισαγωγή, ενημέρωση και εξαγωγή πληροφορίας από αυτήν. Επίσης, διεκπεραιώνει όλες τις αιτήσεις που αφορούν είτε σε δημιουργία νέας βάσης είτε σε διαχείριση υπάρχουσας, αλλά και τις αιτήσεις για εγγραφή νέου χρήστη στο DBMS.

3.2.7 Υποσύστημα διαχείρισης Μηχανής Συλλογιστικής Ανάλυσης

Το Υποσύστημα διαχείρισης Μηχανής Συλλογιστικής Ανάλυσης είναι υπεύθυνο για τη επικοινωνία των υποσυστημάτων του DBRS με τον Reasoner. Στην ουσία, αναλαμβάνει την εκτέλεση όλων εκείνων των λειτουργιών που σχετίζονται με εξαγωγή πληροφορίας από τη βάση γνώσης (στην κύρια μνήμη) του Reasoner.

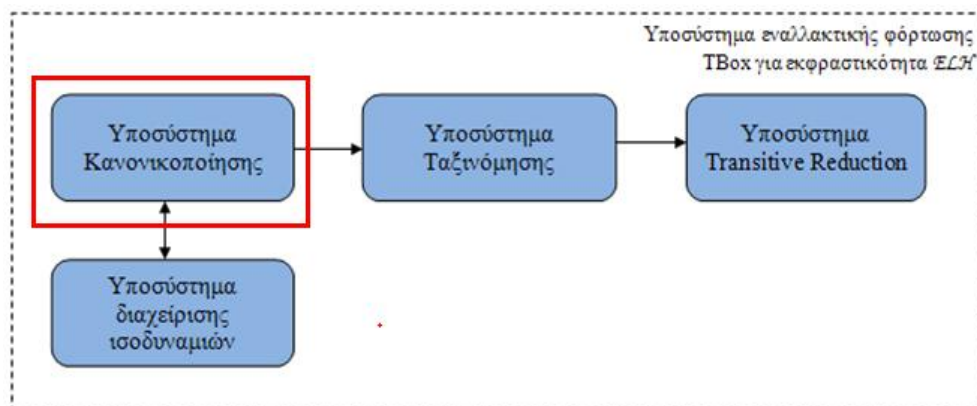
4

Σχεδίαση Συστήματος

Στο κεφάλαιο αυτό περιγράφονται οι εφαρμογές κάθε υποσυστήματος της και, όπου απαιτείται, παρατίθενται διαγράμματα ροής για την καλύτερη κατανόηση των επιμέρους διαδικασιών που εμπλέκονται σε αυτές. Η δομή της παρουσίασης είναι παρόμοια με αυτή του Κεφαλαίου 3, μόνο που εδώ αναφερόμαστε στο επίπεδο των εφαρμογών και θα αναφερθούμε μόνο στα υποσυστήματα που προσθέσαμε στο DBRS. Στο τέλος του Κεφαλαίου περιγράφεται αναλυτικά το σχεσιακό σχήμα της βάσης δεδομένων που υλοποιήθηκε στο DBRS.

4.1 Υποσύστημα κανονικοποίησης

Στο υποσύστημα κανονικοποίησης ανήκουν οι εφαρμογές που αναλαμβάνουν την εφαρμογή των κανόνων κανονικοποίησης στα αξιώματα της οντολογίας και την αποθήκευση της κανονικοποιημένης οντολογίας στο σχεσιακό DBMS. Αρχικά μέσω της εφαρμογής διαχείρισης αξιωμάτων επιτυγχάνεται η μετατροπή των αξιωμάτων της οντολογίας στα κανονικοποιημένα αξιώματα ($A \sqsubseteq B$, $A_1 \sqcap A_2 \sqsubseteq B$, $A \sqsubseteq \exists r.B$, $\exists r.A \sqsubseteq B$, $r \sqsubseteq s$) και η αποθήκευσή τους στις αντίστοιχες προσωρινές δομές σε αρχεία. Στη συνέχεια γίνεται η μεταφορά της οντολογίας από τις προσωρινές δομές στο σχεσιακό DBMS μέσω κλήσης της αντίστοιχης εφαρμογής.



Σχήμα 4.1 Η θέση του υποσυστήματος κανονικοποίησης στο ολικό σύστημα

4.1.1 Εφαρμογή διαχείρισης αξιωμάτων

Η εφαρμογή αυτή αποτελεί τον πυρήνα της εφαρμογής των κανόνων κανονικοποίησης. Σκοπός της διαδικασίας είναι η μετατροπή των αξιωμάτων της οντολογίας στα κανονικοποιημένα αξιώματα και η αποθήκευσή αυτών στις αντίστοιχες προσωρινές δομές. Τα βήματα της διαδικασίας παρατίθενται παρακάτω:

- i) Από το μοντέλο της οντολογίας που έχει κατασκευαστεί προηγουμένως εξαιρούμε τα αξιώματα δήλωσης (declaration). Για τα υπόλοιπα αξιώματα ξεκινάμε την επεξεργασία τους σειριακά.
- ii) Διακρίνεται ο τύπος του κάθε αξιώματος μεταξύ των επιτρεπόμενων:
 - a. αξιώματα υπαγωγής εννοιών (subclass_of)

- b. αξιώματα υπαγωγής ρόλων (sub_object_property)
 - c. αξιώματα ισοδυναμίας εννοιών (equivalent_classes).
- iii) Γίνεται κλήση της εφαρμογής απόδοσης μοναδικού αναγνωριστικού για όλες τις OWL εκφράσεις που συμμετέχουν σε κάθε αξίωμα, δηλαδή τα subClass, superClass στα αξιώματα υπαγωγής εννοιών, τα subProperty, superProperty στα αξιώματα υπαγωγής ρόλων και όλες τις ισοδύναμες OWLclassExpression στα αξιώματα ισοδυναμίας εννοιών με σκοπό την απόδοση των αντίστοιχων id.
 - iv) Το κάθε ζευγάρι αναγνωριστικών (id1-id2) αποθηκεύεται στις αντίστοιχες προσωρινές δομές κανονικοποιημένων αξιωμάτων ($A \sqsubseteq B$ ή $r \sqsubseteq s$) μέσω κλήσης της εφαρμογής προσωρινής αποθήκευσης κανονικοποιημένων σχέσεων.
 - v) Στην περίπτωση αξιωμάτων ισοδυναμίας τα αναγνωριστικά αποθηκεύονται ανά ζεύγη σε μια προσωρινή δομή αναπαράστασης ισοδυναμιών.
 - vi) Για κάθε OWL έκφραση εννοιών (OWLClassExpression) που έχουμε συναντήσει στα αξιώματα υπαγωγής εννοιών ή στα αξιώματα ισοδυναμίας εννοιών καλούμε την εφαρμογή χειρισμού OWL εκφράσεων εννοιών (OWLClassExpression).

4.1.2 Εφαρμογή προσωρινής αποθήκευσης κανονικοποιημένων αξιωμάτων

Η εφαρμογή προσωρινής αποθήκευσης κανονικοποιημένων αξιωμάτων σχετίζεται με την χρήση προσωρινών αρχείων για την αποθήκευση των κανονικοποιημένων αξιωμάτων που προκύπτουν σταδιακά από τη διαδικασία της κανονικοποίησης. Για κάθε έναν τύπο κανονικοποιημένου αξιωματος υπάρχει και ένα αντίστοιχο προσωρινό αρχείο. Επιπλέον υπάρχει ένα ακόμα αρχείο στο οποίο αποθηκεύονται τα ζευγάρια που προκύπτουν από τα αξιώματα ισοδυναμίας και ένα αρχείο που αντιστοιχεί όλες τις OWL εκφράσεις της οντολογίας με το μοναδικό id που τους έχει αντιστοιχηθεί. Βασική λειτουργικότητα αυτή της εφαρμογής είναι η απουσία διπλοεγγραφών μέσα στα προσωρινά αρχεία.

4.1.3 Εφαρμογή απόδοσης μοναδικού αναγνωριστικού

Η εφαρμογή απόδοσης μοναδικού αναγνωριστικού σχετίζεται με την διαδικασία απόδοσης ενός μοναδικού id σε κάθε OWL έκφραση που περιέχεται στην οντολογία.

Βασική επιδίωξη μας είναι να μην χρειάζεται να ανατρέχουμε σε όλα τα id που έχουμε ήδη αποδόσει πριν από την απόδοση id σε μία νέα έκφραση και να είμαστε σίγουροι ότι κάθε φορά που η διαδικασία κανονικοποίησης θα συναντά την ίδια owl έκφραση θα αποδίδει το ίδιο id. Για να γίνει αυτό εφικτό χρειαζόμαστε μία δομή στη μνήμη με τα χαρακτηριστικά ενός hash map που θα αντιστοιχεί το URI κάθε OWL έκφρασης με το αντίστοιχο id. Τα βήματα αυτής της διαδικασίας περιγράφονται παρακάτω και φαίνονται στο διάγραμμα ροής του σχήματος :

- i) Απόδοση προσωρινού id με τη χρήση μιας συνάρτησης κατακερματισμού (hashcode).
- ii) Έλεγχος αν υπάρχει ήδη αποθηκευμένο μέσα στη δομή hash map αυτό το id. Αν αυτό δεν υπάρχει τότε το προσωρινό id γίνεται οριστικό και αποθηκεύεται το αντίστοιχο ζευγάρι URI – id στην δομή hash map. Αν υπάρχει το id ήδη αποθηκευμένο προχωράμε στο επόμενο βήμα.
- iii) Ελέγχουμε αν το URI που βρίσκεται αποθηκευμένο στην δομή hash map στο αντίστοιχο id είναι το ίδιο με αυτό που έχει η υπό μελέτη OWL έκφραση. Αν αυτό συμβαίνει σταματάμε την διαδικασία σε αυτό το σημείο και χρησιμοποιούμε αυτό το id για την υπόλοιπη διαδικασία της κανονικοποίησης. Αν όχι συνεχίζουμε στο επόμενο βήμα.
- iv) Αυξάνουμε το προσωρινό id κατά 1 και επιστρέφουμε πάλι στο βήμα ii.

4.1.4 Εφαρμογή χειρισμού OWL εκφράσεων εννοιών (OWLClassExpression)

Η εφαρμογή χειρισμού εκφράσεων OWL εννοιών (OWLClassExpression) σχετίζεται με τον τρόπο που γίνεται το unfolding των σύνθετων εννοιών έως ότου να έχουν δημιουργηθεί όλα τα κανονικοποιημένα αξιώματα και περιέχει 2 εφαρμογές για τον χειρισμό των κατασκευαστών που υπάρχουν στην εκφραστικότητα ELH. Αν η υπο εξέταση έκφραση OWL κλάσης είναι ήδη μια ατομική ονοματισμένη κλάση (named classes) τότε η διαδικασία δεν συνεχίζει αν όμως η έκφραση OWL κλάσης ταυτοποιηθεί ότι είναι μία έκφραση τομής ή μια έκφραση υπαρξιακού ποσοδείκτη τότε γίνεται κλήση των αντίστοιχων εφαρμογών που περιγράφονται στη συνέχεια.

4.1.4.1 Εφαρμογή χειρισμού του κατασκευαστή της τομής

Η εφαρμογή χειρισμού του κατασκευαστή της τομής σχετίζεται με τον χειρισμό των εκφράσεων OWL κλάσεων όταν έχει ταυτοποιηθεί ότι είναι μία έκφρασης τομής. Τα βήματα της εφαρμογής περιγράφονται παρακάτω:

- i) Αρχικά μέσω της εφαρμογής απόδοσης μοναδικού χαρακτηριστικού αποδίδεται ένα id στο σύνολο της OWL έκφρασης.
- ii) Απομονώνονται όλες οι N OWL εκφράσεις που συμμετέχουν στην αρχική τομή.
- iii) Δημιουργούνται 2 νέες εκφράσεις OWL κλάσεων. Η πρώτη περιέχει την πρώτη έκφραση OWL κλάσης της αρχικής τομής και η δεύτερη τις υπόλοιπες N-1 εκφράσεις.
- iv) Με κλήση της εφαρμογής απόδοσης μοναδικού χαρακτηριστικού αποδίδονται id στις νέες εκφράσεις (id1 , id2)
- v) Μέσω της εφαρμογής προσωρινής αποθήκευσης κανονικοποιημένων αξιωμάτων αποθηκεύονται τα ζεύγη αναγνωριστικών (id-id1 και id-id2) που αντιστοιχούν στις 2 νέες κανονικοποιημένες σχέσεις υπαγωγής ($A \sqsubseteq B$) που υπονοούνται από τον κατασκευαστή της τομής και η τριπλέτα αναγνωριστικών (id1-id2-id) που αντιστοιχεί στην κανονικοποιημένη αναπαράσταση της τομής ($A_1 \sqcap A_2 \sqsubseteq B$).
- vi) Για κάθε μία από τις δύο νέες εκφράσεις OWL κλάσεων καλείται πάλι η εφαρμογή χειρισμού εκφράσεων OWL κλάσεων για την συνέχιση της διαδικασίας “ξεδιπλώματος”

4.1.4.2 Εφαρμογή χειρισμού του υπαρξιακού κατασκευαστή

Η εφαρμογή χειρισμού του υπαρξιακού κατασκευαστή σχετίζεται με τον χειρισμό των εκφράσεων OWL κλάσεων όταν έχει ταυτοποιηθεί ότι είναι μία έκφρασης πλήρους υπαρξιακής ποσοτικοποίησης. Τα βήματα της εφαρμογής περιγράφονται παρακάτω:

- i) Αρχικά μέσω της εφαρμογής απόδοσης μοναδικού χαρακτηριστικού αποδίδεται ένα id στο σύνολο της OWL έκφρασης.

- ii) Απομονώνεται η έκφραση OWL ιδιότητας αντικειμένων και η έκφραση OWL κλάσης που συμμετέχουν στην αρχική έκφραση.
- iii) Με κλήση της εφαρμογής απόδοσης μοναδικού χαρακτηριστικού αποδίδονται id στις νέες εκφράσεις (r_id , c_id)
- iv) Μέσω της εφαρμογής προσωρινής αποθήκευσης κανονικοποιημένων αξιωμάτων αποθηκεύεται η τριπλέτα αναγνωριστικών (r_id-c_id-id) που αντιστοιχεί στην κανονικοποιημένη αναπαράσταση του υπαρξιακού κατασκευαστή ($\exists r.A \sqsubseteq B$). Αν ο υπαρξιακός κατασκευαστής συμμετέχει αρνητικά³⁷ στο αξίωμα τότε εισάγεται και η αντίστοιχη τριπλέτα αναγνωριστικών (id-r_id-c_id) στην κανονικοποιημένη αναπαράσταση του αρνητικού υπαρξιακού κατασκευαστή ($A \sqsubseteq \exists r.B$).
- v) Για την νέα έκφραση OWL κλάσης καλείται πάλι η εφαρμογή χειρισμού εκφράσεων OWL κλάσεων για την συνέχιση της διαδικασίας “ξεδιπλώματος”

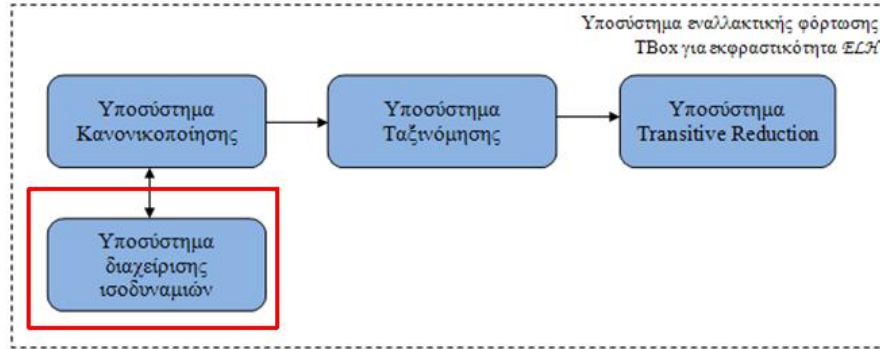
4.1.5 Εφαρμογή αποθήκευσης οντολογίας στο σχεσιακό DBMS

Η εφαρμογή αποθήκευσης της οντολογίας στο σχεσιακό DBMS σχετίζεται με τις διαδικασίες για την μεταφορά των προσωρινών δομών με τα κανονικοποιημένα αξιώματα στο αντίστοιχο σχήμα βάσης που θα χρησιμοποιηθεί και στα επόμενα στάδια.

4.2 Υποσύστημα διαχείρισης ισοδυναμιών

Το υποσύστημα διαχείρισης ισοδυναμιών επιτελεί όλες εκείνες τις λειτουργίες που είναι απαραίτητες για την διαχείριση των ισοδυναμιών. Για την ορθότητα του αλγορίθμου είναι αναγκαίο ισοδύναμες εκφράσεις OWL κλάσεων να σχετίζονται με το ίδιο αναγνωριστικό id σε παρέκκληση όσων ισχύουν για την απόδοση αναγνωριστικώνγενικά. Αυτή η ανάγκη επιτελείται μέσω των δύο εφαρμογών που αποτελούν αυτό το υποσύστημα και αναλύονται στη συνέχεια.

³⁷ Στην συγκεκριμένη εκφραστικότητα αυτό συμβαίνει μόνο όταν συμμετέχει σε σχέσεις ισοδυναμίας



Σχήμα 4.2 Η θέση του υποσυστήματος διαχείρισης ισοδυναμιών στο ολικό σύστημα

4.2.1 Εφαρμογή ομαδοποίησης ισοδυναμιών

Η εφαρμογή ομαδοποίησης ισοδυναμιών σχετίζεται με την χρήση ενός αλγορίθμου ο οποίος ομαδοποιεί όλες τις ισοδύναμες εκφράσεις OWL κλάσεων που μπορεί να εμφανίστηκαν από διαφορετικά αξιώματα ισοδυναμιών στο στάδιο της κανονικοποίησης (π.χ. $A \equiv B$ και $B \equiv C$). Σε κάθε ομάδα ισοδύναμων εκφράσεων αποφασίζεται πιο αναγνωριστικό θα είναι το χαρακτηριστικό αυτής της ομάδας και δημιουργεί ένα mapping.

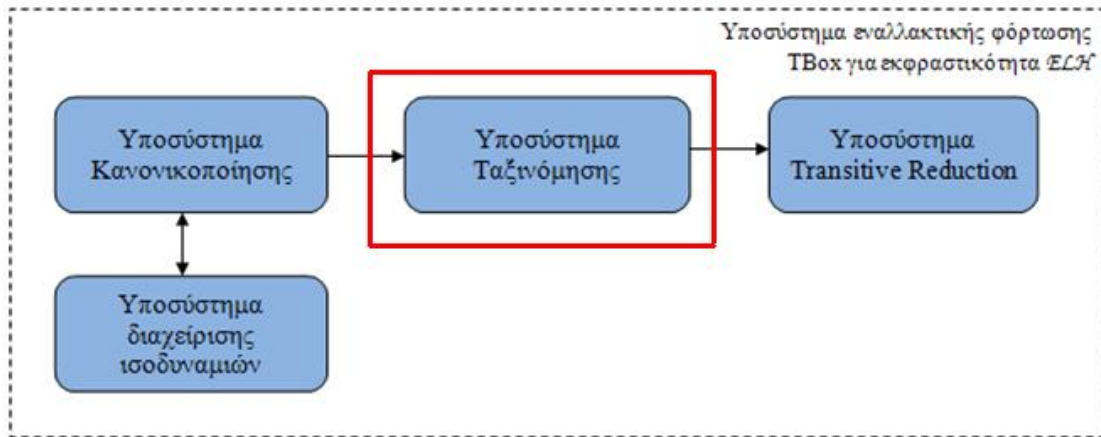
4.2.2 Εφαρμογή ενημέρωσης σχήματος βάσης

Η εφαρμογή ενημέρωσης σχήματος βάσης σχετίζεται με την ενημέρωση του σχήματος βάσης της κανονικοποιημένης οντολογίας σχετικά με την πληροφορία που εξάγεται από τα αξιώματα ισοδυναμίας. Αυτό επιτυγχάνεται μέσω της εφαρμογής μιας σειράς από sql statements στο σχήμα της βάσης.

4.3 Υποσύστημα ταξινόμησης

Στο υποσύστημα ταξινόμησης ανήκουν οι εφαρμογές που αναλαμβάνουν την εξαντλητική εφαρμογή των επαγωγικών κανόνων (fix-point) στο σχήμα βάσης που έχει αποθηκευθεί η ταξινομημένη οντολογία. Ουσιαστικά, το υποσύστημα ταξινόμησης υλοποιεί έναν αλγόριθμο ταξινόμησης που έχει σκοπό την εξασφάλιση της παραγωγής όλων των σχέσεων ιεραρχίας μεταξύ των ατομικών ονοματισμένων κλάσεων (named

classes). Για να είναι αυτό εφικτό απαιτείται ο υπολογισμός όλων των “ενδιάμεσων” σχέσεων ιεραρχίας μεταξύ των εννοιών που προκύπτουν κυρίως από τους κατασκευαστές της εκφραστικότητας. Οι εφαρμογές που αποτελούν το υποσύστημα ταξινόμησης παρουσιάζονται αναλυτικά παρακάτω.

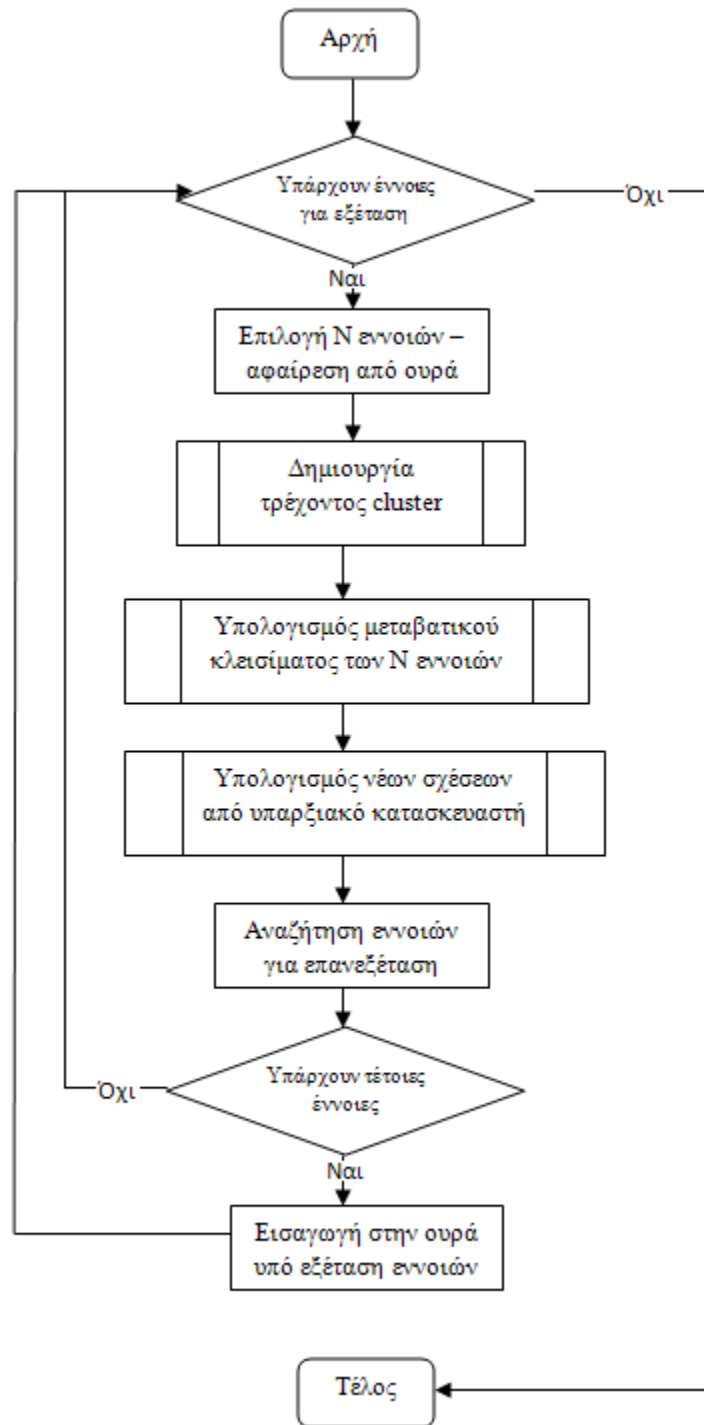


Σχήμα 4.3 Η θέση του υποσυστήματος ταξινόμησης στο ολικό σύστημα

Σε αυτό το σημείο ας αναφέρουμε για μια ακόμα φορά τους επαγωγικούς κανόνες για την εκφραστικότητα ELH , οι οποίοι εφαρμόζονται μετά την μετάφρασή τους σε ένα αντίστοιχο σύνολο sql statement.

7. Αν $A \in T$ τότε $T = T \cup \{A \sqsubseteq A, A \sqsubseteq T\}$
8. Αν $A \sqsubseteq B \in T$ και $B \sqsubseteq C \in T$, τότε $T = T \cup \{A \sqsubseteq C\}$
9. Αν $A \sqsubseteq B \in T$ και $A \sqsubseteq C \in T$ και $B \sqcap C \sqsubseteq D \in T$, τότε $T = T \cup \{A \sqsubseteq D\}$
10. Αν $A \sqsubseteq B \in T$ και $B \sqsubseteq \exists r.C \in T$, τότε $T = T \cup \{A \sqsubseteq \exists r.C\}$
11. Αν $A \sqsubseteq \exists r.B \in T$ και $r \sqsubseteq s \in T$, τότε $T = T \cup \{A \sqsubseteq \exists s.B\}$
12. Αν $A \sqsubseteq \exists r.B \in T$ και $B \sqsubseteq C \in T$ και $\exists r.C \sqsubseteq D \in T$, τότε $T = T \cup \{A \sqsubseteq D\}$

Επιπλέον, πριν από την αναλυτική παρουσίαση των εφαρμογών του υποσυστήματος της κανονικοποίησης παρατίθεται διάγραμμα ροής για όλη τη διαδικασία της κανονικοποίησης στο παρακάτω σχήμα.



Σχήμα 4.4 Διάγραμμα ροής υποσυστήματος ταξινόμησης

4.3.1 Εφαρμογή υπολογισμού μεταβατικού κλεισίματος ιδιοτήτων

Η εφαρμογή υπολογισμού μεταβατικού κλεισίματος ιδιοτήτων (transitive closure) ιδιοτήτων σχετίζεται με τις λειτουργίες που επιτελούνται για τον υπολογισμό νέων σχέσεων υπαγωγής ιδιοτήτων μεταξύ εκφράσεων OWL ιδιοτήτων αντικειμένων από τις κανονικοποιημένες σχέσεις υπαγωγής ιδιοτήτων που έχουν αποθηκευτεί στο σχήμα της βάσης μετά την διαδικασία της κανονικοποίησης. Αυτή η λειτουργία επιτυγχάνεται με την επαναληπτική εφαρμογή ενός sql statement στον αντίστοιχο πίνακα του σχήματος βάσης μέχρι αν μην βγάζει νέες σχέσεις και την αποθήκευση του αποτελέσματος σε έναν νέο πίνακα του σχήματος βάσης.

4.3.2 Εφαρμογή απαλοιφής επαγωγικού κανόνα 5

Η εφαρμογή απαλοιφής του επαγωγικού κανόνα 5 ($\forall x.A \sqsubseteq \exists r.B \in T$ και $r \sqsubseteq s \in T$, τότε $T = T \cup \{A \sqsubseteq \exists s.B\}$) σχετίζεται με τις λειτουργίες που επιτελούνται ώστε να είναι εφικτό ο επαγωγικός κανόνας 5 να εξαιρεθεί από την επαναληπτική εφαρμογή. Δεδομένου ότι έχει προηγηθεί ο υπολογισμός του μεταβατικού κλεισίματος των ιδιοτήτων μπορούμε να εφαρμόσουμε τον επαγωγικό κανόνα 5 μία φορά και να είμαστε σίγουροι ότι δεν χρειάζεται να επαναληφθεί στο μέλλον καθώς θα έχουν εξαχθεί όλες οι σχέσεις ιεραρχίας που θα μπορούσαν να δημιουργηθούν από την εφαρμογή αυτού του κανόνα.

4.3.3 Εφαρμογή δημιουργίας τρέχοντος cluster

Η εφαρμογή δημιουργίας ενός τρέχοντος cluster σχετίζεται με τις απαραίτητες λειτουργίες για την προετοιμασία του σχήματος βάσης πριν από κάθε επανάληψη του αλγορίθμου για την εφαρμογή των επαγωγικών κανόνων σε ένα μόνο τμήμα της οντολογίας. Βασική αρχή του υποσυστήματος της ταξινόμησης είναι ότι οι επαγωγικοί κανόνες δεν εφαρμόζονται στο σύνολο της οντολογίας αλλά σε ένα τμήμα αυτής, το μέγεθος του οποίου καθορίζεται από μία παράμετρο του χρήστη. Ουσιαστικά η παράμετρος αυτή καθορίζει πόσες από τις υπο έξεταση κλάσεις θα εξεταστούν σε κάθε επανάληψη του αλγορίθμου. Επομένως, θα πρέπει να δημιουργηθεί ένα cluster του ολικού σχήματος βάσης που να περιέχει μόνο όσα δεδομένα (σχέσεις υπαγωγής και

κανονικοποιημένες μορφές αναπαράστασης των κατασκευαστών) έχουν κάποια σχέση με αυτές τις επιλεγμένες υπό εξέταση κλάσεις. Αυτή ακριβώς τη λειτουργία επιτελεί η εφαρμογή δημιουργίας τρέχοντος cluster. Επιπλέον σε αυτή την εφαρμογή υλοποιείται και ο επαγωγικός κανόνας 1 με την αρχικοποίηση των αντίστοιχων πινάκων.

$$1. \text{ Αν } A \in T \text{ τότε } T = T \cup \{A \sqsubseteq A, A \sqsubseteq T\}$$

4.3.4 Εφαρμογή υπολογισμού τρέχοντος μεταβατικού κλεισίματος κλάσεων

Η εφαρμογή υπολογισμού τρέχοντος μεταβατικού κλεισίματος κλάσεων σχετίζεται με τις απαραίτητες λειτουργίες για τον υπολογισμό του μεταβατικού κλεισίματος (transitive closure) για τις υπό εξέταση κλάσεις. Ουσιαστικά πρόκειται για την εφαρμογή που υλοποιεί τους επαγωγικούς κανόνες 2, 3 και 4 με την εφαρμογή επαναληπτικών sql statements για την δημιουργία του transitive closure.

$$2. \text{ Αν } A \sqsubseteq B \in T \text{ και } B \sqsubseteq C \in T, \text{ τότε } T = T \cup \{A \sqsubseteq C\}$$

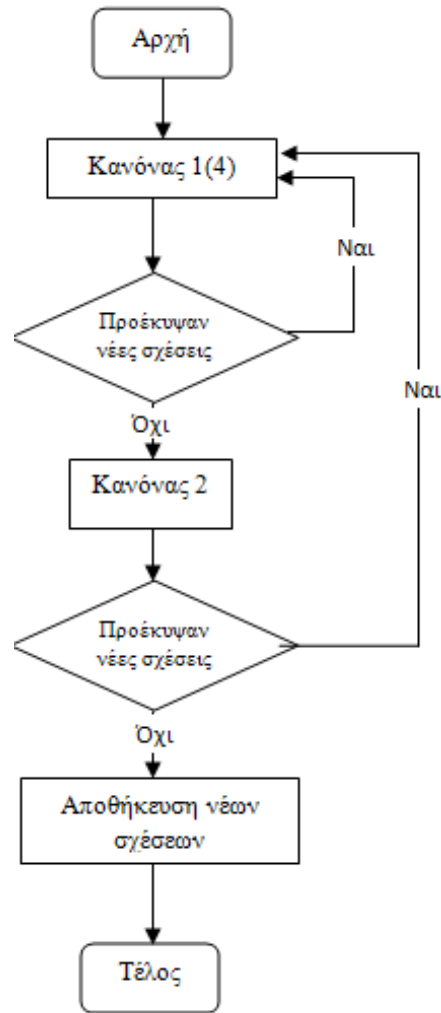
$$3. \text{ Αν } A \sqsubseteq B \in T \text{ και } A \sqsubseteq C \in T \text{ και } B \sqcap C \sqsubseteq D \in T, \text{ τότε } T = T \cup \{A \sqsubseteq D\}$$

$$4. \text{ Αν } A \sqsubseteq B \in T \text{ και } B \sqsubseteq \exists r.C \in T, \text{ τότε } T = T \cup \{A \sqsubseteq \exists r.C\}$$

Με τον τρόπο που έχει δημιουργηθεί η κανονικοποιημένη οντολογία ο κανόνας 4 και ο κανόνας 2 εφαρμόζονται ταυτόχρονα από το ίδιο σύνολο sql statements καθώς στο πίνακα με τις σχέσεις υπαγωγής δεν γίνεται διάκριση των id ανάλογα με το τι αντιπροσωπεύουν. Επίσης πριν από το τέλος αυτής της εφαρμογής οι νέες υπονοούμενες σχέσεις υπαγωγής εισάγονται στον πίνακα των τελικών αποτελεσμάτων. Το διάγραμμα ροής των βημάτων αυτής της διαδικασίας φαίνεται στη συνέχεια ενώ τα βήματα είναι:

- i) Εξαντλητική επαναληπτική εφαρμογή του κανόνα 1 με τρέχων σύνολο των πίνακα που αποθηκεύονται οι σχέσεις ιεραρχίας για τις υπό εξέταση έννοιες και σύνολο βάσης τον πίνακα μερικών αποτελεσμάτων.
- ii) Εφαρμογή του κανόνα 2 από τον οποίο προκύπτουν σχέσεις ιεραρχίας εξαιτίας του κατασκευαστή της τομής

- iii) Αν μετά την εφαρμογή του κανόνα 2 έχουμε νέα αποτελέσματα τότε επανλαμβάνουμε τη διαδικασία από την αρχή αν όχι συνεχίζουμε στο βήμα iv).
- iv) Αποθήκευση των νέων σχέσεων ιεραρχίας που προέκυψαν στον πίνακα των τελικών αποτελεσμάτων.



Σχήμα 4.5 Διάγραμμα ροής εφαρμογής υπολογισμού τρέχοντος μεταβατικού κλεισίματος

4.3.5 Εφαρμογή υπολογισμού σχέσεων ιεραρχίας λόγω του υπαρξιακού κατασκευαστή

Η εφαρμογή αυτή σχετίζεται με τις λειτουργίες που είναι απαραίτητες για τον υπολογισμό νέων σχέσεων ιεραρχίας που προκύπτουν από την σημασιολογία του πλήρους υπαρξιακού ποσοδείκτη. Επομένως, αναφερόμαστε στην περίπτωση εφαρμογή του επαγωγικού κανόνα 6.

$$6. \text{ Αν } A \sqsubseteq \exists r. B \in T \text{ και } B \sqsubseteq C \in T \text{ και } \exists r. C \sqsubseteq D \in T, \text{ τότε } T = T \cup \{A \sqsubseteq D\}$$

Και σε αυτή την περίπτωση ο επαγωγικός κανόνας, δηλαδή τα αντίστοιχα sql statements, δεν εκτελείται σε όλη την οντολογία αλλά σε ένα τμήμα αυτής. Συγκεκριμένα εφαρμόζουμε τον κανόνα 6 μόνο για το σύνολο σχέσεων ιεραρχίας που προέκυψαν από την εφαρμογή υπολογισμού τρέχοντος μεταβατικού κλεισίματος. Τα αποτελέσματα αυτής της εφαρμογής αποθηκεύονται ταυτόχρονα σε έναν προσωρινό και προστίθονται στον πίνακα των μερικών αποτελεσμάτων.

4.3.6 Εφαρμογή εύρεσης εννοιών που πρέπει να επανεξεταστούν

Η εφαρμογή εύρεσης εννοιών που πρέπει να επανεξεταστούν σχετίζεται με τις λειτουργίες που είναι απαραίτητες για τον εντοπισμό εννοιών που έχουν εξεταστεί σε προηγούμενες επαναλήψεις του αλγορίθμου και πρέπει να εξεταστούν ξανά για να είναι πλήρης ο αλγόριθμος ταξινόμησης. Για τον εντοπισμό αυτών των εννοιών επεξεργαζόμαστε τις σχέσεις ιεραρχίας που προέκυψαν από τον επαγωγικό κανόνα 6 σε συνδυασμό με τα τελικά αποτελέσματα μέχρι αυτή τη στιγμή. Για όσες έννοιες υπάρχουν ενδείξεις ότι μπορούν να εμπλακούν κατά τη διάρκεια υπολογισμού του μεταβατικού κλεισίματος με κάποια από τις νέες υπονοούμενες σχέσεις του επαγωγικού κανόνα 6, τις θέτουμε υπό εξέταση. Οι έννοιες που θα βρεθούν από αυτή τη διαδικασία τοποθετούνται στο τέλος της ουράς των προς εξέταση στοιχείων.

5

Βιβλιογραφία

- [OWL EL] OWL 2 EL profile, http://www.w3.org/TR/owl2-profiles/#OWL_2_EL
- [OWL 2] OWL 2 Specification <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/>
- [SNOMED] Snomed CT <http://www.ihtsdo.org/index.php?id=545>
- [GALEN] Galen http://www.openclinical.org/prj_galen.html
- [GO] Go <http://www.geneontology.org/>
- [NCI] NCI Thesaurus http://evs.nci.nih.gov/ftp1/NCI_Thesaurus
- [HER] Hermit Reasoner <http://hermit-reasoner.com/>
- [PEL] Pellet Reasoner <http://clarkparsia.com/pellet>
- [FACT] Fact Reasoner <http://owl.man.ac.uk/factplusplus/>
- [CEL] CEL Reasoner <http://code.google.com/p/cel/>
- [DBRS] DBRS <http://www.dbnet.ece.ntua.gr/pubs/details.php?id=1523&clang=0>
- [OWLAPI] OWLAPI v.3 <http://owlapi.sourceforge.net/index.html>
- [BBL05] F.Baader, S. Brandt and C.Lutz, *Pushing the EL Envelope*, IJCAI, 2005
- [BLS05] F.Baader, C.Lutz and B.Suntisrivaraporn, *Is Tractable Reasoning in Extensions of the Description Logic EL Useful in Practice?*, TU Dresden, 2005
- [DY09] V.Delaitre and Y.Kazakov, *Classifying ELH Ontologies in SQL Databases*, OWLED, 2009
- [K09] Y.Kazakov, *Consequence-Driven Reasoning for Horn SHIQ Ontologies*, IJCAI, 2009
- [CGL09] Andrea Cali, Georg Gottlob, Thomas Lukasiewicz, *PA General Datalog-Based Framework for Tractable-Query Answering over Ontologies*, Proceedings of PODS, 2009
- [K11] Markus Krötzsch, *Efficient Rule-Based Inferencing for OWL EL*, In Proceedings of the 22nd IJCAI, 2011

[KMR10] Markus Krötzsch, Anees ul Mehdi, Sebastian Rudolph, *Orel: Database-Driven Reasoning for OWL 2 Profiles*, 2010

[HMW08] V. Haarslev, R. Møller, S. Wandelt, *The revival of structural subsumption in tableau-based description logic reasoners*, International Workshop on DLs, 2008