Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Συνεπής προσαρμογή οντολογικών γνώσεων σε δεδομένα

Διπλωματικη Εργασια

της

ΛΗΔΑΣ ΠΕΤΡΟΠΟΥΛΟΥ

Επιβλέπων: Γιώργος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Εργαστηριο Ετφυων Συστηματων, Περιεχομενου και Αλληλεπιδρασης

Αθήνα, Ιούλιος 2018

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Ευφυών Συστημάτων, Περιεχομένου και Αλληλεπίδρασης

Συνεπής προσαρμογή οντολογικών γνώσεων σε δεδομένα

Διπλωματικη Εργασια

της

ΛΗΔΑΣ ΠΕΤΡΟΠΟΥΛΟΥ

Επιβλέπων: Γιώργος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 3η Ιουλίου 2018.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Γιώργος Στάμου	Ανδρέας-Γεώργιος Σταφυλοπάτης	Παναγιώτης Τσανάκας
Αν. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2018

(Υπογραφή)

.....

Ληδα Πετροπουλου

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Ευφυών Συστημάτων, Περιεχομένου και Αλληλεπίδρασης

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Γιώργο Στάμου που είχε την επίβλεψη της παρούσας διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.

Επίσης ευχαριστώ ιδιαίτερα τον υποψήφιο διδάκτορα κ. Enrique Matos Alfonso για την βοήθεια και την καθοδήγηση του κατά την εκπόνηση της εργασίας.

Περίληψη

Με την ανάπτυξη του Σημασιολογικού Ιστού, υπάρχει μία αυξανόμενη ανάγκη για οντολογίες σε πολλούς τομείς. Ο εμπλουτισμός οντολογιών χρησιμοποιεί διάφορες τεχνικές που επιτρέπουν την ανακάλυψη νέων λογικών αξιωμάτων. Η εισαγωγή των νέων αυτών αξιωμάτων στην οντολογία μπορεί να προκαλέσει ανεπιθύμητες ασυνέπειες.

Στη παρούσα διπλωματική εργασία, προτείνεται μία μέθοδος που χρησιμοποιεί εξαγωγή κανόνων συσχέτισης (association rule mining) και απάντηση ερωτημάτων (query answering) για να ανακαλύψει νέα λογικά αξιώματα που μπορούν να προστεθούν ασφαλώς σε μία υπάρχουσα οντολογία.

Οι διαδικασίες της εξαγωγής κανόνων συσχέτισης διευκολύνονται από τη χρήση του συστήματος Weka, που παρέχει μια ποικιλία από υλοποιήσεις αλγορίθμων μηχανικής μάθησης. Η απάντηση ερωτημάτων πραγματοποιείται μέσω του συστήματος Completo.

Για την επίτευξη του ασφαλούς εμπλουτισμού της οντολογίας, υλοποιήθηκε ένα πρόγραμμα σε γλώσσα Java, και εκτελέστηκαν αρκετά πειράματα, προκειμένου να μετρηθεί η αποτελεσματικότητα της προτεινόμενης μεθόδου.

Λέξεις Κλειδιά

Οντολογία, Κανόνες Συσχέτισης, Απάντηση ερωτημάτων, ασυνέπειες.

Abstract

As Semantic Web develops, there is a growing need for ontologies in various fields. Ontology enrichment applies different techniques that allow the discovery of new logical axioms. Adding new axioms to the ontology can easily create undesired inconsistencies.

In the presented diploma thesis, we propose a method that uses association rule mining and query answering to discover new logical axioms that can be safely added to the ontology.

The procedures of association rule learning are facilitated by the Weka system that implements a variety of Machine Learning algorithms. Query Answering is facilitated by the Completo system.

A java program was written to carry out the safe enrichment of the Ontology and various experiments were conducted, in order to measure the feasibility of the proposed method.

Keywords

Ontology, Existential Rules, Association Rule Mining, inconsistencies.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Η ανάπτυξη του Σημασιολογικού Ιστού (Semantic Web) [12] συνοδεύεται από την αυξανόμενη ανάγκη για οντολογίες σε διάφορους τομείς εφαρμογών. Η εξέλιξη της τεχνολογίας στην αποθήκευση δεδομένων παρέχει μεγαλύτερο χώρο αποθήκευσης και ταχύτερη πρόσβαση, με αποτέλεσμα την αύξηση του όγκου των διαθέσιμων δεδομένων που διανέμονται μέσω του διαδικτύου. Ωστόσο, τα περισσότερα από τα υπάρχοντα συστήματα παρουσιάζουν έλλειψη εκφραστικών οντολογιών για την εξαγωγή λογικών συμπερασμάτων και την ανακάλυψη μη τετριμμένων συμπερασμάτων.

Ο εμπλουτισμός μίας οντολογίας ορίζεται ως η επέκταση ενός υπάρχοντος οντολογικού σχήματος με αξιώματα που δεν μπορούν να συναχθούν λογικά με το υπάρχον σύνολο αξιωμάτων. Ο εμπλουτισμός της οντολογίας μπορεί να οριστεί ως μια υποενότητα της Οντολογικής Μάθησης που βασίζεται σε εξωτερικά δεδομένα όπως είναι το κείμενο και οι εικόνες, για τη δημιουργία μιας οντολογίας. Ο εμπλουτισμός μίας οντολογίας χρησιμοποιεί μια αρχική οντολογία και αναλύει τα δεδομένα αναζητώντας έννοιες, ρόλους ή λογικά αξιώματα που μπορούν να προστεθούν.

Η εύρεση νέων ορισμών των εννοιών ή των ρόλων είναι ένα πολύ δύσκολο έργο, που σχετίζεται περισσότερο με τον Επαγωγικό Λογικό Προγραμματισμό (Inductive Logic Programming). Η συμπλήρωση της οντολογίας επικεντρώνεται στην εξεύρεση νέων αξιωμάτων ή ισχυρισμών και μπορεί να περιλαμβάνει ανάλυση εννοιών, στατιστική ανάλυση και τεχνικές μηχανικής μάθησης ανάλογα με τον τύπο των αξιωμάτων που πρέπει να ανακαλυφθούν.

Σχετικές εργασίες στην συμπλήρωση οντολογίας επικεντρώνονται κυρίως στην ανακάλυψη νέων λογικών αξιωμάτων και όχι σε ισχυρισμούς. Στο [20], οι συγγραφείς εκτελούν εκμάθηση της έννοιας disjointness. Η Στατιστική Εισαγωγή Σχήματος (Statistical Schema Induction (SSI)) [19] χρησιμοποιεί εξαγωγή κανόνων συσχέτισης για να βρει αξιώματα υποκλάσεων. Μια πιο καθολική μέθοδος προτείνεται στο [7], όπου τα περισσότερα από τα αξιώματα OWL, παράγονται λαμβάνοντας υπόψη τις μεγάλες βάσεις γνώσεων [7].

Η νέα γνώση μπορεί εύκολα να καταστήσει μία οντολογία ασυνεπή, ειδικά λόγω του θορύβου που προκαλεί η μηχανική μάθηση και οι στατιστικές προσεγγίσεις. Οι υπάρχουσες μέθοδοι δεν εκτελούν έλεγχο συνέπειας όταν προσπαθούν να εμπλουτίσουν την οντολογία. Η προσέγγιση που προτείνεται σε αυτή την εργασία είναι παρόμοια με αυτή που υπάρχει στο [19]. Ωστόσο, εστιάζουμε στην ασφαλή προσθήκη των νέων αξιωμάτων στην οντολογία.

Η απάντηση ερωτημάτων (query answering) χρησιμοποιείται για τον έλεγχο της συνέπειας στις οντολογίες. Όμως, η εύρεση της προέλευσης των ασυνεπειών δεν είναι εύκολο να πραγματοποιηθεί απευθείας, αν χρησιμοποιούμε συζευτικά ερωτήματα. Η χρήση των συζευτικών ερωτημάτων με αρνητικά άτομα (Conjunctive queries with negated atoms) [1] επιτρέπει να εκφραστούν οι ασυνέπειες (αντιπαραδείγματα) που σχετίζονται με τα νέα αξιώματα με έναν ευκολότερο τρόπο.

1.1 Αντικείμενο της διπλωματικής

Στο πλαίσιο της διπλωματικής αυτής εργασία, αναπτύσσουμε ένα σύστημα που συνάγει νέα αξιώματα με βάση τους ισχυρισμούς που υπάρχουν σε μια οντολογία. Τα νέα αξιώματα ελέγχονται για να διασφαλιστεί η συνέπεια όταν προστίθενται στην οντολογία. Τα αξιώματα που θα προκαλούσαν ασυνέπειες διορθώνονται, ώστε να αφαιρέσουν τα υπάρχοντα αντιπαραδείγματα που αντιπροσωπεύουν την πηγή των ασυνεπειών.

1.2 Οργάνωση του τόμου

Στο Κεφάλαιο 2 παρουσιάζουμε το θεωρητικό υπόβαθρο όλων των τεχνικών καθώς και τα συστήματα που χρησιμοποιήθηκαν (Weka, Completo) σε αυτή την εργασία. Η θεωρητική προσέγγιση του προβλήματος αναλύεται στο Κεφάλαιο 3, ενώ στο Κεφάλαιο 4 παρουσιάζεται η υλοποίηση του προτεινόμενου μοντέλου που αναπτύχθηκε στην παρούσα διπλωματική. Στο Κεφάλαιο 5 παρουσιάζονται μερικά πειραματικά αποτελέσματα. Τέλος στο Κεφάλαιο 6 θα εξαχθούν συμπεράσματα από τα πειραματικά αποτελέσματα και θα αξιολογηθεί το έργο που επιτεύχθηκε.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Εισαγωγή

Σε αυτή την ενότητα παρέχεται μια εισαγωγή στις Περιγραφικές Λογικές, τον τρόπο με τον οποίο μοντελοποιείται η γνώση στις Περιγραφικές Λογικές καθώς και τα πιο σημαντικά χαρακτηριστικά μοντελοποίησης τους, με σκοπό την κατάληξη στην ομοιότητά και τη χρήση τους στις Οντολογίες.

2.1.1 Περιγραφικές Λογικές

Οι Περιγραφικές Λογικές (ΠΛ) (Description logics - DLs) είναι μία οικογένεια γλωσσών αναπαράστασης που χρησιμοποιούνται ευρέως στην μοντελοποίηση οντολογιών. Το όνομα των Περιγραφικών Λογικών προήλθε από το γεγονός ότι, από τη μία πλευρά, οι σημαντικές έννοιες ενός τομέα περιγράφονται από περιγραφές ιδεών, δηλαδή εκφράσεις που είναι κατασκευασμένες από ατομικές έννοιες (μοναδιαία κατηγορήματα - unary predicates) και ατομικούς ρόλους (δυαδικά κατηγορήματα - binary predicates) χρησιμοποιώντας τους κατασκευαστές εννοιών και ρόλων που παρέχονται από την συγκεκριμένη ΠΛ. Από την άλλη, οι ΠΛ διαφέρουν από τους προκατόχους τους, επειδή είναι εξοπλισμένες με μία επίσημη, βασισμένη στη λογική σημασιολογία [3, 15, 17].

Στις ΠΛ υπάρχουν τρία είδη οντοτήτων: έννοιες, ρόλοι και επονομαζόμενα άτομα (concepts, roles, individual names). Οι έννοιες αντιπροσωπεύουν ομάδες ατόμων, οι ρόλοι δυαδικές σχέσεις μεταξύ των ατόμων και τα επονομαζόμενα άτομα αντιπροσωπεύουν μεμονωμένα άτομα. Για παράδειγμα, μία οντολογία που μοντελοποιεί τον τομέα των ανθρώπων και τις οικογενειακές τους σχέσεις θα μπορούσε να χρησιμοποιεί έννοιες όπως την `Parent` για να αντιπροσωπεύει την ομάδα όλων των γονέων και την `Female` για να αντιπροσωπεύει την ομάδα όλων των γυναικών, ρόλους όπως τον `parentOf` για να αντιπροσωπεύει την δυαδική σχέση μεταξύ των γονέων και των παιδιών τους, και τέλος επονομαζόμενα άτομα όπως `julia` και `john` για να αντιπροσωπεύει τα άτομα

Julia και John.

Μερικοί χαρακτηριστικοί κατασκευαστές θα απεικονιστούν με ένα παράδειγμα. Ας υποθέσουμε ότι θέλουμε να ορίσουμε την έννοια: ΄Ανδρας που είναι παντρεμένος με ιατρό και έχει τουλάχιστον πέντε παιδιά, από τα οποία όλα είναι καθηγητές'. Αυτή η έννοια μπορεί να δηλωθεί με την παρακάτω περιγραφή έννοιας:
`Human`$\sqcap\neg$`Female`$\sqcap(\exists$`married.Doctor`$)\sqcap(\geq$`5hasChild`$)\sqcap(\forall$`hasChild.Professor`$)$.

Η συγκεκριμένη περιγραφή χρησιμοποιεί τον κατασκευαστή τομής (conjuction - $\sqcap$), άρνησης (negation - $\neg$), καθώς και τον υπαρξιακό τελεστή (existential restriction constructor - $\exists$), τον καθολικό τελεστή (value restriction constructor - $\forall$) και τέλος τον τελεστή αριθμητικού περιορισμού (number restriction constructor - $\geq$). Ένα άτομο, ας πούμε ο Bob, ανήκει στην έννοια $\exists$ `married.Doctor` αν υπάρχει κάποιο άτομο το οποίο να είναι παντρεμένο με τον Bob (πιο σωστά, να συνδέεται με τον Bob μέσω της σχέσης `married`) και να είναι ιατρός (δηλαδή, ανήκει στην έννοια `Doctor`). Παρομοίως, ο Bob ανήκει στην έννοια $\geq$ `5hasChild` αν έχει τουλάχιστον πέντε παιδιά, και στην έννοια $\forall$ `hasChild.Professor` αν όλα του τα παιδιά (δηλαδή όλα τα άτομα που συνδέονται με τον Bob μέσω του ρόλου `hasChild`) είναι καθηγητές.

Στην απλούστερη μορφή τους, τα ορολογικά αξιώματα μπορούν να χρησιμοποιηθούν για την εισαγωγή ονομάτων (συντομογραφιών) για σύνθετες περιγραφές. Για παράδειγμα, για την περιγραφή της παραπάνω έννοιας θα μπορούσαμε να εισάγουμε τη συντομογραφία `HappyMan`. Πιο εκφραστικοί φορμαλισμοί αναπαράστασης επιτρέπουν τη δήλωση των περιορισμών όπως το $\exists$ `hasChild.Human` $\sqsubseteq$ `Human`, που λέει ότι μόνο οι άνθρωποι μπορούν να έχουν παιδιά ανθρώπους. Οι έννοιες και οι ρόλοι μπορούν να χρησιμοποιηθούν σε μια βάση γνώσης για την εξαγωγή γνώσης, τόσο σε επίπεδο σώματος ορολογίας (TBox – Terminological Box), όσο και σε επίπεδο σώματος ισχυρισμών (ABox – Assertional Box). Το TBox συνήθως αποτελείται από ένα σύνολο αξιωμάτων που δηλώνουν υπαγωγή ανάμεσα σε έννοιες και ρόλους. Στο ABox, τα αξιώματα αναφέρονται σε γνώση σχετική με επονομαζόμενα άτομα. Όλα αυτά θα αναλυθούν εκτενέστερα στις παρακάτω ενότητες.

2.1.2 Εισαγωγή ισχυρισμών με Αξιώματα ABox

Τα ABox αξιώματα κατέχουν γνώση σχετικά με τα επονομαζόμενα άτομα, δηλαδή τις έννοιες στις οποίες ανήκουν και πώς σχετίζονται μεταξύ τους. Τα πιο συνηθισμένα ABox αξιώματα είναι η εισαγωγή εννοιών όπως το `Mother(julia)`, που ισχυρίζεται ότι η Julia είναι μητέρα, ή πιο ειδικά, ότι το άτομο με το όνομα `julia` είναι στιγμιότυπο της έννοιας `Mother`.

Η εισαγωγή ρόλων περιγράφει την σχέση μεταξύ των επονομαζόμενων ατόμων. Η εισαγωγή του αξιώματος `parentOf(julia, john)` για παράδειγμα, ισχυρίζεται ότι η Julia είναι γονέας του John, ή πιο συγκεκριμένα, ότι το άτομο με το όνομα `julia`

συνδέεται μέσω της σχέσης `parentOf` με το άτομο με το όνομα `john`.

Αν και είναι σαφές ότι η Julia και ο John είναι διαφορετικά άτομα, αυτό το γεγονός δεν προκύπτει λογικά από αυτό που έχουμε δηλώσει μέχρι τώρα. Οι ΠΛ δεν κάνουν την υπόθεση μοναδικού ονόματος (unique name assumption), έτσι διαφορετικά ονόματα μπορεί να αναφέρονται στο ίδιο άτομο εκτός και αν αναφέρεται ρητά κάτι διαφορετικό. Ο ισχυρισμός ανισότητας των ατόμων (individual inequality) $\texttt{julia} \not\approx \texttt{john}$ χρησιμοποιείται για να δηλώσουμε ότι η Julia και ο John είναι πράγματι διαφορετικά άτομα. Από την άλλη, ο ισχυρισμός ισότητας ατόμων (individual equality), όπως ο $\texttt{john} \approx \texttt{johnny}$, δηλώνει ότι δύο διαφορετικά ονόματα είναι γνωστό ότι αναφέρονται στο ίδιο άτομο.

2.1.3 Έκφραση γνώσης με Αξιώματα TBox

Τα αξιώματα TBox περιγράφουν σχέσεις μεταξύ εννοιών. Για παράδειγμα, το γεγονός ότι όλες οι μητέρες είναι γονείς εκφράζεται από την υπαγωγή των εννοιών (concept inclusion) $\texttt{Mother} \sqsubseteq \texttt{Parent}$, στην οποία λέμε ότι η έννοια Mother υπάγεται της έννοιας Parent. Τέτοιου είδους γνώση μπορεί να χρησιμοποιηθεί ώστε να συναχθούν περισσότερα γεγονότα σχετικά με τα άτομα. Η ισοδυναμία εννοιών (concept equivalence) δηλώνει ότι δύο έννοιες έχουν τα ίδια στιγμιότυπα, όπως οι $\texttt{Person} \equiv \texttt{Human}$.

2.1.4 Μοντελοποίηση Σχέσεων μεταξύ Ρόλων με Αξιώματα RBox

Τα RBox αξιώματα αναφέρονται στις ιδιότητες των ρόλων. Όπως και για τις έννοιες, οι ΠΛ υποστηρίζουν την υπαγωγή ρόλων (role inclusion) και την ισοδυναμία ρόλων (role equivalence). Για παράδειγμα, η υπαγωγή $\texttt{parentOf} \sqsubseteq \texttt{ancestorOf}$ δηλώνει ότι ο ρόλος `parentOf` είναι υπορόλος του `ancestorOf`, δηλαδή κάθε ζευγάρι ατόμων που συνδέεται μέσω του `parentOf` θα συνδέεται επίσης μέσω του `ancestorOf`. Στα αξιώματα υπαγωγής ρόλων, η σύνθεση ρόλων μπορεί να χρησιμοποιηθεί ώστε να περιγραφούν ρόλοι όπως ο `uncleOf`. Σαφώς αν ο Charles είναι αδερφός της Julia και η Julia είναι γονέας του John, τότε ο Charles είναι θείος του John. Αυτού του είδους η σχέση μεταξύ των ρόλων `brotherOf, parentOf` και `uncleOf` αναπαρίσταται από το αξίωμα $\texttt{brotherOf} \circ \texttt{parentOf} \sqsubseteq \texttt{uncleOf}$.

Κανείς δε μπορεί να είναι γονιός και παιδί του ίδιου ατόμου ταυτόχρονα, επομένως οι ρόλοι `parentOf` και `childOf` είναι ξένοι (disjoint). Στις ΠΛ μπορούμε να ορίσουμε τους ξένους ρόλους ως εξής: `Disjoint( parentOf, childOf)`. Περισσότερα αξιώματα RBox περιλαμβάνουν χαρακτηριστικά των ρόλων όπως η ανακλαστικότητα (reflexivity), η συμμετρικότητα (symmetry) και η μεταβατικότητα (transitivity) των ρόλων.

2.1.5 Κατασκευαστές εννοιών

Οι κατασκευαστές εννοιών (concept constructors) παρέχουν βασικές λειτουργίες που είναι στενά συνδεδεμένες με τις γνωστές λειτουργίες της τομής, της ένωσης και την άρνηση των συνόλων. Για παράδειγμα στην υπαγωγή εννοιών μπορούμε να δηλώσουμε ότι όλες οι μητέρες είναι γυναίκες και ότι όλες οι μητέρες είναι γονείς, αλλά αυτό που πραγματικά εννοούμε είναι ότι οι μητέρες είναι ακριβώς οι γυναίκες γονείς. Οι ΠΛ υποστηρίζουν τέτοιες δηλώσεις επιτρέποντας μας να δημιουργήσουμε σύνθετες έννοιες όπως η τομή (intersection ή conjuction) $\texttt{Female} \sqcap \texttt{Parent}$, η οποία αναπαριστά το σύνολο των ατόμων που είναι ταυτόχρονα γυναίκες και γονείς. Μία σύνθετη έννοια μπορεί να χρησιμοποιηθεί στα αξιώματα με ακριβώς τον ίδιο τρόπο όπως μία ατομική έννοια, δηλαδή με την ισοδυναμία $\texttt{Mother} \equiv \texttt{Female} \sqcap \texttt{Parent}$. Η ένωση (union ή disjunction) είναι το συμπλήρωμα της τομής. Για παράδειγμα, η έννοια $\texttt{Father} \sqcup \texttt{Mother}$ περιγράφει τα άτομα που είναι είτε πατέρες ή μητέρες. Ξανά, μπορεί να χρησιμοποιηθεί σε ένα αξίωμα ως $\texttt{Parent} \equiv \texttt{Father} \sqcup \texttt{Mother}$, το οποίο δηλώνει ότι ένας γονέας είναι είτε πατέρας ή μητέρα.

Ορισμένες φορές ενδιαφερόμαστε για άτομα που δεν ανήκουν σε μία συγκεκριμένη έννοια, όπως $\neg\texttt{Married}$ που αντιπροσωπεύει το σύνολο των ατόμων που δεν είναι παντρεμένα.

Ακόμη, είναι συχνά χρήσιμο να κάνουμε κάποια δήλωση για όλα τα άτομα, για παράδειγμα να πούμε ότι όλοι είναι είτε άνδρες ή γυναίκες. Αυτό κατορθώνεται με το αξίωμα $\top \sqsubseteq \texttt{Male} \sqcup \texttt{Female}$, όπου η έκφραση έννοιας $\top$ (η αλλιώς Top) είναι μία ειδική έννοια όπου ανήκουν όλα τα άτομα.

Από την άλλη, κανείς δε μπορεί να είναι ταυτόχρονα άνδρας και γυναίκα, έτσι μπορούμε να δηλώσουμε ότι το σύνολο των ανδρών και το αντίστοιχο των γυναικών είναι ασύνδετα μεταξύ τους. Η έννοια $\bot$ (Bottom) είναι η συμπληρωματική της $\top$, και αντιπροσωπεύει την ειδική έννοια όπου κανένα άτομα δεν ανήκει, όπως για παράδειγμα $\text{Male} \sqcap \text{Female} \sqsubseteq \bot$.

2.1.6 Περιορισμοί ρόλων

Ένα από τα πιο ενδιαφέροντα χαρακτηριστικά των ΠΛ είναι η ικανότητα τους να δημιουργούν δηλώσεις οι οποίες συνδέουν έννοιες και ρόλους. Για παράδειγμα, υπάρχει μία προφανής σχέση μεταξύ της έννοιας `Parent` και του ρόλου `parentOf`, δηλαδή, ένας γονέας είναι κάποιος που είναι γονέας τουλάχιστον ενός ατόμου. Στις ΠΛ, η σχέση αυτή μπορεί να διατυπωθεί μέσω της ισοδυναμίας εννοιών $\texttt{Parent} \equiv \exists\ \texttt{parentOf}.\top$ όπου ο υπαρξιακός περιορισμός $\exists\ \texttt{parentOf}.\top$ είναι μία σύνθετη έννοια που περιγράφει το σύνολο των ατόμων τα οποία είναι γονείς τουλάχιστον ενός ατόμου (στιγμιότυπο της $\top$). Για να αναπαραστήσουμε το σύνολο των ατόμων όπου όλα τους τα παιδιά είναι

γυναίκες, χρησιμοποιούμε τον καθολικό περιορισμό $\forall$ `parentOf.Female`.

Οι υπαρξιακοί και καθολικοί περιορισμοί είναι χρήσιμοι σε συνδυασμό με την έννοια $\top$ για να εκφράσουμε περιορισμούς στο πεδίο ορισμού και στο πεδίο τιμών του δοσμένου ρόλου. Για να περιορίσουμε το πεδίο ορισμού του ρόλου `sonOf` στους άνδρες μπορούμε να χρησιμοποιήσουμε το αξίωμα $\exists$ `sonOf`.$\top \sqsubseteq$ `Male` και για να περιορίσουμε το πεδίο τιμών του στους γονείς μπορούμε να γράψουμε $\top \sqsubseteq \forall$ `sonOf.Parent`.

Οι αριθμητικοί περιορισμοί μας επιτρέπουν να περιορίσουμε το πλήθος των ατόμων που μπορούν να προσεγγιστούν μέσω του δοσμένου ρόλου. Για παράδειγμα, μπορούμε να κατασκευάσουμε τον περιορισμό τουλάχιστον (at-least restriction) $\geq$`2childOf.Parent` για να περιγράψουμε το σύνολο των ατόμων που είναι παιδιά τουλάχιστον δύο γονέων, και τον περιορισμό το πολύ (at-most restriction) $\leq$`2childOf.Parent` για αυτούς που είναι παιδιά το πολύ δύο γονέων.

2.2 Γλώσσες Αναπαράστασης Γνώσης

Γίνεται φανερό ότι είναι αναγκαίο να αναπτυχθούν γλώσσες οι οποίες να παρέχουν τις δυνατότητες μοντελοποίησης γνώσης που έχει μία οντολογία, ενώ ταυτόχρονα να είναι εύκολα κατανοητές από τον άνθρωπο αλλά και επεξεργάσιμες από τον υπολογιστή.

Παρακάτω παρουσιάζονται οι γλώσσες RDF, OWL και γίνεται μία σύντομη περιγραφή του SKOS.

RDF Η RDF (Resource Description Framework) είναι η πρώτη γλώσσα που υιοθετήθηκε από τον οργανισμό W3C ως μοντέλο αναπαράστασης μεταδεδομένων [16].

Στην RDF, η βασική ιδέα είναι ότι τα άτομα που πρόκειται να περιγραφούν έχουν κάποιες ιδιότητες για τις οποίες παρουσιάζουν συγκεκριμένες τιμές. Η αναπαράσταση γίνεται με τη μορφή τριάδων που αποτελούνται από ένα υποκείμενο (subject), μία ιδιότητα (property) και ένα αντικείμενο (object). Μία τέτοια τριάδα ονομάζεται πρόταση.

Για την αναπαράσταση της ιδιότητας ενός ατόμου, το άτομο παίζει το ρόλο του υποκειμένου, η ιδιότητα παίζει το ρόλο της ιδιότητας και η τιμή που έχει το άτομο για τη συγκεκριμένη ιδιότητα παίζει το ρόλο του αντικειμένου

Τα υποκείμενα, οι ιδιότητες και τα αντικείμενα καταγράφονται με τη μορφή URIs. Η γλώσσα που χρησιμοποιείται για την καταγραφή προτάσεων είναι η XML. Η ακριβής σύνταξη περιγράφεται με το πρότυπο RDF/XML.

Η RDF είναι μία σχετικά απλή γλώσσα αναπαράστασης γνώσης και λειτουργεί ως βάση για την ανάπτυξη άλλων, πιο πολύπλοκων, αυξημένης λειτουργικότητας, γλωσσών, όπως η OWL που εξετάζεται παρακάτω.

OWL Τα αρχικά OWL προκύπτουν με αναγραμματισμό από τον όρο Web Ontology Language. Όπως υποδηλώνει και το όνομά της, η OWL είναι μία γλώσσα συγγραφής οντολογιών που μπορεί να χρησιμοποιηθεί στο διαδίκτυο [6].

Η OWL αποτελεί μία επέκταση της RDF, σε σχέση με την οποία παρουσιάζει βελτιώσεις τόσο ως προς τη λειτουργικότητα όσο και ως προς την ευχρηστία. Ειδικότερα, η OWL παρέχει επιπλέον της RDF τη δυνατότητα ορισμού κλάσεων μέσω της διάζευξης, σύζευξης ή άρνησης άλλων κλάσεων. Επίσης, λόγω του συντακτικού της, γίνεται εύκολα κατανοητή από τον άνθρωπο και επομένως είναι εύκολη στη χρήση. Ταυτόχρονα, ως επέκταση της RDF, είναι συμβατή με αυτήν και φυσικά με το πρότυπο XML.

Η OWL ορίζει τρεις υπογλώσσες διαφορετικού επιπέδου εκφραστικότητας:

- OWL Lite: Πρόκειται για την απλούστερη έκδοση της OWL. Απευθύνεται σε εφαρμογές που δεν έχουν υψηλές απαιτήσεις ως προς την εκφραστικότητα, το οποίο επιτρέπει την ανάπτυξη αποδοτικότερων εργαλείων που λειτουργούν ταχύτερα σε σχέση με τα εργαλεία που αφορούν σε πιο πλήρεις εκδοχές της OWL.

- OWL DL: Σε αυτή την έκδοση της OWL παρέχεται η μέγιστη εκφραστικότητα, ενώ παράλληλα διατηρούνται σε ικανοποιητικό επίπεδο οι υπολογιστικές δυνατότητες των εργαλείων της γλώσσας.

- OWL Full: Εδώ παρέχεται το ίδιο εκφραστικό επίπεδο με την OWL DL, χωρίς όμως να υπάρχουν συντακτικοί περιορισμοί, ενώ ταυτόχρονα παρέχεται η δυνατότητα της μεταμοντελοποίησης. Αυτά τα χαρακτηριστικά κάνουν τη γλώσσα να είναι μη-αποφασίσιμη (undecidable).

Η γλώσσα OWL αναλύεται περισσότερο στην ενότητα 2.3.

SKOS Τα αρχικά SKOS προκύπτουν από την ονομασία Simple Knowledge Organization System (SKOS). Πρόκειται για ένα μοντέλο δεδομένων που αφορά στο διαμοιρασμό και τη σύνδεση συστημάτων οργάνωσης γνώσης μέσω του διαδικτύου [13].

Το SKOS έχει δημιουργηθεί σύμφωνα με το πρότυπο RDF και μπορεί να χρησιμοποιηθεί σε συνδυασμό με την OWL. Αφορά στην κωδικοποίηση υπαρχόντων θησαυρών, συστημάτων ταξινόμησης, κ.α. και στην ενσωμάτωσή τους στο Σημασιολογικό Ιστό με τη μορφή συνδεδεμένων δεδομένων.

Το μοντέλο SKOS διαθέτει απλό συντακτικό το οποίο γίνεται εύκολα κατανοητό από τον άνθρωπο. Οι κατηγορίες ατόμων θεωρούνται ως έννοιες (concepts). Το SKOS επιτρέπει τη μοντελοποίηση της ιεραρχίας των εννοιών αλλά και τη δήλωση εναλλακτικών ονομασιών για κάθε έννοια.

2.3 OWL 2 - Web Ontology Language

Η τρέχουσα έκδοση της Web Ontology Language είναι η OWL 2, μία σύσταση του W3C (World Wide Web Consorium) από τον Οκτώβριο του 2009 [14, 11]. Η OWL 2 είναι μία γλώσσα αναπαράστασης γνώσης, που αποτελεί μία από τις σημαντικότερες εφαρμογές των ΠΛ σήμερα. Οι κύριοι πυλώνες της OWL είναι πράγματι πολύ παρόμοιοι με εκείνους των ΠΛ, με τη βασική διαφορά ότι οι έννοιες καλούνται κλάσεις (classes) και οι ρόλοι καλούνται ιδιότητες (properties). Δεν αποτελεί έκπληξη λοιπόν, το γεγονός ότι οι ΠΛ είχαν μεγάλη επιρροή στην ανάπτυξη της OWL και στα χαρακτηριστικά που παρέχει.

Η OWL 2 σχεδιάστηκε για να αντιπροσωπεύει πλούσιες και περίπλοκες γνώσεις για πράγματα, ομάδες πραγμάτων και σχέσεις μεταξύ των πραγμάτων. Η OWL 2 βασίζεται στη λογική, έτσι ώστε η γνώση η οποία εκφράζει να μπορεί να χρησιμοποιηθεί από υπολογιστικά προγράμματα για να εξακριβωθεί η συνέπειά της και να καταστεί σαφής η έμμεση γνώση. Τα έγγραφα OWL (οντολογίες) μπορούν να δημοσιευτούν στον Παγκόσμιο Ιστό και αναφέρονται σε (ή από) άλλες οντολογίες. Υπάρχουν αρκετές συντάξεις διαθέσιμες για την OWL που εξυπηρετούν διάφορους σκοπούς. Για τους σκοπούς της παρούσας διπλωματικής, θα χρησιμοποιηθεί η Fynctional-Style Syntax, η οποία έχει σχεδιαστεί για να είναι ευκολότερη για σκοπούς προσδιορισμών και για να παρέχει μια βάση για την εφαρμογή εργαλείων της OWL 2, όπως APIs και reasoners.

Όπως αναφέρθηκε η OWL 2 είναι μία γλώσσα αναπαράστασης γνώσης, που έχει σχεδιαστεί για να διατυπώνει, ανταλλάσσει και να βγάζει συμπεράσματα σχετικά με έναν τομέα ενδιαφέροντος. Μερικές θεμελιώδεις έννοιες πρέπει πρώτα να εξηγηθούν ώστε να κατανοήσουμε πως αναπαρίσταται η γνώση στην OWL 2. Αυτές οι έννοιες είναι:

- Αξιώματα (Axioms): δηλώσεις που μια οντολογία εκφράζει και ισχυρίζεται πως ισχύουν - αυτές περιλαμβάνουν τη συνολική θεωρία που περιγράφει η οντολογία στην περιοχή εφαρμογής της.

- Οντότητες (Entities): έννοιες που αναφέρονται σε αντικείμενα του πραγματικού κόσμου - άτομα, κλάσεις, ιδιότητες

- Εκφράσεις (Expressions): συνδυασμός των οντοτήτων για τον σχηματισμό σύνθετων εννοιών που επιτυγχάνεται συνδυάζοντας βασικές οντότητες με τη χρήση κατασκευαστών.

2.3.1 Κλάσεις και Στιγμιότυπα

Ξεκινάμε με την εισαγωγή των ατόμων στα οποία αναφερόμαστε. Αυτό γίνεται ως εξής:
`ClassAssertion(:Person :Mary).`
Η δήλωση αυτή μιλάει για ένα άτομο που ονομάζεται Mary και δηλώνει ότι το άτομο αυτό είναι άνθρωπος. Πιο σωστά, δηλώνει ότι το άτομο με το όνομα `Mary` είναι στιγμιότυπο της κλάσης `Person`. Γενικότερα, οι κλάσεις χρησιμοποιούνται για να ομαδοποιήσουμε τα άτομα που έχουν κάτι κοινό, ώστε να αναφερόμαστε σε αυτά. Έτσι, οι κλάσεις αναπαριστούν ομάδες ατόμων. Είναι προφανές, ότι η συμμετοχή σε μία κλάση δεν είναι αποκλειστική: καθώς μπορεί να υπάρχουν διαφορετικά κριτήρια για την ομαδοποίηση των ατόμων (όπως φύλο, ηλικία, κ.λ.π.), ένα άτομο μπορεί να ανήκει σε διάφορες κλάσεις ταυτόχρονα. Κάποιες από τις κλάσεις που χρησιμοποιούνται πιο συχνά για την δημιουργία οντολογιών μέσω της OWL είναι:

- `owl:Thing`: Αποτελεί την αρχική κλάση, τη ρίζα στην ιεραρχία των κλάσεων. Όλες οι υπόλοιπες κλάσεις είναι υποκλάσεις της, άρα όλα τα άτομα ανήκουν σε αυτή την κλάση.

- `owl:Nothing`: Αποτελεί την κενή κλάση, δηλαδή την κλάση στην οποία δεν ανήκει κανένα άτομο.

2.3.2 Ιεραρχία των Κλάσεων

Ας σκεφτούμε δύο διαφορετικές κλάσεις `Person` και `Woman`. Για έναν άνθρωπο είναι προφανές ότι η κλάση `Person` είναι πιο γενική από την `Woman`, εννοώντας ότι όποτε γνωρίζουμε ότι ένα άτομο είναι γυναίκα, αυτό το άτομο θα πρέπει να είναι και άνθρωπος. Για να μπορέσει ένα σύστημα να βγάλει αυτό το συμπέρασμα, πρέπει να ενημερωθεί για τη συσχέτιση μεταξύ των δύο κλάσεων. Αυτό επιτυγχάνεται με το αξίωμα υπαγωγής κλάσεων:
`SubClassOf( :Woman :Person).`

Η ύπαρξη του αξιώματος αυτού, επιτρέπει στον reasoner να εξάγει για κάθε άτομο που ορίζεται ως στιγμιότυπο της κλάσης `Woman` ότι είναι και στιγμιότυπο της κλάσης `Person`.

Επίσης, ορισμένες κλάσεις μπορεί να αναφέρονται στην ίδια ομάδα, έτσι η OWL παρέχει ένα μηχανισμό όπου αυτές οι κλάσεις ορίζονται ως ισοδύναμες. Δύο κλάσεις θεωρούνται ισοδύναμες αν έχουν ακριβώς τα ίδια άτομα ως στιγμιότυπα. Για παράδειγμα για να δείξουμε ότι κάθε στιγμιότυπο της κλάσης `Person` είναι και της κλάσης `Human` χρησιμοποιούμε το αξίωμα:
`EquivalentClasses( :Person :Human).`

2.3.3 Ασυμφωνία Κλάσεων

Σε κάποιες περιπτώσεις, η συμμετοχή σε μία κλάση, απαγορεύει τη συμμετοχή σε κάποια άλλη. Για παράδειγμα, γνωρίζουμε ότι κανένα άτομο δε μπορεί να είναι ταυτόχρονα στιγμιότυπο των κλάσεων `Man` και `Woman`. Αυτό αναφέρεται ως ασυμφωνία κλάσεων και επιτυγχάνεται όπως παρακάτω:
`DisjointClasses( :Woman :Man)`.

2.3.4 Ιδιότητες

Παραπάνω αναφέραμε πως οι κλάσεις σχετίζονται μεταξύ τους. Όμως πιο συχνά, μία οντολογία ορίζει πως τα άτομα συνδέονται με άλλα άτομα. Οι οντότητες που περιγράφουν με ποιόν τρόπο συνδέονται τα άτομα, ονομάζονται ιδιότητες (properties). Αν θέλουμε να δείξουμε ότι η `Mary` είναι η γυναίκα του `John`, θα λέμε:
`ObjectPropertyAssertion( :hasWife :John :Mary)`.

Επίσης μπορούμε να δηλώσουμε και ότι δύο άτομα δεν συνδέονται με μία ιδιότητα. Για παράδειγμα το `NegativeObjectPropertyAssertion( :hasWife :Bill :Mary)` μας δείχνει ότι η `Mary` δεν είναι γυναίκα του `Bill`.

2.3.5 Ιεραρχία Ιδιοτήτων

Όπως και στις κλάσεις, η δήλωση
`SubObjectPropertyOf( :hasWife :hasSpouse)`
δηλώνει ότι όποτε γνωρίζουμε πως ο B είναι γυναίκα του A, τότε είναι και σύζυγος του A. Ομοίως, υπάρχει αξίωμα που δηλώνει την ισότητα ιδιοτήτων.

2.3.6 Περιορισμοί Πεδίου Ορισμού και Πεδίου Τιμών

Συχνά, η πληροφορία ότι δύο άτομα συνδέονται μέσω μίας ιδιότητας μπορεί να οδηγήσει σε παραπάνω συμπεράσματα σχετικά με τα ίδια τα άτομα. Πιο συγκεκριμένα, μπορούμε να πάρουμε γνώση για τη συμμετοχή των ατόμων σε κάποια κλάση. Για παράδειγμα η δήλωση ότι ο B είναι γυναίκα του A, μας δείχνει ότι ο B είναι γυναίκα ενώ ο A είναι άντρας. Στο συγκεκριμένο παράδειγμα, θα χρησιμοποιούσαμε τα αξιώματα:
`ObjectPropertyDomain( :hasWife :Man)`
`ObjectPropertyRange( :hasWife :Woman)`

2.3.7 Ισότητα και Ανισότητα Ατόμων

Από τις πληροφορίες που έχουμε ως τώρα, μπορούμε να συμπεράνουμε ότι ο `John` και η `Mary` δεν αναφέρονται στο ίδιο άτομο, αφού αποτελούν στιγμιότυπα των ξένων

κλάσεων Man και Woman, αντίστοιχα. Όμως, αν προσθέσουμε πληροφορίες για κάποιο άλλο οικογενειακό μέλος, ας πούμε τον Bill, και δηλώσουμε ότι είναι στιγμιότυπο της κλάσης Man, τότε τίποτα δε μας λέει ότι ο John και ο Bill δεν είναι το ίδιο άτομο. Η OWL δεν κάνει την υπόθεση μοναδικού ονόματος. Για αυτό το λόγο, αν θέλουμε να εξαλείψουμε την επιλογή ο John και ο Bill να αποτελούν το ίδιο άτομο, θα πρέπει να κάνουμε την παρακάτω δήλωση:
`DifferentIndividuals( :John :Bill).`

Ομοίως, μπορούμε να δηλώσουμε ότι δύο ονόματα αναφέρονται πράγματι στα ίδια άτομα. Για παράδειγμα, λέμε ότι ο James και ο Jim είναι τα ίδια άτομα αν υπάρχει η δήλωση:
`SameIndividual( :James :Jim).`

2.3.8 Σύνθετες Κλάσεις

Η OWL παρέχει γλωσσικά στοιχεία για τα λογικά και (and), ή (or) και όχι (not). Οι αντίστοιχοι όροι της OWL δανείζονται από τη θεωρία συνόλων: τομή (intersection), ένωση (union) και συμπλήρωμα (complement). Αυτοί οι κατασκευαστές συνδυάζουν ατομικές κλάσεις σε σύνθετες κλάσεις.

Η τομή δύο κλάσεων αποτελείται ακριβώς από τα άτομα που είναι στιγμιότυπα και των δύο κλάσεων. Το παρακάτω παράδειγμα δηλώνει ότι η κλάση Mother αποτελείται από τα αντικείμενα που είναι στιγμιότυπα των Woman και Parent ταυτόχρονα:
`EquivalentClasses(:Mother ObjectIntersectionOf( :Woman :Parent)).`

Η ένωση δύο κλάσεων αποτελείται από όλα τα άτομα που είναι στιγμιότυπα σε τουλάχιστον μία από τις δύο κλάσεις. Με αυτόν τον τρόπο θα μπορούσαμε να χαρακτηρίσουμε την κλάση των γονέων ως ένωση των κλάσεων Mother και Father:
`EquivalentClasses(:Parent ObjectUnionOf( :Mother :Father)).`

Το συμπλήρωμα μίας κλάσης αντιστοιχεί στη λογική άρνηση: αποτελείται ακριβώς από τα άτομα τα οποία δεν ανήκουν στην κλάση αυτή. Για παράδειγμα, οι άνθρωποι που δεν έχουν παιδιά, μπορούν να αναπαρασταθούν από: `EquivalentClasses(:ChildlessPerson ObjectIntersectionOf(:Person ObjectComplementOf( :Parent))).`

Όλα τα παραπάνω παραδείγματα δείχνουν τη χρήση των κατασκευαστών με σκοπό να ορίσουμε νέες κλάσεις σε συνδυασμό άλλων. Όμως, είναι επίσης πιθανό να χρησιμοποιούμε τέτοιους κατασκευαστές μαζί με δήλωση υπαγωγής κλάσεων. Για παράδειγμα η παρακάτω δήλωση μας δείχνει ότι κάθε άτομο που ανήκει στην κλάση Grandfather είναι ταυτόχρονα άντρας και πατέρας (παρόλο που το αντίθετο δεν ισχύει):
`SubClassOf(:Grandfather ObjectIntersectionOf( :Man :Parent)).`

2.3.9 Χαρακτηριστικά Ιδιοτήτων

Ορισμένες φορές μπορούμε να λάβουμε νέες ιδιότητες αλλάζοντας την κατεύθυνση της ιδιότητας, δηλαδή αντιστρέφοντας την. Για παράδειγμα, η ιδιότητα `hasParent` ορίζεται ως η αντίστροφη ιδιότητα της `hasChild`:
`InverseObjectProperties( :hasParent :hasChild)`.

Αυτό μας επιτρέπει να εξάγουμε το συμπέρασμα ότι για τα άτομα A και B, όπου το A συνδέεται με το B μέσω της ιδιότητας `hasChild`, τότε το B και το A συνδέονται επίσης μέσω της ιδιότητας `hasParent`. Φυσικά, δεν είναι απαραίτητο να θέσουμε ένα όνομα στην αντίστροφη ιδιότητα αν απλά θέλουμε να την χρησιμοποιήσουμε μέσα σε μία άλλη έκφραση. Για παράδειγμα η έκφραση:
`ObjectInverseOf( :hasChild )`
είναι ισοδύναμη με την `hasParent`.

2.3.10 Εκφραστικότητα OWL

Είναι σημαντικό να τονίσουμε ότι η εκφραστικότητα της OWL είναι μεγάλη. Όμως η μεγάλη εκφραστικότητα της γλώσσας αυτής συνοδεύται και από προβληματικά υπολογιστικά χαρακτηριστικά για τα συστήματα συλλογιστικής, γεγονός που σε ορισμένες εφαρμογές μπορεί να καταστήσει τη χρήση οντολογιών απαγορευτική.

Για το σκοπό αυτό, στο πλαίσιο ορισμού της OWL έχει οριστεί ένα σύνολο από εκφραστικά υποσύνολα της που ονομάζονται προφίλ (OWL profiles) και είναι τα εξής:

- Η OWL QL, που αποτελεί εκφραστικό αντίστοιχο μιας διαλέκτου της γλώσσας DL-Lite, και η οποία έχει πολύ καλά υπολογιστικά χαρακτηριστικά για την επίλυση του προβλήματος της απάντησης ερωτημάτων, κυρίως με τη μέθοδο της επαναγραφής.

- Η OWL RL, που αποτελεί εκφραστικό αντίστοιχο των περιγραφικών λογικών που αποτελούν υποσύνολα της $\mathcal{SHIQ}$ και έχουν την ιδιότητα ότι επιτρέπουν τη διατύπωση οριστικής γνώσης (αποτελούν υποσύνολα της Horn λογικής πρώτης τάξης). Η OWL RL έχει πολύ καλά υπολογιστικά χαρακτηριστικά για ένα σύνολο προβλημάτων συλλογιστικής, μέσω αλγορίθμων που στηρίζονται κυρίως στη μέθοδο της υλοποίησης.

- Η OWL EL, η οποία αποτελεί εκφραστικό αντίστοιχο μιας επεκταμένης έκδοσης της περιγραφικής λογικής $\mathcal{EL}$, που δίνει επιπλέον τη δυνατότητα για ανάστροφους ρόλους, ονοματικές έννοιες και τύπους δεδομένων, με κάποιους περιορισμούς. Στην OWL EL είναι ιδιαίτερα αποτελεσματικοί αλγόριθμοι ταξινόμησης εννοιών που στηρίζονται στον αλγόριθμο δομικής υπαγωγής.

Στην παρούσα διπλωματική εργασία, καθώς θέλουμε να μετατρέψουμε την οντολογία σε μία βάση γνώσης βασισμένη στους Υπαρξιακούς Κανόνες, δε μπορούμε να χρησιμοποιήσουμε όλες τις οντολογίες που είναι σε μορφή OWL. Για αυτό το λόγο, εστιάζουμε στο προφίλ OWL ER [4], στο οποίο μπορούν να γραφούν όλα τα αξιώματα από τα υπάρχοντα προφίλ που αναφέρθηκαν παραπάνω, και είναι εφικτό να μετατραπούν όλα τα αξιώματα μίας οντολογίας σε μία γνωστική βάση Υπαρξιακών Κανόνων (Existential Rules knowledge base).

2.4 Υπαρξιακοί Κανόνες (Existential Rules)

Ένας υπαρξιακός κανόνας r είναι μία λογική συνέπεια:

$$b_1(\mathbf{x}_1) \wedge \ldots \wedge b_m(\mathbf{x}_m) \rightarrow h(\mathbf{y}) \ ,$$

όπου τα άτομα $b_i(\mathbf{x}_i)$ που βρίσκονται από τα αριστερά του συμβόλου $\rightarrow$ είναι το σώμα του κανόνα (body) και $h(\mathbf{y})$ είναι η κεφαλή (head). Οι μεταβλητές που βρίσκονται τόσο στην κεφαλή όσο και στο σώμα είναι γνωστές ως *συνοριακές μεταβλητές (frontier variables)*. Οι μεταβλητές που είναι παρούσες μόνο στην κεφαλή του κανόνα είναι γνωστές ως *υπαρξιακές μεταβλητές (existential variables)*. Ένας υπαρξιακός κανόνας είναι ισοδύναμος με τον ακόλουθο τύπο λογικής πρώτης τάξης:

$$\forall \mathbf{X} \forall \mathbf{Z} \exists \mathbf{Y}^{+} \ (b_1(\mathbf{x}_1) \wedge \ldots \wedge b_m(\mathbf{x}_m) \rightarrow (\mathbf{y})) \ .$$

όπου $\mathbf{X}$ είναι το σύνολο των συνοριακών μεταβλητών, $\mathbf{Y}^{+}$ είναι αυτό των υπαρξιακών μεταβλητών και $\mathbf{Z}$ είναι το σύνολο των υπολοίπων μεταβλητών που παίρνουν μέρος στον κανόνα.

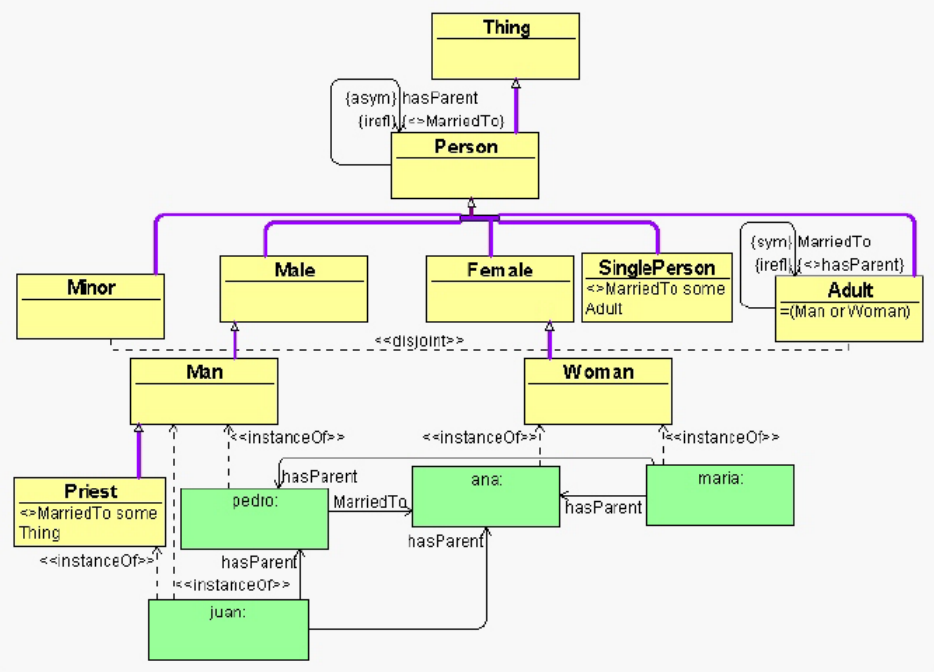
Όταν ένας κανόνας δεν έχει κεφαλή, καλείται αρνητικός περιορισμός (negative constraint) και αναπαρίσταται ως:

$$b_1(\mathbf{x}_1) \wedge \ldots \wedge b_m(\mathbf{x}_m) \rightarrow \bot \ .$$

Σημασιολογικά, οι περιορισμοί μπορούν να θεωρηθούν ως ερωτήματα τα οποία δεν πρέπει να ικανοποιούνται ώστε να υπάρχει συνέπεια.

Ορίζουμε ένα *γεγονός (fact)* ως το άτομο $a_i(\mathbf{t_i})$ και μία βάση δεδομένων $\mathcal{D}$ είναι ένα σύνολο από γεγονότα.

Στο σχήμα 2.1 παρουσιάζεται μία βάση γνώσης που περιέχει όλα τα στοιχεία που αναφέρθηκαν παραπάνω. Το διάγραμμα ακολουθεί τα πρότυπα της UML και περιλαμβάνει πρόσθετη σημειογραφία για να εκφράσει ιδιότητες όπως η συμμετρία (sym), η ασυμμετρία (asym) και η μη-ανακλαστικότητα (irefl). Οι ισοδυναμίες εκφράζονται με την ένδειξη ισότητας και η έννοια των ξένων κλάσεων (disjointness) εκφράζεται με το



Σχήμα 2.1: Διάγραμμα οντολογίας OWL.

σύμβολο ΄<>΄. Είναι εύκολο να καταλάβουμε ποια στοιχεία αντιπροσωπεύουν κανόνες, περιορισμούς και γεγονότα με τις βασικές γνώσεις της UML, όμως θα αναφέρουμε ορισμένα από αυτά:

- Κανόνες
  - $Adult(X) \rightarrow Person(X)$ (Υποκλάση)
  - $hasParent(X, Y) \rightarrow Person(X)$ (Πεδίο ορισμού)
  - $MarriedTo(X, Y) \rightarrow MarriedTo(Y, X)$ (Συμμετρία)
- Περιορισμοί
  - $Adult(X), Minor(X) \rightarrow \bot$ (Ξένες έννοιες)
  - $hasParent(X, Y), MarriedTo(X, Y) \rightarrow \bot$ (Ξένοι ρόλοι)
  - $Priest(X), MarriedTo(X, Y) \rightarrow \bot$
- Γεγονότα
  - $Woman(ana)$ (Ισχυρισμός έννοιας)
  - $hasParent(maria, pedro)$ (Ισχυρισμός ρόλου)

Ένα *συζευτικό ερώτημα (conjunctive query)* $q(\mathbf{x}) = a_1(\mathbf{x}_1), \ldots, a_n(\mathbf{x}_n)$ είναι μία συνένωση των ατόμων $a_i(\mathbf{x}_i)$ όπου $\mathbf{x}$ είναι ελεύθερες μεταβλητές (που καλούνται *μεταβλητές απάντησης (answer variables)*) και οι υπόλοιπες μεταβλητές ($\mathbf{Y} = (\cup_i^n \mathbf{x}_1) \setminus \mathbf{x}$) είναι ποσοτικά καθορισμένες:

$$q(\mathbf{x}) \equiv \exists \mathbf{Y} \ (a_1(\mathbf{x}_1) \wedge \ldots \wedge a_n(\mathbf{x}_n)) \ .$$

Μία *ένωση συζευτικών ερωτημάτων* (*union of conjunctive queries* (UCQ)), που δηλώνεται ως ένα σύνολο $\{q_1(\mathbf{x}), \ldots, q_{n'}(\mathbf{x})\}$ από συζευτικά ερωτήματα, αντιπροσωπεύει μία αποσύζευξη των συζευτικών ερωτημάτων $q_1(\mathbf{x}) \vee \ldots \vee q_{n'}(\mathbf{x})$.

Ένα ζεύγος (tuple) $\mathbf{t}$ είναι μία απάντηση του ερωτήματος $q(x)$ σε σχέση με ένα σύνολο κανόνων $\mathcal{R}$ και ένα σύνολο γεγονότων $\mathcal{D}$, αν το ερώτημα ικανοποιείται όταν αντικαθιστούμε τις μεταβλητές απάντησης με τις σταθερές στο $\mathbf{t}$:

$$\mathcal{D}, \mathcal{R} \models q(\mathbf{t}) \ .$$

Για ένα δοσμένο σύνολο κανόνων $\mathcal{R}$, ένα σύνολο από *UCQ-rewritings* ενός συζευτικού ερωτήματος (ή UCQ) $q(\mathbf{x})$ ορίζεται ως ένα UCQ $\mathcal{R}^*_{q(\mathbf{x})}$ τέτοιο ώστε για κάθε βάση δεδομένων $\mathcal{D}$:

$$\exists i \ q_i(\mathbf{x}) \in \mathcal{R}^*_q \text{ τέτοιο ώστε } \mathcal{D} \models q_i(\mathbf{x}) \text{ συνεπάγεται ότι } \mathcal{R}, \mathcal{D} \models q(\mathbf{x}) \ . \tag{2.1}$$

Αν ισχύει το αντίστροφο της (2.1) δηλαδή

$$\mathcal{R}, \mathcal{D} \models q(\mathbf{x}) \text{ συνεπάγεται ότι } \exists i \ q_i(\mathbf{x}) \in \mathcal{R}^*_{q(\mathbf{x})} \text{ τέτοιο ώστε } \mathcal{D} \models q_i(\mathbf{x}) \ ,$$

το σύνολο $\mathcal{R}^*_q$ είναι ένα *ολοκληρωμένο (complete)* UCQ-rewriting του q σε σχέση με το $\mathcal{R}$. Κάθε στοιχείο ενός UCQ-rewriting συνόλου καλείται *επαναγραφή* του αρχικού ερωτήματος σε σχέση με το $\mathcal{R}$.

Στη βάση γνώσης που χρησιμοποίησαμε και προηγουμένως (Σχήμα 2.1), αν επαναγράψουμε το ερώτημα $Q(X) : -Person(X)$ θα λάβουμε το εξής UCQ-rewriting:

$$\begin{aligned} Q(X) = & Adult(X) \vee Female(X) \vee Male(X) \vee Man(X) \vee Minor(X) \vee \\ & Person(X) \vee Priest(X) \vee SinglePerson(X) \vee Woman(X) \vee \\ & MarriedTo(Y, X) \vee MarriedTo(X, Y) \vee hasParent(Y, X) \vee \\ & hasParent(X, Y), \end{aligned}$$

με απαντήσεις $\{ana, maria, jua, pedro\}$.

Οι αλγόριθμοι επαναγραφής (rewriting algorithms) μας επιτρέπουν να περιορίσουμε το πρόβλημα της συλλογιστικής σε σχέση με ένα σύνολο κανόνων $\mathcal{R}$ και της βάσης δεδομένων $\mathcal{D}$, στο πρόβλημα της απάντησης ερωτημάτων σε σχέση με το $\mathcal{D}$.

Πίνακας 2.1: Μετάφραση των ΠΛ εκφράσεων σε Υπαρξιακούς κανόνες

Κατασκευαστής F και μεταβλητή X	Μετάφραση F^X
Έννοια C, D	Μετάφραση C^X (D^X)
A (όνομα έννοιας)	$A(X)$
$\exists R$	$R(X, \_)$
$\exists R^-$	$R(\_, X)$
Ρόλοι R, μεταβλητές X και Y	Μετάφραση $R^{(X,Y)}$
R (ρόλος)	$R(X, Y)$
R^- (ανάστροφος ρόλος)	$R(Y, X)$
A-Box Αξίωμα	Γεγονός
$A(a)$	$A(a)$
$\exists R(a)$	$R(a, \_)$
$\exists R^-(a)$	$R(\_, a)$
$R(a, b)$	$R(a, b)$
$R^-(a, b)$	$R(b, a)$
T-Box Αξίωμα	Κανόνας
$C \sqsubseteq D$	$C^X \rightarrow D^X$
$C \sqsubseteq \neg D$	$C^X, D^X \rightarrow \bot$
$R_1 \sqsubseteq R_2$	$R_1^{(X,Y)} \rightarrow R_2^{(X,Y)}$
$Dis(R_1, R_2)$	$R_1^{(X,Y)}, R_2^{(X,Y)} \rightarrow \bot$
$Asym(R_1)$	$R_1(X, Y), R_1(Y, X) \rightarrow \bot$
$Irr(R_1)$	$R_1(X, X) \rightarrow \bot$
$Sym(R_1)$	$R_1(X, Y) \rightarrow R_1(X, Y)$

Το πεδίο των υπαρξιακών κανόνων σχετίζεται με ορισμένα κομμάτια των ΠΛ. Πιο συγκεκριμένα, τα αξιώματα που ανήκουν στην $\mathsf{DL-Lite}_{\mathsf{core}}^{\mathcal{H}}$ μπορούν επίσης να εκφραστούν ως υπαρξιακοί κανόνες, περιορισμούς και γεγονότα. Στην $\mathsf{DL-Lite}_{\mathsf{core}}^{\mathcal{H}}$ μία οντολογία είναι το ζεύγος $< \mathcal{T}, \mathcal{A} >$ όπου $\mathcal{T}$ είναι το T-Box και $\mathcal{A}$ το A-Box. Ένα A-Box περιέχει αξιώματα σχετικά με έννοιες και ρόλους της μορφής $C(a)$ και $R(a, b)$ τα όποια είναι τα ίδια όπως τα γεγονότα στους υπαρξιακούς κανόνες. Ένα T-Box περιέχει υπαγωγή εννοιών $C \sqsubseteq D$ αλλά και υπαγωγή ρόλων $R_1 \sqsubseteq R_2$. Επιπρόσθετα, τα DL-Lite TBoxes μπορούν να περιέχουν περιορισμούς ρόλων για να εκφράσουν γνωστές ιδιότητες όπως διαφωνία (disjointness $Dis(R_1, R_2)$), ασυμμετρία (asymmetry $Asym(R)$), συμμετρία (symmetry $Sym(R)$) κλπ.

Ο πίνακας (2.1) παρουσιάζει τα στοιχεία από το T-Box μίας οντολογίας και την αντίστοιχη έκφραση τους στο πλαίσιο των υπαρξιακών κανόνων. Στο πρώτο μέρος ορίζεται ο τρόπος με τον οποίο, οι έννοιες των ΠΛ μεταφράζονται σε εκφράσεις υπαρξιακών κανόνων. Το '_' αντιπροσωπεύει την ανώνυμη μεταβλητή. Στο δεύτερο μέρος ορίζεται πως οι ρόλοι και οι αντίστροφοί τους μεταφράζονται. Στη συνέχεια, οι εκφράσεις των A-Box και T-Box παρουσιάζονται με την μετάφρασή τους σε γεγονότα, κανόνες και περιορισμούς αντίστοιχα.

2.5 Ερωτήματα με αρνητικά άτομα (Queries with negated atoms)

Ένα συζευτικό ερώτημα με αρνητικά άτομα ($\mathsf{CQ}^{\neg}$) μπορεί να αναπαρασταθεί ως:

$$q(\mathbf{x}) = a_1(\mathbf{x}_1), \ldots, a_n(\mathbf{x}_n), \neg p_1(\mathbf{y}_1), \ldots, \neg p_m(\mathbf{y}_m)$$

όπου η άρνηση χρησιμοποιεί σημασιολογία *Ανοικτού Κόσμου* (Open-World semantics), δηλαδή ένα άτομο $\neg p_i(\mathbf{t}_i)$ προέρχεται από την οντολογία αν και μόνο αν για όλες τις ερμηνείες $\mathcal{I}$ των ισχυρισμών και των δεδομένων, ισχύει ότι $p_i(\mathbf{t}_i) \notin \mathcal{I}$. Αυτή η έννοια βρίσκεται πιο κοντά στην έκφραση 'δε μπορεί', δηλαδή το ερώτημα:

$$q = Person(X), Person(Y), \neg married(X, Y)$$

διαβάζεται ως: ' υπάρχει κάποιος άνθρωπος που να μη μπορεί να παντρευτεί κάποιον άλλο άνθρωπο·'. Κάποιος μπορεί επίσης να ενδιαφέρεται για το ερώτημα 'άνθρωποι που δεν μπορούν να παντρευτούν':

$$q(X) = Person(X), \neg married(X, Y).$$

Όταν έχουμε περιορισμούς σε μία βάση δεδομένων, μπορούμε εύκολα να μετατρέψουμε το πρόβλημα $\mathcal{D}, \mathcal{R}, \mathcal{C} \models q$ στον έλεγχο συνέπειας του $\mathcal{D}, \mathcal{R}, \mathcal{C}, \neg q$. Με αυτό τον τρόπο έχουμε ότι:

$$\mathcal{D}, \mathcal{R}, \mathcal{C} \models q \text{ εάν και μόνο εάν } \mathcal{D}, \mathcal{R}, \mathcal{C}, \neg q \models \bot$$

Ένα $\mathsf{CQ}^{\neg}$ βασίζεται σε *ασφαλή* άρνηση ($\mathsf{CQ}^{\neg s}$) εάν όλες οι μεταβλητές που υπάρχουν στα αρνητικά άτομα, είναι επίσης παρούσες σε θετικά άτομα του ερωτήματος. Το πρόβλημα συνεπαγωγής για $\mathsf{CQ}^{\neg s}$ δεν είναι αποφάνσιμο, ακόμα και με μια βάση γνώσης $\mathsf{DL-Lite}^{\mathcal{H}}_{\mathsf{core}}$ [9]. Τα συνδετικά ερωτήματα με προστατευμένα αρνητικά άτομα (guarded negated atoms $\mathsf{CQ}^{\neg g}$) έχουν ένα θετικό άτομο, στο οποίο περιέχονται όλες οι μεταβλητές που παίρνουν μέρος στα αρνητικά άτομα του ερωτήματος.

Τα περισσότερα από τα συστήματα που απαντούν τα $\mathsf{CQ}^{\neg}$ βασίζονται στους αλγορίθμους tableau, οι οποίοι δεν αποδίδουν καλά σε οντολογίες με μεγάλο αριθμό ισχυρισμών. Παρόλα αυτά, ο E. Matos πρότεινε έναν αλγόριθμο επαναγραφής για $\mathsf{CQ}^{\neg}$ βασισμένο σε ανάλυση λογικής πρώτης τάξης και σε επαναγραφή των ερωτημάτων. Ο αλγόριθμος είναι ολοκληρωμένος για μερικούς τύπους των $\mathsf{CQ}^{\neg}$.

Η άρνηση ενός υπαρξιακού κανόνα r μπορεί να εκφραστεί ως ένα $\mathsf{CQ}^{\neg}$ με τα άτομα του σώματος του κανόνα ακολουθούμενα από το άτομο στην κεφαλή του κανόνα σε αρνητική μορφή:

$$q_r(\mathbf{X}) = a_1(\mathbf{x}_1), \ldots, a_n(\mathbf{x}_n), \neg h(\mathbf{y}) \ .$$

Το ερώτημα $q_r(\mathbf{X})$ αντιπροσωπεύει τα αντιπαραδείγματα του κανόνα r, δηλαδή τα ζεύγη για τα οποία ο κανόνας δεν ισχύει.

2.6 Μηχανική Μάθηση

Πολλοί διαφορετικοί τύποι αλγορίθμων μηχανικής μάθησης χρησιμοποιούνται για να ανακαλύψουν μοτίβα σε μεγάλα δεδομένα που οδηγούν σε πρακτικές γνώσεις. Σε υψηλό επίπεδο, αυτοί οι διαφορετικοί αλγόριθμοι μπορούν να ταξινομηθούν σε δύο ομάδες βάσει του τρόπου με τον οποίο 'μαθαίνουν' τα δεδομένα για να κάνουν προβλέψεις: μάθηση με επίβλεψη και χωρίς επίβλεψη.

2.6.1 Μάθηση με επίβλεψη

Η μάθηση με επίβλεψη είναι η πιο συχνά χρησιμοποιούμενη μεταξύ των δύο. Σε αυτή, ένα σύνολο δεδομένων εκπαίδευσης εισόδου και οι σχετικές έξοδοι δίνονται στο πρόγραμμα έτσι ώστε να του διδάξουν πως μια συγκεκριμένη είσοδος οδηγεί σε μια συγκεκριμένη έξοδο. Το πρόγραμμα αναλύει τα δεδομένα εκπαίδευσης και παράγει μια συνάρτηση συνεπαγωγής, η οποία μπορεί αργότερα να εφαρμοστεί σε νέα δεδομένα, προκειμένου να χαρτογραφήσει νέες περιπτώσεις.

Καλείται μάθηση με επίβλεψη διότι η διαδικασία ενός αλγορίθμου που μαθαίνει από τα δεδομένα εκπαίδευσης, μπορεί να θεωρηθεί ως δάσκαλος που επιβλέπει τη διαδικασία μάθησης. Γνωρίζουμε τις σωστές απαντήσεις, ο αλγόριθμος κάνει επανειλημμένα προβλέψεις για τα δεδομένα εκπαίδευσης και διορθώνεται από τον δάσκαλο. Η μάθηση σταματά όταν ο αλγόριθμος επιτύχει ένα αποδεκτό επίπεδο απόδοσης. Για παράδειγμα, ένας αλγόριθμος ταξινόμησης θα μάθει να αναγνωρίζει τα ζώα αφού εκπαιδευτεί σε ένα σύνολο δεδομένων εικόνων που φέρουν σωστή επισήμανση με τα είδη του ζώου και κάποια χαρακτηριστικά ταυτοποίησης.

2.6.2 Μάθηση χωρίς επίβλεψη

Από την άλλη, στη μάθηση χωρίς επίβλεψη το πρόγραμμα δεν έχει κάποιο στοιχείο σχετικά με το πώς θα πρέπει να μοιάζει η επιθυμητή έξοδος, καθώς δεν δίνεται σύνολο δεδομένων εκπαίδευσης, και επομένως ψάχνει για συσχετίσεις ανάμεσα στα δεδομένα ώστε να αποκαλύψει μια κρυμμένη δομή σε μη επισημασμένα δεδομένα.

Καλείται μάθηση χωρίς επίβλεψη διότι αντίθετα από τη μάθηση με επίβλεψη δεν υπάρχουν σωστές απαντήσεις και δεν υπάρχει δάσκαλος. Οι αλγόριθμοι αφήνονται ώστε να ανακαλύψουν μόνοι τους και να παρουσιάσουν την ενδιαφέρουσα δομή στα δεδομένα.

2.6.3 Κανόνες Συσχέτισης

Η εκμάθηση κανόνων συσχέτισης είναι μια διαδικασία ανακάλυψης σχέσεων μεταξύ μεταβλητών σε ένα σύνολο δεδομένων. Εάν ένα σύνολο δεδομένων αποτελείται από n διαφορετικά χαρακτηριστικά/στοιχεία I = { i_1 , i_2 , ... , i_n } και m διαφορετικές καταγραφές/συναλλαγές T = { t_1 , t_2 , ... , t_n }, όπου κάθε καταγραφή υποδηλώνει με αληθές ή ψευδές εάν περιλαμβάνει κάθε στοιχείο και, ως αποτέλεσμα, κάθε συναλλαγή στο T έχει ένα μοναδικό ID και περιλαμβάνει ένα υποσύνολο των στοιχείων του I, τότε ένας κανόνας υποδεικνύει πως

$$X \Rightarrow \Upsilon$$

όπου $X, \Upsilon \subseteq I$ και $X \cap \Upsilon = \varnothing$. Αυτό σημαίνει πως εάν κάθε χαρακτηριστικό που αντιστοιχεί σε ένα στοιχείο στο σύνολο X (antecedent) είναι αληθές σε μία συναλλαγή, τότε τα χαρακτηριστικά που αντιστοιχούν στα στοιχεία του συνόλου Υ (consequent) είναι εξίσου αληθή.

Η εξαγωγή ικανοποιητικών κανόνων συσχέτισης εξαρτάται κυρίως από την εφαρμογή ενός ελαχίστου ορίου στήριξης (minimum support threshold) για την εύρεση όλων των συχνών συνόλων στοιχείων σε ένα σύνολο δεδομένων και ένα ελάχιστο όριο εμπιστοσύνης (minimum confidence constraint) για τον σχηματισμό κανόνων στα συχνά σύνολα στοιχείων. Η απόδοση ενός κανόνα μετράται με βάση τις παρακάτω μετρικές (metrics):

- Support
- Confidence
- Lift
- Conviction

Support ενός συνόλου στοιχείων X (supp(X)) σε σχέση με ένα σύνολο συναλλαγών T, είναι το ποσοστό των συναλλαγών στο σύνολο δεδομένων που περιέχουν το σύνολο στοιχείων X.

Confidence ενός κανόνα X⇒ Υ (conf(X⇒ Y)) σε σχέση με ένα σύνολο συναλλαγών T, είναι το ποσοστό των συναλλαγών που περιέχουν τα X και Υ ταυτόχρονα. Υπολογίζεται ως ακολούθως:

$$\text{conf(X}\Rightarrow \text{Y)} = \frac{supp(X \cup Y)}{supp(X)}$$

Lift ενός κανόνα X⇒ Υ (lift(X⇒ Y)) είναι η αναλογία του support που παρατηρήθηκε προς του προσδοκώμενου εάν τα X και Υ ήταν ανεξάρτητα:

$$\text{lift(X}\Rightarrow \text{Y)} = \frac{supp(X \cup Y)}{supp(X) \times supp(Y)}$$

Τέλος, *conviction* ενός κανόνα X⇒ Υ ορίζεται ως ακολούθως:

$$\text{conv(X}\Rightarrow \text{Y)} = \frac{1 - supp(Y)}{1 - conf(X \Rightarrow Y)}$$

Στην παρούσα διπλωματική γίνεται χρήση μίας τροποποιημένης έκδοσης του αλγορίθμου εκμάθησης κανόνων συσχέτισης Apriori. Ο αλγόριθμος αυτός αναγνωρίζει τα αντικείμενα που εμφανίζονται συχνά σε ένα σύνολο δεδομένων και τα επεκτείνει σε όλο και μεγαλύτερα σύνολα στοιχείων μέχρις ότου τα σύνολα που προκύπτουν να εμφανίζονται αρκετά συχνά στα δεδομένα συναλλαγών με βάση το όριο της εμπιστοσύνης που έχουμε θέσει. Η τροποποίηση που έχουμε εισάγει στον συγκεκριμένο αλγόριθμο, είναι ότι ο αλγόριθμος σταματάει να επεκτείνει τα σύνολα σε μεγαλύτερα όταν φτάσει σε ένα μέγιστο μήκος για τα στοιχειοσύνολα το οποίο έχει καθορίσει ο χρήστης.

2.7 Το εργαλείο Weka

Το Weka (Waikato Environment for Knowledge Analysis) είναι ένα πρόγραμμα ανάπτυξης εφαρμογών μηχανικής μάθησης και εξόρυξης γνώσης από δεδομένα (data mining), το οποίο αναπτύχθηκε στο τμήμα Επιστήμης Υπολογιστών του Waikato της Νέας Ζηλανδίας [10]. Πρόκειται για πακέτο λογισμικού ανοιχτού κώδικα υλοποιημένο σε Java, και χρησιμοποιείται ευρέως τόσο για ερευνητικούς και εκπαιδευτικούς λόγους όσο και για εφαρμογές που σχετίζονται με τον τομέα της εξόρυξης δεδομένων.

Το Weka περιλαμβάνει εργαλεία για προεπεξεργασία δεδομένων, κατηγοριοποίηση, παλινδρόμηση, συσταδοποίηση, κανόνες συσχέτισης και απεικόνιση, ενώ είναι κατάλληλο και για την ανάπτυξη νέων σχημάτων μηχανικής εκμάθησης.

Το Weka δέχεται ως είσοδο αρχεία τύπου ARFF (Attribute-Relation File Format), τα οποία είναι απλά αρχεία κειμένου, όπου περιέχουν σειρές από στιγμιότυπα (instances) κάποιων χαρακτηριστικών (attributes). Ειδικότερα ένα αρχείο τύπου ARFF αποτελείται από δύο μέρη: α) την περιοχή της επικεφαλίδας, όπου περιγράφονται όλα τα χαρακτηριστικά που χρησιμοποιούνται (πχ μεταβλητές ή ιδιότητες σε ένα πρόβλημα) αλλά και ο τύπος δεδομένων τους και β) την περιοχή των δεδομένων, όπου κάθε παράδειγμα του συνόλου δεδομένων αντιστοιχεί σε μια γραμμή με τα χαρακτηριστικά ταξινομημένα σύμφωνα με την προκαθορισμένη σειρά και διαχωρισμένα με κόμμα.

Πιο συγκεκριμένα, η περιοχή της επικεφαλίδας περιλαμβάνει την έκφραση @relation η οποία δεν μπορεί να παραληφθεί και περιγράφει το όνομα του αρχείου. Στην συνέχεια, ακολουθεί η δήλωση όλων των χαρακτηριστικών που περιγράφουν το συγκεκριμένο σύνολο παραδειγμάτων. Η δήλωση γίνεται χρησιμοποιώντας την παρακάτω σύμβαση:

@attribute attributeName datatype,

όπου attributeName είναι το όνομα του χαρακτηριστικού και datatype καθορίζει τον τύπο του χαρακτηριστικού.

Το Weka υποστηρίζει 4 διαφορετικούς τύπους δεδομένων:

- Αριθμητικά (numeric), τα οποία μπορεί να είναι είτε πραγματικοί είτε ακέραιοι αριθμοί
- Ονομαστικά (nominal), δεδομένα που δηλώνουν μία συγκεκριμένη λίστα με όλες τις πιθανές τιμές στον ορισμό τους
- Αλφαριθμητικά (string), τα οποία επιτρέπουν τη δημιουργία αυθαίρετων αλφαριθμητικών δομών
- Ημερομηνίες (dates), όπου περιέχουν τιμές ημερομηνίας και αν ορισθεί με συγκεκριμένη μορφοποίηση

Στην περιοχή των δεδομένων, υπάρχει μια γραμμή δήλωσης των δεδομένων η οποία σηματοδοτεί την έναρξη του συνόλου των δεδομένων και έχει τη μορφή @data. Έπειτα ακολουθούν τα δεδομένα ταξινομημένα σύμφωνα με την σειρά που έχει καθοριστεί στην επικεφαλίδα, διαχωρισμένα με κόμμα. Εάν κάποια τιμή απουσιάζει, τότε εκπροσωπείται από ένα μόνο ερωτηματικό (Missing Value - ?).

2.8 Completo Software

Το σύστημα Completo [1] υλοποιήθηκε αρχικά χρησιμοποιώντας το σύστημα Rapid [18] ως εξωτερικό επανεγραφέα (rewriter) και μία σύνδεση σε μία βάση δεδομένων (MySQL) για την αποτελεσματική ανάκτηση στιγμιοτύπων. Ένα TBox χρησιμοποιείται για την επαναγραφή των αρχικών περιορισμών στο σύστημα. Στη συνέχεια,

οι ισχυρισμοί που θα μπορούσαν να κωδικοποιηθούν αρχικά σε μορφή OWL μεταφράζονται σε μία βάση δεδομένων. Τέλος, οι διαδικασίες επαναγραφής και ανάκτησης των παραδειγμάτων μπορούν να πραγματοποιηθούν με τη χρήση της επαναγραφής των περιορισμών, της βάσης δεδομένων και των ερωτημάτων που πρέπει να επαναγραφούν και να απαντηθούν.

Η αναβαθμισμένη έκδοση του Completo χρησιμοποιεί ως εξωτερικό επανεγραφέα το σύστημα Graal [5], το οποίο έχει τη δυνατότητα να επαναγράφει συζευτικά ερωτήματα χρησιμοποιώντας υπαρξιακούς κανόνες. Επιπλέον, για την αναπαράσταση των ABox ισχυρισμών της οντολογίας, αντί για βάση δεδομένων MySQL, χρησιμοποιείται η βάση δεδομένων H2.

Στην παρούσα διπλωματική εργασία, χρησιμοποιείται η αναβαθμισμένη έκδοση του Completo, καθώς χρησιμοποιούμε υπαρξιακούς κανόνες.

Το Completo δέχεται ως είσοδο ένα αρχείο με ερωτήματα (queries) τα οποία είναι της μορφής:

$$\text{Q(?X)} \leftarrow \text{Person(?X)}, \text{–married(?X)}.$$

Κεφάλαιο 3

Υλοποίηση της μεθόδου

Σε αυτό το κεφάλαιο, παρουσιάζεται το προτεινόμενο μοντέλο της παρούσας διπλωματικής εργασίας.

3.1 Θεωρητική προσέγγιση

Το σύνολο των ισχυρισμών μίας οντολογίας μπορεί να μεταφραστεί σε ένα σύνολο δεδομένων όπου οι συναλλαγές αντιστοιχούν στις σταθερές του A-Box και τα αντικείμενα σε όλες τις πιθανές έννοιες που μπορούμε να συναντήσουμε στην οντολογία, δηλαδή ατομικές έννοιες A, ή έννοιες της μορφής $\exists R$, είτε έννοιες της μορφής $\exists R.D$, όπου το D αποτελεί ατομική έννοια. Στην παρούσα διπλωματική εργασία, εστιάζουμε στις πιθανές έννοιες $A, \exists R$ και $\exists R^-$ για όλες τις πιθανές κλάσεις A, τους ρόλους R και τους αντίστροφους ρόλους R^- που υπάρχουν στην οντολογία. Με αυτό το σύνολο δεδομένων μπορούμε να παράγουμε Κανόνες Συσχέτισης και να ανακαλύψουμε πιθανούς κανόνες που περιγράφουν κρυμμένες σχέσεις στα υπάρχοντα δεδομένα. Η μεταφορά από οντολογία σε σύνολο δεδομένων Weka υλοποιήθηκε μέσω ενός άλλου συστήματος [21] το οποίο χρησιμοποιεί το Weka, προκειμένου να εφαρμόσει αλγορίθμους μηχανικής μάθησης σε οντολογίες.

Φυσικά, για να παράγουμε τους κανόνες συσχέτισης χρησιμοποιούμε το εργαλείο Weka [10]. Οι κανόνες που παίρνουμε από το Weka έχουν την ακόλουθη μορφή:

$$l_1, \ldots, l_n \rightarrow l_{n+1}, \ldots, l_m$$

όπου το l_i αναφέρεται σε ένα αντικείμενο C_i που είτε είναι παρόν στα δεδομένα ($C_i = true$) είτε όχι ($C_i = false$). Η απουσία του αντικειμένου ($C_i = false$) σε μία συναλλαγή α αντιπροσωπεύει την απουσία του ισχυρισμού $C_i(\alpha)$ στην οντολογία. Όμως, αυτό δεν είναι αρκετό για να θεωρήσουμε ότι το $\neg C_i(\alpha)$ ισχύει στην παρούσα οντολογία. Για να αποφύγουμε αυτή τη σύγχυση με τους διαφορετικούς τύπους άρνησης, εστιάζουμε μόνο στους κανόνες που ασχολούνται με την παρουσία των αντικειμένων

($C_i = true$). Επιπλέον, οι κανόνες που παράγονται έχουν μόνο ένα στοιχείο στη συνέπεια ($m = n + 1$). Για να αποφύγουμε την εξερεύνηση άλλων τύπων κανόνων, τροποποιήσαμε τον αλγόριθμο Apriori που προσφέρει το Weka. Με αυτό τον τρόπο, μπορούμε να ελέγξουμε το μέγεθος στο σώμα των κανόνων.

Οι νέοι κανόνες που παράγει το εργαλείο Weka μπορούν να προστεθούν στην οντολογία, ώστε να εμπλουτιστεί η παρούσα γνώση. Γενικότερα, για έναν κανόνα συσχέτισης:

$$C_1 = true, \ldots, C_n = true \rightarrow C_{n+1} = true$$

το αντίστοιχο αξίωμα που θα προστεθεί στην οντολογία είναι:

$$r = C_1 \sqcap \ldots \sqcap C_n \sqsubseteq C_{n+1}$$

Όμως, ο νέος αυτός κανόνας μπορεί να καταστήσει την οντολογία ασυνεπή ανεξάρτητα με την τιμή της εμπιστοσύνης (confidence) του αντίστοιχου κανόνα συσχέτισης. Το ερώτημα

$$q_r(X) = C_1^X, \ldots, C_n^X, \neg C_{n+1}^X$$

της άρνησης του κανόνα που πρόκειται να προστεθεί, ορίζει αν ο κανόνας είναι ασφαλής για την συγκεκριμένη οντολογία. Ένας κανόνας r είναι *συνεπής (consistent)* σε σχέση με την οντολογία $\mathcal{K}$ αν και μόνο αν $ans(q_r(X)) = \emptyset$. Διαφορετικά, ο r λέγεται *ασυνεπής (inconsistent)*. Μόνο οι συνεπείς κανόνες μπορούν να προστεθούν τελικά στην οντολογία.

Σχετικά με τα UCQ-rewritings ενός ερωτήματος $q_r(X)$, μπορούμε να θεωρήσουμε μία μετρική $f_m(q_r(X))$ ώστε να κατατάξουμε τους υποψήφιους κανόνες. Όσο υψηλότερη είναι η τιμή της μετρικής $f_m(q_r(X))$, τόσο καλύτερος είναι και ο υποψήφιος κανόνας r. Η τιμή της μετρικής είναι ευθέως ανάλογη με την τιμή της εμπιστοσύνης του αντίστοιχου κανόνα συσχέτισης. Με αυτόν τον τρόπο, δίνεται προτεραιότητα στους κανόνες που έχουν καλύτερη τιμή εμπιστοσύνης. Προκειμένου να συμπεριληφθούν οι επαναγραφές της άρνησης του αντίστοιχου κανόνα, και το σχήμα που έχουν, εισάγουμε πιθανότητες για τις έννοιες που παίρνουν μέρος στις επαναγραφές αυτές. Η πιθανότητα να ανήκει ένα άτομο σε μία έννοια C ορίζεται ως:

$$p(C) = \frac{|\{a \mid C(a)\}|}{|D|},$$

όπου D είναι το σύνολο όλων των ατόμων που ανήκουν στην οντολογία. Με την παραδοχή ότι οι ισχυρισμοί της υπάρχουσας οντολογίας είναι αντιπροσωπευτικοί για κάθε άλλη πιθανή εκδοχή της οντολογίας, η τιμή $p(C)$ είναι ένας καλός δείκτης για την πιθανότητα να ανήκει ένα άτομο στην έννοια C. Για ένα άτομο, η πιθανότητα να αποτελεί απάντηση του ερωτήματος $q(X) = C_1^X, \ldots, C_n^X$ μπορεί να υπολογισθεί ως:

$$p(q(X)) = \Pi_{i=1}^{n} p(C_i)$$

Τέλος, η πιθανότητα να ανήκει ένα άτομο στο UCQ $q(X)$ με ερωτήματα $q_j(X)$ χρησιμοποιούμε μία εκτίμηση:

$$p(q(X)) = \sum_{j=1}^{m} p(q_j(X))$$

και με αυτό τον τρόπο αποφεύγουμε τον υπολογισμό της πιθανότητας ένα άτομο να αποτελεί απάντηση σε παραπάνω από ένα ερώτημα ταυτόχρονα, για όλα τα υποσύνολα του UCQ δηλαδή $\pi_{j=1}^{m'} p(q_{s_j}(X))$ για κάθε $\{s_1, \ldots, s_{m'}\} \subseteq \{1, \ldots, m\}$.

Τελικά, η μετρική $f_m(q_r(X))$ υπολογίζεται ως ακολούθως:

$$f_m(q_r(X)) = \frac{conf(r)}{1 + p(q_r(X))}$$

3.1.1 Βελτίωση των κανόνων

Οι κανόνες που είναι ασυνεπείς σύμφωνα με την υπάρχουσα οντολογία, δεν επιτρέπεται να προστεθούν σε αυτή. Μπορούμε λοιπόν να τους μετατρέψουμε σε συνεπείς, προσθέτοντας επιπλέον έννοιες στο σώμα τους, διότι με τη νέα αυτή προσθήκη, μπορούμε να 'φιλτράρουμε' τα αντιπαραδείγματα του κανόνα. Οι νέες έννοιες D είναι είτε ονόματα εννοιών A είτε ποσοτικά καθορισμένοι ρόλοι $\exists R$ ή $\exists R^-$. Αν ο τελικός κανόνας:

$$r' = C_1 \sqcap \ldots \sqcap C_n \sqcap D \sqsubseteq C_{n+1}$$

είναι συνεπής σε σχέση με την οντολογία, λέμε ότι ο αρχικός κανόνας r *διορθώθηκε (fixed)* εισάγοντας την έννοια D. Οι αρχικοί κανόνες μπορούν να διορθωθούν εισάγοντας πολλές διαφορετικές έννοιες και λαμβάνοντας υπόψιν την μετρική $f_m/1$ που ορίσαμε προηγουμένως, μπορούμε να επιλέξουμε τον καταλληλότερο κανόνα για να τον εισάγουμε στην οντολογία.

3.1.2 Εισαγωγή των κανόνων στην οντολογία

Όπως αναφέρθηκε, στη μετρική που υπολογίσαμε προηγουμένως συμπεριλάβαμε τις επαναγραφές της άρνησης του κανόνα. Για το λόγο αυτό, με την προσθήκη ενός κανόνα στην αρχική οντολογία, προσθέτουμε ταυτόχρονα και τις επαναγραφές αυτές, ως περιορισμούς της οντολογίας, ώστε να βεβαιωθούμε ότι ο κανόνας αυτός δε θα προκαλέσει κάποια ασυνέπεια στο μέλλον.

Κεφάλαιο 4

Υλοποίηση της εφαρμογής

Στο κεφάλαιο αυτό παρουσιάζονται τα βασικά σημεία της υλοποίησης του κώδικα που παράχθηκε ώστε να πραγματοποιηθεί το παρόν μοντέλο.

4.1 Προαπαιτούμενα

Το λογισμικό περιέχεται σε ένα αρχείο τύπου jar και εκτελείται από έναν μεταγλωττιστή java [1] μέσω της εντολής:

```
java -jar thesis.jar
```

Επιπλέον, για τη λειτουργία του η h2 database [2] πρέπει να είναι εγκατεστημένη.

Για την εκτέλεση του προγράμματος, πρέπει να έχουμε αποθηκευμένη την επιθυμητή οντολογία (αρχείο τύπου *.owl) στα αρχεία του συστήματος ώστε να παρέχουμε την απαραίτητη είσοδο στο σύστημα. Επιπροσθέτως, χρειαζόμαστε ένα αρχείο ρύθμισης (αποθηκευμένο με το όνομα sqlconfig.properties) το οποίο περιέχει ορισμένα σημαντικά δεδομένα για να επιτευχθεί η σύνδεση με τη βάση δεδομένων. Το αρχείο αυτό έχει την ακόλουθη μορφή:

```
username = [username]
password = [password]
url = jdbc:h2:~/testh2
```

4.2 Εκτέλεση του προγράμματος

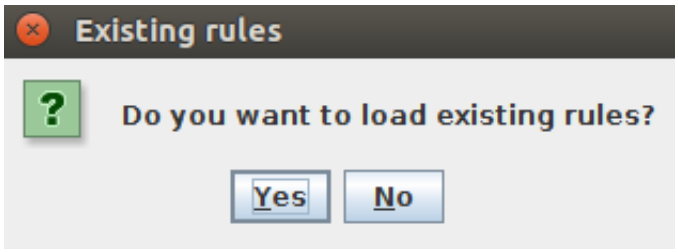
Η ενότητα αυτή παρέχει μία λεπτομερής περιγραφή του τρόπου χρήσης του συστήματος.

[1] έκδοση 1.8 ή νεότερη

[2] http://www.h2database.com/html/main.html

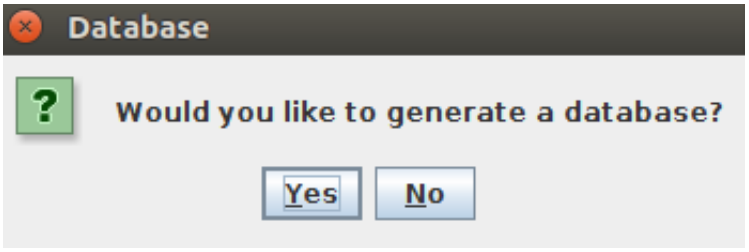
4.2.1 Φόρτωση της οντολογίας

Η πρώτη εργασία που εκτελεί το πρόγραμμα είναι η φόρτωση της επιθυμητής οντολογίας. Ένα παράθυρο εμφανίζεται για να επιλέξει ο χρήστης την οντολογία στην οποία θέλει να προσθέσει νέα αξιώματα. Το αρχείο της οντολογίας πρέπει να είναι της μορφής (*.owl).



Σχήμα 4.1: Επιλογή υπαρχόντων κανόνων

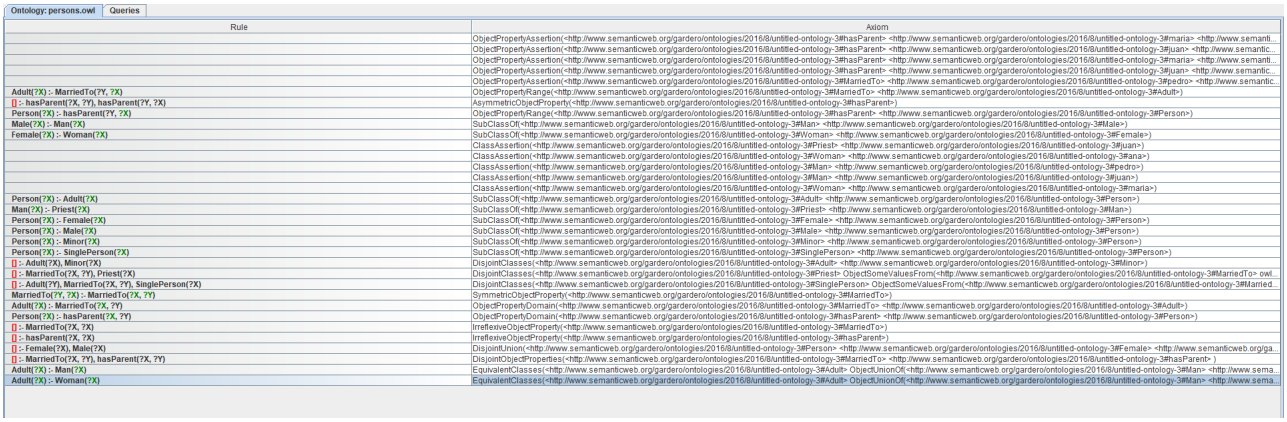
Στη συνέχεια ο χρήστης επιλέγει αν θέλει να φορτώσει υπάρχοντες κανόνες (Σχήμα 4.1). Σε αυτό το σημείο, ο χρήστης επιλέγει 'Yes', μόνο αν έχει ήδη τρέξει το πρόγραμμα μία φορά και τα απαραίτητα αρχεία έχουν δημιουργηθεί.



Σχήμα 4.2: Δημιουργία βάσης δεδομένων.

Αν ο χρήστης επιλέξει 'No' στο προηγούμενο παράθυρο, θα ερωτηθεί για το αν επιθυμεί τη δημιουργία μίας νέας βάσης δεδομένων (Σχήμα 4.2). Αν επιλέξει 'Yes', δημιουργείται μια βάση δεδομένων μέσω του συστήματος Completo. Αυτή η βάση δεδομένων χρησιμοποιείται στην διαδικασία ανάκτησης στιγμιοτύπων. Αντιθέτως αν ο χρήστης έχει ήδη δημιουργήσει μία βάση δεδομένων χειροκίνητα μέσω του συστήματος Completo, μπορεί να επιλέξει 'No'. Σε αυτή την περίπτωση, το όνομα της υπάρχουσας βάσης δεδομένων θα πρέπει να είναι ίδιο με το όνομα της οντολογίας. Για παράδειγμα αν έχουμε φορτώσει το αρχείο με όνομα *'persons.owl'* τότε το αντίστοιχο όνομα της βάσης δεδομένων πρέπει να είναι *'persons'*.

Στη συνέχεια ένας μηχανισμός εξαγωγής συμπερασμάτων εκκινείται με βάση τη δομένη οντολογία για να εξάγει κάθε ισχυρισμό που αφορά κλάσεις και ιδιότητες αντικειμένων. Για κάθε άτομο που αναφέρεται στην οντολογία, εξάγονται όλες οι κλάσεις που ανήκει το άτομο και αποθηκεύονται σε μία δομή HashMap. Ακολούθως, οι ιδιότητες αντικειμένων που συμμετέχει το συγκεκριμένο άτομο εξάγονται και αποθηκεύονται.



(α′)

```
@relation persons
@attribute Individuals string
@attribute "Male(X)" {true, false}
@attribute "Person(X)" {true, false}
@attribute "Adult(X)" {true, false}
@attribute "Priest(X)" {true, false}
@attribute "Woman(X)" {true, false}
@attribute "Female(X)" {true, false}
@attribute "Man(X)" {true, false}
@attribute "hasParent(X, Y)" {true, false}
@attribute "hasParent(Y, X)" {true, false}
@attribute "MarriedTo(X, Y)" {true, false}
@attribute "MarriedTo(Y, X)" {true, false}
@DATA
"ana", false, true, true, false, true, true, false, false, true, true, true
"juan", true, true, true, true, false, false, true, true, false, false, false
"maria", false, true, true, false, true, true, false, true, false, false, false
"pedro", true, true, true, false, false, false, true, false, true, true, true
```

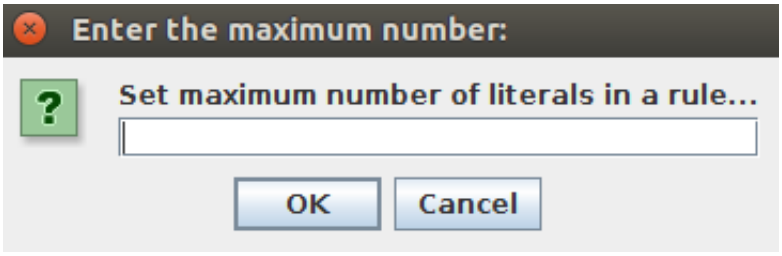
(β′)

Σχήμα 4.3: Παράδειγμα Μετατροπής Οντολογίας σε Αρχείο ARFF

Μία άλλη μέθοδος καλείται, η οποία δημιουργεί το επιθυμητό ARFF αρχείο από τη δομή HashMap που δημιουργήθηκε με τη συλλογή αντικειμένων από τη δοθείσα οντολογία. Το πρώτο attribute είναι πάντα τα άτομα (individuals) τα οποία αποθηκεύονται ως strings. Τα επόμενα attributes είναι τα ονόματα των κλάσεων (classes) και των ιδιοτήτων αντικειμένων (object properties) που αποθηκεύονται ως nominal attributes με τις πιθανές τιμές 'true' και 'false'. Η παραπάνω διαδικασία ακολουθείται για τη δημιουργία της επικεφαλίδας (header) του αρχείου ARFF. Το κομμάτι των δεδομένων του αρχείου δημιουργείται όπως ακολούθως: Αρχικά, το όνομα του ατόμου που αναφέρεται εισάγεται ως τιμή για το attribute *Individuals*. Στη συνέχεια, για κάθε attribute ερευνάται η δομή HashMap για το κατά πόσο το άτομο ανήκει στην κλάση ή συμμετέχει στον συγκεκριμένο ισχυρισμό ρόλου, και ανάλογα με τα ευρήματα, η τιμή ορίζεται ως 'true' ή 'false'.

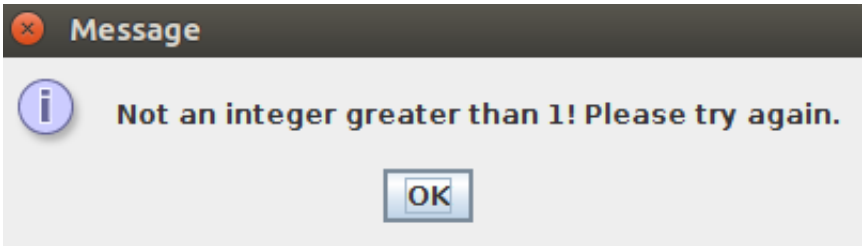
Το σχήμα 4.3 είναι ένα παράδειγμα μετατροπής της οντολογίας persons.owl στο αρχείο ARFF που δημιουργείται ώστε να παρουσιαστεί η μορφή των attributes στην επικεφαλίδα καθώς και το πως κατασκευάζονται οι αντίστοιχες τιμές στην ενότητα των

δεδομένων.



Σχήμα 4.4: Επιλογή μέγιστου αριθμού λεκτικών.

Στη συνέχεια αφού αποθηκευθεί το αρχείο ARFF, και φορτωθεί στο σύστημα Weka, εμφανίζεται ένα μήνυμα όπου ζητάει από το χρήστη να εισάγει τον μέγιστο αριθμό των λεκτικών στους αρχικούς κανόνες (εικόνα 4.4). Για παράδειγμα, αν ο χρήστης εισάγει τον αριθμό '3', οι αρχικοί κανόνες θα είναι της μορφής *A AND B* ⇒ *C*, όπου τα A, B, C αντιστοιχούν σε attributes. Καθώς κάθε κανόνας περιέχει σίγουρα ένα χαρακτηριστικό στο σώμα (body) και ένα στην κεφαλή (head) του κανόνα, ο αριθμός που θα εισάγει ο χρήστης πρέπει να είναι ακέραιος και μεγαλύτερος του 2.



Σχήμα 4.5: Εισαγωγή λάθος τιμής.

Σε περίπτωση που δεν ικανοποιείται η παραπάνω προϋπόθεση, ο χρήστης ενημερώνεται ότι έχει επιλέξει λανθασμένη τιμή (εικόνα 4.5) και το προηγούμενο μήνυμα επανεμφανίζεται για την εισαγωγή νέας τιμής.

Ως τελικό βήμα, καλείται μία τροποποιημένη έκδοση του αλγορίθμου Apriori από το σύστημα Weka με σκοπό να παραχθούν οι αρχικοί μας κανόνες. Ο αλγόριθμος Apriori μπορεί να λειτουργήσει μόνο με nominal attributes, επομένως για την παραγωγή κανόνων συσχέτισης και την αποφυγή προβλημάτων, το attribute Individuals με τα ονόματα των ατόμων διαγράφεται για κάθε στιγμιότυπο. Σε αυτό το σημείο, κάθε attribute διερευνάται και στην περίπτωση που όλες οι τιμές του είναι ίδιες για όλα τα στιγμιότυπα (instances), τότε το attribute διαγράφεται καθώς δεν έχει τίποτα να προσφέρει στην διαδικασία εξαγωγής κανόνων. Ταυτόχρονα διαγράφεται και από την δομή HashMap που είναι αποθηκευμένο.

Όπως έχει αναφερθεί στο κεφάλαιο 2, η τροποποίηση που εισάγαμε στον αρχικό αλγόριθμο, σχετίζεται με τη διακοπή του ελέγχου για νέα σύνολο στοιχείων αν το

μήκος τους ξεπερνάει το μέγιστο αριθμό που εισήγαγε ο χρήστης.

Επίσης ως μετρική για την εξαγωγή των κανόνων από το σύστημα WEKA έχει οριστεί η confidence και η ελάχιστη τιμή της έχει τεθεί σε 0.

Τέλος, οι συσχετισμοί οικοδομούνται και παράγονται κανόνες οι οποίοι αποθηκεύονται σε μία δομή HashMap μαζί με την αντίστοιχη τιμή confidence τους. Στη σπάνια περίπτωση που δεν παράγεται κανένας κανόνας επειδή δεν υπάρχουν μεγάλα σύνολα στοιχείων (itemsets) στα δεδομένα ο χρήστης ενημερώνεται με ένα παράθυρο και το σύστημα τερματίζει.

Στη συνέχεια, καλείται μία άλλη μέθοδος, η οποία μετατρέπει τους κανόνες στα αντίστοιχα ερωτήματα που χρειαζόμαστε ως είσοδο για το σύστημα COMPLETO. Για παράδειγμα, αν έχουμε τον ακόλουθο κανόνα:

hasPart(X, Y) ->City(X)

θα μεταφραστεί σε:

Q(?X) <– hasPart(?X, ?Y), –City(?X).

Η άρνηση στην κεφαλή του κανόνα χρησιμοποιείται ώστε να λάβουμε ως απαντήσεις τα αντιπαραδείγματα του. Όλα τα ερωτήματα λοιπόν, γράφονται σε ένα αρχείο και το σύστημα COMPLETO καλείται για κάθε ένα από αυτά. Επίσης σε μία δομή HashMap εισάγονται όλοι οι κανόνες μαζί με την τιμή της εμπιστοσύνης που τους αντιστοιχεί.

Αφού λάβουμε τα αποτελέσματα που προκύπτουν μετά την κλήση του COMPLETO, δημιουργείται ένας πίνακας που περιέχει τους κανόνες, την τιμή εμπιστοσύνης τους, το μέγεθος των αντιπαραδειγμάτων τους και το μέγεθος των επαναγραφών. Επιπλέον, δημιουργείται μία δομή HashMap η οποία περιέχει τους κανόνες μαζί με όλα τα αξιώματα που θα πρέπει να γράψουμε μαζί με αυτούς στην νέα οντολογία.

Ταυτόχρονα δημιουργείται ένα αρχείο csv με τα ίδια χαρακτηριστικά, καθώς και ένα αρχείο που περιέχει όλους τους κανόνες με τα αξιώματα που πρέπει να επαναγραφούν στην οντολογία σε περίπτωση εισαγωγής του συγκεκριμένου κανόνα. Τα δύο αυτά αρχεία χρησιμοποιούνται σε περίπτωση που ο χρήστης θέλει να εκτελέσει το πρόγραμμα και άλλη φορά με την ίδια οντολογία. Το όνομα του πρώτου αρχείου εξαρτάται από τον αριθμό που έχει εισάγει ο χρήστης για τα λεκτικά (αφού με διαφορετικούς αριθμούς παράγονται και διαφορετικοί κανόνες) και φυσικά από το όνομα της οντολογίας. Για παράδειγμα, αν η οντολογία ονομάζεται 'travel' και ο χρήστης έχει εισάγει τον αριθμό 2 τότε το αρχείο αποθηκεύεται με το όνομα 'rules2_travel.csv'. Αντίστοιχα, το αρχείο με τις επαναγραφές (rewritings) θα αποθηκευτεί ως 'rewr2_travel.txt'.

Στην περίπτωση όπου ο χρήστης επιθυμεί να φορτώσει την ίδια οντολογία στο πρόγραμμα σε επόμενη χρήση, μπορεί να φορτώσει τα δύο αυτά αρχεία, προκειμένου να αποφευχθεί η διαδικασία εξαγωγής νέων κανόνων, καθώς είναι χρονοβόρα, και να δημιουργηθεί απευθείας ο πίνακας με τους αρχικούς κανόνες και τα χαρακτηριστικά τους. Ένα παράδειγμα του αρχείου που περιέχει τους αρχικούς κανόνες παρουσιάζεται

```
"rule";"confidence";"cexamples";"rwcexamples"
"Woman(X) => Female(X)";1.0;0;1
"Female(X) => Woman(X)";1.0;0;1
"Priest(X) => Male(X)";1.0;0;0
"MarriedTo(X, Y) => hasParent(Y, X)";1.0;0;0
"Man(X) => Male(X)";1.0;0;0
"Priest(X) => hasParent(X, Y)";1.0;0;0
"Priest(X) => Man(X)";1.0;0;0
"hasParent(Y, X) => MarriedTo(Y, X)";1.0;0;1
"MarriedTo(X, Y) => MarriedTo(Y, X)";1.0;0;1
"MarriedTo(Y, X) => MarriedTo(X, Y)";1.0;0;4
"hasParent(Y, X) => MarriedTo(X, Y)";1.0;1;3
"MarriedTo(Y, X) => hasParent(Y, X)";1.0;0;0
"Male(X) => Man(X)";1.0;0;1
"MarriedTo(X, Y) => Man(X)";0.5;1;4
"hasParent(X, Y) => Male(X)";0.5;1;3
"hasParent(Y, X) => Female(X)";0.5;1;3
"MarriedTo(Y, X) => Man(X)";0.5;1;5
```

Σχήμα 4.6: Παράδειγμα αρχείου που περιέχει τους αρχικούς κανόνες.

στην εικόνα 4.6

4.2.2 Διόρθωση των αρχικών κανόνων

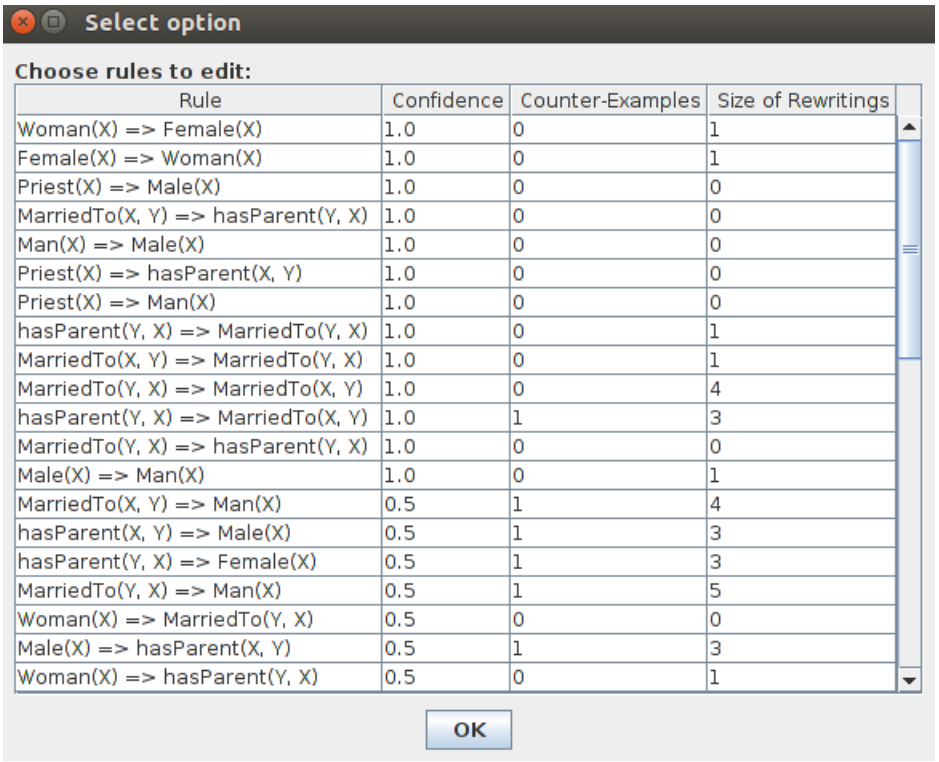
Σε αυτό το σημείο, οι αρχικοί κανόνες που παράχθηκαν από τον εξαγωγέα συσχετισμών παρουσιάζονται στον χρήστη, δηλαδή εμφανίζεται ο πίνακας που δημιουργήθηκε προηγουμένως, και ο χρήστης έχει τη δυνατότητα να επιλέξει ποιους κανόνες επιθυμεί να διορθώσει (Σχήμα 4.7). Αν ένας επιλεγμένος κανόνας δεν έχει κανένα αντιπαράδειγμα, τότε εισάγεται σε μία λίστα με τους τελικούς κανόνες που μπορούν να εισαχθούν στην οντολογία χωρίς να την καταστήσουν ασυνεπής. Αντιθέτως, αν ένας κανόνας έχει τουλάχιστον ένα αντιπαράδειγμα σύμφωνα με το σύστημα Completo, τότε εισάγεται σε μία άλλη λίστα με όλους τους κανόνες που χρειάζονται διόρθωση και έχει επιλέξει ο χρήστης.

Στη συνέχεια, εμφανίζεται ένα παράθυρο ζητώντας από τον χρήστη να επιλέξει τον τύπο του χαρακτηριστικού που θα προστεθεί στον κανόνα, δηλαδή αν θα είναι κλάση ή ιδιότητα (Σχήμα 4.8). Έπειτα, μία μέθοδος που προσθέτει όλα τα πιθανά χαρακτηριστικά καλείται και οι νέοι κανόνες δημιουργούνται. Εδώ πρέπει να τονίσουμε ότι, για την αποφυγή δημιουργίας 'άχρηστων' κανόνων, οι νέοι κανόνες που προκύπτουν θα πρέπει να ικανοποιούνται από τουλάχιστον ένα στιγμιότυπο. Δηλαδή αν δημιουργήσουμε τον παρακάτω κανόνα:

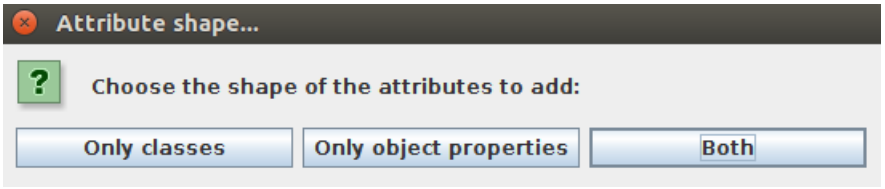
$$A\ AND\ B \Rightarrow C$$

τουλάχιστον ένα άτομο θα πρέπει ανήκει ή να παίρνει μέρος (αναλόγως αν είναι κλάση ή ιδιότητα) στα χαρακτηριστικά A,B,C ταυτόχρονα.

Το σύστημα, μετατρέπει τους νέους κανόνες σε ερωτήματα, με τον ίδιο τρόπο όπως και προηγουμένως και το σύστημα Completo καλείται ξανά. Όταν το Completo



Σχήμα 4.7: Εμφάνιση των αρχικών κανόνων στον χρήστη.



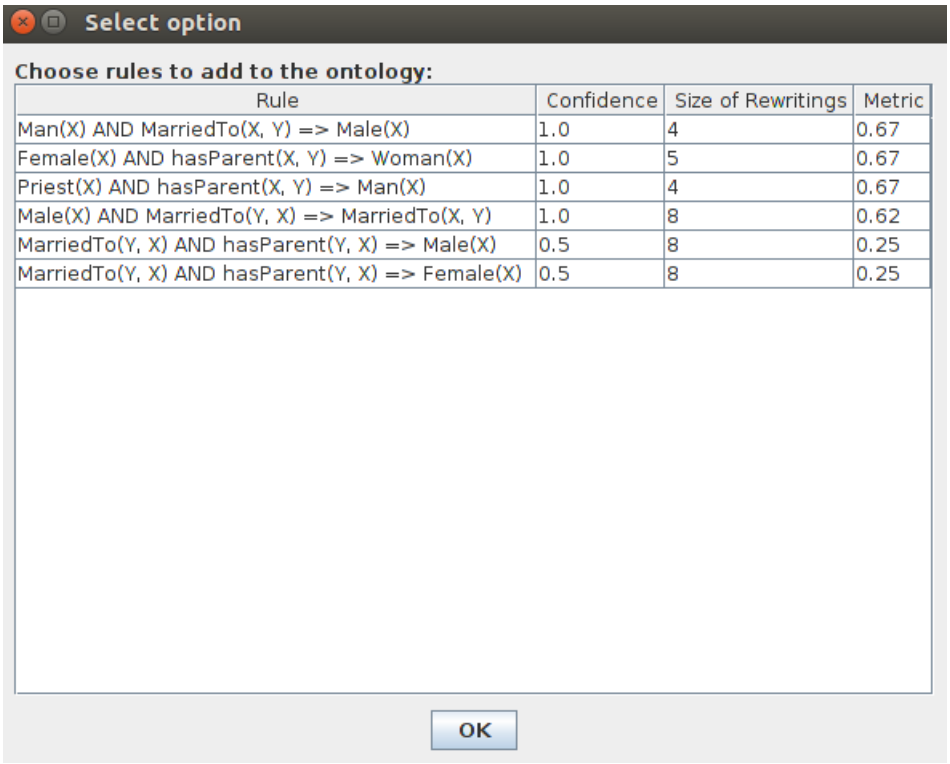
Σχήμα 4.8: Επιλογή τύπου του νέου attribute.

ολοκληρωθεί και εξάγει τα απαραίτητα αρχεία, τότε το σύστημα κρατάει μόνο τους κανόνες που δεν έχουν κανένα αντιπαράδειγμα και τους εισάγει στη λίστα με τους τελικούς κανόνες που μπορούν να προστεθούν στην οντολογία. Επίσης οι επαναγραφές των κανόνων αυτών προσθέτονται στην αντίστοιχη HashMap δομή.

4.2.3 Τελικοί κανόνες

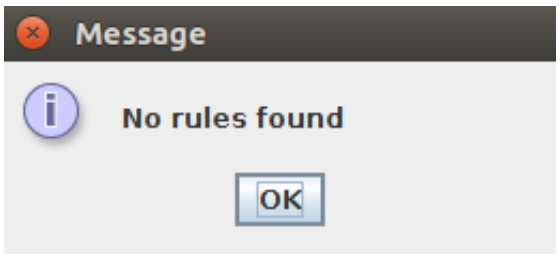
Με τον ίδιο τρόπο όπως και πριν, κατασκευάζεται ένας πίνακας που περιέχει τους κανόνες, την τιμή confidence τους και το μέγεθος των επαναγραφών για κάθε κανόνα. Όμως, αντί για το μέγεθος των αντιπαραδειγμάτων, μία μετρική που ορίσαμε προηγουμένως (3.1) εμφανίζεται στον χρήστη.

Ο χρήστης στη συνέχεια καλείται να επιλέξει τους τελικούς κανόνες που θέλει να εισάγει στην οντολογία, όπως φαίνεται και στο σχήμα 4.9. Στην περίπτωση που το



Σχήμα 4.9: Παράδειγμα πίνακα με τους τελικούς συνεπείς κανόνες.

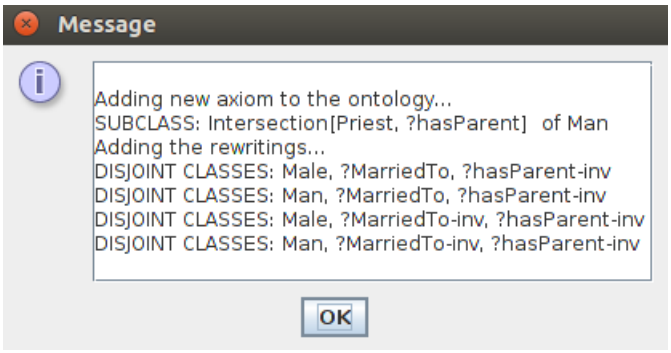
σύστημα δεν έχει βρει κανέναν κανόνα, ο οποίος να μπορεί να εισαχθεί στην οντολογία χωρίς να προκαλέσει ασυνέπεια, τότε ο χρήστης ενημερώνεται αντίστοιχα και το πρόγραμμα τερματίζει (4.10).



Σχήμα 4.10: Αποτυχία εύρεσης κανόνων.

4.2.4 Προσθήκη των Κανόνων στην Οντολογία

Αφού λάβουμε τους τελικούς κανόνες που επέλεξε ο χρήστης, η αρχική οντολογία μπορεί να εμπλουτιστεί με ισχυρισμούς που βασίζονται στο αντικείμενο που αντιπροσωπεύεται από κάθε attribute που συμμετέχει στον κανόνα που προέκυψε. Ταυτόχρονα η οντολογία εμπλουτίζεται με τις επαναγραφές της άρνησης του κανόνα ως περιορισμούς. Αυτή η διαδικασία εμπλουτίζει το TBox της οντολογίας.



Σχήμα 4.11: Εμφάνιση των νέων ισχυρισμών στο χρήστη.

Οι ισχυρισμοί εξαρτώνται από την ποσότητα των attributes που παίρνουν μέρος σε κάθε κανόνα και στα αντικείμενα της οντολογίας στα οποία αντιστοιχούν. Παρατηρήθηκαν οι ακόλουθοι τύποι κανόνων:

Εάν ο κανόνας περιέχει ένα μόνο attribute σε κάθε μέρος του:

- If $attribute_1$ = true, then $attribute_2$ = true

Σε αυτή την περίπτωση οι ισχυρισμοί κλάσης που προκύπτουν από τα δύο attributes, `clExpression1` και `clExpression2` αντίστοιχα, εισάγονται στην οντολογία με την μορφή του ακόλουθου ισχυρισμού:

$$\texttt{OWLSubClassOfAxiom}(clExpression_1,\ clExpression_2)$$

Εάν ο κανόνας περιέχει παραπάνω από ένα attribute στο σώμα του κανόνα:

- If $attribute_1$ = true and $attribute_2$ = true and ... , then $attribute_N$ = true

Σε αυτή την περίπτωση η τομή των attributes του σώματος του κανόνα δημιουργείται με τον ακόλουθο ισχυρισμό:

$$\texttt{OWLObjectIntersectionOf}(concepts)$$

όπου η έκφραση concepts περιέχει τους ισχυρισμούς κλάσεων που προκύπτουν από τα attributes: attribute_1, ..., $\text{attribute}_{(N-1)}$. Όμοια με πριν, το αξίωμα εισάγεται στην οντολογία ως:

$$\texttt{OWLSubClassOfAxiom}(intersection,\ clExpression_N)$$

Κάθε ισχυρισμός κλάσης (class expression) προέκυψε όπως περιγράφεται ακολούθως, εξαρτώμενος από τα attributes και τα αντικείμενα που αντιστοιχούν:

Εάν το attribute αντιστοιχεί σε μία κλάση, τότε ο ισχυρισμός κλάσης είναι η `OWLClass` που αναφέρεται στο όνομα του attribute.

Εάν το attribute αντιστοιχεί σε μία ιδιότητα αντικειμένων, τότε ο ισχυρισμός κλάσης είναι ο ακόλουθος:

$$\texttt{OWLObjectSomeValuesFrom}(objProperty,\ owl:Thing)$$

όπου `objProperty` είναι η `OWLObjectProperty` που αναφέρεται στο όνομα του attribute.

Εάν το attribute αντιστοιχεί σε μία αντίστροφη ιδιότητα αντικειμένων, τότε ο ισχυρισμός κλάσης είναι ο ακόλουθος:

`OWLObjectSomeValuesFrom(invObjProperty, owl:Thing)`

όπου `invObjProperty` είναι η `OWLObjectInverseOf(objProperty)` που αναφέρεται στο όνομα του attribute.

Εν τέλει, οι ισχυρισμοί που προκύπτουν εισάγονται στην οντολογία, και η νέα εμπλουτισμένη οντολογία αποθηκεύεται. Ο χρήστης ενημερώνεται από ένα μήνυμα για κάθε ισχυρισμό που εισάγεται στην οντολογία (4.11), και στη συνέχεια επιλέγει το όνομα και την τοποθεσία που θέλει να αποθηκευτεί η νέα οντολογία.

Κεφάλαιο 5

Πειραματική Αξιολόγηση

Στο κεφάλαιο αυτό, περιγράφονται τα δεδομένα που χρησιμοποιήθηκαν ως είσοδος, και αξιολογείται πειραματικά το σύστημα που υλοποιήθηκε.

5.1 Οντολογίες Εισόδου

Το σύστημα που παρουσιάστηκε προηγουμένως εκτελείται φορτώνοντας αρχικά μία οντολογία, η οποία τελικά θα εμπλουτιστεί με συνεπείς κανόνες.

Οι οντολογίες που έχουν χρησιμοποιηθεί για τα πειράματα είναι δύο ειδών. Το πρώτο είδος αποτελείται από την οντολογία LUBM[1], η οποία αναφέρεται στο πεδίο της πανεπιστημιακής εκπαίδευσης και προέρχεται από το εργαλείο συγκριτικής ανάλυσης συστημάτων βάσεων γνώσης του Πανεπιστημίου Lehigh [8]. Το συγκεκριμένο εργαλείο είναι ευρέως διαδεδομένο και αποτελεί ίσως το σημαντικότερο μέτρο σύγκρισης μεταξύ διαφορετικών συστημάτων. Μερικά αξιώματα αφαιρέθηκαν προκειμένου η οντολογία να είναι συμβατή με το σύστημα Completo. Επίσης έχουν προστεθεί κάποιοι περιορισμοί που δηλώνουν ότι μερικές ατομικές έννοιες είναι ξένες (disjoint) μεταξύ τους. Το σύνολο των οντολογιών LUBM που χρησιμοποιήθηκε, αποτελείται από το ίδιο σύνολο αξιωμάτων, αλλά από διαφορετικό αριθμό ισχυρισμών που σχετίζονται με διαφορετικό αριθμό πανεπιστημίων (1, 5 και 10), ο οποίος δόθηκε ως παράμετρος στη γεννήτρια LUBM [8]. Το δεύτερο είδος οντολογίας που χρησιμοποιήθηκε είναι η Travel[2] [2], η οποία είναι ένα παράδειγμα μίας τουριστικής οντολογίας που έχει αναπτυχθεί από το εργαλείο Protégé[3] και αποτελεί μια αναφορά που περιγράφει γενικότερα τον τομέα του τουρισμού. Στις οντολογίες LUBM χρησιμοποιήθηκαν 52 έννοιες ενώ στην οντολογία travel 20. Οι οντολογίες αυτές επιλέχθηκαν διότι περιείχαν ικανοποιητικό αριθμό ατόμων. Οι κανόνες συσχέτισης που παράχθηκαν για κάθε οντολογία, ήταν 86

[1] image.ntua.gr/~gardero/completo2.0/ontofile.zip

[2] http://www.owl-ontologies.com/travel.owl

[3] https://protege.stanford.edu/

για την οντολογία Travel, και περίπου 700 για τις διάφορες οντολογίες LUBM.

5.2 Αξιολόγηση οντολογιών

Πίνακας 5.1: Πιθανότητες των εννοιών στην οντολογία Travel

Concept	Probability
Accomodation(X)	0.071
AccomodationRating(X)	0.214
BackpackersDestination(X)	0.143
Beach(X)	0.143
Capital(X)	0.143
City(X)	0.214
Destination(X)	0.714
Hotel(X)	0.071
LuxuryHotel(X)	0.071
NationalPark(X)	0.143
RetireeDestination(X)	0.143
RuralArea(X)	0.286
Town(X)	0.071
UrbanArea(X)	0.286
hasAccomodation(X,Y)	0.071
hasAccomodation(Y,X)	0.071
hasPart(X,Y)	0.071
hasPart(Y,X)	0.143
hasRating(X,Y)	0.071
hasRating(Y,X)	0.071

Πίνακας 5.2: Πιθανότητες των εννοιών στην οντολογία LUBM 10

Concept	Probability
AssistantProfessor(X)	0.005
AssociateProfessor(X)	0.007
Chair(X)	0.005
Course(X)	0.062
Department(X)	0.088

Συνεχίζεται στην επόμενη σελίδα

Πίνακας 5.2 – συνέχεια από την προηγούμενη σελίδα

Concept	Probability
Employee(X)	0.095
Faculty(X)	0.02
FullProfessor(X)	0.005
GraduateCourse(X)	0.031
GraduateStudent(X)	0.076
Lexturer(X)	0.003
Organization(X)	0.564
Person(X)	0.366
Professor(X)	0.016
Publication(X)	0.009
ResearchAssistan(X)	0.076
ResearchGroup(X)	0.009
Student(X)	0.346
TeachingAssistant(X)	0.073
UndergraduateStudent(X)	0.27
University(X)	0.466
Work(X)	0.062
advisor(X,Y)	0.329
advisor(Y,X)	0.016
degreeFrom(X,Y)	0.096
degreeFrom(Y,X)	0.466
doctoralDegreeFrom(X,Y)	0.02
doctoralDegreeFrom(Y,X)	0.466
hasAlumnus(X,Y)	0.466
hasAlumnus(Y,X)	0.096
headOf(X,Y)	0.005
headOf(Y,X)	0.088
mastersDegreeFrom(X,Y)	0.02
mastersDegreeFrom(Y,X)	0.465
member(X,Y)	0.088
member(Y,X)	0.366
memberOf(X,Y)	0.366
memberOf(Y,X)	0.088
publicationAuthor(X,Y)	0.009
publicationAuthor(Y,X)	0.096

Συνεχίζεται στην επόμενη σελίδα

Πίνακας 5.2 – συνέχεια από την προηγούμενη σελίδα

Concept	Probability
subOrganizationOf(X,Y)	0.097
subOrganizationOf(Y,X)	0.093
takesCourse(X,Y)	0.346
takesCourse(Y,X)	0.062
teacherOf(X,Y)	0.02
teacjerOf(Y,X)	0.062
teachingAssistantOf(X,Y)	0.073
teachingAssistantOf(Y,X)	0.03
undergraduateDegreeFrom(X,Y)	0.096
undergraduateDegreeFrom(Y,X)	0.466
worksFor(X,Y)	0.02
worksFor(Y,X)	0.088

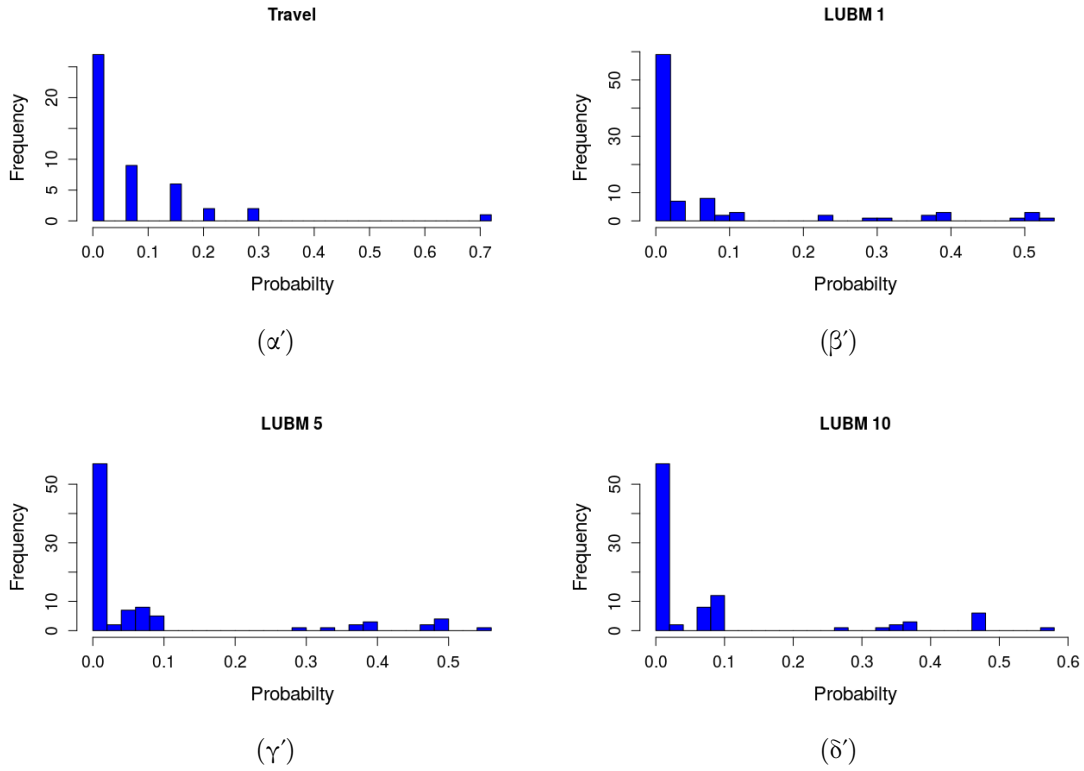
Όπως αναφέρθηκε στην ενότητα 3.1, υπολογίσθηκε η πιθανότητα να ανήκει κάποιο άτομο σε μία έννοια, για όλες τις έννοιες που συμμετέχουν στην οντολογία. Στον Πίνακα 5.1 παρουσιάζονται οι πιθανότητες της οντολογίας Travel, ενώ στον 5.2 οι αντίστοιχες της LUBM 10.

Συγκεντρωτικά η κατανομή της πιθανότητας των εννοιών στις οντολογίες που χρησιμοποιήθηκαν, παρουσιάζεται στο Σχήμα 5.1

Η κατανομή της πιθανότητας μας βοηθά να καταλάβουμε πόσο αντιπροσωπευτικό είναι το κάθε ABox. Καθώς παρατηρήθηκε ότι στις διαφορετικές εκδοχές της οντολογίας LUBM, υπήρχαν διαφορές στις κατανομές των πιθανοτήτων τους, υπολογίσθηκαν οι έννοιες που διέφεραν μεταξύ των τριών εκδοχών, όπου και παρουσιάζονται στο Σχήμα 5.2. Όπως φαίνεται, η πλειοψηφία των εννοιών, έχει διαφορά μικρότερη από 0.05, γεγονός που δείχνει ότι οι τρεις αυτές εκδοχές της συνολικής οντολογίας, δεν παρουσιάζουν σημαντικές διαφορές και, άρα είναι και οι τρεις αντιπροσωπευτικές της αρχικής μας οντολογίας.

5.3 Αξιολόγηση της μετρικής

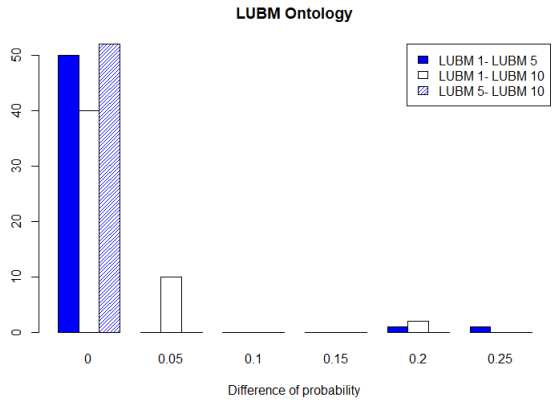
Προκειμένου να ελέγξουμε αν η μετρική που δημιουργήσαμε θα αποτελούσε χρήσιμο μέτρο σύγκρισης για τους κανόνες, δηλαδή αν θα μας έδινε νέα πληροφορία πέρα από αυτή που παίρνουμε από την τιμή confidence, υπολογίστηκε ο συντελεστής συσχέτισης Pearson μεταξύ της τιμής confidence των κανόνων συσχέτισης και της πιθανότητας ύπαρξης αντιπαραδειγμάτων του αντίστοιχου ερωτήματος, όπως αυτή προέκυψε από τις



(α′) (β′)

(γ′) (δ′)

Σχήμα 5.1: Κατανομή πιθανότητας



Σχήμα 5.2: Διαφορά των πιθανοτήτων στις οντολογίες LUBM.

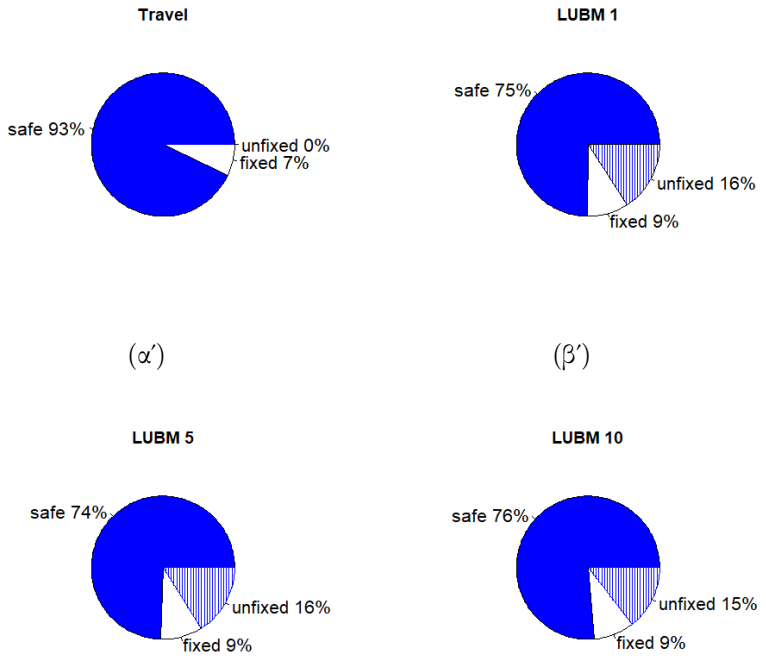
επαναγραφές. Ο συντελεστής Pearson λαμβάνεται διαιρώντας τη συνδιακύμανση των δύο μεταβλητών με το γινόμενο των τυπικών τους αποκλίσεων και ορίζεται μόνο αν και οι δύο τυπικές αποκλίσεις είναι διάφορες του μηδενός. Η συσχέτιση Pearson είναι +1 σε περίπτωση μίας τέλειας, αυξανόμενης γραμμικής σχέσης, -1 σε περίπτωση μίας τέλειας, φθίνουσας γραμμικής σχέσης, και κάποια τιμή μεταξύ -1 και 1 σε κάθε άλλη περίπτωση, καταδεικνύοντας το βαθμό γραμμικής εξάρτησης μεταξύ των μεταβλητών. Καθώς πλησιάζει το μηδέν υπάρχει όλο και λιγότερη σχέση. Όσο πιο κοντά είναι

ο συντελεστής είτε στο -1, είτε στο 1, τόσο πιο δυνατή είναι η συσχέτιση των δύο μεταβλητών. Στην προκειμένη περίπτωση, επιθυμούμε η τιμή μας να προσεγγίζει το μηδέν. Πράγματι, ο συντελεστής συσχέτισης μεταξύ των δύο τιμών που επιλέξαμε είχε τιμές μεταξύ του διαστήματος (-0.4,0) για όλες τις οντολογίες που χρησιμοποιήσαμε.

5.4 Παραγόμενοι κανόνες

Πίνακας 5.3: Κανόνες που δημιουργήθηκαν για τις αξιολογούμενες οντολογίες

Οντολογία	Κανόνες Συσχέτισης	Ασυνεπείς Κανόνες	Διορθωμένοι Κανόνες	Νέοι Κανόνες
Travel	86	6	6	21
LUBM 1	694	174	64	139
LUBM 5	701	179	66	148
LUBM 10	702	166	64	149



(α′) (β′)

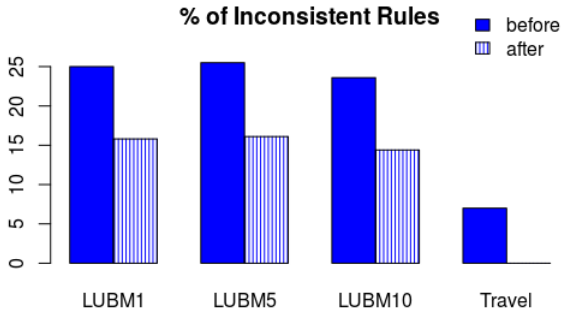
(γ′) (δ′)

Σχήμα 5.3: Κατανομή πιθανότητας

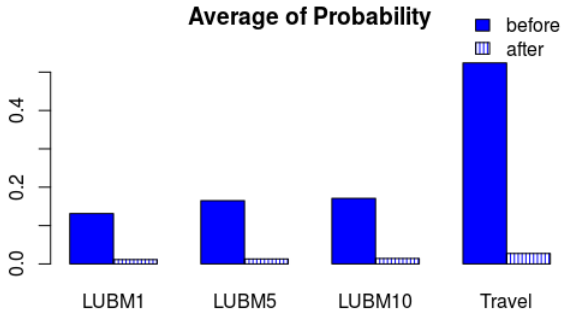
Το σημαντικότερο κομμάτι της εργασία μας, αποτέλεσε η διόρθωση των κανόνων. Για αυτό, αφού παράχθηκαν οι αρχικοί κανόνες από το σύστημα Weka, υπολογίσθηκε

πόσοι από αυτούς ήταν ασυνεπείς. Στη συνέχεια, αφού το πρόγραμμα μας, δοκίμασε να τους διορθώσει, μετρήθηκε ο αριθμός των κανόνων που τελικά μετατράπηκαν σε συνεπείς, καθώς και ο συνολικός αριθμός των νέων κανόνων που κατασκευάστηκαν και δε θα προκαλούσαν καμία ασυνέπεια στην οντολογία. Τα αποτελέσματα παρουσιάζονται στον Πίνακα 5.3.

Στο Σχήμα 5.3 παρουσιάζεται το ποσοστό των συνεπών (safe) , διορθωμένων (fixed) και μη διορθωμένων (unfixed) κανόνων, όπου ένας διορθωμένος (μη διορθωμένος) κανόνας είναι ένας ασυνεπής κανόνας ο οποίος παράγει (δεν μπορεί να παράξει) τουλάχιστον ένα συνεπή κανόνα, μετά την προσθήκη μίας έννοιας στο σώμα του.



Σχήμα 5.4: Ποσοστό των ασυνεπών κανόνων πριν και μετά την διόρθωση των κανόνων.



Σχήμα 5.5: Μέση τιμή της πιθανότητας πρόκλησης ασυνέπειας.

Το ποσοστό των κανόνων, οι οποίοι είναι ασυνεπείς, πριν και μετά τη διόρθωση τους, μπορεί να φανεί στο Σχήμα 5.4. Αντίστοιχα στο Σχήμα 5.5, βλέπουμε τη μέση τιμή της πιθανότητας να υπάρχει άτομο, το οποίο να είναι στιγμιότυπο κάποιας από τις επαναγραφές της άρνησης του κανόνα (δηλαδή να αποτελεί αντιπαράδειγμα).

Όπως ήταν αναμενόμενο το ποσοστό των ασυνεπών κανόνων μειώνεται μετά την εφαρμογή της μεθόδου μας. Επιπλέον, παρατηρείται σημαντική μείωση της μέσης τιμής

της πιθανότητας, μετά τη διόρθωση των κανόνων, γεγονός που μας δείχνει ότι οι νέοι παραγόμενοι κανόνες έχουν μικρότερη πιθανότητα να προκαλέσουν ασυνέπεια σε άλλο ABox της εξεταζόμενης οντολογίας σε σχέση με τους αρχικούς.

Στη συνέχεια παρουσιάζονται μερικοί από τους διορθωμένους κανόνες της οντολογίας Travel. Ο ασυνεπής κανόνας

$$Destination(X) \rightarrow RuralArea(X),$$

ο οποίος έχει τιμή confidence 0.4, μετατράπηκε στον συνεπή κανόνα:

$$Destination(X), BackpackersDestination(X) \rightarrow RuralArea(X),$$

με τιμή confidence 1 και πιθανότητα 0.102 για τις επαναγραφές της άρνησης του κανόνα (άρα η τιμή της μετρικής που υπολογίσαμε στην ενότητα (3.1) είναι ίση με 0.90). Επιπλέον, παρουσιάζονται και οι επαναγραφές αυτές, οι οποίες πρέπει να προστεθούν ως περιορισμοί στην οντολογία:

- $BackpackersDestination(X), Capital(X) \rightarrow \bot$
- $BackpackersDestination(X), City(X) \rightarrow \bot$
- $BackpackersDestination(X), Town(X) \rightarrow \bot$
- $BackpackersDestination(X), UrbanArea(X) \rightarrow \bot$

Παρομοίως, ο ασυνεπής κανόνας

$$\text{Destination}(X) \rightarrow \text{City}(X),$$

με τιμή confidence 0.3, διορθώθηκε στον συνεπή κανόνα:

$$\text{Destination}(X), \text{UrbanArea}(X) \rightarrow \text{City}(X),$$

ο οποίος έχει τιμή confidence 0.75 και τιμή πιθανότητας 0 για τις επαναγραφές της άρνησης του κανόνα (αντίστοιχα η τιμή της μετρικής είναι ίση με 0.75).

Αντίστοιχα, παρουσιάζεται ένα παράδειγμα και για την οντολογία LUBM 10. Ο ασυνεπής κανόνας:

$$advisor(Y_1, X) \rightarrow teacherOf(X, Y_2),$$

ο οποίος έχει τιμή confidence 1, μετατράπηκε στον συνεπή κανόνα:

$$advisor(Y_1, X), memberOf(X, Y_3) \rightarrow teacherOf(X, Y_2),$$

με τιμή confidence 1 και πιθανότητα ασυνέπειας 0.0017 (δηλαδή η τιμή τη μετρικής είναι ίση με 0.99). Η επαναγραφή της άρνησης του κανόνα που πρέπει να προστεθεί ως περιορισμός είναι η:

- $Program(Y_1), advisor(Y_2, X), headOf(X, Y_1) \rightarrow \bot$

Κεφάλαιο 6

Επίλογος

Στο κεφάλαιο αυτό, εξάγονται συμπεράσματα σχετικά με το προτεινόμενο μοντέλο βασισμένα στα πειραματικά αποτελέσματα και στο τέλος προτείνονται δυνατές επεκτάσεις που θα μπορούσαν να βελτιώσουν το σύστημα.

6.1 Σύνοψη και συμπεράσματα

Η παρούσα διπλωματική προτείνει ένα μοντέλο που εμπλουτίζει ασφαλώς μια οντολογία. Τεχνικές Μηχανικής Μάθησης χρησιμοποιήθηκαν με σκοπό να παραχθούν κανόνες συσχέτισης από τα δεδομένα της οντολογίας. Οι κανόνες που βρέθηκαν, ελέγχθηκαν αν είναι συνεπείς σε σχέση με τη δοσμένη οντολογία, και διορθώθηκαν, αν ήταν απαραίτητο, προσθέτοντας νέες έννοιες στο σώμα του κανόνα. Οι επαναγραφές των αντιπαραδειγμάτων κάθε κανόνα, μπορούν να προστεθούν στην οντολογία ως περιορισμοί. Το προτεινόμενο μοντέλο υλοποιήθηκε χρησιμοποιώντας το σύστημα Weka για την παραγωγή των κανόνων συσχέτισης, και το σύστημα Completo για την επαναγραφή των ερωτημάτων με αρνητικά άτομα. Το υλοποιημένο σύστημα αξιολογήθηκε σε διαφορετικές οντολογίες.

Τα πειράματα που εκτελέστηκαν έδειξαν ότι τα αποτελέσματα είναι παρόμοια για οντολογίες με διαφορετικό αριθμό ισχυρισμών (που βασίζονται στο ίδιο TBox), όταν οι έννοιες έχουν παρόμοια αναλογία στοιχείων. Με αυτό τον τρόπο, μία οντολογία με μικρό αλλά αντιπροσωπευτικό ABox προτιμάται από οντολογίες με μεγαλύτερα ABoxes. Το προτεινόμενο μέτρο κατάταξης των κανόνων είναι σχετικό με την ποιότητά τους. Επιπλέον, οι πληροφορίες που εξετάζονται δεν είναι περιττές, καθώς έχουν μικρό συντελεστή συσχέτισης. Τα αποτελέσματα υποστηρίζουν τη σκοπιμότητα της προτεινόμενης μεθόδου για εμπλουτισμό οντολογιών. Τέλος, ο έλεγχος συνέπειας που πραγματοποιείται, επιτρέπει την διόρθωση των κανόνων και την ασφαλή προσθήκη τους στην οντολογία.

6.2 Μελλοντικές επεκτάσεις

Υπάρχει πληθώρα επεκτάσεων του συστήματος που υλοποιήθηκε. Οι επεκτάσεις αυτές αφορούν την αύξηση των ικανοτήτων του συστήματος.

Σχετικά με την εξερεύνηση νέων κανόνων, θα ήταν χρήσιμο το σύστημα να παράγει πιο εκφραστικούς κανόνες, καθώς δεν ήταν πάντοτε δυνατή η εύρεση κανόνων συσχέτισης σε οντολογίες με μικρό όγκο δεδομένων. Αυτό αποτελεί τροχοπέδη που θα μπορούσε να αντιμετωπιστεί σε κάποια μελλοντική επέκταση. Επιπλέον, θα ήταν χρήσιμο να εξερευνηθούν περισσότερες έννοιες, όπως η $\exists R.C$, οι οποίες στη συνέχεια θα μετατραπούν στα αντίστοιχα δεδομένα για τα αρχεία εισόδου του Weka, ώστε τα αποτελέσματα των κανόνων συσχέτισης να προσφέρουν περισσότερες πληροφορίες.

Όσον αφορά τα δεδομένα εισόδου, μία μελλοντική επέκταση που πιθανότατα θα βελτίωνε σημαντικά το σύστημα, θα ήταν ο χειρισμός μεγαλύτερου όγκου οντολογιών. Καθώς το πρόγραμμα φορτώνει τις οντολογίες, απαιτεί πολλή μνήμη, γεγονός που προκαλεί την καθυστέρηση της ολοκλήρωσης του προγράμματος σε περίπτωση που η οντολογία εισόδου είναι μεγάλη. Μία καλύτερη υλοποίηση θα μείωνε σημαντικά τον συνολικό χρόνο εκτέλεσης του προγράμματος.

Βιβλιογραφία

[1] Alfonso, E. M., and Stamou, G. *Rewriting Queries with Negated Atoms*. Springer International Publishing, Cham, 2017, pp. 151–167.

[2] Alonso, K., Zorrilla, M., Confalonieri, R., Vázquez-Salceda, J., Inan, H., Palau, M., Calle, F., and Castro, E. Ontology-based tourism for all recommender and information retrieval system for interactive community displays, 01 2012.

[3] Baader, F., Horrocks, I., and Sattler, U. *Description Logics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 3–28.

[4] Baget, J.-F., Gutierrez, A., Leclère, M., Mugnier, M.-L., Rocher, S., and Sipieter, C. Datalog+, RuleML and OWL 2: Formats and Translations for Existential Rules. In *RuleML: Web Rule Symposium* (Berlin, Germany, Aug. 2015).

[5] Baget, J.-F., Leclère, M., Mugnier, M.-L., Rocher, S., and Sipieter, C. *Graal: A Toolkit for Query Answering with Existential Rules*. Springer International Publishing, Cham, 2015, pp. 328–344.

[6] Bechhofer, S., Özsu, M., and Liu, L. *OWL: Web Ontology Language*. Reference. Springer, Germany, 2009. In Press.

[7] Bühmann, L., and Lehmann, J. Universal owl axiom enrichment for large knowledge bases. In *Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management* (Berlin, Heidelberg, 2012), EKAW'12, Springer-Verlag, pp. 57–71.

[8] Guo, Y., Pan, Z., and Heflin, J. Lubm: A benchmark for owl knowledge base systems. *Web Semant. 3*, 2-3 (Oct. 2005), 158–182.

[9] Gutiérrez-Basulto, V., Ibañez-García, Y., Kontchakov, R., and Kostylev, E. V. *Conjunctive Queries with Negation over DL-Lite: A Closer Look*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 109–122.

[10] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The weka data mining software: An update. *SIGKDD Explor. Newsl. 11*, 1 (Nov. 2009), 10–18.

[11] Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., and Rudolph, S., Eds. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation, 27 October 2009. Available at `http://www.w3.org/TR/owl2-primer/`.

[12] Hitzler, P., Krötzsch, M., and Rudolph, S. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2009.

[13] Isaac, A., and Summers, E., Eds. *SKOS Simple Knowledge Organization System Primer*. W3C Recommendation, 18 August 2009. Available at `http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/`.

[14] Krötzsch, M. *OWL 2 Profiles: An Introduction to Lightweight Ontology Languages*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 112–183.

[15] Krötzsch, M., Simancik, F., and Horrocks, I. A description logic primer.

[16] Pan, J. Z. *Resource Description Framework*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 71–90.

[17] Rudolph, S. *Foundations of Description Logics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 76–136.

[18] Trivela, D., Stoilos, G., Chortaras, A., and Stamou, G. Optimising resolution-based rewriting algorithms for owl ontologies.

[19] Völker, J., and Niepert, M. Statistical schema induction. In *The Semantic Web: Research and Applications* (Berlin, Heidelberg, 2011), Springer Berlin Heidelberg, pp. 124–138.

[20] Völker, J., Vrandečić, D., Sure, Y., and Hotho, A. Learning disjointness. In *The Semantic Web: Research and Applications* (Berlin, Heidelberg, 2007), Springer Berlin Heidelberg, pp. 175–189.

[21] Ζαφειρογδη, Κ. Εμπλουτισμός Οντολογιών με Τεχνικές Μηχανικής Μάθησης, 2016. Διπλωματική Εργασία.