

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ



ΜΕΘΟΔΟΙ ΚΑΙ ΕΡΓΑΛΕΙΑ ΓΙΑ ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ
ΣΥΣΤΗΜΑΤΩΝ ΔΟΣΟΛΗΨΙΩΝ ΣΕ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΚΟΥΒΑΡΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΗΡΑΚΛΕΙΟ ΚΡΗΤΗΣ ΣΕΠΤΕΜΒΡΗΣ 2000

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Μέθοδοι και Εργαλεία για Ανάπτυξη και Σχεδίαση Συστημάτων
Δοσοληψιών σε Βάσεις Δεδομένων

Κούβαρης Κωνσταντίνος

Μεταπτυχιακή Εργασία

Ηράκλειο Κρήτης, Σεπτέμβρης 2000

Μέθοδοι και Εργαλεία για Ανάπτυξη και Σχεδίαση Συστημάτων Δοσοληψιών σε Βάσεις Δεδομένων

Εργασία που υποβλήθηκε στην επιτροπή μεταπτυχιακών σπουδών
του Τμήματος Επιστήμης Υπολογιστών
της Σχολής Θετικών Επιστημών
του Πανεπιστημίου Κρήτης
ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση του
ΔΙΠΛΩΜΑΤΟΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Κούβαρης Κωνσταντίνος
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

Πλεξουσάκης Δημήτριος, Επόπτης
(Επίκουρος Καθηγητής)

Κωνσταντόπουλος Παναγιώτης, Μέλος
(Καθηγητής)

Μαρκάτος Ευάγγελος, Μέλος
(Επίκουρος Καθηγητής)

Δεκτή:

Κωνσταντόπουλος Παναγιώτης, Μέλος
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών
(Καθηγητής)

Ηράκλειο Κρήτης Σεπτέμβρης 2000

Μέθοδοι και Εργαλεία για Ανάπτυξη και Σχεδίαση Συστημάτων Δοσοληψιών σε Βάσεις Δεδομένων

Κούβαρης Κωνσταντίνος
Μεταπτυχιακή Εργασία

Τμήματος Επιστήμης Υπολογιστών
Πανεπιστημίου Κρήτης

Περίληψη

Η εξασφάλιση της σημασιολογικής ορθότητας των πληροφοριών που βρίσκονται αποθηκευμένες σε ένα σύστημα Βάσεων Δεδομένων, έχει αναγνωριστεί ως ένας από του θεμέλιους λίθους για την ανάπτυξη πληροφοριακών συστημάτων. Επίσης, είναι μία από τις σημαντικότερες αιτίες χρονικής καθυστέρησης της λειτουργίας ενός συστήματος και επομένως μία από τις κύριες αιτίες μειωμένης απόδοσής του.

Η *σημασιολογική ορθότητα*(semantic correctness) προσδιορίζεται σε σχέση με του *περιορισμούς ακεραιότητας*(integrity constraints), οι οποίοι είναι δηλωτικοί προσδιορισμοί ιδιοτήτων του πεδίου που καλούμαστε να μοντελοποιήσουμε και που δεν πρέπει ποτέ να παραβιαστούν από καμία διαδικασία ενημέρωσης της Βάσης Δεδομένων που διαθέτουμε.

Η πλειοψηφία των εμπορικών συστημάτων Βάσεων Δεδομένων δεν παρέχουν τίποτα περισσότερο από την επιβολή πολύ περιορισμένων τύπων περιορισμών ακεραιότητας. Για τον λόγο αυτό, είναι πολύ εμφανής η έλλειψη εργαλείων που θα βοηθούν τους σχεδιαστές των Βάσεων Δεδομένων στον προσδιορισμό σωστών δοσοληψιών.

Στην παρούσα εργασία μας ενδιαφέρει η *διατήρηση των περιορισμών μέσω δοσοληψιών*(maintenance by transaction). Η προσέγγιση αυτή, τροποποιεί τις δοσοληψίες κατά τέτοιο τρόπο, ώστε να υπάρχει η εγγύηση ότι οι περιορισμοί ακεραιότητας δεν παραβιάζονται, όποια και αν είναι η ακολουθία των δοσοληψιών που θα εκτελεστούν στην Βάση Δεδομένων του συστήματος.

Μέθοδοι και Εργαλεία για Ανάπτυξη και Σχεδίαση Συστημάτων Δοσοληψιών σε Βάσεις Δεδομένων

Κούβαρης Κωνσταντίνος
Μεταπτυχιακή Εργασία

Τμήματος Επιστήμης Υπολογιστών
Πανεπιστημίου Κρήτης

Abstract

Ensuring the semantic correctness of the information stored in database systems has been recognized as one of cornerstone issues in the development of information systems. It is also one of the major bottlenecks in the systems' performance.

Semantic correctness is specified in terms of *integrity constraints*, which are declarative specifications of properties of the domain being modeled that should never be violated by any operation that updates the database.

The majority of commercial database systems do not provide but for the enforcement of very limited types of constraints. The task of ensuring, that transaction execution will not violate any integrity constraint is left to the transaction designer. In large, semantically rich, multi-user and possibly distributed databases, this task becomes extremely difficult: a transaction designer may not be familiar with all the specified constraints and multiple transactions may affect more than one constraint at a time.

Our research has followed one avenue in addressing the problem of ensuring data integrity, which is called *maintenance by transactions*. Maintenance by transactions modifies transaction specifications so that database integrity constraints are guaranteed to hold in any executable sequence of transactions.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα αν ευχαριστήσω όλους όσους συνέβαλαν στην πραγματοποίηση της εργασίας αυτής. Πρώτον από όλους τον καθηγητή μου, κ. Δημήτριο Πλεξουσάκη(Επίκουρο Καθηγητή του Πανεπιστημίου Κρήτης), για την συνολική του καθοδήγηση στην εργασία αυτή, αλλά και σε κάθε άλλη ακαδημαϊκή μου δραστηριότητα.

Θα ήθελα, επίσης, να ευχαριστήσω τα μέλη της εισηγητικής επιτροπής κ. Παναγιώτη Κωνσταντόπουλο(Καθηγητή του Πανεπιστημίου Κρήτης) και κ. Ευάγγελο Μαρκάτο(Επίκουρο Καθηγητή του Πανεπιστημίου Κρήτης) για το ενδιαφέρον που έδειξαν.

Επίσης, θα επιθυμούσα να ευχαριστήσω το Πανεπιστήμιο Κρήτης και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για την οικονομική και υλικοτεχνική τους υποστήριξη.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου για τα εφόδια που μου έδωσαν, ώστε να φθάσω στο σημείο που βρίσκομαι, καθώς και τους φίλους μου που τόσο πολύ μου συμπαραστάθηκαν καθ' όλη την διάρκεια της εκπόνησης της εργασίας αυτής.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	III
Abstract	IV
ΕΥΧΑΡΙΣΤΙΕΣ	V
ΠΕΡΙΕΧΟΜΕΝΑ	VI
1. Εισαγωγή.....	1
1.1 Περιγραφή του προβλήματος.....	1
1.2 Κίνητρα	3
1.3 Περιεχόμενα και στόχοι.....	4
1.4 Ερευνητικά Θέματα	6
1.5 Συμπεράσματα	8
2. Παρελθοντική δουλειά.....	10
2.1 Διατήρηση περιορισμών μέσω δοσοληπιών	11
2.2 Παράλληλη βιβλιογραφία	19
3. Πρόβλημα πλαισίου και επιπτώσεων.....	22
3.1 Επιπτώσεις(Ramifications)	24
3.2 Πρόβλημα πλαισίου	28
4. Η δική μας προσέγγιση	41
4.1 Περιγραφή της αρχιτεκτονικής του συστήματος.....	43
4.2 Περιγραφή των αλγορίθμων επεξεργασίας των δεδομένων	47
4.3 Περιγραφή των δεδομένων εισόδου	55
4.4 Περιγραφή των δεδομένων εξόδου.....	58
4.5 Περιγραφή του αλγορίθμου μετατροπής σε Δ.Κ.Μ.	58
5. Παρουσίαση αποτελεσμάτων – Παράδειγμα.....	61
6. Επεκτάσεις, Βελτιώσεις	79
6.1 Επεκτάσεις του συστήματος	79
6.2 Βελτιώσεις του συστήματος	80
Συμπεράσματα	82
Παράρτημα Α.....	85
Βιβλιογραφία	94

Κεφάλαιο 1^ο

Εισαγωγή

1.1 Περιγραφή του προβλήματος

Η εξασφάλιση της σημασιολογικής ορθότητας των πληροφοριών που βρίσκονται αποθηκευμένες σε ένα σύστημα Βάσεων Δεδομένων, έχει αναγνωριστεί ως ένας από τους θεμέλιους λίθους για την ανάπτυξη πληροφοριακών συστημάτων. Επίσης, λόγω του μεγάλου πλήθους των απαιτούμενων πράξεων, αποτελεί μία από τις σημαντικότερες αιτίες χρονικής καθυστέρησης της λειτουργίας ενός τέτοιου συστήματος και επομένως μία από τις κύριες αιτίες μειωμένης απόδοσής του.

Η *σημασιολογική ορθότητα* (semantic correctness) προσδιορίζεται σε σχέση με τους *περιορισμούς ακεραιότητας* (integrity constraints) [Ple96]. Οι περιορισμοί ακεραιότητας αποτελούν συναρτησιακές εξαρτήσεις. Σύμφωνα με τον ορισμό που συναντάμε στο παραπάνω έγγραφο, αποτελούν δηλωτικούς προσδιορισμούς ιδιοτήτων, πάνω στο πεδίο που καλούμαστε να μοντελοποιήσουμε. Οι περιορισμοί αυτοί δεν πρέπει ποτέ να παραβιαστούν από καμία διαδικασία ενημέρωσης της Βάσης Δεδομένων που μας ενδιαφέρει. Με άλλα λόγια, προσδιορίζουν τις έγκυρες καταστάσεις μίας Βάσης Δεδομένων, καθώς επίσης και τις επιτρεπτές μεταβάσεις μεταξύ των καταστάσεων αυτών. Θα μπορούσαμε να πούμε ότι οι περιορισμοί ακεραιότητας συνιστούν ένα μέσο ελέγχου της ποιότητας της αποθηκευμένης πληροφορίας σε μία Βάση Δεδομένων ή σε μία Βάση Γνώσεων.

Οι περιορισμοί ακεραιότητας χωρίζονται σε τρεις κύριες κατηγορίες: τους στατικούς, τους δυναμικούς και τους μεταβατικούς. Το μεγαλύτερο μέρος της βιβλιογραφίας, σχετικά με επαλήθευση και διατήρηση των περιορισμών ακεραιότητας, αφορά τους στατικούς περιορισμούς ακεραιότητας, ενώ μόνο μία μικρή ομάδα ερευνητών έχουν ασχοληθεί με την κατηγορία των δυναμικών περιορισμών. Ορίζοντας τον όρο *επαλήθευση περιορισμών*, μία Βάση Δεδομένων ΒΔ λέμε ότι *επαληθεύει* ένα περιορισμό ΠΑ, όταν και μόνο όταν από την ΒΔ, μετά την εκτέλεση ενός βήματος, προκύπτει ο ΠΑ. Δηλαδή, όταν και μόνο όταν ισχύει:

« $BD \models PA$ » [Rei95].

Οι στατικοί περιορισμοί είναι συναρτησιακές εξαρτήσεις των οποίων όλα τα κατηγορήματα ή οι συναρτήσεις που λαμβάνουν μέρος στον ορισμό τους, αναφέρονται στην τρέχουσα κατάσταση του συστήματος και μόνο σ' αυτήν, αδιαφορώντας για παρελθοντικές ή μελλοντικές καταστάσεις του συστήματος που διαθέτουμε [Ple96]. Ένα παράδειγμα τέτοιων περιορισμών είναι το ακόλουθο: « $(\forall st, c, g, g'). \text{grade}(st, c, g, S_0) \wedge \text{grade}(st, c, g', S_0) \supset g = g'$ ». Στο παράδειγμα αυτό, η S_0 είναι η αρχική κατάσταση του συστήματος, st είναι ο φοιτητής, c είναι η τάξη που αυτός ανήκει, ενώ g και g' είναι οι βαθμοί που έχει αποκτήσει. Στην έκφραση του περιορισμού αυτού, βλέπουμε ότι σε κάθε κατάσταση του συστήματος αντιστοιχεί μοναδικός βαθμός, ανεξάρτητος από τυχόν επόμενες ή προηγούμενες τιμές του [Rei95].

Αντίθετα, οι δυναμικοί περιορισμοί είναι συναρτησιακές εξαρτήσεις των οποίων μερικά από τα κατηγορήματα ή τις συναρτήσεις που λαμβάνουν μέρος στον ορισμό τους, αναφέρονται τόσο στην τρέχουσα, όσο και σε παρελθοντικές ή ακόμα και μελλοντικές καταστάσεις του συστήματος [Ple96]. Ένα παράδειγμα τέτοιου περιορισμού είναι ο μισθός ενός εργαζόμενου, ο οποίος σε κάθε κατάσταση του συστήματος, πρέπει να είναι μεγαλύτερος από την τιμή που είχε στην προηγούμενη κατάσταση [Rei95]. Σε αντίθεση με τους στατικούς, οι δυναμικοί περιορισμοί σχετίζονται και με αναφορές σε τιμές γνωρισμάτων στην κατάσταση πριν, αλλά και μετά την ενημέρωση.

Τέλος, μία επιπλέον κατηγορία περιορισμών είναι οι μεταβατικοί περιορισμοί οι οποίοι αναφέρονται είτε σε μία καθορισμένη κατάσταση είτε σε ζεύγη συσχετιζόμενων καταστάσεων της Βάσης Δεδομένων [Ple96].

Τα τελευταία χρόνια, οι ερευνητές έχουν ακολουθήσει δύο κύριες προσεγγίσεις, όσον αφορά την διερεύνηση των προβλημάτων που προκύπτουν από την εξασφάλιση της σημασιολογικής ορθότητας. Η πρώτη αποκαλείται *παρακολούθηση των περιορισμών* (integrity monitoring), ενώ η δεύτερη *διατήρηση μέσω δοσοληψιών* (maintenance by transactions) [Ple96]. Πιο αναλυτικά, η πρώτη προσπαθεί να εκμεταλλευτεί το γεγονός ότι, εν γέννη, οι ενημερώσεις δεν επηρεάζουν του σημασιολογικούς περιορισμούς. Ωστόσο, η δεύτερη προσέγγιση τροποποιεί τις δοσοληψίες κατά τέτοιο τρόπο, ώστε το σύστημα να εγγυάται την μη παραβίαση των

περιορισμών ακεραιότητας, κάτω από την εκτέλεση οποιασδήποτε ακολουθίας δοσοληψιών, πάνω στην Βάση Δεδομένων. Στην παρούσα εργασία μας ενδιαφέρει η υλοποίηση ενός τμήματος της δεύτερης προσέγγισης. Το θεωρητικό τμήμα της προσέγγισης αυτής, απασχόλησε πολλούς ερευνητές σε εργασίες όπως η [Ple96], η [LR94] και τέλος η [BMR93]. Η επιλογή της προσέγγισης αυτής έγινε επειδή μας παρέχει το κατάλληλο περιβάλλον επίλυσης του προβλήματος τις αναδρομικής σχεδίασης, σε συνεργασία με την διατήρηση των τύπων περιορισμών ακεραιότητας.

1.2 Κίνητρα

Μέχρι σήμερα, στα εμπορικά πακέτα ευρείας κυκλοφορίας, εν γένη, δεν κρίθηκε επιβεβλημένη η χρήση μηχανισμών επιβολής περίπλοκων περιορισμών ακεραιότητας, κατά την μοντελοποίηση ενός πεδίου. Αυτό συμβαίνει, ενδεχομένως, επειδή, η πολυπλοκότητα των υπολογισμών που προκύπτουν είναι πρακτικά πολύ μεγάλη, ενώ το κέρδος, που αποκομίζουμε από αυτούς, είναι πρακτικά μικρό. Έτσι, η πλειοψηφία των εμπορικών συστημάτων Βάσεων Δεδομένων παρέχουν την επιβολή πολύ περιορισμένων τύπων περιορισμών ακεραιότητας.

Παρά την εκτεταμένη έρευνα της τελευταίας δεκαετίας, γύρω από την εξασφάλιση της σημασιολογίας των περιορισμών, δεν υπάρχουν ακόμα πρακτικά επιτεύγματα. Για την κατάσταση αυτή, ενδεχομένως, ευθύνεται η έλλειψη αποτελεσματικών μεθόδων ελέγχου της επαλήθευσης, των περιορισμών ακεραιότητας. Για τον λόγο αυτό, είναι πολύ εμφανής η έλλειψη εργαλείων, που σκοπό έχουν να βοηθήσουν τους σχεδιαστές των Βάσεων Δεδομένων στην δημιουργία σωστών δοσοληψιών, ώστε να απλοποιηθεί το πρόβλημα της εξασφάλισης της σημασιολογικής ορθότητας.

Η διαδικασία εξασφάλισης ότι η εκτέλεση μίας δοσοληψίας δεν παραβιάζει μερικούς, από τους πιθανόν πολυάριθμους, περιορισμούς ακεραιότητας του πεδίου που μοντελοποιείται, επαφίεται στον μηχανικό που είναι υπεύθυνος για την σχεδίαση των δοσοληψιών αυτών. Σε μεγάλες, με πλούσια σημασιολογία, πολυχρηστικές και πιθανόν κατανεμημένες Βάσεις Δεδομένων, η διαδικασία γίνεται εξαιρετικά δύσκολη. Έτσι, ο σχεδιαστής των δοσοληψιών πρέπει να είναι πλήρως εξοικειωμένος με όλους τους περιορισμούς ακεραιότητας, επειδή είναι υπαρκτό το ενδεχόμενο,

ορισμένες δοσοληψίες να επηρεάσουν περισσότερους από ένα περιορισμούς, κατά την ίδια χρονική στιγμή, στην τρέχουσα κατάσταση της Βάσης Δεδομένων.

Η ανάγκη μοντελοποίησης εξελισσόμενων πεδίων έχουν δώσει ώθηση σε ερευνητικά θέματα, που σχετίζονται με την ενσωμάτωση του χρόνου στα συστήματα που περιέχουν Βάσεις Δεδομένων. Στην περίπτωση αυτή, οι έως τώρα, αποκαλούμενοι στατικοί και δυναμικοί περιορισμοί ακεραιότητας δίνουν την θέση τους στους χρονικούς περιορισμούς ακεραιότητας [Ple96].

Έτσι, με τη εισαγωγή του χρονικού στοιχείου στο δυναμικό περιβάλλον του συστήματός μας, εισάγουμε μία επιπλέον διάσταση στα προβλήματα, των οποίων η επίλυση εξασφαλίζει την μη παραβίαση των περιορισμών. Η διάσταση αυτή συνίσταται, κυρίως, στην παρακολούθηση των χρονικά εξαρτημένων ιδιοτήτων των δεδομένων μίας τέτοιας Βάσης. Οι χρονικές Βάσεις και οι επεκτάσεις τους, γνωστές με τον όρο Βάσεις Γνώσεων, απαιτούν ισχυρούς μηχανισμούς προκειμένου να εξασφαλίσουν ότι οι χρονικά εξαρτημένες ιδιότητες δεν παραβιάζονται εξαιτίας μερικών ενημερώσεων, πάνω στην Βάση Γνώσεων ή εξαιτίας της παρόδου του χρόνου.

Οι ιδιότητες που χρειάζονται εξασφάλιση μπορεί να συνδέουν πολλαπλές καταστάσεις, που αναφέρονται στο παρόν, στο παρελθόν, ή ακόμα και στο μέλλον. Για τον λόγο αυτό, η επιβεβαίωση των ιδιοτήτων, που εκφράζονται με περιορισμούς ακεραιότητας, μπορεί να αφορούν πολλαπλές καταστάσεις που είναι διαθέσιμες σε όλες τις χρονικές στιγμές. Όπως γίνεται πλέον, εύκολα, κατανοητό, η πολυπλοκότητα της επαλήθευσης τέτοιων χρονικών περιορισμών είναι, σε σημαντικό βαθμό, υψηλότερη από αυτή που αφορά την επαλήθευση στατικών περιορισμών.

1.3 Περιεχόμενα και στόχοι

Οι θεωρητικές αρχές για μία μεθοδική αντιμετώπιση του ελέγχου ορθότητας της περιγραφής δοσοληψιών σε ένα σύστημα Βάσεως Γνώσεων, μελετήθηκε, λεπτομερώς, στις εργασίες [Ple96] και [PM96], ενώ η [Ple95b] παρουσιάζει μία εφαρμογή της προτεινόμενης μεθόδου. Η προσέγγιση υιοθετεί μερικές βασικές ιδέες άλλων επιστημονικών πεδίων, με σημαντικότερες αυτές του *προβλήματος του πλαισίου* (frame problem) και του *προβλήματος των επιπτώσεων* (ramification

problem). Είναι απαραίτητη η παρατήρηση ότι η συστηματική επίλυση των προβλημάτων αυτών μπορεί να χρησιμοποιηθεί κατά την διάρκεια της σχεδίασης των δοσοληψιών της Βάσης Δεδομένων. Στις εργασίες αυτές προτάθηκε μία τεχνική επεξεργασίας πάνω στις προδιαγραφές των δοσοληψιών, η οποία τις τροποποιεί ενσωματώνοντας μέσα σε αυτές, συνθήκες ικανές για την ικανοποίηση των περιορισμών ακεραιότητας.

Το πρόβλημα του πλαισίου αναφέρεται στην ρητή δήλωση των δεδομένων που παραμένουν αναλλοίωτα μετά την ενημέρωση της Βάσης Δεδομένων, ενώ το πρόβλημα των επιπτώσεων αφορά την περιληπτική δήλωση των έμμεσων αποτελεσμάτων μίας εκτέλεσης[BMR95]. Αρχίζοντας από ένα σύνολο τελικών προσδιορισμών, με μία μορφή *προαπαιτούμενων συνθηκών*, ή αλλιώς προσυνθήκες (precondition), καθώς επίσης και συνθηκών που ισχύουν στην επόμενη κατάσταση της Βάσης Δεδομένων, ή αλλιώς μετασυνθήκες(postcondition), μαζί με ένα σύνολο αναλλοίωτων συνθηκών(invariant), η προτεινόμενη, από τις προαναφερθείσες εργασίες, μέθοδος ενισχύει τις μετασυνθήκες, ώστε οι αναλλοίωτες συνθήκες να παραμένουν σαν αποτέλεσμα αυτών των ενισχυμένων συνθηκών.

Η διαδικασία βασίζεται στην επίλυση των προβλημάτων του πλαισίου και των επιπτώσεων. Οι προδιαγραφές των δοσοληψιών αναλύονται και ενισχύονται κατά την διάρκεια της μετάφρασης του προγράμματος του συστήματος. Στο σημείο αυτό, εάν νέες αναλλοίωτες συνθήκες ή νέοι προσδιορισμοί προστεθούν, μπορούν να εξυπηρετηθούν σταδιακά, χωρίς την υποχρεωτική, εκ νέου, μετάφραση ολόκληρου του συνόλου των προσδιορισμών, επιτυγχάνοντας μεγάλη εξοικονόμηση στον απαιτούμενο χρόνο απόκρισης του συστήματος, καθώς επίσης και μείωση του κόστους συντήρησής του.

Επιπλέον, η διαδικασία εφαρμόζεται σε σύνολα στατικών και μεταβατικών περιορισμών ακεραιότητας, μεγάλου μεγέθους. Τέτοια σύνολα περιορισμών συναντάμε, συνήθως, σε συστήματα πολλών χιλιάδων εγγραφών, που έχουν υψηλές απαιτήσεις στην περιγραφή των λειτουργιών τους. Ένα παράδειγμα τέτοιας εφαρμογής, είναι το σύστημα ενός παγκόσμιου οργανισμού υγείας, όπου για την μοντελοποίηση του πεδίου του χρειάζομαστε πολλούς και διαφορετικής φύσεως περιορισμούς.

Βασιζόμενοι στην θεωρητική μελέτη της εργασίας [Pl96], υλοποιήσαμε ένα σύστημα, το οποίο ακολουθεί μία μέθοδο συστηματικής επίλυσης των προβλημάτων του πλαισίου και των επιπτώσεων. Με ορισμένες επεκτάσεις και τροποποιήσεις του

συστήματός μας, τις οποίες περιγράφουμε στο Κεφάλαιο 7, το σύστημα μπορεί να παράσχει πολύτιμη βοήθεια στον σχεδιαστή των δοσοληψιών, με μηχανισμούς ανάδρασης, είτε με την μορφή μίας προτεινόμενης, από το σύστημα, ενίσχυσης των περιγραφών των δοσοληψιών, είτε με την μορφή εντοπισμού ασυνεπειών στην Βάση Δεδομένων. Επιπλέον, με την υλοποίηση της νέας αυτής μεθόδου διατήρησης των περιορισμών ακεραιότητας, σε μεγάλες Βάσεις Δεδομένων, το σύστημα επιτυγχάνει ικανοποιητική απόδοση, κατά την διάρκεια της εκτέλεσης.

Με άλλα λόγια, το εργαλείο που παρουσιάζουμε, θα βοηθήσει ένα σχεδιαστή δοσοληψιών στον καθορισμό προδιαγραφών σωστών δοσοληψιών. Το εργαλείο αυτό δέχεται περιγραφές προδιαγραφών σε μία γλώσσα που περιλαμβάνει και την χρονική διάσταση, σε περιβάλλον Βάσης Δεδομένων, όπως η Sybase. Η γλώσσα που επινοήσαμε, μας επιτρέπει να δηλώνουμε τα σύνολα προσυνθηκών/μετασυνθηκών με κάποιο προκαθορισμένο τύπο. Λαμβάνοντας σαν είσοδο ένα σύνολο δοσοληψιών, σύμφωνα με ένα συγκεκριμένο τρόπο αναπαράστασης, το εργαλείο, που υλοποιήσαμε, θα παράγει σαν έξοδο ένα σύνολο νέων κανόνων και προσδιορισμών, οι οποίοι αποτελούν αναγκαίες συνθήκες για την εξασφάλιση της ορθότητας των περιορισμών. Οι κανόνες αυτοί, μετά από κατάλληλη επεξεργασία, μπορούν να αποτελέσουν και ικανές συνθήκες, για την εξασφάλιση της ορθότητας των περιορισμών, όπως προαναφέραμε. Μία, περισσότερο αναλυτική, περιγραφή της λειτουργίας του εργαλείου μας παρουσιάζεται στο Κεφάλαιο 4.

1.4 Ερευνητικά Θέματα

Το πρόβλημα της διατήρησης των περιορισμών ακεραιότητας σχετίζεται με το *πρόβλημα ενημέρωσης όψεων* (view update problem) [Ple96]. Οι περιορισμοί μπορούν να θεωρηθούν όψεις, που πρέπει να παραμείνουν κενές. Για τον λόγο αυτό, οι περιορισμοί θα μπορούσαν να εξασφαλιστούν, ακολουθώντας διαδικασία παρόμοια με αυτήν της διαγραφής καταχωρήσεων από μία όψη. Οι μέχρι σήμερα γνωστές τεχνικές ενημέρωσης όψεων, ωστόσο, δεν παρέχουν μία αποδοτική επίλυση στο πρόβλημα της διατήρησης των περιορισμών. Οι εργασίες, που προαναφέραμε, όπως και η υλοποίησή τους μέσα από την εργασία μας, βοηθούν στην κατασκευή αποδοτικών μεθόδων εξασφάλισης χρονικών περιορισμών ακεραιότητας, σε ένα

δομημένο αντικειμενοστραφές πλαίσιο εργασίας, με την παρουσία *χρονικών συμπερασματικών κανόνων* (temporal deductive rules).

Η εισαγωγή της χρονικής διάστασης σε μία Βάση Γνώσεων, έχει ως κύριο σκοπό την διατήρηση της ιστορικής συνέχειας των γεγονότων και των πράξεων. Δημιουργεί νέα ζητήματα που αφορούν τόσο την σημασιολογία που δηλώνει το γνώρισμα του χρόνου, όσο και την αντίστοιχη οντολογία του. Ο χρόνος, σαν πεδίο πιθανών τιμών, είναι αόριστος. Οι χρονικές προτάσεις, επίσης, μπορεί να αναφέρονται σε αυθαίρετες χρονικές στιγμές στο παρελθόν ή στο μέλλον. Ωστόσο, οι χρονικοί περιορισμοί ακεραιότητας, πρέπει να μεταφράζονται πάνω σε προκαθορισμένες, υπαρκτές δομές, όπως είναι οι Βάσεις Γνώσεων. Επιπλέον, τα αντικείμενα μπορεί να αρχίζουν ή να τερματίζεται η ύπαρξή τους, σε καθορισμένα χρονικά διαστήματα (time intervals), κατά την μοντελοποίηση του πεδίου εφαρμογής που επιθυμούμε να σχεδιάσουμε.

Προτείνουμε, λοιπόν, να επεκτείνουμε την ιδέα της ικανοποίησης ενός περιορισμού, πάνω από μια ιστορικά καθορισμένη ακολουθία συμβάντων, ώστε, να λάβουμε υπόψη μας την διπλή παρουσία του χρόνου, σαν χρονικό διάστημα, δηλαδή την ιστορία του, είτε σαν χρονική στιγμή, δηλαδή το στιγμιότυπα, στους τύπους που χρησιμοποιούμε στην εφαρμογή μας.

Έτσι, οι περιορισμοί προσδιορίζουν ποιες ιδιότητες πρέπει να μείνουν σε έγκυρες καταστάσεις και έγκυρες μεταβάσεις στην Βάση Γνώσεων και πώς μπορούν να εξασφαλιστούν αυτές οι ιδιότητες. Η διαδικασία επιβεβαίωσης των περιορισμών ακεραιότητας, αποτελείται από εσωτερικές διαδικασίες αποφάσεων, εάν όλοι οι περιορισμοί ικανοποιούνται σε μία κατάσταση που προήλθε από μία ενημέρωση. Ωστόσο, υπάρχουν μεγάλα εμπόδια στην διαδικασία επιβεβαίωσης των περιορισμών. Τα εμπόδια αυτά οφείλονται στον μεγάλο αριθμό περιορισμών ακεραιότητας και συμπερασματικών κανόνων, καθώς και στην πολυπλοκότητα της συμπερασματικής διαδικασίας, με την χρήση λογικής πρώτης τάξης.

Κατά την διάρκεια εκτέλεσης ενός συνόλου ενημερώσεων, οι μέθοδοι απλοποίησης των περιορισμών, που πρέπει να επαληθεύσει το σύστημα, έχουν σαν στόχο την παραγωγή απλούστερων εκφράσεων για να τους αναπαραστήσουν. Οι μέθοδοι αυτές, μπορούν να ταξινομηθούν σε δύο κύριες κατηγορίες. Το κριτήριο διάκρισης μεταξύ των δύο αυτών κατηγοριών είναι ο χρόνος κατά τον οποίο οι απλοποιήσεις συμβαίνουν. Έτσι, συναντάμε τις μεθόδους στις οποίες η απλοποίησης συντελείται κατά την διάρκεια της εκτέλεσης και σε αυτές τις οποίες η απλοποίησης

συντελείται κατά την διάρκεια της μετάφρασης. Στην πρώτη κατηγορία, δηλαδή, η απλοποίηση συμβαίνει κατά την διάρκεια της ενημέρωσης, ενώ στην δεύτερη κατά την διάρκεια του ορισμού της Βάσης.

Οι περισσότερες υπαρκτές μέθοδοι, δεν είναι πολύ αποδοτικές, γιατί ελέγχουν τους περιορισμούς μετά από κάθε ενημέρωση. Οι επαυξημένες μέθοδοι, όμως, επικεντρώνονται μόνο σε ένα υποσύνολο των περιορισμών, δηλαδή αυτούς που επηρεάζονται από μία ειδική κατηγορία ενημερώσεων. Οι μέθοδοι αυτές βασίζονται στην προϋπόθεση ότι η Βάση Γνώσεων δεν παραβιάζει κανένα περιορισμό στην κατάσταση που προηγείται της ενημέρωσης.

Στην εργασία μας θα ασχοληθούμε με αυτή την κατηγορία μεθόδων. Προτείνουμε μία μέθοδο απλούστευσης, κατά την διάρκεια της μετάφρασης, για χρονικούς περιορισμούς και συμπερασματικούς κανόνες. Πιο συγκεκριμένα θα ασχοληθούμε με την γραμμή έρευνας που σχετίζεται με την διατήρηση των περιορισμών με την βοήθεια των δοσοληψιών, όπως αναφέραμε και στην §1.1.

Η εξασφάλιση των περιορισμών, που εκφράζουν μία ιδιότητα η οποία πρέπει να ικανοποιείται συνεχώς, είναι απαραίτητο να δοκιμάζεται σε κάθε κατάσταση. Οι περιορισμοί αυτής της κατηγορίας ονομάζονται *περιορισμοί κατάστασης* (state constrains) και αναφέρονται σε μία μόνο κατάσταση της Βάσης Γνώσεων. Από την άλλη μεριά, οι δυναμικοί περιορισμοί, όπως προαναφέραμε, εκφράζουν ιδιότητες, που αναφέρονται σε οποιοδήποτε αριθμό συνεχών καταστάσεων και η επιβεβαίωσή τους απαιτεί την διατήρηση της ιστορικής μνήμης των μεταβάσεων μεταξύ των καταστάσεων στην Βάση Γνώσεων.

1.5 Συμπεράσματα

Τα αποτελέσματα της έρευνας αυτής, αναμένεται να έχουν σημαντική επίδραση τόσο στον ακαδημαϊκό κόσμο όσο και στον εμπορικό. Είναι φανερό ότι τα πληροφοριακά συστήματα γίνονται, συνεχώς, μεγαλύτερα, με ολοένα και περισσότερο πολύπλοκη δομή και σημασιολογία. Επίσης, οι χρήστες τους προέρχονται από, ολοένα και περισσότερο, ξένα μεταξύ τους επιστημονικά πεδία, διαθέτοντας διαφορετική παιδεία, μόρφωση, αλλά και εξοικείωση με τους ηλεκτρονικούς υπολογιστές, οπότε η ανομοιογένεια μεταξύ τους γίνεται συνεχώς

μεγαλύτερη. Λόγω όλων των προαναφερθέντων, γίνεται συνεχώς μεγαλύτερη η πιθανότητα μετά την εκτέλεση μίας εντολής του χρήστη να οδηγήσει σε λανθασμένη, μη αποδεκτή κατάσταση, την υπάρχουσα Βάση Δεδομένων. Στα συστήματα αυτά, λοιπόν, η χρήση μεθοδικών εργαλείων υποστήριξης για σχεδίαση δοσοληψιών καθίσταται πολύτιμη. Πρέπει να σημειωθεί ότι αυτή είναι η πρώτη προσπάθεια υλοποίησης μίας συστηματικής τεχνικής εξασφάλισης της διατήρησης των περιορισμών ακεραιότητας. Επίσης το αποτέλεσμα της εργασίας αυτής αποδεικνύει, πρακτικά, την ορθότητα και την χρησιμότητα των μεθόδων που κατασκευάστηκαν από τις εργασίες [Pl96], [PM96] και [Pl95b], τις οποίες παρουσιάσαμε προηγουμένως.

Κεφάλαιο 2^ο

Παρελθοντική δουλειά

Στο κεφάλαιο αυτό επικεντρώνουμε την προσοχή μας σε θεωρητικές μελέτες που έχει γίνει πάνω σε συστήματα διατήρησης των περιορισμών ακεραιότητας, για σχεσιακές, χρονικές, αντικειμενοστραφείς Βάσεις Δεδομένων και Βάσεις Γνώσεων. Η ενσωμάτωση γενικών περιορισμών σε συστήματα τόσο Βάσεων Δεδομένων, όσο και Βάσεων Γνώσεων είναι μία πραγματική απαίτηση των σύγχρονων εφαρμογών. Έτσι, θα μελετήσουμε μεθόδους που χειρίζονται αυτούς τους γενικούς περιορισμούς.

Ο έλεγχος της μη παραβίασης όλων των περιορισμών, μετά το πέρας κάθε ενημέρωσης, καθίσταται πρακτικά ανεπαρκής διαδικασία. Η ανεπάρκεια αυτή συνίσταται, κατά κύριο λόγο στο υψηλό κόστος της αντίστροφης διαδικασίας της ενημέρωσης, σε περίπτωση που κάποιος περιορισμός έχει παραβιαστεί ή κάποιο σφάλμα έχει συμβεί κατά την διάρκεια της ενημέρωσης. Υπάρχει, επίσης, η ανάγκη για έλεγχο πολλών στοιχείων κατά την διάρκεια της εκτέλεσης, όπως αναφέραμε και στο εισαγωγικό κεφάλαιο. Για τον λόγο αυτό, υιοθετούμε την ιδέα του σταδιακού ελέγχου των περιορισμών, που μας οδηγεί σε σημαντική απλοποίηση της διαδικασίας ελέγχου.

Οι περισσότερες μέθοδοι επιβολής των περιορισμών, στις παραπάνω κατηγορίες συστημάτων, μοιράζονται κάποιες κοινές αρχές που εκφράζουν την σημασία του σταδιακού ελέγχου, όπως προαναφέραμε. Μερικές από αυτές βρίσκονται διατυπωμένες στην εργασία [Ple96]:

- Η κατάσταση της Βάσης Δεδομένων πριν την ενημέρωση θεωρείται έγκυρη.
- Κάθε αποδεκτή μέθοδος πρέπει να εκμεταλλεύεται τις γνώσεις που παρέχονται από το σύστημα, που σχετίζονται με τις ενημερώσεις και τους περιορισμούς.
- Οι ενημερώσεις είναι ενσωματωμένες στους αρχικούς περιορισμούς, έτσι ώστε να υπάρχει η δυνατότητα οι περιορισμοί να αποτιμώνται στην τρέχουσα κατάσταση της Βάσης.
- Οι περιορισμοί είναι ενσωματωμένοι στις δοσοληψίες, που είναι υπεύθυνες για τις ενημερώσεις, έτσι ώστε οι τελευταίες να καθίστανται ασφαλείς.

2.1 Διατήρηση περιορισμών μέσω δοσοληψιών

Στην ενότητα αυτή θα εξαντλήσουμε τις μεθόδους που αποτέλεσαν τις σημαντικότερες επιρροές για την εργασία μας, στο πεδίο της διατήρησης των περιορισμών.

Οι περιορισμοί ακεραιότητας προσδιορίζουν τις έγκυρες καταστάσεις των δεδομένων μίας Βάσης Γνώσεων, όπως, επίσης, τις επιτρεπτές μεταβάσεις μεταξύ των καταστάσεων [F88]. Οι δομικοί περιορισμοί ακεραιότητας εκφράζουν ιδιότητες του μοντέλου δεδομένων που χρησιμοποιείται για αναπαράσταση γνώσεων. Οι σημασιολογικοί περιορισμοί ακεραιότητας, από την άλλη, ορίζονται από τον χρήστη και χρησιμοποιούνται για να εκφράσουν ιδιότητες του μοντελοποιούμενου πεδίου. Η διατήρηση των σημασιολογικών περιορισμών ακεραιότητας αποτελούν τροχοπέδη για την ταχύτητα εκτέλεσης εφαρμογών σε συστήματα διαχείρισης Βάσεων Δεδομένων. Η πλειοψηφία των εμπορικών συστημάτων δεν παρέχουν τίποτα άλλο εκτός από την επιβολή πολύ περιορισμένου αριθμού και είδους περιορισμών.

Η διατήρηση γενικών περιορισμών ακεραιότητας καθίσταται αρμοδιότητα του χρήστη ο οποίος υποβάλει τις επερωτήσεις στην Βάση Δεδομένων. Σε μεγάλες και σημασιολογικά πλούσιες Βάσεις Δεδομένων το πρόβλημα γίνεται ακόμα δυσκολότερο εξαιτίας του μεγάλου αριθμού περιορισμών και των αυθαίρετων ενημερώσεων. Η επεξεργασία των περιορισμών και των μεταβάσεων, κατά την διάρκεια της μετάφρασης, μειώνει το κόστους επιβολής των περιορισμών ακεραιότητας, κατά την διάρκεια της εκτέλεσης και αποτελεί μία ελπιδοφόρα μέθοδο για την επίτευξη αποδεκτών χρόνων εκτέλεσης.

Το πρόβλημα της διατήρησης των περιορισμών ακεραιότητας έχει δεχθεί αρκετά μεγάλη προσοχή στην βιβλιογραφία των προηγούμενων δύο δεκαετιών. Οι περισσότερες προτεινόμενες τεχνικές, για την διατήρηση των περιορισμών, βασίζονται στην παρακολούθηση από ένα πρόγραμμα γενικής χρήσης εφαρμογή ανεξάρτητο της εκάστοτε εφαρμογής που μας επιβεβαιώνει ότι οι ενημερώσεις δεν παραβιάζουν τους σημασιολογικούς περιορισμούς[BDM88], [JJ91], [Ple93], [Ple95].

2.1.1 Σχεσιακές Βάσεις Δεδομένων

Η πρώτη προσπάθεια δημιουργίας μίας διαδικασίας τροποποίησης δοσοληψιών, διατυπώθηκε στο άρθρο [Sto75]. Η εργασία αυτή πρότεινε την υλοποίηση μίας μεθόδου που εφαρμόζεται σε συστήματα, των οποίων οι περιορισμοί εκφράζονται σε γλώσσα QUEL. Οι περιορισμοί αυτοί είναι εκφράσεις που εισάγουν τις μεταβλητές τους με την βοήθεια καθολικών ποσοδεικτών. Η χρησιμότητα των ποσοδεικτών είναι μεγάλη, καθώς κάθε μεταβλητή εκφράζει κάποιο πεδίο του μοντέλου του συστήματος. Έτσι, κάθε περιορισμός μπορεί να ισχύει είτε για κάθε τιμή του πεδίου αυτού, είτε για κάποιο περιορισμένο εύρος τιμών του αντίστοιχου γνωρίσματος. Κάθε ενημέρωση μετασχηματίζεται, έτσι ώστε να απορροφάται από αυτήν η σημασιολογία των περιορισμών, που την αφορούν, αφού πρώτα δεχθούν κάποιες τροποποιήσεις. Με τον τρόπο αυτό, η μέθοδος εγγυάται ότι οι περιορισμοί δεν θα παραβιαστούν από την ενημέρωση αυτή.

Στο σημείο αυτό οφείλουμε να παρατηρήσουμε ότι σε κάθε ενημέρωση είμαστε υποχρεωμένοι να ελέγχουμε όλες τις πλειάδες, ώστε καμία να μην παραβιάζει τους περιορισμούς στους οποίους υπόκειται. Ωστόσο, όταν συμπεριλάβουμε τις συνθήκες όλων των σχετικών περιορισμών, σε κάθε ενημέρωση, επιτυγχάνουμε, πρακτικά, το ίδιο αποτέλεσμα με τον εξαντλητικό έλεγχο όλων των πλειάδων. Παρά το γεγονός ότι με την προσέγγιση αυτή η διατήρηση της συνέπειας της Βάσης -δηλαδή η επαλήθευση των περιορισμών- είναι εγγυημένη, οι εκφράσεις που περιγράφουν τους περιορισμούς δεν υφίστανται καμία απλοποίηση, οπότε καθίσταται εξαιρετικά πολύπλοκος ο υπολογισμός τους. Οι περιορισμοί, δηλαδή είναι απαραίτητο να απλοποιηθούν. Επιπλέον, το γεγονός ότι οι περιορισμοί, εξ ορισμού, ικανοποιούνται στην κατάσταση πριν από την ενημέρωση, δεν λαμβάνεται, καθόλου, υπόψη από την μέθοδο αυτή, ενώ, παράλληλα, δεν χρησιμοποιείται καμία γνώση των ιδιοτήτων της μοντελοποίησης του πεδίου εφαρμογής ή τους τύπους των δοσοληψιών. Τέλος, η τροποποίηση των περιορισμών, όπως προαναφέραμε, γίνεται κατά την διάρκεια της επεξεργασίας των επερωτήσεων, οδηγώντας το σύστημα σε μειωμένη απόδοση, κατά την εκτέλεσή του. Η μέθοδος αυτή ολοκληρώθηκε και δοκιμάστηκε για έλεγχο περιορισμών στο σύστημα Βάσης Δεδομένων της **Ingress**.

Παράλληλα, η εργασία [LTW93] παρουσιάζει το συντακτικό και τις ιδιότητες μίας γλώσσας, που επινοήθηκε για δήλωση και περιγραφή προδιαγραφών για

δοσοληψίες πάνω σε συστημάτων Βάσεων Δεδομένων. Μία βασική, όσο και πρωτοποριακή, έννοια της γλώσσας αυτής είναι η έννοια της *αδύναμης προαπαιτούμενης συνθήκης* (weakest precondition ή wp). Μία τέτοια συνθήκη παράγεται για κάθε ενημέρωση και κάθε περιορισμό, έτσι ώστε, εάν η συνθήκη αυτή είναι αληθής στην κατάσταση που προηγείται της ενημέρωσης, το σύστημα να παρέχει την εγγύηση ότι οι περιορισμοί θα είναι αληθείς στην κατάσταση που έπεται της ενημέρωσης αυτής.

Αν και οι παραγόμενες wp υπόκεινται, κατά περίπτωση, σε απλοποιήσεις, από το σύστημα, δεν υπάρχει συστηματική διαδικασία βελτιστοποίησης των συνθηκών αυτών. Το γεγονός αυτό οδηγεί σε ανεπιθύμητες καταστάσεις το σύστημα και υποχρεώνει τον χρήστη, ενδεχομένως, σε προσωπική παρέμβαση, ώστε να επιτευχθεί η βελτιστοποίηση των πολύπλοκων αυτών εκφράσεων. Η μόνη συστηματική μέθοδος βελτιστοποίησης, με την οποία είναι εφοδιασμένο ένα τέτοιο σύστημα, εφαρμόζεται μόνο σε ένα περιορισμένο υποσύνολο των στατικών περιορισμών. Τέλος, δεν υπάρχει καμία αναφορά στην εργασία αυτή για κατασκευή αδύναμων συνθηκών, όταν πολλαπλοί περιορισμοί σχετίζονται με μία ενημέρωση, κατάσταση πολύ επιθυμητή, που συναντάται αρκετά συχνά, στα συστήματα Βάσεων Δεδομένων.

Μία επιπλέον εργασία που μελετήσαμε, η εργασία [SS89], περιγράφει έναν μηχανισμό απόδειξης της ασφάλειας των δοσοληψιών λαμβάνοντας υπόψη το σύνολο των περιορισμών ακεραιότητας του δοθέντος συστήματος. Ενημερώσεις, που αποτελούν συνεπαγωγές άλλων, καθώς και δυναμικοί περιορισμοί, δεν λαμβάνονται υπόψη. Είναι φανερό, λοιπόν, ότι οι δύο αυτοί περιορισμοί στην χρήση του συστήματος, το καθιστούν ανεπαρκές για εφαρμογές των οποίων το πεδίο μοντελοποιείται με την χρήση χρονικών Βάσεων, όπου οι δυναμικοί περιορισμοί κατέχουν πρωτεύοντα ρόλο. Αν και η ασφάλεια των δοσοληψιών είναι αποδεδειγμένη κατά την διάρκεια της μετάφρασης, με την υλοποίηση της μεθόδου αυτής, στο αντίστοιχο σύστημα, ο προαναφερθείς μηχανισμός θα μπορούσε να εκμεταλλευτεί τις πληροφορίες που του παρέχει το σύστημα μετά την επεξεργασία των ενημερώσεων, οι οποίες λαμβάνουν μέρος σε κάθε δοσοληψία, με απώτερο στόχο, της επεξεργασίας αυτής, την απλοποίηση της διαδικασίας απόδειξης και ενδεχομένως, την πρόταση πιθανών αλλαγών για τον καθορισμό των προδιαγραφών της εκάστοτε δοσοληψίας.

Στον μηχανισμό αυτό, το πρόβλημα του πλαισίου, όπως είχαμε αναφέρει και στο προηγούμενο κεφάλαιο, αντιμετωπίζεται με την δέουσα προσοχή, όσον αφορά τα

2.1 Διατήρηση περιορισμών μέσω δοσοληψιών

κατηγορήματα που αλλάζουν από την ενημέρωση. Επιπλέον, ακολουθεί απαλοιφή των αδρανών όρων, από τα θεωρήματα που πρέπει να αποδείξουμε, προκειμένου να επιβεβαιώσουμε τους περιορισμούς.

Με το πρόβλημα των επιπτώσεων, εξάλλου, ασχολήθηκαν πολλές εργασίες τα τελευταία χρόνια. Σε μία ακόμα εργασία, στην θεωρητική δουλειά [PM96], εισάγεται ο όρος *επιπτώσεις* (ramification). Στην εργασία αυτή αντιμετωπίζεται το πρόβλημα της διατήρησης των περιορισμών με την βοήθεια δοσοληψιών συστημάτων Βάσεων Δεδομένων μεγάλου μεγέθους, με την βοήθεια νέων τεχνικών τροποποίησης των δοσοληψιών. Οι τροποποιήσεις αυτές, συντελούνται κατά την διάρκεια της μετάφρασής τους, προκειμένου να επιτευχθεί οικονομία χρόνου, και, κατά συνέπεια, καλύτερη απόδοση του συστήματος. Υιοθετήθηκαν, επίσης, μερικές βασικές έννοιες από τον χώρο της Τεχνητής Νοημοσύνης.

Η διαδικασία που περιγράφεται, στηρίζεται στην παρατήρηση ότι μία λύση στα προβλήματα του πλαισίου και των επιπτώσεων θα μπορούσε να χρησιμοποιηθεί κατά την διάρκεια της σχεδίασης δοσοληψιών. Η τεχνική που υιοθετήθηκε, συγχωνεύει τους προσδιορισμούς των δοσοληψιών με συνθήκες, απαραίτητες για την επαλήθευση των περιορισμών που αφορούν τις δοσοληψίες αυτές. Με κάθε δοσοληψία συγχωνεύονται ορισμένες μετασυνθήκες, οι οποίες είναι αληθείς μετά την εκτέλεση των δοσοληψιών αυτών και έχουν σαν αποτέλεσμα την διατήρηση των αντίστοιχων περιορισμών. Οι συνέπειες των περιορισμών ανιχνεύονται από ένα μηχανισμό που ονομάζεται *προσδιοριστής δοσοληψιών* (transaction specifier), με αποτέλεσμα να αποφεύγεται η διαδικασία απόδειξης της ορθότητας των δοσοληψιών, εφόσον οι συνθηκών που προστίθενται, είναι εγγυημένα αληθείς. Επιπλέον, στην εργασία αυτή περιγράφεται η επεξεργασία τόσο των άμεσων όσο και των έμμεσων αποτελεσμάτων των διαφόρων πράξεων (direct-indirect effect), με σκοπό την πιο αποτελεσματική επίλυση του προβλήματος του πλαισίου και των επιπτώσεων.

2.1.2 Αντικειμενοστραφείς Βάσεις Δεδομένων

Στην εργασία [FCT88], εφαρμόστηκαν ορισμένες θεμελιώδεις αρχές σχεδίασης αντικειμενοστραφών Βάσεων με σκοπό την διατήρηση της συνέπειας της Βάσης. Η λογική συνέπεια της Βάσης μπορεί να εξασφαλιστεί, εάν ο μόνος τρόπος ενημέρωσής της είναι διαμέσου ορισμένων τελεστών, ειδικής χρήσης, που

ονομάζονται *τελεστές διατήρησης περιορισμών*. Οι τελεστές αυτοί ενσωματώνονται στα αντικείμενα και στους περιορισμούς, επιτρέποντας μόνο όσες ενημερώσεις δεν παραβιάζουν κάποιον(-ους) από τους περιορισμούς που αφορούν το συγκεκριμένο αντικείμενο.

Παράλληλα, στην εργασία [Cas89], παρουσιάζεται ένα τυπικό σύστημα δημιουργίας διεργασιών από συμπερασματικούς κανόνες και περιορισμούς, που ονομάζονται *δαίμονες*(demons), πάνω σε αντικειμενοστραφείς Βάσεις. Οι κανόνες και οι περιορισμοί προσδιορίζονται με την χρήση κατηγορηματικού λογισμού και μετασχηματίζονται σε εκφράσεις σχεσιακής άλγεβρας. Ένας μεταφραστής, εφοδιασμένος με λογική, κατασκευάζει για τον χρήστη, κατάλληλες διαδικασίες επίλυσης των προβλημάτων του. Το εργαλείο αυτό μπορεί να παράγει κώδικα κατάλληλο να χρησιμοποιηθεί για την διαδικασία ενημέρωσης ή ακόμα και επιβολής των περιορισμών πάνω στο σύστημα που εφαρμόζεται. Ωστόσο, μέχρι σήμερα μετά από πειράματα πάνω στο σύστημα αυτό, οι μόνες διαδικασίες που υποστηρίζονται είναι η εισαγωγή και η ανάκτηση πληροφορίας από μία συσχέτιση.

2.1.3 Βάσεις Γνώσεων

Στην κατηγορία των Βάσεων Γνώσεων, η έρευνα στον τομέα της διατήρησης των περιορισμών είναι ακόμα σε πολύ πρώιμα στάδια. Το πρώτο σύστημα που μελετήσαμε ονομάζεται **Taxis**[CRZNM88]. Στο σύστημα αυτό ο όρος *διαβεβαιώσεις*(assertion) περιγράφει συνθήκες που προσδιορίζεται σαν αναλλοίωτες συνθήκες κλάσεων που αποτελούνται από δεδομένα ή διαδικασίες. Οι περιορισμοί ακεραιότητας μεταφράζονται όπως ακριβώς οι προαπαιτούμενες συνθήκες, στην περίπτωση των δοσοληψιών, τις οποίες έχουμε περιγράψει σε προηγούμενη ενότητα και θα αναφερθούμε και σε επόμενο κεφάλαιο.

Είναι κατανοητό ότι εάν το σύστημα αυτό παρείχε, στους χρήστες του, άπειρη εκφραστικότητα για την δήλωση των περιορισμών ακεραιότητας, δεν θα μπορούσε να επιτύχει αποτελεσματικές διαδικασίες επιβολής των περιορισμούς του. Το πρόβλημα αυτό προσπαθεί να επιλύσει η αντίστοιχη γλώσσα διαβεβαιώσεων επιβάλλοντας κάποια επιθυμητά όρια, όσον αφορά την δυνατότητα έκφρασης των περιορισμών ακεραιότητας του συστήματος. Ειδικότερα, δεν επιτρέπεται ποσοδείκτες, ενώ όσα γνωρίσματα λαμβάνουν μέρος στην επιλογή μονοπατιών για

2.1 Διατήρηση περιορισμών μέσω δοσοληψιών

κάποια αναζήτηση ή ενημέρωση μέσα στην Βάση, δεν επιτρέπεται να ανήκουν σε μετακλάσεις, γιατί αναμένεται η έκτασή τους να είναι μεγάλη, δημιουργώντας προβλήματα λόγω της μεγάλης πολυπλοκότητας που ενδεχομένως να εισάγουν στους υπολογισμούς.

Στο σημείο αυτό, ένας μεταφραστής, παράγει ένα σύνολο από διαδικασίες, για κάθε γνώρισμα και κάθε λειτουργία, κατά την διάρκεια της εκτέλεσης κάθε εντολής του συστήματος. Επιπλέον, ο μεταφραστής παράγει συνθήκες, οι οποίες εγγυώνται την ικανοποίηση των περιορισμών, κατ' αντιστοιχία με τις προστιθέμενες συνθήκες που περιγράψαμε, σε εργασίες που αναφέραμε προηγουμένως. Επιπλέον, το μεταφραστικό αυτό εργαλείο παράγει κώδικα, που βελτιστοποιεί τις εκφράσεις ελέγχου των διαβεβαιώσεων, μειώνοντάς, έτσι, τον χρόνο απόκρισης του συστήματος. Κατόπιν, ένα τμήμα του τελικού εργαλείου, που λειτουργεί με *μηχανισμούς πυροδότησης σκανδάλης (triggering)*, συσχετίζει τον παραγόμενο κώδικα με τα αντικείμενα, στα οποία αντιστοιχεί.

Παρόμοια, το σύστημα **KRISYS**, που περιγράφεται στις εργασίες [Mat91] και [Des90], χρησιμοποιεί ειδικής χρήσης περιορισμούς (ad-hoc constrains), για να εμπλουτίσει τους ήδη υπάρχοντες, με μία ειδική γλώσσα, που επίσης περιγράφεται στην εργασία αυτή. Οι δαίμονες που δημιουργούνται, από το σύστημα, εκτελούνται αυτόματα όταν συμβούν κάποια γεγονότα που επηρεάζουν το αντικείμενα με τα οποία είναι συσχετισμένοι, είτε όταν τα γεγονότα είναι εξωτερικά, όπως η εκτέλεση κάποιας δοσοληψίας, είτε ακόμα όταν τα γεγονότα είναι εσωτερικά, όπως η αλλαγή της τιμής ενός πεδίου σε περίπτωση που αυτή επηρεάζει και άλλα γνωρίσματα. Καμία βελτιστοποίηση δεν παρατηρείται κατά την εκτέλεση του συστήματος αυτού, ενώ ο σωστός ορισμός των δαιμόνων παραμένει αποκλειστική αρμοδιότητα του σχεδιαστή τους, επιτρέποντας διαφορετικές προσεγγίσεις στο ίδιο σύνολο περιορισμών, που σαν αποτέλεσμα έχουν την έλλειψη ομοιομορφίας σε συγγενικά συστήματα ή ακόμα και μέσα στο ίδιο σύστημα.

2.1.4 Χρονικές Βάσεις Δεδομένων

Μία πρώτη προσπάθεια για την χρησιμοποίηση μηχανισμών πυροδότησης σκανδάλης, με σκοπό την διατήρηση των περιορισμών με την βοήθεια δοσοληψιών, έγινε στην εργασία [EL84]. Οι περιορισμοί προσδιορίζονται με την βοήθεια χρονικού

λογισμού. Παράλληλα, μετασηματίζονται σε δηλώσεις που εισάγονται από μία γλώσσα και έχουν την μορφή $\langle \text{on } \phi \text{ do } \text{op} \rangle$, όπου η ϕ είναι ένας τύπος που εκφράζει μία συνθήκη, ενώ op είναι μία εντολή προγράμματος. Η δήλωση αυτή, δηλαδή, εκφράζει το γεγονός ότι εάν ικανοποιηθεί η συνθήκη ϕ , τότε, αυτόματα, το πρόγραμμα θα εκτελέσει την εντολή op .

Άλλες μέθοδοι έχουν αναπτυχθεί βασιζόμενες στην ιδέα της παραγωγής μηχανισμών πυροδότησης σκανδάλης από ένα σύνολο χρονικών περιορισμών. Στην εργασία [GL93], οι συνθήκες πυροδότησης παράγονται από ένα γράφο μεταβάσεων που χτίζεται για να αναπαραστήσει τον αποδεκτό χρόνο ζωής, των διαφόρων αντικειμένων της Βάσης, λαμβάνοντας υπόψη και τους περιορισμούς που αντιστοιχούν στο καθένα. Ο γράφος μας δείχνει ποιες είναι οι αποδεκτές μεταβάσεις μεταξύ των καταστάσεων του συστήματος και κάτω από ποιες συνθήκες αυτές οι μεταβάσεις είναι όχι μόνο εφικτές, αλλά και επιθυμητές.

Η ανάλυση των περιορισμών χρησιμοποιείται για να παράγει τους μηχανισμούς πυροδότησης σκανδάλης, που σκοπό έχουν την διατήρηση των περιορισμών ακεραιότητας των αντίστοιχων δοσοληψιών. Βλέπουμε στο σημείο αυτό την ομοιότητα στις κύριες επιδιώξεις, ανάμεσα στις πολλές, διαφορετικές μεταξύ τους προσεγγίσεις, που περιγράψαμε ως τώρα. Σε όλες αυτές τις προσεγγίσεις κύριο μέλημα είναι η εύρεση ενός μηχανισμού διατήρησης των περιορισμών ακεραιότητας, είτε αυτή γίνεται μέσω εμπλουτισμού των προδιαγραφών, όπως στην § 2.1.1, είτε γίνεται με την βοήθεια διαδικασιών δαιμόνων, όπως στην §2.1.2.

Αναφερόμενοι πάλι στην μέθοδο που περιγράφουμε, πρέπει να επισημάνουμε ότι η μέθοδος αυτή εκμεταλλεύεται το δεδομένο ότι οι δυναμικοί περιορισμοί ικανοποιούνται στην κατάσταση που προηγήθηκε της εκτέλεσης της δοσοληψίας. Αυτή η γνώση μπορεί να χρησιμοποιηθεί για να παραχθούν απλοποιημένες εκδόσεις των συνθηκών που περιγράφουν τους περιορισμούς και παραμένουν ανεπιβεβαιώτες. Με τον τρόπο αυτό, μπορούν να δημιουργηθούν συστηματικά όλοι οι μηχανισμοί πυροδότησης σκανδάλης, με την χρήση του γράφου μεταβάσεως καταστάσεων. Οι περιορισμοί σχετίζονται άμεσα με τη ύπαρξη χρονικών διαστημάτων ζωής των αντικειμένων, περιορίζοντας, έτσι, τις καταστάσεις στις οποίες είναι απαραίτητο να επιβεβαιωθούν οι περιορισμοί, όπου φυσικά, τα αντικείμενα είναι ορισμένα. Το μειονέκτημα της μεθόδου αυτής είναι ότι πρέπει να συντηρείται ένας μεγάλος αριθμός γράφων μεταβάσεων, δημιουργώντας έτσι πρόβλημα στη απαιτούμενη

2.1 Διατήρηση περιορισμών μέσω δοσοληψιών

μνήμη ενώ οι εγγυημένοι μηχανισμοί πυροδότησης σκανδάλης δεν είναι οι καλύτεροι δυνατοί.

Η διατήρηση των περιορισμών που περιλαμβάνουν και την χρονική παράμετρο, διαμέσου τροποποιήσεων των δοσοληψιών, εξετάζεται πιο λεπτομερώς και στην εργασία [Lip90]. Οι δυναμικοί περιορισμοί προσδιορίζονται με την χρήση Χρονικής Προτασιακής Λογικής. Οι περιορισμοί μετασχηματίζονται σε γράφους ακολουθώντας τον παραδοσιακό τρόπο αναπαράστασης. Οι γράφοι αυτοί περιγράφουν την διάρκεια ζωής των αντικειμένων της Βάσης, λαμβάνοντας υπόψη τους αντίστοιχους περιορισμούς. Οι διαδρομές μέσα στους γράφους αυτούς, αντιστοιχούν σε μία ακολουθία έγκυρων μεταβάσεων μεταξύ των καταστάσεων. Μία τέτοια ακολουθία μεταβάσεων ονομάζεται έγκυρη, όταν και μόνο όταν είναι σύμφωνη με τους περιορισμούς που έχουν επιβληθεί από το σύστημα, μέχρι την τρέχουσα κατάσταση. Στο τελικό στάδιο, οι γράφοι μεταβάσεων μετασχηματίζονται, με την αντίστροφη διαδικασία παραγωγής τους, σε περιγραφές δοσοληψιών, έτσι ώστε, οι τελευταίες, να μπορούν να εκτελεστούν. Σύμφωνα με αυτό τον τρόπο αναπαράστασης, οι δοσοληψίες προσδιορίζονται από ζεύγη προσυνθηκών και μετασυνθηκών (precondition και post condition αντίστοιχα). Τις δύο αυτές κατηγορίες συνθηκών τις συναντήσαμε και σε προηγούμενες εργασίες, γεγονός που δηλώνει την πρωταρχική σημασία τους στον ορισμό και στην εκτέλεση δοσοληψιών. Οι συνθήκες, αυτές εκφράζονται σε Κατηγορηματική Λογική.

Τέλος, στις εργασίες [SW95a] και [SW95b], για την εξασφάλιση της επαλήθευσης των περιορισμών ακεραιότητας χρησιμοποιήθηκαν χρονικοί μηχανισμοί πυροδότησης σκανδάλης, που εκφράζονται σε Μελλοντική Χρονική Λογική (FTL). Στις εργασίες αυτές, προτείνεται μία μέθοδος αποτίμησης των συνθηκών πυροδότησης σκανδάλης με παράλληλο και όχι σειριακό τρόπο, όπως συμβαίνει στα περισσότερα συστήματα Βάσεων Δεδομένων. Προτείνεται, δηλαδή, η αποτίμηση περισσότερων του ενός τμήματος των συνθηκών, ταυτόχρονα, όπου και εάν αυτή η διαδικασία είναι εφικτή.

Οι εκφράσεις, που αναπαριστούν τους περιορισμούς, μεταφράζονται σαν ένα σύνολο καταστάσεων, λαμβάνοντας υπόψη και την ιστορική συνέχεια της Βάσης που μοντελοποιείται. Κάθε έγκυρη κατάσταση απορρέει από την προηγούμενη μετά από την εκτέλεση μίας ενημέρωσης. Αν θεωρήσουμε σε ένα υποτιθέμενο γράφο, σαν κόμβους τις έγκυρες καταστάσεις του συστήματος, τότε οι ακμές που ενώνουν τους

κόμβους, δηλώνουν την εκτέλεση των αντίστοιχων ενημερώσεων. Η χρονική παράμετρος, που αντιστοιχεί σε κάθε κατάσταση, δηλώνει την χρονική στιγμή που έγινε η ενημέρωση. Έτσι, για τον έλεγχο επαλήθευσης των συνθηκών, που επιβάλλονται από τον κάθε μηχανισμό πυροδότησης σκανδάλης, το σύστημα πρέπει να αποτιμήσει κάθε προγενέστερη συσχετιζόμενη συνθήκη, αρχίζοντας από την πρώτη κατάσταση την οποία ο μηχανισμός, πυροδότησης σκανδάλης, επηρεάζει και τελειώνοντας στην τρέχουσα κατάσταση.

Για βελτίωση του χρόνου απόκρισης του συστήματος, εις βάρος όμως του κόστους αποθήκευσης, οι επερωτήσεις, που εμφανίζονται στις συνθήκες, αποτελούν υλοποιημένες όψεις των δεδομένων του συστήματος.

Στην ίδια εργασία, υιοθετείται και ο όρος *σύνολα απαιτήσεων*. Τα σύνολα αυτά διατηρούνται για κάθε μηχανισμό πυροδότησης σκανδάλης, ξεχωριστά και περιέχουν συνθήκες οι οποίες, εάν ικανοποιηθούν από μία μελλοντική αλληλουχία μεταβάσεων μεταξύ των μεταβάσεων, θα προκαλέσουν ενεργοποίηση των μηχανισμό πυροδότησης της αντίστοιχης σκανδάλης. Βασική προϋπόθεση είναι ότι τουλάχιστον ένα από τα σύνολα απαιτήσεων πρέπει να ικανοποιείται. Ο αλγόριθμος αντικαθιστά, με συστηματικό τρόπο, τις τρέχουσες συνθήκες με υποσυνθήκες, έτσι ώστε, σταδιακά, μερικές από τις συνθήκες να επαληθεύονται στην τρέχουσα κατάσταση, ενώ άλλες να επαληθεύονται σε μεταβάσεις μεταξύ μελλοντικών μεταβάσεων. Για να βελτιστοποιηθεί η διαδικασία επαλήθευσης των συνθηκών, απαιτήσεις, που δεν ικανοποιούνται, διαγράφονται.

2.2 Παράλληλη βιβλιογραφία

2.2.1 Καταστατικός λογισμός (Situation calculus)

Υπάρχουν ορισμένες εργασίες πάνω σε ένα πολύ σημαντικό είδος λογισμού, τον Καταστατικό. Μία από τις πλέον ενδεικτικές, είναι η εργασία [Rei95]. Στην συγκεκριμένη εργασία, μεγάλο ενδιαφέρον παρουσιάζει η αναπαράσταση των δοσοληψιών μίας Βάσης Δεδομένων, σύμφωνα με εκφράσεις αυτού του είδους Λογισμού. Το ενδιαφέρον του συγγραφέα επικεντρώνεται στη αποτύπωση της

εξέλιξης της Βάσης, κάτω από την επίδραση μίας αυθαίρετης ακολουθίας ενημερώσεων.

Παρέχεται, εξάλλου, μία λύση στο πρόβλημα του πλαισίου, όπως αυτό εμφανίζεται στην περιοχή εφαρμογής των δοσοληψιών των Βάσεων Δεδομένων, βασισμένη στην αντίστοιχη επίλυση του προβλήματος αυτού, στο πεδίο της Τεχνητής Νοημοσύνης. Η προσέγγιση αυτή πλησιάζει πολύ με την προτεινόμενη διαδικασία της προαναφερθείσας εργασίας [Ple96], η οποία είχε στόχο, επίσης, την προσπάθεια επίλυσης μίας έκφρασης του ίδιου προβλήματος. Η προσέγγιση αυτή αποτελεί λύση για ένα περιορισμένο, αλλά, ουσιαστικά, το σημαντικότερο σύνολο ενημερώσεων. Η λύση αυτή αποδεικνύει την μη παραβίαση τόσο των στατικών, όσο και των δυναμικών περιορισμών, με την χρήση μαθηματικών μεθόδων. Παράγεται ένα σύνολο συναρτήσεων ικανών να αποφασίσουν αν μία ακολουθία ενημερώσεων οδηγεί την Βάση σε ασυνεπής κατάσταση. Επίσης παράγει συναρτήσεις για αποτίμηση ορισμένων επερωτήσεων, πάνω στην, ήδη ενημερωμένη, Βάση. Ένα ενδεικτικό παράδειγμα τυποποίησης περιορισμών είναι το ακόλουθο [Rei95]: $\forall s, s', \forall em, sal_1, sal_2 S_0 \leq s \wedge s' \leq s' \wedge sal(em, s, sal_1) \wedge sal(em, s, sal_2) \Rightarrow sal_1 \leq sal_2$, όπου s, s' είναι καταστάσεις του συστήματος, ενώ S_0 είναι η αρχική κατάσταση της Βάσης που διαθέτουμε. Ο συγκεκριμένος περιορισμός ακεραιότητας, δηλώνει ότι για κάθε εργαζόμενο, που αντιστοιχεί στην ελεύθερη μεταβλητή em , σε κάθε κατάσταση της Βάσης, s' , αντιστοιχεί μισθός sal_2 , μεγαλύτερος από τον μισθό sal_1 που του αντιστοιχούσε στην προηγούμενη κατάσταση s , της Βάσης. Η ικανοποίηση περιορισμών, όπως ο προηγούμενος, μπορούν να αποδειχθούν λαμβάνοντας υπόψη απλά και μόνο την αρχική κατάσταση της Βάσης. Ωστόσο, η προσέγγιση αυτή δεν εκμεταλλεύεται την πρότερη γνώση, ότι, δηλαδή, οι περιορισμοί επαληθεύονται σε όλες τις προηγούμενες καταστάσεις πριν την εκτέλεση της δοσοληψίας, οδηγώντας το σύστημα σε εκτέλεση περιττών πράξεων πάνω στις καταστάσεις της Βάσης. Έτσι, για την απόδειξη κάθε περιορισμού, η διαδικασία απόδειξης αρχίζει από την αρχική κατάσταση της Βάσης και όχι από την προηγούμενη, όπως κάνουν πολλές παρόμοιες μέθοδοι.

Τέλος, μία εργασία, προγενέστερη της προαναφερθείσας, είναι η [Rei91]. Η εργασία αυτή ασχολήθηκε, με την επίλυση του προβλήματος του πλαισίου στην περιοχή των Βάσεων Δεδομένων, στα πλαίσια του Καταστατικού Λογισμού. Η εργασία αυτή έχει σαν κύριο στόχο την αναλυτική παρουσίαση του προβλήματος του

πλαisiού, στον χώρο των Βάσεων Δεδομένων, καλύπτοντας όλες του τις διαστάσεις. Επιπλέον, ένα ενδιαφέρον συμπέρασμα, της εργασίας αυτής, είναι οτι μπορούμε να επιλύσουμε το παραπάνω πρόβλημα, με αυτοματοποιημένο τρόπο, με την συστηματική δημιουργία κατάλληλων αξιωμάτων, όπως μας απέδειξε η εργασία αυτή.

2.2.2. Άλλες εργασίες

Μία, εξίσου σημαντική, εργασία στον χώρο της διατήρησης των περιορισμών ακεραιότητας, είναι η εργασία [BMR95], η οποία δείχνει γιατί το πρόβλημα του πλαisiού είναι ιδιαίτερα σημαντικό στην περιγραφή προδιαγραφών αντικειμενοστραφών Βάσεων. Η ιδιαιτερότητα που συναντάμε στην περίπτωση αυτή είναι οτι οι δοσοληψίες περιγράφονται από υπερκλάσεις και κληρονομούνται σε υποκλάσεις. Τα αποτελέσματα επεξεργασίας των κανόνων, πρέπει να ληφθούν υπόψη και να ελεγχθούν σύμφωνα με τους περιορισμούς ακεραιότητας. Σύμφωνα με εκτεταμένη έρευνα γύρω από την σχετική βιβλιογραφία, δεν υπάρχει καμία προηγούμενη προσπάθεια σύνδεσης του προβλήματος των επιπτώσεων, με το πρόβλημα ασφάλειας των δοσοληψιών. Στα πρώτα βήματα της μεθοδολογίας, που παρουσιάζεται στην εργασία αυτή, οι προδιαγραφές των δοσοληψιών μετασχηματίζονται, ώστε να εμπεριέχουν την σημασιολογία των περιορισμών. Την τακτική αυτή την συναντήσαμε ακόμα και στις σχεσιακές Βάσεων Δεδομένων, αλλά και στις Βάσεις Γνώσεων. Εάν οι νέες προδιαγραφές δεν επαληθεύονται, τότε οι περιορισμοί είναι βέβαιο οτι θα παραβιαστούν κατά την εκτέλεση των αντίστοιχων δοσοληψιών. Με την ενσωμάτωση της σημασιολογίας αυτής στις νέες προδιαγραφές, δεν είναι πλέον απαραίτητος ο έλεγχος των περιορισμών κατά την εκτέλεση του προγράμματος, ενώ το κέρδος στην απόδοση του συστήματος είναι σημαντικό.

Κεφάλαιο 3^ο

Πρόβλημα πλαισίου και επιπτώσεων

Το πρόβλημα της διατήρησης των περιορισμών ακεραιότητας, όπως αναφέραμε και σε προηγούμενη ενότητα, έχει δεχθεί αρκετά μεγάλη προσοχή στην βιβλιογραφία των προηγούμενων δύο δεκαετιών. Οι περισσότερες προτεινόμενες τεχνικές, για την διατήρηση των περιορισμών, βασίζονται στην παρακολούθηση από ένα πρόγραμμα γενικής χρήσης εφαρμογή ανεξάρτητο της εκάστοτε εφαρμογής που μας επιβεβαιώνει ότι οι ενημερώσεις δεν παραβιάζουν τους σημασιολογικούς περιορισμούς.

Μία εξίσου δημοφιλής προσέγγιση βασίζεται στην διατήρηση μέσω των δοσοληψιών, η οποία αποτελεί πηγή έμπνευσης και για την παρούσα εργασία [Li90], [CW90]. Η πρώτη τεχνική απαλλάσσει τον χρήστη από την επιβάρυνση της υλοποίησης δοσοληψιών προς την Βάση, ώστε κανένας περιορισμός να μην παραβιάζεται. Η δεύτερη τεχνική απαιτεί οι προδιαγραφές των δοσοληψιών ενημέρωσης να δίνονται εκ των προτέρων. Οι δοσοληψίες τροποποιούνται, έτσι, ώστε να υπάρχει εγγύηση επαλήθευσης των περιορισμών της Βάσης μετά από την εκτέλεση κάθε ακολουθίας δοσοληψιών.

Σε προηγούμενες εργασίες [Ple93] και [Ple95], παρουσιάστηκε μία μέθοδος απλοποίησης κατά την διάρκεια του χρόνου μετάφρασης για Βάσεις Δεδομένων, όπου οι περιορισμοί ακεραιότητας, οι οποίοι προσδιορίζονται από μία χρονική γλώσσα [BMJ90], [Ple93b], εξειδικεύονται και απλοποιούνται λαμβάνοντας υπόψη τις προβλεπόμενες δοσοληψίες. Στην εργασία αυτή θα παρουσιάσουμε μία διπλή προσέγγιση, που ονομάζεται *τροποποίηση δοσοληψιών κατά την μετάφραση* (compile-time transaction modification), σε περιβάλλον συστημάτων Βάσεων Γνώσεων που περιέχουν περιορισμούς ακεραιότητας και παραγωγικούς κανόνες. Πιο συγκεκριμένα, εξετάζουμε πώς οι δοσοληψίες, που περιγράφονται από το ζεύγος προσυνθηκών και μετασυνθηκών (precondition-post condition), μπορεί να τροποποιηθεί, ούτως ώστε οι περιορισμοί, που έμμεσα ή άμεσα επηρεάζουν, διατηρούνται και στην κατάσταση που οδηγείται το σύστημα μετά την εκτέλεση της δοσοληψίας.

Προσδιορίζοντας τις δοσοληψίες της Βάσης Δεδομένων με τις προαναφερθείσες συνθήκες, επιτρέπεται στον σχεδιαστή να δηλώσει **ποιο** θα είναι το αποτέλεσμα μίας δοσοληψίας και όχι **πως** αυτό θα επιτευχθεί. Τα σύνολα προσδιορισμών των Βάσεων Δεδομένων θα πρέπει να δίνονται σε μία προκαθορισμένη μορφή. Η γλώσσα που προσδιορίζει την μορφή αυτή θα πρέπει να συνοδεύεται από τον απαραίτητο μηχανισμό απόδειξης των ιδιοτήτων των προδιαγραφών. Έτσι, επιλέξαμε την Χρονική Λογική Πρώτης Τάξεως(ή T.F.O.L.), για να εκφράσουμε τις συνθήκες που προαναφέραμε, τους περιορισμούς και τους παραγωγικούς κανόνες. Αντιμετωπίζουμε το πρόβλημα της απόδειξης της διατήρησης των περιορισμών σχετίζοντάς το με το πρόβλημα του πλαισίου και των επιπτώσεων.

Στην πλειοψηφία των υπάρχοντων μεθόδων για διατήρηση περιορισμών ακεραιότητας μέσω δοσοληψιών, το πρόβλημα του πλαισίου και των επιπτώσεων έχει είτε αγνοηθεί, είτε προσπεραστεί με την έννοια αυθαίρετων συμπερασμάτων που εκφράζουν ότι «...τίποτα άλλο δεν αλλάζει, εκτός από αυτά που έχουν ρητά δηλωθεί ότι αλλάζουν κατά την διάρκεια μίας ενημέρωσης...». Με δεδομένο, λοιπόν, ένα σύνολο προδιαγραφών για δοσοληψίες, το πρόβλημα της περιληπτικής δήλωσης ότι «...και τίποτα άλλο δεν αλλάζει...», ονομάζεται *πρόβλημα πλαισίου*(frame problem). Το *πρόβλημα των επιπτώσεων*(ramification problem) αναφέρεται στην ανεύρεση μίας μεθόδου αποφυγής της ανάγκης επιπλέον προσδιορισμού έμμεσων αποτελεσμάτων μίας ενέργειας. Πολλές προσεγγίσεις επίλυσης, αυτών των προβλημάτων, δημοσιεύθηκαν τα τελευταία χρόνια, κυρίως στο επιστημονικό πεδίο της Τεχνητής Νοημοσύνης.

Το άρθρο [LR94] παρουσιάζει μία λύση στο πρόβλημα του πλαισίου με εφαρμογές σε συστήματα ενημερώσεων Βάσεων Δεδομένων. Η λύση περιέχει την συστηματική παραγωγή ενός συνόλου αξιωμάτων που ονομάζονται αξιώματα επόμενης κατάστασης(*successor state axioms*), τα οποία περιγράφουν ολοκληρωτικά πώς ορισμένα κατηγορήματα μεταβάλλουν την τιμή αληθείας τους, από κατάσταση σε κατάσταση, σαν αποτέλεσμα της εκτέλεσης ορισμένων δοσοληψιών. Μία γεννήτρια δημιουργίας τέτοιων αξιωμάτων, με την παρουσία περιορισμών και συγκεκριμένων ορισμών για μη τερματικά κατηγορήματα, προτείνεται στην εργασία [Pi94]. Στην εργασία μας συσχετίζουμε την λύση αυτή, με το πρόβλημα της διατήρησης των περιορισμών.

Στις εργασίες [BMR93] και [BMR95], εξηγούνται οι λόγοι για τους οποίους το πρόβλημα του πλαισίου γίνεται τόσο έντονο στους προσδιορισμούς

αντικειμενοστραφών Βάσεων, όπου οι δοσοληψίες περιγράφονται από υπερκλάσεις και κληρονομούνται σε υποκλάσεις. Τα επιχειρήματα που παρουσιάζονται εδώ, εφαρμόζονται απευθείας στον προσδιορισμό δοσοληψιών Βάσεων δεδομένων. Τα αποτελέσματα της αποτίμησης ή πυροδότησης των κανόνων πρέπει να ληφθούν υπόψη και να ελεγχθούν για τους περιορισμούς ακεραιότητας. Επιπλέον, οι προδιαγραφές των δοσοληψιών μπορούν να προσδιοριστούν από υπερκλάσεις σε υποκλάσεις και να συγχωνευτούν, ώστε να σχηματίσουν σύνθετες δοσοληψίες. Τα αποτελέσματα των νέων αυτών δοσοληψιών πρέπει να χαρακτηριστούν επακριβώς.

Σύμφωνα με τα υπάρχοντα δεδομένα, δεν έχει γίνει καμία προηγούμενη προσπάθεια σύνδεσης του προβλήματος των επιπτώσεων με την απόδειξη της ασφάλειας των δοσοληψιών. Στην πραγματικότητα, μία επίλυση του προβλήματος των επιπτώσεων μπορεί να προτείνει μία στρατηγική για την διατήρηση των περιορισμών ακεραιότητας. Αρχικά εμπλουτίζουμε τις προδιαγραφές των δοσοληψιών ώστε να περιλαμβάνει μία υλοποίηση των περιορισμών(οι οποίοι μπορεί να είναι απλούστερες μορφές εκφράσεων). Εάν οι νέες προδιαγραφές δεν επαληθεύονται, τότε οι περιορισμοί παραβιάζονται. Η διαδικασία εμπλουτισμού δηλώνει ότι οι περιορισμοί δεν είναι απαραίτητο να ελεγχθούν κατά την διάρκεια της εκτέλεσης της εφαρμογής.

Η επόμενη παράγραφος προσπαθεί να επιτύχει μία ανασκόπηση των θεμάτων σχετικά με τις *επιπτώσεις*(ramification) και περιγράφει μία επέκταση στην επίλυση των προβλημάτων του πλαισίου και των επιπτώσεων, για την περίπτωση συγκεκριμένων προδιαγραφών δοσοληψιών.

3.1 Επιπτώσεις(Ramifications)

Στην ενότητα αυτή, παρέχουμε το απαραίτητο θεωρητικό υπόβαθρο για την κατανόηση της μεθόδου επεξηγήσεων στο πεδίο της τροποποίησης των δοσοληψιών. Θα κάνουμε μία ανασκόπηση στα διάφορα θέματα και στις ιδιότητες των επεξηγήσεων. Το υλικό της ανασκόπησης βασίζεται στις εργασίες [Ple96], [F88] και [Pi94], όπως οι ορισμοί αυτοί μπορούν να αντιστοιχιστούν στο πεδίο των Βάσεων Δεδομένων.

3.1.1 Θεμελιώδεις Γνώσεις

Ας υποθέσουμε ότι έχουμε μία Βάση Γνώσεων, ΒΓ, η οποία περιλαμβάνει ένα σύνολο κατηγορημάτων που αναπαριστούν αληθή γεγονότα στην ΒΓ, ένα σύνολο R συμπερασματικών κανόνων και ένα σύνολο I περιορισμών ακεραιότητας. Οι περιορισμοί και οι κανόνες εκφράζονται σαν, περιορισμένου εύρους, μορφές εκφράσεων Λογικής Πρώτης Τάξεως(Λ.Π.Τ.), στην οποία αναγνωρίζουμε έναν χρονικό τύπο Time, για χρονικά σημεία. Για συντομία θα αναφερόμαστε από εδώ και στο εξής στην λογική αυτή σαν ΧΛΠΤ(Χρονική Λογική Πρώτης Τάξεως). Ο χρονικός τύπος μεταφράζεται σαν να είναι *σχετικός, γραμμικός, διακριτός & άπειρος*. Ο τύπος αυτός μπορεί να θεωρηθεί σαν ένα σύνολο ισομορφικό με τους ακέραιους.

Διαισθητικά, μία *επίπτωση*(ramification) μίας έκφραση(formulae) φ, είναι μία έκφραση N, τέτοια ώστε η N να είναι αναπόφευκτα αληθής εάν και μόνο εάν η φ είναι αληθής[F88]. Αυτός ο ορισμός υπόκειται σε διαφορετικές μεταφράσεις, για διαφορετικά μοντέλα του κόσμου μας. Εάν το μοντέλο, στην προκειμένη περίπτωση η KB, εκφράζεται σαν ένα θεώρημα Λογικής Πρώτης Τάξεως(ΛΠΤ), τότε η έννοια της επίπτωσης μπορεί να αποδοθεί σαν μία συνεπαγωγή ΛΠΤ.

Μία έκφραση φ βρίσκεται σε κανονική μορφή τύπου prenex, εάν είναι της μορφής: $\varphi \equiv Q_1x_1/S_1, \dots, Q_mx_m/S_m \varphi^*$, όπου Q_1, \dots, Q_m είναι είτε καθολικοί είτε υπαρξιακοί ποσοδείκτες(\forall, \exists αντίστοιχα), S_1, \dots, S_m είναι κατηγορίες στις οποίες μπορεί μία μεταβλητή από τις $x_1 \dots x_m$, να ανήκει, ενώ, τέλος, φ^* είναι μία έκφραση απαλλαγμένη από ποσοδείκτες. Πιο αναλυτικά ακολουθεί ένας χρήσιμος ορισμός:

Ορισμός Νο1(Επίπτωση): Με δεδομένη μία Βάση Γνώσεων ΒΓ και μία έκφραση φ σε prenex κανονική μορφή με μεταβλητές $x_1 \dots x_m$ που εισάγονται από ποσοδείκτες, μία έκφραση N με ελεύθερες μεταβλητές $y_1 \dots y_n$ ανάμεσα στις $x_1 \dots x_m$, είναι μία *επίπτωση* της φ στην ΒΓ, εάν ισχύει η σχέση $B\Gamma \cup \{\varphi\} \models \forall y_1, \dots, \forall y_n N$.

Από τον ορισμό αυτό, όπως βρίσκεται καθαρά διατυπωμένος στην [Ple965], είναι εύκολο να επιβεβαιώσουμε τις παρακάτω ιδιότητες των επιπτώσεων:

1. Εάν μία επίπτωση N, μίας έκφρασης φ, είναι γνωστό ότι δεν ικανοποιείτε, τότε η ίδια η έκφραση φ δεν ικανοποιείτε

2. Οι επιπτώσεις i , μίας έκφρασης φ , μπορούν να περιορίσουν το εύρος του χώρου έρευνα για την ικανοποίηση της ίδιας της φ
3. Διάφοροι μετασχηματισμοί μπορεί να είναι εφαρμόσιμοι σε κάποιες από τις επιπτώσεις των εκφράσεων φ , χωρίς, όμως, να είναι εφαρμόσιμοι στις ίδιες τις εκφράσεις.

Για τον λόγο αυτό, εάν υπάρχει ένας συστηματικός τρόπος παραγωγής επιπτώσεων, από ένα σύνολο τέτοιων εκφράσεων όπως η φ , οι παραγόμενες επιπτώσεις μπορούν να χρησιμοποιηθούν για βελτιστοποίηση της αποτίμησης των ίδιων των εκφράσεων.

Η δημιουργία επιπτώσεων από μία έκφραση φ , ουσιαστικά περιλαμβάνει μία τμηματική περιγραφή μίας κατάστασης, όπου η φ στηρίζεται από επιπρόσθετες περιγραφές που πηγάζουν από την γνώση για την συνέπεια της κατάστασης. Μία διαδικασία δημιουργίας επιπτώσεων μπορεί να απαιτεί ένα αυθαίρετο αριθμό συμπερασμάτων. Από τις διαδικασίες συνεπαγωγής της Λ.Π.Τ., έπεται ότι το πρόβλημα της ανεύρεσης επιπτώσεων, είναι δυσεπίλυτο, ενώ σε μερικές περιπτώσεις ακόμα και άλυτο. Μία ειδικότερη, αλλά πολύ αποτελεσματική, αντιμετώπιση του προβλήματος αυτού, μπορεί να επιτευχθεί περιορίζοντας την κλάση των εκφράσεων για τις οποίες μπορούμε να δημιουργήσουμε επιπτώσεις, τεχνική που έχουμε συναντήσει και σε εργασίες που προαναφέραμε στη Κεφάλαιο 2.

Επιπλέον, δεν είναι όλες οι παραγόμενες, από μία έκφραση, επιπτώσεις, χρήσιμες για την διαδικασία απλοποίηση της απόδειξης της ευστάθειας της έκφρασης αυτής. Για τον λόγο αυτό, η γεννήτρια πρέπει να καθοδηγείται ώστε να παράγει μόνο τις χρήσιμες επιπτώσεις παρέχοντας τις απαραίτητες κλάσεις εισόδου. Στην πραγματικότητα, η διαδικασία που θα περιγράψουμε σε επόμενα στάδια της εργασίας μας παράγει μόνο τις εκφράσεις που μας είναι απολύτως απαραίτητες.

Η διαδικασία τροποποίησης ενός αρχικού στόχου, με την σύζευξή του με επιπρόσθετους περιορισμούς, έτσι ώστε η νέα, τροποποιημένη έκφραση να είναι λιγότερο δύσκολο στο να ελεγχθεί από την αρχική, ονομάζεται *εκλέπτυνση*(supersumption)[F88]. Οι μέθοδοι, που προτείνονται στην εργασία μας, εφαρμόζονται μόνο σε συζευγμένες εκφράσεις. Είναι εύκολο να καθορίσουμε την ορθότητα της διαδικασίας εκλέπτυνσης για εκφράσεις σε Διαζευκτική Κανονική Μορφή(ΔΚΜ ή σύμφωνα με την αγγλική ορολογία DNF). Το Πρόγραμμα2 είναι μία

απευθείας συνέπεια του ορισμού της επίπτωσης, ενώ το Πόρισμα3 είναι αποτελεί μία συνέπεια των ιδιοτήτων της συνεπαγωγής της ΛΠΤ. Στην εργασία [Ple965], φαίνονται πιο αναλυτικά τα ακόλουθα πορίσματα:

Πόρισμα 2: Εάν N είναι μία επίπτωση της έκφρασης φ σε μία ΒΓ και το N δεν ισχύει(είναι πάντα ψευδής) στην ΒΓ, τότε το ίδιο ισχύει και για την φ .

Πόρισμα 3: Ας υποθέσουμε ότι φ είναι μία έκφραση σε ΔΚΜ, δηλαδή είναι της μορφής $\varphi \equiv \varphi_1 \cup \dots \cup \varphi_n$, όπου κάθε ένα από τα φ_i αποτελείται από συζευγμένα κατηγορήματα. Επίσης, υποθέτουμε ότι N_i είναι μία επίπτωση του φ_i , $\forall i = 1, \dots, n$ και ισχύει η σχέση $B\Gamma \models (\varphi_i \Rightarrow N_i)$. Τότε εάν ισχύει η σχέση $B\Gamma \models \neg N_i$, τότε ισχύει και η σχέση $B\Gamma \models \neg \varphi$.

Σύμφωνα με το Πόρισμα 2, δοθείσας μίας έκφρασης φ σε ΔΚΜ, μπορούμε να παράγουμε επιπτώσεις από τις συζεύξεις της έκφρασης και τότε να δοκιμάσει εάν όλες οι επιπτώσεις παραμένουν ψευδείς. Εάν αυτό ισχύει, τότε η αρχική έκφραση είναι ψευδής. Η στρατηγική αυτή είναι χρήσιμη για την περίπτωση που ο χρήστης ενδιαφέρεται να παρακολουθήσει πότε η έκφραση παραβιάζεται, παρά για την περίπτωση που ενδιαφέρεται να αποδείξει ότι η έκφραση είναι πάντα αληθής. Όπως θα αποδείξουμε και στο επόμενο κεφάλαιο, η στρατηγική αυτή είναι επαρκής για την παρακολούθηση των παραβιάσεων των περιορισμών ακεραιότητας. Το επόμενο Πόρισμα(Πόρισμα 4) είναι διατυπωμένο, αναλυτικότερα, στην εργασία [Ple965] και προσδιορίζει την πληρότητα της διαδικασίας εκλέπτυνσης. Το αποτέλεσμα αυτό, αποδεικνύει την ορθότητα της διαδικασίας τροποποίησης, όσον αφορά στην ικανοποίηση της παραγόμενης έκφρασης.

Πόρισμα 4(Πληρότητα της απλούστευσης): Ας υποθέσουμε ότι φ είναι μία έκφραση της μορφής $\varphi \equiv Q_1x_1, \dots, Q_mx_m \varphi^*$, όπου Q_1, \dots, Q_m είναι είτε καθολικοί είτε υπαρξιακοί ποσοδείκτες(\forall, \exists αντίστοιχα), καθένας από τους οποίους εισάγει την αντίστοιχη μεταβλητή από το σύνολο $x_1 \dots x_m$. Επίσης, φ^* είναι μία έκφραση σε ΔΚΜ απαλλαγμένη από ποσοδείκτες, όπου κάθε μία από τις $x_1 \dots x_m$ εμφανίζεται σε τουλάχιστον μία από της διαζεύξεις της φ . Τέλος, N είναι μία επίπτωση της φ στην ΒΓ. Τότε ισχύει η σχέση $(B\Gamma \cup \{\varphi\}) \models (Q_1x_1, \dots, Q_mx_m (\varphi^* \cap N))$.

3.2 Πρόβλημα πλαισίου

Τα επιχειρήματα υπέρ της άποψης ότι είναι προτιμότερη η προετοιμασία του τυπικού προσδιορισμού ενός προγράμματος, πριν την υλοποίησή του σε μία γλώσσα προγραμματισμού, είναι έως τώρα πολύ γνωστά και αποδεκτά από ένα ευρύ φάσμα της επιστημονικής κοινότητας. Οι προδιαγραφές των προγραμμάτων δηλώνουν, περισσότερο, **τι** υποτίθεται ότι θα κάνει το επιθυμητό πρόγραμμα και λιγότερο **πώς** θα το επιτύχει, με την χρήση κατάλληλων μηχανισμών περιγραφής προσδιορισμών, τους οποίους καλούμε *γλώσσες προδιαγραφών* (specification languages). Η επάρκεια των γλωσσών αυτών απορρέει από διάφορα καθολικά κριτήρια όπως είναι η εκφραστικότητα αλλά και

- *καταλληλότητα των μηχανισμών περιγραφής*, που σημαίνει ο βαθμός στον οποίο μία τέτοια γλώσσα επιτρέπει στον σχεδιαστή να εκφράσει τον σκοπό μίας διαδικασίας, με ένα ακριβή, απλό, σαφή, κατανοητό, τμηματικό και εύκολα τροποποιήσιμο τρόπο.
- *Δυνατότητα υποστήριξης τυπικής μεταχείρισης*, δηλαδή η έκταση μέσα στην οποία μία γλώσσα προδιαγραφών παρέχει μία θεμελίωση και οδηγεί στην υποστήριξη μίας μεθοδολογίας για μία τυπική απόδειξη ιδιοτήτων που σχετίζονται με το πρόγραμμα και τους σχεδιαστές του.

Το πρώτο κριτήριο επικεντρώνεται σε θέματα μηχανικής, σε αντιδιαστολή με θέματα που αφορούν στην εκφραστικότητα. Η σημασία τέτοιων θεμάτων υποστηρίζεται πολύ από εργασίες της σύγχρονης έρευνας πάνω σε γλώσσες προδιαγραφών. Μετά από όλα αυτά, κάποιος θα μπορούσε δηλώσει οτιδήποτε χρειάζεται για μία τυπική περιγραφή προδιαγραφών, χρησιμοποιώντας έννοιες ενός ικανοποιητικά πλούσιου συνόλου θεωριών, ή λογική κατηγορημάτων.

Το δεύτερο κριτήριο μετράει τον βαθμό στον οποίο η επιστημότητα στις εκφράσεις οδηγεί σε μία μεθοδολογία, κατά προτίμηση να υποστηρίζεται από εργαλεία, για την απόδειξη ιδιοτήτων σχετικά με προδιαγραφές. Ένα οικείο είδος αποδείξεων που αφορά προδιαγραφές, είναι να δείξουμε ότι σε κάποιες υλοποιήσεις οι προδιαγραφές αυτές επαληθεύονται. Ένα άλλο είδος *υποχρέωσης απόδειξης* (proof obligation), η οποία εμφανίζεται με την χρήση τεχνικών προδιαγραφών βασισμένες σε κάποιο μοντέλο, όπως η VDM που παρουσιάζεται στην εργασία [BMR95], απαιτεί ο

σχεδιαστής να αποδείξει ότι κάθε διαδικασία διατηρεί τις καθολικές σταθερές του προγράμματος, που είναι μέρος των προδιαγραφών.

Οι σταθερές καταστάσεων (State invariant), όπως για παράδειγμα «μόνο όσοι μαθητές έχουν πληρώσει τα δίδακτρα μπορούν να εμφανιστούν στις λίστες μαθημάτων», είναι γνωστές και ως περιορισμοί ακεραιότητας στον κόσμο των Βάσεων Δεδομένων. Τέτοιοι περιορισμοί δηλώνονται, συνήθως, στον ορισμό της Βάσης Δεδομένων, ανεξάρτητα από διαδικασίες που έχουν πρόσβαση ή ενημερώνουν την Βάση. Αυτό συμβαίνει, τόσο εξαιτίας του γεγονότος ότι η Βάση παραμένει συνεπής και ανάμεσα σε διαδοχικές εκτελέσεις των διαδικασιών όσο και εξαιτίας της απαίτησης να εισάγονται στο σύστημα νέες διαδικασίες συνεχώς.

Έστω μία διαδικασία p η οποία προσδιορίζεται με όρους που περιέχονται σε ένα ζεύγος από συνθήκες, δηλαδή ένα ζεύγος της μορφής $\langle \text{precondition}, \text{post condition} \rangle$ ή αλλιώς $\langle \text{pre-}p, \text{post-}p \rangle$. Το ζεύγος αυτό αναφέρεται στις συνθήκες που πρέπει να ισχύουν τόσο πριν ($\text{pre-}p$), όσο και μετά ($\text{post-}p$) από την εκτέλεση της p . Υποθέτουμε επίσης ότι η έκφραση I αποτελεί μία σταθερά που θα πρέπει να ισχύει πάντα. Τότε η υποχρέωση απόδειξης θα είναι της μορφής: $I_{\text{before}} \wedge \text{pre-}p \wedge \text{post-}p \Rightarrow I_{\text{after}}$

Στην σχέση αυτή, η I_{before} δίνει τιμή στην σταθερά I σύμφωνα με την κατάσταση του προγράμματος πριν από την εκτέλεση της p (η κατάσταση στην οποία η $\text{pre-}p$ έχει πάρει τιμή) και I_{after} αντίστοιχα αναφέρεται στην επόμενη κατάσταση και την συνθήκη $\text{post-}p$. Στο σημείο αυτό θα πρέπει να παρουσιάσουμε μία οικογένεια προβλημάτων που υπάρχουν στην περιοχή της περιγραφής προδιαγραφών οι οποίες χρησιμοποιούν τις έννοιες του ζεύγους συνθηκών $\langle \text{precondition}, \text{post condition} \rangle$ και σχετίζονται με ένα πρόβλημα που τοποθετήθηκε στον χώρο της Τεχνητής Νοημοσύνης εδώ και πολλά χρόνια [McCarthy69]. Το πρόβλημα αυτό είναι αυτό που εν συντομία παρουσιάσαμε προηγουμένως, το αποκαλούμενο πρόβλημα πλαισίου.

Ορισμός: Η ανεύρεση του ελάχιστου συνόλου συνθηκών και εκφράσεων που απαιτούνται ώστε να δηλώσουμε σε ένα σύστημα, όπως για παράδειγμα μία Βάση Δεδομένων, ότι «...και τίποτα άλλο δεν αλλάζει...», αποκαλείται *πρόβλημα του πλαισίου*.

3.2.1 Γενικά για το πρόβλημα του πλαισίου

Στο σημείο αυτό θα προσπαθήσουμε να παρουσιάσουμε το πρόβλημα του πλαισίου χρησιμοποιώντας για παράδειγμα μία Βάση Δεδομένων. Το πρόβλημα του πλαισίου δεν υφίσταται μόνο στο πεδίο των Βάσεων Δεδομένων, αλλά η απεικόνισή του στο πεδίο αυτό είναι ένας από τους αρχικούς στόχους της εργασίας αυτής. Για την περιγραφή αυτή χρησιμοποιήσαμε την Προτασιακή Λογική Πρώτης Τάξης(Π.Λ.Π.Τ), ώστε να εκφράσουμε τις συνθήκες με τον πιο κατανοητό, για τον αναγνώστη, τρόπο.

Υποθέτουμε ότι διαθέτουμε ένα τμήμα των αρχικών προδιαγραφών για την απλή διαδικασία πρόσληψης ενός υπαλλήλου, σε μία εταιρία. Την διαδικασία αυτή την ονομάζουμε *hire* και έχει σαν ορίσματα τον υπάλληλο (*em* από την αντίστοιχη λέξη *employee*) και το τμήμα της εταιρίας που ο υπάλληλος θα εργάζεται(*dep* από την αντίστοιχη λέξη *department*), καθώς και την ίδια την εταιρία(*cmp* από την αντίστοιχη λέξη *company*). Έχουμε, δηλαδή, την διαδικασία *hire(em, dep, cmp)*. Ο ορισμός της διαδικασίας αυτής, έχει, σύμφωνα με όσα προαναφέραμε, ως εξής:

hire(em, dep, cmp)

Pre: $dep_size(dep, cmp) < max_size(dep, cmp) \wedge \neg WorkIn(em, dep, cmp)$

Post: $dep_size(dep, cmp) = dep_size(dep, cmp) + 1 \wedge @WorkIn(em, dep, cmp)$

Στο παράδειγμά αυτό, οι συναρτήσεις *dep_size(dep, cmp)* και *max_size(dep, cmp)* δηλώνουν τον τρέχον, καθώς και τον μέγιστο επιτρεπόμενο αριθμό υπαλλήλων που έχουν προσληφθεί στο τμήμα *dep*, της εταιρίας *cmp* αντίστοιχα, ενώ το κατηγορημα *WorkIn(em, dep, cmp)* δηλώνει ότι ο υπάλληλος *em* εργάζεται στο τμήμα *dep*, της εταιρίας *cmp*. Ο μέγιστος αριθμός δεν είναι απαραίτητο να υπάρχει αφού, εν γέννη, οι ανάγκες κάθε εταιρίας είναι διαφορετικές. Για να αναφερθούμε στις τιμές των μεταβλητών αμέσως πριν καθώς και αμέσως μετά της εκτέλεση της διαδικασίας υιοθετούμε τις έννοιες *τονισμένο* και *μη τονισμένο*(*primed/unprimed*) και τις εκφράζουμε με την παρουσία ή την απουσία αντίστοιχα του συμβόλου *@*. Αν και στην διεθνή βιβλιογραφία το αντίστοιχο σύμβολο είναι το «'», η επιλογή του συγκεκριμένου συμβόλου δεν είναι τυχαία. Επιθυμούμε να αναπαραστήσουμε το σύμβολο αυτό με ένα πλήκτρο του αγγλικού πληκτρολογίου, ώστε να μπορεί να

χρησιμοποιηθεί από τον χρήστη του συστήματος που υλοποιήσαμε. Έτσι, το κατηγορήμα @WorkIn(em, dep, cmp) δηλώνει ότι ο υπάλληλος em θα εργάζεται στο τμήμα dep, της εταιρίας cmp, αμέσως μετά την εκτέλεση της διαδικασίας hire(em, dep, cmp).

Σύμφωνα, λοιπόν, με τον ορισμό της διαδικασίας αυτής, απαραίτητη συνθήκη για την πρόσληψη ενός εργαζόμενου σε ένα τμήμα μίας εταιρίας, είναι η ύπαρξη διαθέσιμων θέσεων, καθώς και η αποφυγή διπλής εγγραφής του υπαλλήλου σε ένα τμήμα. Η συνθήκη που επιβάλλεται να ισχύει μετά την εκτέλεση της διαδικασίας αυτής(post condition), προσδιορίζει τα αποτελέσματα της διαδικασίας στην τελική κατάσταση, περιλαμβάνοντας την επαύξηση του μετρητή των προσληφθέντων υπαλλήλων στο τμήμα της εταιρίας, καθώς και αλλάζοντας τιμή αληθείας στο κατηγορήμα WorkIn(em, dep, cmp), κάνοντάς το αληθές από ψευδές.

Δυστυχώς, οι προδιαγραφές της διαδικασίας αυτής υπόκειται σε, τουλάχιστον δύο, διαφορετικές προσεγγίσεις. Επιπλέον, η πιο διαισθητική προσέγγιση είναι η δυσκολότερη για να δημιουργηθεί σε Λογική Κατηγορημάτων. Για να γίνει εμφανές το πρόβλημα αυτό, υποθέτουμε δύο υλοποιήσεις των προδιαγραφών. Η πρώτη φέρει ως αποτέλεσμα δύο ακριβώς αλλαγές στην κατάσταση του προγράμματος, όπως απαιτείται από την post-p. Η δεύτερη κάνει αυτές τις αλλαγές, όμως επίσης, αλλάζει μερικά άλλα κατηγορήματα, όπως το WorkIn ενδεχομένως με διαφορετικά γνωρίσματα.

Πιο αναλυτικά, μια πιο διαισθητική και πιο «σφικτή», προσέγγιση των προδιαγραφών είναι συνεπής με την δεύτερη υλοποίηση, επειδή η υλοποίηση αυτή κάνει περισσότερες λειτουργίες από όσες είναι απαραίτητες ώστε να καθίσταται η συνθήκη post-p αληθής. Με άλλα λόγια, η διαισθητική προσέγγιση περιλαμβάνει μία έκφραση της μορφής «...και τίποτα άλλο δεν αλλάζει...», για να περιορίσει τις τροποποιήσεις που μια διαδικασία συνεπάγεται. Η δήλωση αυτή είναι μία μορφή του προβλήματος του πλαισίου.

Μία δεύτερη, η οποία είναι και αρκετά χαλαρή, προσέγγιση δεν περιλαμβάνει δήλωση, της μορφής που προαναφέραμε και είναι συνεπής και με τις δύο υλοποιήσεις που προαναφέραμε. Η προσέγγιση αυτή είναι λιγότερο διαισθητική στο πρόβλημα αυτό, ενώ θέτει ένα χαμηλότερο εύρος αποτελεσμάτων της διαδικασίας που προσδιορίζει. Ωστόσο, έχει δύο πλεονεκτήματα. Είναι λιγότερο περιοριστική προσφέροντας μεγαλύτερη ελευθερία στην υλοποίηση, που είναι μια επιθυμητή ιδιότητα των προσδιορισμών, καθώς επίσης παρέχει ένα απλό φορμαλισμό σε Λογική

Κατηγορημάτων: Κατά πρώτον, βλέπει τις καταστάσεις σαν μοντέλα για προτάσεις που αφορούν unprimed κατηγορήματα και συναρτήσεις-οι μεταβλητές που το πρόγραμμα θα ενημερώσει.

Η τελική υλοποίηση πρέπει να εγκαταστήσει μία κατάσταση, όπου η έκφραση της post-p να γίνεται αληθής, με τα primed κατηγορήματα να αποτιμώνται στην τελική κατάσταση, ενώ τα unprimed κατηγορήματα να αποτιμώνται στην αρχική κατάσταση. Κατά δεύτερον, με την βοήθεια της διαδικασίας δημιουργίας primed/unprimed κατηγορημάτων, μπορούμε να συνδέσουμε, στην έκφραση των pre-p και post-p, κατηγορήματα που αναφέρονται τόσο στην αρχική, όσο και την τελική κατάσταση.

Είναι αξιοσημείωτο ότι η σχέση μεταξύ των κατηγορημάτων R και @R, είναι ολοκληρωτικά συμβατική(στην έκφραση της συνθήκης post-p)-μπορούν να αποκαλεστούν επίσης με εντελώς διαφορετικά ονόματα. Πιο συγκεκριμένα, κάθε υποχρέωση απόδειξης, της προαναφερθείσας μορφής, που περιλαμβάνει μία σταθερά κατάστασης I, του προγράμματος και μία διαδικασία p, μαζί με ένα ζεύγος συνθηκών $\langle \text{pre-p}(x), \text{post-p}(x) \rangle$, μπορεί να οριστεί ως εξής:

$$\forall x [I \wedge \text{pre-p}(x) \wedge \text{post-p}(x)] \Rightarrow @I$$

όπου το @I αναπαριστά την σταθερά κατάστασης I, όπως την περιγράψαμε παραπάνω, με όλα τα κατηγορήματά του να αναφέρονται στην επόμενη κατάσταση της εκτέλεσης. Ας υποθέσουμε, τώρα, ότι έχουμε μερικές απλές σταθερές καταστάσεων $\forall y.Q(y)$, όπου το Q είναι ένα κατηγορήμα. Τότε η υποχρέωση απόδειξης εκφράζεται από την παρακάτω έκφραση:

$$\begin{aligned} &\forall em, dep, cmp [\forall x.Q(x) \wedge \\ &(\text{dep_size}(dep, cmp) < \text{max_size}(dep, cmp) \wedge \neg \text{WorkIn}(em, dep, cmp)) \wedge \\ &(@\text{dep_size}(dep, cmp) = \text{dep_size}(dep, cmp) + 1 \wedge \text{WorkIn}(em, dep, cmp)) \\ &\Rightarrow \forall x.Q(x)] \end{aligned}$$

Αυτό δεν μπορεί να αποδειχθεί, γιατί η post-p της διαδικασίας δεν μας πληροφορεί για το κατηγορήμα @Q. Για τον λόγο αυτό, τέτοιες υποχρεώσεις απόδειξης δεν μπορούν να επιτευχθούν χρησιμοποιώντας την πιο χαλαρή προσέγγιση

των προδιαγραφών, εκτός εάν ο χρήστης πάρει την ευθύνη να προσδιορίσει επακριβώς στις προδιαγραφές, όχι μόνο όσα κατηγορήματα αλλάζουν από την διαδικασία, αλλά και όσα παραμένουν αναλλοίωτα. Έτσι, εάν ο χρήστης επιθυμεί να δηλώσει ότι «...και τίποτα άλλο δεν αλλάζει...», θα πρέπει να το δηλώσει ρητά στις προδιαγραφές, προσθέτοντας μερικές φράσεις της μορφής:

$$\forall x[(P(x) \Rightarrow @P(x)) \wedge (\neg P(x) \Rightarrow \neg @P(x))]$$

ή ισοδύναμα:

$$\forall x[P(x) \equiv @P(x)]$$

για κάθε κατηγορήμα P εκτός του EnrolledIn, ενώ πρέπει, επίσης, να προστεθεί μία φράση της μορφής: $\forall x[f(x) \equiv @f(x)]$, για κάθε συνάρτηση f, εκτός της size. Επιπλέον θα πρέπει να υπάρχουν φράσεις που δηλώνουν ότι το κατηγορήμα WorkIn και η συνάρτηση dep_size, παραμένουν αναλλοίωτα για κάθε γνώρισμα εκτός από τα em, dep, cmp:

$$\forall x \quad [x \neq \text{dep} \vee y \neq \text{cmp} \Rightarrow \text{dep_size}(x, y) = @\text{dep_size}(x, y)]$$

$$\forall x, y, z [x \neq \text{em} \vee y \neq \text{dep} \vee z \neq \text{cmp} \Rightarrow (\text{WorkIn}(x, y, z) \equiv @\text{WorkIn}(x, y, z))]$$

Υποθέτοντας ότι οι καταστάσεις του προγράμματος περιγράφονται μόνο από δύο συναρτήσεις, την dep_size και την max_size και από δύο κατηγορήματα, το WorkIn και το Q, μία πλήρης περιγραφή των προδιαγραφών της διαδικασίας hire που να δηλώνει ρητά ότι «...και τίποτα άλλο δεν αλλάζει...», θα πρέπει να είναι της ακόλουθης μορφής:

hire(em, dep, cmp)

Pre: dep_size(dep, cmp) < max_size(dep, cmp) \wedge \neg WorkIn(em, dep, cmp)

Post: @dep_size(dep, cmp) = dep_size(dep, cmp) + 1 \wedge WorkIn(em, dep, cmp) \wedge

$$\forall x \quad [x \neq \text{dep} \vee y \neq \text{cmp} \Rightarrow \text{dep_size}(x, y) = @\text{dep_size}(x, y)] \quad \wedge$$

$$\forall x, y, z [x \neq \text{em} \vee y \neq \text{dep} \vee z \neq \text{cmp} \Rightarrow (\text{WorkIn}(x, y, z) \equiv @\text{WorkIn}(x, y, z))] \wedge$$

$$\forall x, y \quad [\text{max_size}(x, y) = @\text{max_size}(x, y)] \quad \wedge$$

$$\forall x \quad [Q(x) \equiv @Q(x)]$$

Οι επιπλέον εκφράσεις που απαιτούνταν για την ρητή δήλωση, ότι «...και τίποτα άλλο δεν αλλάζει...», ονομάζονται *αξιώματα πλαισίου*(frame axioms), ενώ το γενικότερο πρόβλημα της ρητής δήλωσης αυτής ονομάζεται, όπως προαναφέραμε, *πρόβλημα πλαισίου*.

3.2.2 Αξιώματα αποτελέσματος(effect axioms)

Στην παράγραφο αυτή θα περιγράψουμε μερικές κατηγορίες που χαρακτηρίζουν τα αποτελέσματα των πράξεων πάνω στα διάφορα κατηγορήματα ή συναρτήσεις, που εμφανίζονται στην εκάστοτε εφαρμογή. Επίσης, παρέχουμε ένα σύνολο αξιωμάτων, συνδυάζοντας διαφορετικές ιδέες πάνω στο πρόβλημα. Το υλικό που μας είναι απαραίτητο για την πλήρη περιγραφή του πεδίου με τους ορισμούς και τα αποτελέσματά τους, βρίσκονται στο [Rei91]. Στο σημείο αυτό κρίνεται σκόπιμη μία μικρή αναφορά σε βασικές έννοιες πάνω στον καταστατικό λογισμό.

Ο Καταστατικός Λογισμός(situation calculus [McCarthy69]), είναι μία γλώσσα που σχεδιάστηκε για να αναπαραστήσει, δυναμικά μεταβαλλόμενους, κόσμους, όπου όλες οι αλλαγές είναι αποτέλεσμα κάποιων εξωτερικών ή εσωτερικών ερεθισμάτων, που ονομάζονται *ενέργειες*(actions). Ο κόσμος θεωρείτε πως βρίσκεται σε μία κατάσταση s , η οποία μπορεί να αλλάξει μόνο με την παρέμβαση κάποιου εξωτερικού φορέα(agent), ο οποίος μπορεί είτε να είναι ένας άνθρωπος, χρήστης του συστήματος, είτε μία αυτόματη μηχανή που σαν σκοπό έχει την χρήση του εν λόγω συστήματος. Ο φορέας αυτός για να αλλάξει την κατάσταση του συστήματος πρέπει να εκτελέσει μία ενέργεια. Εάν υποθέσουμε ότι a είναι μία τέτοια ενέργεια, τότε η επόμενη κατάσταση(successor state) του συστήματος, που είναι αποτέλεσμα της ενέργειας a πάνω στην κατάσταση s , συμβολίζεται με την συνάρτηση $do(a, s)$.

Γενικά, οι ενέργειες μπορούν να έχουν και διάφορες παραμέτρους. Για παράδειγμα η ενέργεια ρίψης ενός αντικείμενου x , από ένα βραχίονα r , όπως θα αναφέρουμε παρακάτω, συμβολίζεται με την συνάρτηση $drop(r, x)$. Έτσι, συμβολίζουμε με $do(drop(r, x), s)$, την κατάσταση που προκύπτει εάν από την κατάσταση του συστήματος s , ο βραχίονας r ρίξει το αντικείμενο x . Οι σχέσεις, όπως συναρτήσεις ή κατηγορήματα, των οποίων η τιμή αλλάζει από αληθής σε ψευδής και αντίστροφα, από την μία κατάσταση στην άλλη, ονομάζονται *fluents*.

Υποθέτουμε ότι τα αποτελέσματα των πράξεων πάνω στα κατηγορήματα προσδιορίζονται από τα αξιώματα αποτελέσματος της ακόλουθης μορφής:

- ◆ **Θετικά Αξιώματα αποτελέσματος**(Positive effect axioms) για το κατηγορήμα R και την πράξη a .

Για κάθε πράξη $a(x)$ και κατηγορήμα R, υπάρχει ένα αξίωμα της μορφής:

$$\pi_a(x, s) \wedge \varepsilon_R^+(x, y, s) \supset R(y, \text{do}(a(x)), s).$$

Εδώ, οι μεταβλητές x , y και s εισάγονται, αναμφίβολα, από καθολικούς ποσοδείκτες. Οι μεταβλητές αυτές x , y είναι διαφορετικές μεταξύ τους. Η πρόταση $\pi_a(x, s)$ δηλώνει τις προσυνθήκες της διαδικασίας(action precondition), δηλώνει δηλαδή την προαπαιτούμενη συνθήκη της $a(x)$. Αυτές είναι οι προϋποθέσεις που πρέπει να ισχύουν για να εκτελεστεί η πράξη $a(x)$ και εξαρτώνται μόνο από την a , όχι από τα R. Η $\varepsilon_R^+(x, y, s)$ δηλώνει τις μετασυνθήκες(fluent precondition), κάτω από τις οποίες η εκτέλεση της $a(x)$ θα οδηγήσει το R σε τιμή αληθή, για την μεταβλητή y , στην διαδοχική κατάσταση $\text{do}(a(x), s)$, που προκύπτει μετά την εκτέλεση της $a(x)$.

Για παράδειγμα, τα αποτελέσματα, πάνω στο κατηγορήμα broken, που προκύπτει από την εκτέλεση της διαδικασίας drop έχει ως εξής:

$$(\text{holding}(r, x, s) \wedge (y = x \wedge \text{fragile}(y))) \supset \text{broken}(y, \text{do}(\text{drop}(r, x), s)).$$

Στο παράδειγμα αυτό, σαν $\pi_{\text{drop}}(r, x, s)$ έχουμε το $\text{holding}(r, x, s)$, που είναι η προϋπόθεση ώστε, ο βραχίονας του robot r , να ρίξει το αντικείμενο x στην κατάσταση s . Η έκφραση $\varepsilon_R^+(r, x, y, s)$ είναι η $(y = x \wedge \text{fragile}(y))$, που αποτελεί την αναγκαία και ικανή συνθήκη, ώστε το y να σπάσει στην κατάσταση $\text{do}(\text{drop}(r, x), s)$, που οδηγείται το σύστημα, όταν το robot ρίξει το αντικείμενο όταν βρίσκεται στην κατάσταση s .

- ◆ **Αρνητικά Αξιώματα αποτελέσματος**(Negative effect axioms) για το κατηγορήμα R και την πράξη a .

Για κάθε πράξη $a(x)$ και κατηγορήμα R, υπάρχει ένα αξίωμα της μορφής:

$$\pi_a(x, s) \wedge \varepsilon_R^-(x, y, s) \supset \neg R(y, \text{do}(a(x)), s).$$

Όπως και προηγουμένως, η πρόταση $\pi_a(x, s)$ δηλώνει τις συνθήκες που είναι απαραίτητες για την εκτέλεση της διαδικασίας, δηλώνει δηλαδή την προαπαιτούμενη συνθήκη της $a(x)$. Αυτές είναι οι προϋποθέσεις που πρέπει να ισχύουν για να εκτελεστεί η πράξη $a(x)$ και εξαρτώνται μόνο από την a , όχι από τα R . Η $\varepsilon_R^-(x, y, s)$ δηλώνει τις απαραίτητες μετασυνθήκες, κάτω από τις οποίες η εκτέλεση της $a(x)$ θα οδηγήσει το R σε τιμή ψευδή, για την μεταβλητή y , στην διαδοχική κατάσταση $\text{do}(a(x), s)$, που προκύπτει μετά την εκτέλεση της $a(x)$.

Στο ακόλουθο παράδειγμα, τα αποτελέσματα, πάνω στο κατηγορημα *broken*, που προκύπτει από την εκτέλεση της διαδικασίας *repair* έχει ως εξής:

$$(\text{has_glue}(r, s) \wedge \text{broken}(x, s) \wedge (y = x)) \supset \neg \text{broken}(y, \text{do}(\text{repair}(r, x), s)).$$

Στο παράδειγμα αυτό, σαν $\pi_{\text{repair}}(r, x, s)$ έχουμε το $\text{has_glue}(r, x, s)$, που είναι η προϋπόθεση ώστε, ο βραχίονας του robot r , να επιδιορθώσει το αντικείμενο x , στην κατάσταση s . Η έκφραση $\varepsilon_R^-(r, x, y, s)$ είναι η $(y = x)$, που αποτελεί την αναγκαία και ικανή συνθήκη, ώστε το y να επιδιορθωθεί στην κατάσταση $\text{do}(\text{repair}(r, x), s)$, που οδηγείται το σύστημα, όταν το robot ρίξει το αντικείμενο όταν βρίσκεται στην κατάσταση s .

3.2.3 Αξιώματα πλαισίου (frame axioms)

Όπως έχει ήδη αναγνωριστεί, από την εργασία [McCarthy69], επιπλέον αξιώματα εκτός από τα αξιώματα αποτελέσματος απαιτούνται για τον σχηματισμό και την τεκμηρίωση δυναμικών μοντέλων για τους διάφορους κόσμους που καλούμαστε να αντιμετωπίσουμε σε μία εφαρμογή. Αυτά τα αξιώματα ονομάζονται *αξιώματα πλαισίου* (frame axioms) και προσδιορίζουν τις σταθερές συνθήκες των διαδικασιών που εκτελούνται στις διάφορες δοσοληψίες στο πεδίο εφαρμογής μιας Βάσης Δεδομένων.

Για παράδειγμα η ρίψη πραγμάτων δεν έχει κανένα αποτέλεσμα πάνω στο χρώμα του ίδιου του αντικειμένου. Την παρατήρηση αυτή μπορούμε να την δηλώσουμε με μία έκφραση όπως η ακόλουθη:

$$\text{holding}(r, x, s) \wedge \text{color}(y, c, s) \supset \text{color}(y, c, \text{do}(\text{drop}(r, x), s))$$

ή το γεγονός ότι δεν σπάμε ένα αντικείμενο εκφράζεται ως εξής:

$$\text{holding}(r, x, s) \wedge \neg \text{broken}(y, s) \wedge [y \neq x \vee \neg \text{fragile}(y)] \supset \neg \text{broken}(y, \text{do}(\text{drop}(r, x), s))$$

Γενικά, ένα αξιώματα πλαισίου έχει μία από τις παρακάτω συντακτικές μορφές [Rei91]:

➤ **Θετικά Αξιώματα Πλαισίου** (Positive Frame axioms) για το κατηγορημα R και την πράξη α: $\pi_a(x,s) \wedge \varphi_R^+(x,y,s) \wedge R(y, s) \supset R(y, \text{do}(a(x)), s)$.

όπου, οι προϋποθέσεις για την εκτέλεση της α ικανοποιούνται, οι αναγκαίες και ικανές συνθήκες $\varphi_R^+(x, y, s)$ αληθεύουν και το R παραμένει αληθές μετά από την εκτέλεση της πράξης α(x).

➤ **Αρνητικά Αξιώματα Πλαισίου** (Negative Frame axioms) για το κατηγορημα R και την πράξη α: $\pi_a(x,s) \wedge \varphi_R^-(x,y,s) \wedge \neg R(y, s) \supset \neg R(y, \text{do}(a(x)), s)$.

όπου, οι προϋποθέσεις για την εκτέλεση της α ικανοποιούνται, οι αναγκαίες και ικανές συνθήκες $\varphi_R^-(x,y,s)$ αληθεύουν και το R παραμένει ψευδές μετά από την εκτέλεση της πράξης α(x).

3.2.4 Μία απλή επίλυση του προβλήματος του πλαισίου

Στην εργασία του [Rei91], ο R. Reiter έκανε μία προσπάθεια να επιλύσει το πρόβλημα του πλαισίου. Η βασική ιδέα, της πρότασης αυτής, φαίνεται πιο αναλυτικά με την βοήθεια ενός παραδείγματος. Ορισμένες ιδέες της πρότασης αυτή αποτελούν πηγή έμπνευσης και για την δική μας προσπάθεια επίλυσης του ίδιου προβλήματος σε συγκεκριμένη εφαρμογή Βάσεως Δεδομένων.

Πιο αναλυτικά, ας υποθέσουμε το παράδειγμα που παρουσιάστηκε και προηγουμένως με το σύνολο των κατηγορημάτων που μοντελοποιούν την συμπεριφορά ενός αντικειμένου που το χειρίζεται ένας μηχανικός βραχίονας. Ας υποθέσουμε, επίσης, ότι έχουμε δύο θετικά αξιώματα αποτελεσμάτων, για το κατηγορημα broken. Έχουμε δηλαδή τις εκφράσεις:

Ισχύει δηλαδή η σχέση

$$\text{holding}(r, x, s) \wedge (y = x) \wedge \text{fragile}(y) \supset \text{broken}(y, \text{do}(\text{drop}(r, x), s)),$$

αλλά και η σχέση

$$\text{bomb}(b) \wedge \text{nexto}(b, y, s) \supset \text{broken}(y, \text{do}(\text{explode}(b), s)).$$

Οι δύο αυτές σχέσεις μπορούν να μετασχηματιστούν στην ακόλουθη:

$$\begin{aligned} & \{[\text{holding}(r, x, s) \wedge a = \text{drop}(r, x) \wedge y = x \wedge \text{fragile}(y)] \\ & \quad \vee [\text{bomb}(b) \wedge \text{nexto}(b, y, s) \wedge a = \text{explode}(b)]\} \\ & \quad \supset \text{broken}(y, \text{do}(a, s)) \end{aligned}$$

Αυτές οι σχέσεις μπορούν να εμφανιστούν πιο συμπαγείς, εισάγοντας ένα νέο κατηγορήμα, που θα ονομάσουμε $\text{Poss}(a, s)$ και δηλώνει ότι η πράξη a είναι πιθανόν να εκτελεστεί στην κατάσταση s . Έτσι έχουμε την νέα σχέση:

$$\begin{aligned} & \text{Poss}(a, s) \wedge [(\exists r, x) a = \text{drop}(r, x) \wedge y = x \wedge \text{fragile}(y) \\ & \quad \vee (\exists b) a = \text{explode}(b) \wedge \text{nexto}(b, y, s)] \quad (1) \\ & \quad \supset \text{broken}(y, \text{do}(a, s)), \end{aligned}$$

$$\text{holding}(r, x, s) \supset \text{Poss}(\text{drop}(r, x), s),$$

$$\text{bomb}(b) \supset \text{Poss}(\text{explode}(r, x), s),$$

Ομοίως, αν υποθέσουμε ότι έχουμε δύο αρνητικά αξιώματα αποτελεσμάτων για το κατηγορήμα broken , μπορούμε να γράψουμε τις σχέσεις

$$(\text{has_glue}(r, s) \wedge \text{broken}(x, s) \wedge (y = x)) \supset \neg \text{broken}(y, \text{do}(\text{repair}(r, x), s)).$$

Με τον ίδιο ακριβώς τρόπο μπορούμε να γράψουμε τις ακόλουθες εκφράσεις:

$$(\text{Poss}(a, s) \wedge (\exists r, x) a = \text{repair}(r, x) \wedge (y = x)) \supset \neg \text{broken}(y, \text{do}(a, s)),$$

$$(\text{has_glue}(r, x, s) \wedge \text{broken}(x, s)) \supset \text{Poss}(\text{repair}(r, x), s), \quad (2)$$

Το αξίωμα (1) χαρακτηρίζει όλες τις συνθήκες κάτω από τις οποίες οι εκτελέσεις, των διαφόρων πράξεων, οδηγούν, το κατηγορήμα broken , σε κατάσταση αληθή. Τότε, εάν τα $\text{Poss}(a, s)$, $\neg \text{broken}(y, s)$ και $\text{broken}(y, \text{do}(a, s))$ είναι όλα αληθή, η τιμή αληθείας πρέπει να έχει αλλάξει επειδή ισχύει ότι είναι αληθής η σχέση:

$$\begin{aligned} & (\exists r, x) a = \text{drop}(r, x) \wedge y = x \wedge \text{fragile}(y) \\ & \quad \vee (\exists b) a = \text{explode}(b) \wedge \text{nexto}(b, y, s). \end{aligned}$$

Η σύλληψη, πίσω από αυτή την παρατήρηση, μπορεί να εκφραστεί μετά από μερικές λογικές απλοποιήσεις, από τα ακόλουθα αξιώματα επεξήγησης της κλειστότητας(explanation closure axioms):

$$\text{Poss}(a, s) \wedge \neg \text{broken}(y, s) \wedge \text{broken}(y, \text{do}(a, s)) \supset \\ (\exists r)a = \text{drop}(r, y) \wedge \text{fragile}(y) \vee (\exists b)a = \text{explode}(b) \wedge \text{nexto}(b, y, s).$$

Ομοίως, στο αξίωμα (2) αντιστοιχεί το ακόλουθο αξιώματα επεξήγησης της κλειστότητας: $\text{Poss}(a, s) \wedge \text{broken}(y, s) \wedge \neg \text{broken}(y, \text{do}(a, s)) \supset (\exists r)a = \text{repair}(r, y)$.

Το προηγούμενο παράδειγμα εύκολα μπορεί να γενικευτεί. Υποθέτουμε ως δοθέντα, για κάθε κατηγορημα ή συνάρτηση R, τα ακόλουθα γενικευμένα αξιώματα αποτελεσμάτων:

- **Γενικευμένο Θετικό Αξίωμα αποτελέσματος**(General Positive Effect axioms) για το κατηγορημα R: $\text{Poss}(a, s) \wedge \gamma_R^+(a, s) \supset R(\text{do}(a, s))$. (3)
- **Γενικευμένο Αρνητικό Αξίωμα αποτελέσματος**(General Negative Effect axioms) για το κατηγορημα R: $\text{Poss}(a, s) \wedge \gamma_R^-(a, s) \supset \neg R(\text{do}(a, s))$. (4)

Τα δύο αυτά αξιώματα δημιουργούνται, συμμετρικά, από όλα τα θετικά αξιώματα αποτελέσματος(αντίστοιχα και για τα αρνητικά) για το κάθε κατηγορημα R, με μια διαδικασία βασισμένα στην ίδια την διαδικασία που παρουσιάσαμε στο προηγούμενο παράδειγμα. Το κατηγορημα Poss θα πρέπει, επίσης, να έχει δηλωθεί σαν κατηγορημα της Βάσης Γνώσεων ή της Βάσης Δεδομένων της εφαρμογής μας.

Τα αξιώματα (3) και (4), αντίστοιχα, χαρακτηρίζουν όλες τις συνθήκες κάτω από τις οποίες κάθε πράξη a μπορεί να οδηγήσει το κατηγορημα R σε κατάσταση αληθή(ψευδή κατά αντίστοιχο τρόπο) στην επόμενη κατάσταση. Έτσι, εάν η εκτέλεση της πράξης a είναι πιθανή και η τιμή αληθείας της R αλλάζει από ψευδή(false) σε αληθή(true), σαν αποτέλεσμα της εκτέλεσης της a, τότε η πρόταση $\gamma_R^+(a, s)$ πρέπει να είναι αληθής. Παρόμοια, εάν η τιμή αληθείας της R αλλάζει από αληθή σε ψευδή, τότε η $\gamma_R^-(a, s)$ πρέπει να είναι αληθής. Πιο συγκεκριμένα έχουμε τα δύο ακόλουθα αξιώματα επεξήγησης της κλειστότητας(explanation closure axioms):

$$\text{Poss}(a, s) \wedge R(s) \wedge \neg R(\text{do}(a, s)) \supset \gamma_R^-(a, s) \quad (5)$$

$$\text{Poss}(a, s) \wedge \neg R(s) \wedge R(\text{do}(a, s)) \supset \gamma_R^+(a, s) \quad (6)$$

Επίσης, σαν αποτέλεσμα των αξιωμάτων (5) και (6), αν υποθέσουμε ότι T είναι μία έκφραση, σε Λογική Πρώτης Τάξεως, που συνεπάγεται την σχέση $\neg \exists (\text{Poss}(a, s) \wedge \gamma_R^+(a, s) \wedge \gamma_R^-(a, s))$, όπου το σύμβολο \exists δηλώνει την υπαρξιακή κλειστότητα της πρότασης σε αυτό το επίπεδο. Τότε από την T συνεπάγεται ότι τα αξιώματα (3) και (4), μαζί με τα αξιώματα (5) και (6) είναι λογικά ισοδύναμα με την έκφραση:

$$\text{Poss}(a, s) \supset [R(\text{do}(a, s)) \equiv \gamma_R^+(a, s) \vee R(s) \wedge \neg \gamma_R^-(a, s)] \quad (7)$$

Τα αξιώματα που έχουν μορφή όμοια με την πρόταση (7) αποκαλούνται και *αξιώματα επόμενης κατάστασης* (successor state axioms). Η απαίτηση ότι η T συνεπάγεται την σχέση $\neg \exists (\text{Poss}(a, s) \wedge \gamma_R^+(a, s) \wedge \gamma_R^-(a, s))$ εγγυάται την ακεραιότητα των αξιωμάτων αποτελεσμάτων της μορφής (3) και (4). Έτσι, είναι αδύνατο να παράγονται ταυτόχρονα και το $R(\text{do}(a, s))$ αλλά και το $\neg R(\text{do}(a, s))$.

Συμπερασματικά, θα μπορούσαμε να πούμε ότι για μια πλήρη περιγραφή του πεδίου της εφαρμογής που καλούμαστε να μοντελοποιήσουμε, μας χρειάζονται συνολικά $F + A$ αξιώματα. Στον τύπο αυτό, το A αντιπροσωπεύει τον συνολικό αριθμό των πράξεων, ενώ F είναι ο συνολικός αριθμός των κατηγορημάτων. Μπορούμε πολύ εύκολα να συγκρίνουμε τον αριθμό αυτό με τον $2 * A * F$ που θα ήταν απαραίτητος σε κάθε άλλη περίπτωση, οπότε διαπιστώνουμε την μεγάλη οικονομία σε χρόνο και κόστος εκτέλεσης στην περίπτωση της δικής μας προσέγγισης.

Κεφάλαιο 4^ο

Η δική μας προσέγγιση

Όπως έχουμε ήδη αναφέρει, η εξασφάλιση της σημασιολογικής ορθότητας των πληροφοριών που βρίσκονται αποθηκευμένες σε ένα σύστημα Βάσεων Δεδομένων, έχει αναγνωριστεί ως ένας από τους θεμέλιους λίθους για την ανάπτυξη πληροφοριακών συστημάτων. Επίσης, λόγω του μεγάλου πλήθους των απαιτούμενων πράξεων, αποτελεί μία από τις σημαντικότερες αιτίες χρονικής καθυστέρησης της λειτουργίας ενός τέτοιου συστήματος και επομένως μία από τις κύριες αιτίες μειωμένης απόδοσής του. Η σημασιολογική ορθότητα έχει σχέση με τους περιορισμούς ακεραιότητας.

Στον τομέα της εξασφάλισης των περιορισμών ακεραιότητας, η εργασία μας ακολούθησε και υλοποίησε, μία σχετικά πρόσφατη προτεινόμενη προσέγγιση στην διερεύνηση των προβλημάτων, που προκύπτουν, στην προσπάθεια εξασφάλισης της σημασιολογικής ορθότητας, όπως προαναφέραμε. Η μέθοδος που ακολουθήσαμε αποκαλείται *διατήρηση μέσω δοσοληψιών*. Πιο αναλυτικά, τροποποιεί τις δοσοληψίες κατά τέτοιο τρόπο, ώστε να υπάρχει η εγγύηση ότι οι περιορισμοί ακεραιότητας δεν παραβιάζονται, όποια και αν είναι η ακολουθία των δοσοληψιών που θα εκτελεστούν στην Βάση Δεδομένων του συστήματος, αρκεί να είναι αιτιοκρατική. Επίσης, η μέθοδος αυτή παρέχει το κατάλληλο περιβάλλον επίλυσης του προβλήματος τις επανασχεδίασης, σε συνεργασία με την διατήρηση των τύπων περιορισμών ακεραιότητας.

Επιπροσθέτως, υπάρχει η ανάγκη για ενσωμάτωση του χρόνου στα συστήματα που περιέχουν Βάσεις Δεδομένων, εξαιτίας σύγχρονων εφαρμογών που απαιτούν την υποστήριξη της ιστορικής συνέχειας των πεπραγμένων πάνω στην Βάση που διαθέτουμε. Στην περίπτωση ενσωμάτωσης του χρόνου, οι έως τώρα, αποκαλούμενοι στατικοί και δυναμικοί περιορισμοί ακεραιότητας δίνουν την θέση τους στους χρονικούς περιορισμούς ακεραιότητας, όπως προαναφέραμε στο Κεφάλαιο 1. Έτσι, πρέπει να παρακολουθούμε και τις χρονικά εξαρτημένες ιδιότητες των δεδομένων μίας τέτοιας Βάσης, προκειμένου να εξασφαλίσουμε τους χρονικούς περιορισμούς. Οι χρονικές Βάσεις και οι επεκτάσεις τους, που αναφέρονται σαν Βάσεις Γνώσεων, χρειάζονται ισχυρούς μηχανισμούς για να εξασφαλίσουν ότι οι

χρονικά εξαρτημένες ιδιότητες δεν παραβιάζονται εξαιτίας της αποτίμησης της Βάσης Γνώσεων ή εξαιτίας του περάσματος του χρόνου.

Οι ιδιότητες που χρειάζονται εξασφάλιση μπορεί να συνδέουν πολλαπλές καταστάσεις, που αναφέρονται στο παρόν, στο παρελθόν, ή ακόμα και στο μέλλον. Για τον λόγο αυτό, η επιβεβαίωση των ιδιοτήτων, που εκφράζονται με περιορισμούς ακεραιότητας, μπορεί να αφορούν πολλαπλές καταστάσεις που είναι διαθέσιμες σε όλες τις χρονικές στιγμές. Όπως γίνεται πλέον, εύκολα, κατανοητό, η πολυπλοκότητα της επαλήθευσης τέτοιων χρονικών περιορισμών είναι, σε σημαντικό βαθμό, υψηλότερη από αυτή που αφορά την επαλήθευση στατικών περιορισμών.

Οι θεωρητικές αρχές για μία μεθοδική αντιμετώπιση του ελέγχου ορθότητας της περιγραφής δοσοληψιών σε ένα σύστημα Βάσεως Γνώσεων, μελετήθηκε, λεπτομερώς, στις εργασίες [Ple96] και [PM96], ενώ η [Ple95b] παρουσιάζει μία εφαρμογή της προτεινόμενης μεθόδου. Η προσέγγιση υιοθετεί μερικές βασικές ιδέες άλλων επιστημονικών πεδίων, με σημαντικότερες αυτές του προβλήματος του πλαισίου και του προβλήματος των επιπτώσεων, στα οποία αναφερθήκαμε αναλυτικά στο προηγούμενο κεφάλαιο. Είναι απαραίτητη η παρατήρηση ότι η συστηματική επίλυση των προβλημάτων αυτών μπορεί να χρησιμοποιηθεί κατά την διάρκεια της σχεδίασης των δοσοληψιών της Βάσης Δεδομένων.

Αρχίζοντας με ένα σύνολο τελικών προσδιορισμών, με μία μορφή προσυνθηκών, μετασυνθηκών, καθώς και αναλλοίωτων συνθηκών (*invariant*), η προτεινόμενη μέθοδος ενισχύει τις μετασυνθήκες, έτσι ώστε οι αναλλοίωτες συνθήκες να διατηρούνται αληθείς, σαν λογική συνέπεια αυτών των ενισχυμένων συνθηκών. Η διαδικασία είναι καθαρά συντακτική και βασίζεται στην επίλυση των προβλημάτων του πλαισίου και των επιπτώσεων. Οι προδιαγραφές των δοσοληψιών αναλύονται και ενισχύονται κατά την διάρκεια της μετάφρασης. Στο σημείο αυτό, εάν νέες αναλλοίωτες συνθήκες ή νέοι προσδιορισμοί προστεθούν, τότε αυτοί μπορούν να εξυπηρετηθούν σταδιακά χωρίς την υποχρέωση να ξαναμεταφραστεί ολόκληρο το σύνολο των προσδιορισμών.

Επιπλέον, η διαδικασία εφαρμόζεται σε σύνολα στατικών και μεταβατικών περιορισμών ακεραιότητας, σχετικά μεγάλων τάξεων. Οι μεταβατικοί περιορισμοί είναι τύποι που αναφέρονται σε μία καθορισμένη κατάσταση ή σε ζεύγη συσχετιζόμενων καταστάσεων της Βάσης Δεδομένων.

Στο σημείο αυτό ακολουθεί μια περιληπτική περιγραφή της ανάπτυξης ενός εργαλείου, που θα βοηθήσει ένα σχεδιαστή δοσοληψιών στον καθορισμό

4.1. Περιγραφή της αρχιτεκτονικής του συστήματος

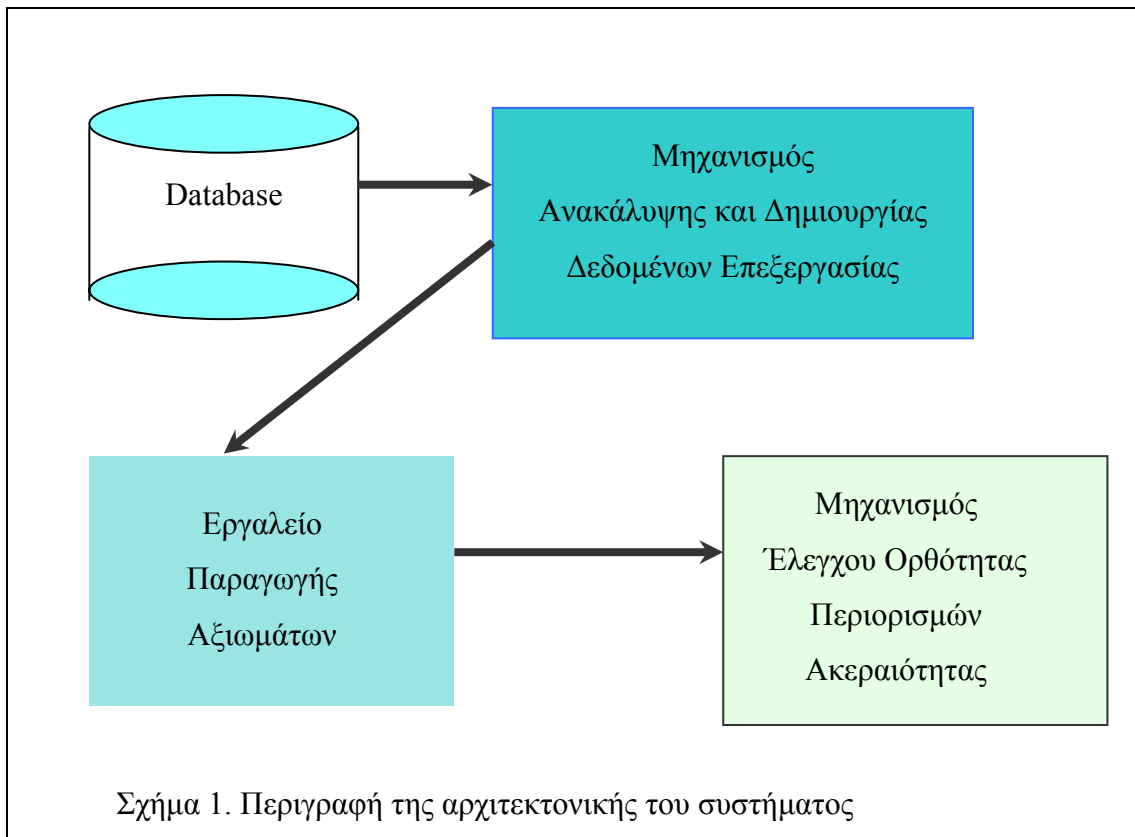
προδιαγραφών για σωστές δοσοληψίες. Με άλλα λόγια, το εργαλείο που παρουσιάζουμε, θα βοηθήσει ένα σχεδιαστή δοσοληψιών στον καθορισμό προδιαγραφών σωστών δοσοληψιών. Το εργαλείο αυτό δέχεται περιγραφές προδιαγραφών σε μία γλώσσα που περιλαμβάνει και την χρονική διάσταση, σε περιβάλλον ενός Σχεσιακού Συστήματος Διαχείρισης Βάσης Δεδομένων, όπως η Sybase. Η γλώσσα που επιλέξαμε, μας επιτρέπει να δηλώνουμε τα σύνολα προσυνθήκες/μετασυνθήκες με κάποιο προκαθορισμένο τύπο.

4.1 Περιγραφή της αρχιτεκτονικής του συστήματος

Όπως έγινε ήδη σαφές, από την εισαγωγική, κιάλας, του κεφαλαίου αυτού, υπάρχει έλλειψη αποτελεσματικών μεθόδων για τον έλεγχο της επαλήθευσης ή μη, των περιορισμών ακεραιότητας. Επίσης, είναι πολύ εμφανής η έλλειψη εργαλείων που θα βοηθούν τους σχεδιαστές των Βάσεων Δεδομένων στον προσδιορισμό σωστών δοσοληψιών.

Ένα τέτοιο εργαλείο, λοιπόν, αποφασίσαμε να υλοποιήσουμε στα πλαίσια της μεταπτυχιακής αυτής εργασίας. Το εργαλείο αυτό θα αποτελεί ένα σημαντικό μέρος ενός μεγαλύτερου συστήματος, το οποίο φαίνεται σε παρακάτω σχήμα(Σχήμα 1).

Στο σχήμα αυτό, μπορούμε να διακρίνουμε τέσσερα βασικά τμήματα. Τέσσερα, δηλαδή, διαφορετικά ημιαυτόνομα υποσυστήματα που η σύνθεσή τους παράγει το τελικό εργαλείο σχεδίασης και εκτέλεσης ενεργειών ενημέρωσης, εισαγωγής, διαγραφής, ή οτιδήποτε άλλων επιθυμητών πράξεων, από ένα χρήστη-σχεδιαστή δοσοληψιών, πάνω σε ένα ολοκληρωμένο σύστημα Βάσεων Δεδομένων ή ακόμα και Βάσεων Γνώσεων.



Πιο αναλυτικά, το πρώτο σύστημα είναι μία απλή εφαρμογή Βάσεων Δεδομένων. Είναι δηλαδή, ο φυσικός χώρος μνήμης, που βρίσκεται αποθηκευμένη η Βάση της εφαρμογής μας. Στην προκειμένη περίπτωση, η Βάση μας θα μπορούσε να είναι μία Σχεσιακή Βάση Δεδομένων. Ο χώρος αυτός θα μπορούσε να είναι κάποια απομακρυσμένη μονάδα μνήμης, όπως για παράδειγμα ο σκληρός δίσκος του υπολογιστή του χρήστη ή ακόμα και ένας κεντρικό υπολογιστής μίας εταιρίας που χρησιμοποιεί το εργαλείο, αν διαθέτει ένα κατακευματισμένο δίκτυο υπολογιστών. Ο λόγος που δεν συνυπάρχουν στην ίδια μονάδα μνήμης το εκτελέσιμο πρόγραμμα της εφαρμογής μας, με την Βάση, είναι προφανές και έχει να κάνει με την ταχύτητα εκτέλεσης και τις απαιτήσεις σε χρόνο απόκρισης του συστήματος. Επειδή οι απαιτήσεις για την εφαρμογή μας, συνήθως, θα είναι πολύ μεγάλες, είναι επιτακτική ανάγκη να υπάρχει σχεδόν μόνιμα, κατά την διάρκεια της εκτέλεσης, στην κεντρική μνήμη του εκάστοτε υπολογιστή.

Επίσης, το δεύτερο τμήμα της εφαρμογής, που ονομάζεται Μηχανισμός Ανακάλυψης και Δημιουργίας Δεδομένων Επεξεργασίας, είναι ένα εργαλείο που λειτουργεί, κατά κάποιο τρόπο, σαν διεπιφάνεια χρήσης μεταξύ της Βάσης

4.1 Περιγραφή της αρχιτεκτονικής του συστήματος

Δεδομένων και του τρίτου και σημαντικότερου μέρους του εργαλείου. Το σύστημα αυτό, διαβάζει τα δεδομένα που βρίσκονται αποθηκευμένα στην Βάση και τα μεταφράζει σύμφωνα με το μοντέλο του εργαλείου της εργασίας μας. Μία από τις αρμοδιότητές του είναι η μετατροπή κάθε συσχέτισης στο αντίστοιχο κατηγορημα της εφαρμογής μας και η ταυτόχρονη αποτίμησή του.

Έτσι, ως υποθέσουμε, για παράδειγμα, τις σχέσεις Employee, Department, Company και WorkIn, όπως τους είχαμε ορίσει στο προηγούμενο κεφάλαιο. Σύμφωνα με το σχεσιακό μοντέλο οι αντίστοιχοί πίνακες των οντοτήτων αυτών θα είναι οι εξής:

Employee (<i>Surname, First name, St_Id</i>).
Department (<i>Name, Dep_Id, Cmp_Id</i>).
Company (<i>Name, Address, Cmp_Id</i>)
WorkIn (<i>St_Id, Dep_Id, Cmp_Id</i>).

Αν υποθέσουμε τώρα ότι βάζουμε τις τιμές στους πίνακες ως εξής:

Employee

Kefalakis	Paulos	1
Kouvaris	Costas	2

Department

Software	1	1
Hardware	2	1

Company

Company1	1
Company2	2

WorkIn

1	1	1
2	1	1

Τότε, το δεύτερο τμήμα, παίρνοντας σαν είσοδο τους πίνακες που προαναφέραμε, θα παράγει σαν δεδομένα εξόδου τις σχέσεις: WorkIn(1,1,1) = Yes, WorkIn(2,1,1) = Yes. Οι υπόλοιπες σχέσεις που εξ ορισμού είναι ψευδείς, δεν είναι υποχρεωτικό να αναφερθούν. Η δουλειά όμως του τμήματος αυτού δεν τελειώνει εδώ. Μερικά άλλα εξίσου σημαντικά δεδομένα, που μας είναι χρήσιμα στο επόμενο στάδιο, όπως η κλάση(οντότητα) που ανήκει το κάθε όρισμα, της εκάστοτε

δοσοληψίας, είναι απαραίτητο να γίνουν στο στάδιο αυτό, ώστε να τα έχουμε έτοιμα όταν τα χρειαστούμε.

Επιπλέον, το τρίτο τμήμα, το επονομαζόμενο και Εργαλείο Παραγωγής Αξιωμάτων, είναι και το πλέον κρίσιμο για το εργαλείο μας. Στην ουσία είναι το τμήμα του συστήματος που υλοποιήσαμε στην εργασία αυτή. Τα υπόλοιπα, είτε είναι έτοιμα εκ των προτέρων, όπως η Βάση Δεδομένων, είτε η υλοποίησή τους δεν εξετάστηκε στην παρούσα φάση της εργασίας μας, όπως το δεύτερο και το τέταρτο τμήμα.

Στο τμήμα αυτό, κάναμε πράξη όσα είχαμε αρχικά πει σχετικά με τις μεθόδους επίλυσης του προβλήματος του πλαισίου και των επιπτώσεων. Σαν δεδομένο, λοιπόν, έχουμε ένα σύνολο από κατηγορήματα, συναρτήσεις, αλλά και δοσοληψίες, με πλήρη περιγραφή όσων αφορά τα ορίσματά τους, τις προσυνθήκες αλλά και τις μετασυνθήκες για την εκτέλεσή τους, αλλά και τις αναλλοίωτες συνθήκες. Για κάθε τέτοιο κατηγορημα, που με κάποιο τρόπο αποφασίζουμε ότι μας ενδιαφέρει, παράγουμε το αντίστοιχο αξίωμα επόμενης κατάστασης (βλέπε §3.2.4), ώστε, συγκεντρώνοντας όλα αυτά τα αξιώματα μαζί, να μπορέσουμε να αποφασίσουμε, στο επόμενο στάδιο, ποιοι περιορισμοί παραβιάζονται, αν παραβιάζονται και ποιοι παραμένουν αληθείς. Στο τμήμα αυτό της εργασίας μας επινοήσαμε μία γλώσσα για εισαγωγή και ορισμό δοσοληψιών πάνω σε χρονικές Βάσεις Δεδομένων. Η γλώσσα αυτή αποτελεί μία από τις καινοτομίες της εργασίας μας και εισάγει μερικές νέες ιδέες πάνω στον ορισμό δοσοληψιών.

Όπως, λοιπόν, αναφέραμε ήδη, το τέταρτο κομμάτι είναι το τελευταίο τμήμα του συστήματος και ονομάζεται Μηχανισμός Έλεγχου Ορθότητας Περιορισμών Ακεραιότητας. Η είσοδος του είναι τα αξιώματα που παραγάγαμε στο προηγούμενο στάδιο, ενώ αποκλειστικός σκοπός του είναι -σύμφωνα με κάποιες ιδιότητες των αξιωμάτων αυτών, που θα αναφέρουμε στην επόμενη ενότητα- η επαλήθευση ή όχι της αλήθειας των περιορισμών ακεραιότητας του συστήματος. Το τελικό αποτέλεσμα θα φαίνεται απευθείας στον χρήστη, για να τον προειδοποιεί για ενδεχόμενο λάθος στον τρόπο που διατύπωσε τους περιορισμούς τόσο συντακτικά, όσο και λογικά.

Με τον αυτόματο αυτό τρόπο ελέγχου, το σύστημα θα λειτουργεί προληπτικά, αποφεύγοντας την διαδικασία ανάκτησης της πρότερης πληροφορίας, σε περίπτωση που διαπιστωθεί παραβίαση κάποιου(-ων) περιορισμού ακεραιότητας, από μια ακολουθία ενημερώσεων ή γενικά από μία ακολουθία δοσοληψιών.

4.2 Περιγραφή των αλγορίθμων επεξεργασίας των δεδομένων

Στην παράγραφο αυτή, παρουσιάζουμε την υλοποίηση της λύσης του προβλήματος του πλαισίου και του προβλήματος των επιπτώσεων, όπως το παρουσιάσαμε σε προηγούμενη ενότητα, για μία κλάση περιορισμών ακεραιότητας που περικλείουν τόσο στατικούς, όσο και μεταβατικούς περιορισμούς. Η υλοποίηση αυτή αποτελεί και την υλοποίηση του τρίτου τμήματος του συστήματος που περιγράψαμε στην προηγούμενη παράγραφο.

Η επίλυση του προβλήματος του πλαισίου, προτάθηκε αρχικά από την εργασία [Rei91], στο περιβάλλον του Καταστατικού Λογισμού. Η λύση βασίζεται στην αυτόματη δημιουργία ολοκληρωμένου χαρακτηρισμού των συνθηκών, κάτω από τις οποίες, τα κατηγορήματα ή οι συναρτήσεις, μπορούν να αλλάξουν τιμή αληθείας, σαν αποτέλεσμα εκτέλεσης δοσοληψιών. Μία πιο αποτελεσματική μέθοδος επίλυσης των ίδιων προβλημάτων παρουσιάστηκε, όπως προαναφέραμε στην [Ple96]. Προκειμένου να υλοποιήσουμε τον αλγόριθμο που περιγράφεται στην εργασία [Ple96], κάναμε, πάνω σε αυτόν, ορισμένες αλλαγές και τροποποιήσεις. Στην εργασία μας, λοιπόν, παρουσιάζουμε μια επέκταση της μεθόδου [Ple96], για χειρισμό τέτοιων περιορισμών δοσοληψιών.

Στην εργασία αυτή, ακολουθούμε μία μέθοδο η οποία, πρώτα εκτελεί μία διαδικασία λεξικογραφικής και συντακτικής ανάλυσης, πάνω σε ένα σύνολο που αποτελείται από τους ορισμούς των δοσοληψιών, και κατόπιν, δημιουργεί αξιώματα επόμενης κατάστασης(βλ. §3.2.4), από ένα σύνολο αξιωμάτων αποτελέσματος(βλ. §3.2.2), με την παρουσία μίας περιορισμένης κλάσης περιορισμών και δηλώσεων μη πρωταρχικών κατηγορημάτων.

Για παράδειγμα, στην περίπτωση του κατηγορήματος WorkIn(βλέπε §4.1), ένα άμεσο αποτέλεσμα είναι ότι το πλήθος των εργαζομένων σε ένα τμήμα κάποιας εταιρίας, αυξάνεται αυτόματα μετά από κάθε εισαγωγή, νέου εργαζόμενου. Τα αξιώματα επόμενης κατάστασης χαρακτηρίζουν όλες τις συνθήκες κάτω από τις οποίες τα κατηγορήματα ή οι συναρτήσεις, μπορούν να αλλάξουν τιμή αληθείας, σαν αποτέλεσμα εκτέλεσης δοσοληψιών. Τέτοια αξιώματα μας διευκολύνουν στην τυποποίηση, σε ένα κλειστό κόσμο, των υποθέσεων μας, για τις δοσοληψίες, περισσότερο παρά για την ίδια την Βάση Γνώσεων. Οι περιορισμοί ακεραιότητας έχουν την μορφή:

$$\forall x_1/S_1, \dots, \forall x_k/S_k, \forall t_1, t_2/\text{Time } \varphi(x_1, \dots, x_k, t_1, t_2) \vee \\ (\neg)p_1(x_1, \dots, x_k, t_1) \vee (\neg)p_2(x_1, \dots, x_k, t_2)$$

όπου p_1, p_2 είναι κατηγορήματα με $(k + 1)$ ορίσματα, τα οποία δηλώνουν ένταση ή έκταση, οι S_1, \dots, S_k είναι κατηγορίες αντικειμένων, ενώ η φ είναι μία πρόταση στην οποία οι παράμετροι είναι απαλλαγμένοι από ποσοδείκτες, και δεν αναφέρουν άλλα κατηγορήματα, εκτός από αυτά που λαμβάνουν τιμή. Αυτή η κατηγορία περιορισμών επιτρέπει σε χρονικές μεταβλητές να υπάρξουν μέσα σε τέτοιου είδους κατηγορήματα. Περιλαμβάνει στατικούς περιορισμούς, καθώς και περιορισμούς μεταβάσεων, οι οποίοι αναφέρονται σε δύο διαδοχικές καταστάσεις της Βάσης Γνώσεων, αλλά δεν περιλαμβάνει, εν γένη, δυναμικούς περιορισμούς.

Μία δοσοληψία T , με παραμέτρους X , προσδιορίζεται από το ζεύγος $\langle \text{pre}_T(X), \text{post}_T(X) \rangle$, όπου δηλώνουν, αντίστοιχα, τις απαραίτητες προσυνθήκες, αλλά και μετασυνθήκες της δοσοληψίας T . Η λύση υποθέτει ότι οι παραγωγικοί κανόνες δεν είναι αναδρομικοί. Επιπλέον, υποτίθεται ότι είναι διαστρωματωμένοι. Επίσης, προϋποθέτει την ύπαρξη συμπερασματικών κανόνων, που περιγράφουν τα άμεσα αποτελέσματα των δοσοληψιών. Αυτοί οι συμπερασματικοί κανόνες εκφράζονται και ως *ευθεία αξιώματα αποτελέσματος* (direct effect axioms). Τα αξιώματα αυτά, για μια δοσοληψία T (ζεύγος $\langle \text{pre}_T(X), \text{post}_T(X) \rangle$) έχουν την μορφή [Ple96]:

$$\forall x/S \text{ Occur}(T(X) \Rightarrow (\text{pre}_T(X) \wedge @\text{post}_T(X))).$$

όπου, το κατηγορήμα Occur είναι μία παραλλαγή του αντίστοιχου do , της προηγούμενης παραγράφου, ενώ με το σύμβολο $@$ δηλώνουμε την επόμενη κατάσταση, στην οποία θα μεταβεί το σύστημα μετά την εκτέλεση της δοσοληψίας T . Δοθέντος κάθε προσδιορισμού μίας δοσοληψίας, τα αξιώματα αποτελέσματος παράγονται, ανεξάρτητα, από κάθε άλλο προσδιορισμό. Με τον τρόπο αυτό, αποφεύγουμε την ανάγκη να περιγράψουμε τα αξιώματα, σε μία αφηρημένη μορφή Λογικής, ώστε να μεταφράσουμε το κατηγορήμα Occur σε Λογική Πρώτης Τάξεως. Τα παραπάνω αξιώματα μπορούν να ξαναγραφούν ως εξής:

$$\forall x/S T(X) \Rightarrow (\text{pre}_T(X) \wedge @\text{post}_T(X)).$$

Πρώτη ενέργεια της εργασίας μας, στο σημείο αυτό, είναι η μετατροπή της έκφρασης των συνθηκών $\text{pre}_T(X)$ και $\text{post}_T(X)$ σε Διαζευκτική Κανονική

4.2 Περιγραφή των αλγορίθμων επεξεργασίας των δεδομένων

Μορφή(ΔΚΜ), όπως είπαμε και στην §3.1.1. Η μορφή αυτή μας είναι ιδιαίτερα χρήσιμη, γιατί, όπως φαίνεται παρακάτω, όλοι οι έλεγχοι ως προς την μορφή των συνθηκών αυτών γίνονται με βάση την μορφή αυτή. Για παράδειγμα, στο Βήμα 3 και Βήμα 4 του ακόλουθου αλγορίθμου, ελέγχουμε αν το $\text{post}_T(X)$ έχει την μορφή

$$\gamma(X) \Rightarrow P(X)$$

η οποία σύμφωνα με μία ιδιότητα του τελεστή \Rightarrow , μπορεί να γραφτεί και ως

$$\neg\gamma(X) \vee P(X)$$

που είναι σε ΔΚΜ. Η διαδικασία μετατροπής οποιασδήποτε έκφρασης, σε ΔΚΜ είναι σχετικά απλή στην σύλληψή της, αλλά παρουσιάζει ορισμένες τεχνικές δυσκολίες στην υλοποίηση. Έχει, σαν ιδέα, απασχολήσει πολλούς ερευνητές, ανά την υφήλιο, εδώ και πολλά χρόνια, αλλά δεν υπάρχει κάποιο έτοιμο διαθέσιμο εμπορικό εργαλείο που να την αυτοματοποιεί. Έτσι, ένα τμήμα της εργασίας μας ήταν και η υλοποίησή της. Όλες οι απαραίτητες ρουτίνες που υλοποιήσαμε για να μας βοηθήσουν στην μετατροπή αυτή φαίνονται στην τελευταία παράγραφο του Κεφαλαίου 4.

Από τα άμεσα αυτά αξιώματα μπορούμε εύκολα να παράγουμε *θετικά* και *αρνητικά* αξιώματα αποτελεσμάτων(βλέπε §3.2.2)[BMR95], για κάθε κατηγορία P , που λαμβάνει μέρος στις συνθήκες pre_T και post_T , όπως θα περιγράψουμε στα παρακάτω βήματα. Η λογική είναι να περιγράψουμε, επακριβώς, όλες τις συνθήκες που είναι απαραίτητα αληθείς όταν ένα κατηγορημα αλλάζει τιμή από Ψευδής σε Αληθείς(ή το αντίστροφο, αντίστοιχα). Ο αλγόριθμος που ακολουθεί είχε αρχικά διατυπωθεί στην [Ple96], αλλά κρίθηκαν απαραίτητες ορισμένες προσθήκες και τροποποιήσεις ώστε να μπορέσουμε να τον υλοποιήσουμε. Έτσι, για κάθε κατηγορημα ή συνάρτηση P θα εκτελέσουμε τα ακόλουθα Βήματα:

1. Κατασκευάζουμε τα ακόλουθα θετικά και αρνητικά αξιώματα για το P :

$$\text{Θετικό:} \quad \forall X/S \quad \neg P(X) \wedge @P(X) \wedge \neg T(X) \Rightarrow \text{False} \quad (1)$$

$$\text{Αρνητικό:} \quad \forall X/S \quad P(X) \wedge \neg @P(X) \wedge \neg T(X) \Rightarrow \text{False} \quad (2)$$

2. Εάν το $\text{post}_T(X)$ περιλαμβάνει μόνο το $@P(X)$ (αντίστοιχα, μόνο το $\neg @P(X)$), τότε πρόσθεσε στο θετικό αξίωμα(αντίστοιχα στο αρνητικό) για το P , του βήματος 1, την τιμή True , με τη μορφή διάζευξης

3. Εάν το $\text{post}_T(X)$ είναι της μορφής $\gamma(X) \Rightarrow (\neg)@P(X)$, όπου το γ δεν περιλαμβάνει κανένα όρο που αναφέρεται στην επόμενη κατάσταση, πρόσθεσε με τη μορφή διάζευξης, τον όρο $\gamma(X)$, στο θετικό(αντίστοιχα στο αρνητικό) αξίωμα για το P .
4. Εάν το $\text{post}_T(X)$ είναι της μορφής $\exists Z (\gamma(X,Z) \Rightarrow (\neg)@P(W))$, όπου το W αποτελείται από σταθερές και μεταβλητές από το X, Z , πρόσθεσε με τη μορφή διάζευξης, τον όρο $\exists Z (\gamma(X,Z) \wedge (X = W))$, στο θετικό(αντίστοιχα στο αρνητικό) αξίωμα για το P .

Η διαδικασία αυτή παράγει ένα σύνολο T_{ef} από αξιώματα της μορφής:

$$\forall X/S \neg P(X) \wedge \neg \Phi_{1P} \Rightarrow \neg @P(X) \quad (3)$$

$$\forall X/S P(X) \wedge \neg \Phi_{2P} \Rightarrow @P(X) \quad (4)$$

Στον αλγόριθμο αυτό, θα ήταν σκόπιμο να κάνουμε ορισμένες επεξηγήσεις. Κατά πρώτον, είναι πολύ σημαντικό να τονίσουμε ότι μας ενδιαφέρουν τα αξιώματα **μόνο** όσων κατηγορημάτων ή συναρτήσεων αναφέρονται στην επόμενη κατάσταση της εκτέλεσης μίας δοσοληψίας. Για τον λόγο αυτό μόνο αυτά τα κατηγορήματα θα λαμβάνουμε υπόψη στην αναδρομική εκτέλεση των σταδίων του αλγορίθμου. Επίσης, μία εξίσου σημαντική παρατήρηση είναι ότι δεν επιτρέπεται σε ένα κατηγορήμα ή συνάρτηση που βρίσκεται στην συνθήκη $\text{pre}_T(X)$, να αναφέρεται στην επόμενη κατάσταση του συστήματος, δηλαδή την κατάσταση που το σύστημα οδηγείται μετά την εκτέλεση της εν λόγω δοσοληψίας. Επομένως, θεωρούμε ότι το $\text{pre}_T(X)$ περιέχει μόνο όρους που αναφέρονται στην παρούσα κατάσταση του συστήματος, ενώ στο $\text{post}_T(X)$ περιέχονται και όροι που αναφέρονται στην επόμενη κατάσταση.

Επιπλέον, πριν ακολουθήσουμε το Βήμα 1, θα πρέπει να αποφασίσουμε ποιο από τα δύο είδη αξιωμάτων (θετικό ή αρνητικό) αποτελέσματος θα δημιουργήσουμε κάθε φορά. Έτσι, έχουμε τις ακόλουθες περιπτώσεις:

- Όταν τόσο στο $\text{pre}_T(X)$, όσο και $\text{post}_T(X)$ εμφανίζεται το συγκεκριμένο κατηγορήμα ή συνάρτηση, στην ίδια μορφή, δηλαδή είτε και τα δύο εμφανίζονται σε μορφή συμπληρώματος είτε και τα δύο σε απλή μορφή, τότε δεν λαμβάνουμε υπόψη το αντίστοιχο P και προχωράμε στο επόμενο κατηγορήμα.

4.2 Περιγραφή των αλγορίθμων επεξεργασίας των δεδομένων

- Όταν το $P(X)$ δεν εμφανίζεται καθόλου στο $\text{pre}_T(X)$, είτε εμφανίζεται, με μορφή συμπληρώματος, ενώ το $@P(X)$ βρίσκεται σε απλή μορφή στο $\text{post}_T(X)$, λαμβάνουμε μόνο το θετικό αξίωμα, δηλαδή το αξίωμα της μορφής (1).
- Στην ακριβώς αντίστροφη περίπτωση, όπου το $P(X)$ δεν εμφανίζεται καθόλου στο $\text{pre}_T(X)$, είτε εμφανίζεται, με απλή μορφή, ενώ το $@P(X)$ βρίσκεται σε μορφή συμπληρώματος στο $\text{post}_T(X)$, λαμβάνουμε μόνο το αρνητικό αξίωμα, δηλαδή το αξίωμα της μορφής (2).
- Όταν τόσο το $P(X)$ στο $\text{pre}_T(X)$, όσο και το $@P(X)$ στο $\text{post}_T(X)$, εμφανίζονται και σε απλή μορφή, αλλά και σε μορφή συμπληρώματος, τότε λαμβάνουμε και τα δύο αξιώματα.
- Σε κάθε άλλη περίπτωση δεν μας ενδιαφέρει κανένα αξίωμα.

Επιπλέον, για το Βήμα 2 πρέπει να κατανοήσουμε τι σημαίνει για το παραγόμενο αξίωμα, που σχετίζεται με το P , να του προσθέσουμε την τιμή True , με τη μορφή διάζευξης. Όπως είναι γνωστό από τις λογικές πράξεις, ισχύει ότι " $A \vee \text{True} \equiv \text{True}$ ". Επομένως, αυτό που κάνουμε στην περίπτωση αυτή, είναι να δηλώσουμε έμμεσα, ότι δεν μας ενδιαφέρει πλέον αυτό το αξίωμα γιατί δεν μας δίνει επιπλέον πληροφορίες.

Τα αξιώματα αυτά, περιγράφουν επακριβώς πως οι δοσοληψίες επηρεάζουν, άμεσα, την τιμή αληθείας των κατηγορημάτων. Μένει μόνο να περιγράψουμε πλέον τα έμμεσα αποτελέσματα, εξαιτίας της παρουσίας των περιορισμών ακεραιότητας.

Ας υποθέσουμε ότι έχουμε:

$$C: \forall X/S \varphi(X) \vee P(X) \vee @Q(X) \quad (I)$$

το οποίο είναι ένας περιορισμός ακεραιότητας που πρέπει να ικανοποιείται πάντα. Τότε, για κάθε αξίωμα της μορφής (3), για το P , δημιουργούμε το ακόλουθο αξίωμα που σχετίζεται με το Q :

$$\forall X/S \neg P(X) \wedge \neg \Phi_{1P}(X) \wedge \neg \varphi(X) \Rightarrow @Q(X), \quad (II)$$

όπου το $\Phi_{1P}(X)$ παράγεται από την σχέση (3) που γράψαμε προηγουμένως. Το αξίωμα αυτό εκφράζει την ιδιότητα, ότι εάν το κατηγορημα P είναι μη αληθές, στην κατάσταση που προηγείται της εκτέλεσης μιας δοσοληψίας και ο περιορισμός C ικανοποιείται στην ίδια κατάσταση, τότε -εάν η συνθήκη, που οδήγησε το P σε

αλλαγή της τιμής του, από Ψευδές σε Αληθές, δεν ικανοποιείται- το $(@Q(X) \vee \varphi(X))$ πρέπει να είναι αληθές για να παραμένει έγκυρος ο περιορισμός, στην επόμενη κατάσταση της εκτέλεσης της εν λόγω δοσοληψίας.

Συμμετρικά, για κάθε αξίωμα της μορφής (3), για το κατηγορήμα Q, έχουμε το ακόλουθο αξίωμα:

$$\forall X/S \neg Q(X) \wedge \neg \Phi_{1Q}(X) \wedge \neg \varphi(X) \Rightarrow @P(X).$$

Η αντίστοιχη διαδικασία λαμβάνει χώρο εάν οι περιορισμοί περιέχουν αναιρούμενα κατηγορήματα(κατηγορήματα που δεν είναι αληθή, αλλά πριν από αυτά υπάρχει το σύμβολο \neg). Στο σημείο αυτό, κάτω από την υπόθεση ότι οι δοθέντες προσδιορισμοί χαρακτηρίζουν όλες τις δοσοληψίες, μπορούμε να δημιουργήσουμε το σύνολο T_{ss} των αξιωμάτων επόμενης κατάστασης όπως φαίνεται παρακάτω. Ας ορίσουμε τις ακόλουθες εκφράσεις:

$$\Psi_P(X) = \neg \Phi_{1Q}(X) \wedge \neg \varphi(X). \quad (5) \quad \Psi_Q(X) = \neg \Phi_{1P}(X) \wedge \neg \varphi(X). \quad (7)$$

$$\Psi_{\neg P}(X) = \neg \Phi_{2Q}(X) \wedge \neg \varphi(X). \quad (6) \quad \Psi_{\neg Q}(X) = \neg \Phi_{2P}(X) \wedge \neg \varphi(X). \quad (8)$$

Τότε, τέλος, τα αξιώματα επόμενης κατάστασης, για το P, είναι της ακόλουθης μορφής:

$$\forall X/S \Psi_P(X) \vee (\Psi_{\neg P}(X) \wedge P). \quad (9)$$

Στο σημείο αυτό θα αναφέρουμε ένα παράδειγμα εφαρμογής του αλγορίθμου που υλοποιήσαμε. Ας υποθέσουμε τις δοσοληψίες DropProfDep και BecomeAdvisor, των οποίων η πλήρης περιγραφή γίνεται στο Κεφάλαιο 5. Οι περιγραφές τους, φαίνονται στους ακόλουθους τύπους:

DropProfDep(prof, dep)

Precondition: **pr_belongs_dep**(prof, dep)

Postcondition: not **@ pr_belongs_dep** (prof, dep) and not **@Is_advisor**(prof, dep)
and (**@num_prof**(dep) = **num_prof**(dep) -1)

Invariant: **num_prof**(dep) >0 and

(not exist dep₁, dep₂/Department(**pr_belongs_dep**(prof, dep₁) and **pr_belongs_dep**(prof, dep₂) and (dep₁ != dep₂)))

BecomeAdvisor(prof, st, dep)

Precondition: **postgraduate**(st, dep) and not **has_advisor**(st, prof, dep)

Postcondition: **@has_advisor**(st, prof, dep) and **@is_advisor**(prof, dep)

Invariant **@is_advisor**(prof, dep) or
is_advisor(prof, dep)

Αναλύοντας την πρώτη περιγραφή, έχουμε σαν όνομα δοσοληψίας το **DropProfDep**(prof, dep), σαν pre_T την έκφραση **pr_belongs_dep**(prof, dep), σαν $post_T$ την έκφραση (not **@pr_belongs_dep**(prof, dep) and not **@Is_advisor**(prof, dep) and (**@num_prof**(dep) = **num_prof**(dep) - 1). Αντίστοιχα για την δεύτερη δοσοληψία, έχουμε τις εκφράσεις **BecomeAdvisor**(prof, st, dep), (**postgraduate**(st, dep) and not **has_advisor**(st, prof, dep)) και (**@has_advisor**(st, prof, dep) and **@is_advisor**(prof, dep)).

Σύμφωνα με τον αλγόριθμό μας, στο πρώτο βήμα θα παράγουμε για την δοσοληψία **DropProfDep**, ως προς το κατηγορημα **prof_bel_dep**, το αρνητικό αξίωμα αποτελέσματος με τύπο τον εξής:

for all prof/Professor for all dep/Department (
DropProfDep(prof, dep) or
not **prof_bel_dep**(prof, dep) or **@prof_bel_dep**(prof, dep))

Το αξίωμα πρέπει να είναι αρνητικό γιατί το κατηγορημα **prof_bel_dep** εμφανίζεται σαν αληθές στην τρέχουσα κατάσταση της Βάσης και σαν ψευδές στην επόμενη. Αντίστοιχα, ως προς το κατηγορημα **is_advisor**, παράγεται το αρνητικό αξίωμα αποτελέσματος με τύπο:

for all prof/Professor for all dep/Department (
DropProfDep(prof, dep) or
not **is_advisor**(prof, dep) or **@ is_advisor**(prof, dep)).

Κατά αντιστοιχία ο αλγόριθμος καθορίζει και τον τύπο του αξιώματος της δοσοληψίας **BecomeAdvisor**, ως προς το κατηγορημα **has_advisor**, το οποίο πρέπει να είναι θετικό και έχει τον ακόλουθο τύπο:

for all prof/Professor for all st/Student for all dep/Department(
BecomeAdvisor(prof, st, dep) or

not **has_advisor**(st, prof) or **@has_advisor**(st, prof))

Κατά ανάλογο τρόπο, επίσης, ως προς το κατηγορημα **is_advisor** έχουμε την έκφραση:

for all prof/Professor for all st/Student for all dep/Department(
BecomeAdvisor(prof, st, dep) or
 not **@is_advisor**(prof, dep) or **is_advisor**(prof, dep)).

Στο επόμενο βήμα του αλγορίθμου, ενώ έχουμε φτιάξει σχέσεις της μορφής (1) και (2) αντίστοιχα, προσπαθούμε να κατατάξουμε τις εκφράσεις αυτές σύμφωνα με τις μετασυνθήκες σε μία από τις τρεις κατηγορίες του αλγορίθμου. Τόσο στην περίπτωση του **DropProfDep**, όσο και στην περίπτωση του **BecomeAdvisor**, βλέπουμε ότι δεν ανήκουν σε κάποια κατηγορία του αλγορίθμου, οπότε και παραμένουν ως έχουν. Ένα παράδειγμα δοσοληψίας, που ανήκει στην περίπτωση 3, είναι αυτή της οποίας το post_T είναι η έκφραση: for all x/X , for all y/Y for all z/Z $Q(x) \Rightarrow P(x, z, y)$. Παρόμοια για την περίπτωση 4, θα μπορούσαμε να έχουμε την έκφραση: for all x/X , for all y/Y for all z/Z $Q(x, z, y) \Rightarrow P(x)$.

Στην επόμενη φάση του αλγορίθμου μας θα πρέπει να φτιάξουμε αξιώματα της μορφής (3) ή (4). Για το παράδειγμά μας, στην περίπτωση του **DropProfDep**, θα έχουμε, ως προς το κατηγορημα **prof_bel_dep**, το αξίωμα:

for all prof/Professor for all dep/Department (
DropProfDep(prof, dep) or
 not **prof_bel_dep**(prof, dep) or **@prof_bel_dep**(prof, dep)).

όπου, $P(X) \equiv \text{prof_bel_dep}(\text{prof}, \text{dep})$ και η $\Phi_{2P} \equiv \text{DropProfDep}(\text{prof}, \text{dep})$, ενώ ως προς το κατηγορημα **is_advisor**(st, prof), κατά παρόμοιο τρόπο έχουμε:

for all prof/Professor for all dep/Department (
DropProfDep(prof, dep) or
 not **is_advisor**(prof, dep) or **@is_advisor**(prof, dep))

οπότε, επίσης, $\Phi_{2P} \equiv \text{DropProfDep}(\text{prof}, \text{dep})$.

4.3 Περιγραφή των δεδομένων εισόδου

Κατά, ακριβώς αντίστοιχο τρόπο, για την δοσοληψία **BecomeAdvisor**(prof, st, dep) έχουμε, ως προς το **is_advisor**, $\Phi_{1P} \equiv \mathbf{BecomeAdvisor}(\text{prof}, \text{st}, \text{dep})$.

Στο τελικό στάδιο του αλγορίθμου μας, παράγουμε τα αξιώματα επόμενης κατάστασης σύμφωνα με την περιγραφή που προηγήθηκε. Για την περίπτωση της δοσοληψίας **DropProfDep**(prof, dep), η αναλλοίωτη συνθήκη δεν είναι στην μορφή που επιθυμούμε, δηλαδή δεν έχει τουλάχιστον ένα κατηγορήματα το οποίο αναφέρονται στην επόμενη κατάσταση, οπότε περιορίζουμε την περιγραφή μας μόνο στην δεύτερη δοσοληψία, την **BecomeAdvisor**(prof, st, dep). Στην περιγραφή αυτή ως αναλλοίωτη συνθήκη έχουμε την έκφραση: (**@is_advisor**(prof, dep) or **is_advisor**(prof, dep)), όπου σύμφωνα με την σχέση (I), προκύπτει ότι $\varphi(X) \equiv \text{False}$, δηλαδή η τιμή του $\varphi(X)$ είναι η ψευδής για να μην επηρεάζει την έκφραση της αναλλοίωτης συνθήκης. $P(X) \equiv \mathbf{is_advisor}(\text{prof}, \text{dep})$, $Q(X) \equiv \mathbf{@is_advisor}(\text{prof}, \text{dep})$. Σύμφωνα με την περιγραφή του αλγορίθμου αυτού, έχοντας ότι $\Phi_{1Q} \equiv \mathbf{BecomeAdvisor}(\text{prof}, \text{st}, \text{dep})$ και $\Phi_{2Q} \equiv \mathbf{DropProfDep}(\text{prof}, \text{dep})$, ακολουθώντας τους τύπους (5)-(8) έχουμε σαν $\Psi_P \equiv \text{not } \mathbf{BecomeAdvisor}(\text{prof}, \text{st}, \text{dep})$, $\Psi_{\neg P} \equiv \text{not } \mathbf{DropProfDep}(\text{prof}, \text{dep})$. Έτσι, το αξίωμα επόμενης κατάστασης που φαίνεται στον τύπο (9), γίνεται:

for all prof/Professor for all st/Student for all dep/Department(
not **DropProfDep**(prof, dep) and **is_advisor**(prof, dep) or
not **BecomeAdvisor**(prof, st, dep)).

Η έκφραση αυτή σημαίνει ότι για κάθε καθηγητής, που είναι εγγεγραμμένος σε ένα πανεπιστημιακό τμήμα, δεν θα έχει εκτελεστεί η διαγραφή του(not **DropProfDep**(prof, dep)). Επίσης, για να μην θεωρείται Ακαδημαϊκός Σύμβουλος κανενός φοιτητή(που σημαίνει ότι **is_advisor**(prof, dep) = False), δεν θα πρέπει να έχει δηλωθεί ως Ακαδημαϊκός Σύμβουλος(δηλαδή not **BecomeAdvisor**(prof, st, dep)).

4.3 Περιγραφή των δεδομένων εισόδου

Σε προηγούμενη παράγραφο(βλέπε §4.1), είχαμε επισημάνει μερικά από τα απαραίτητα δεδομένα εισόδου του προγράμματός μας. Το δεύτερο τμήμα

του συστήματος(Μηχανισμός Ανακάλυψης και Δημιουργίας Δεδομένων Επεξεργασίας) μας παρέχει τα απαραίτητα δεδομένα εισόδου το οποία είναι τα εξής:

- 1) Μία ακολουθία από δηλώσεις δοσοληψιών, με όλες τις συνθήκες που αναφέραμε σε προηγούμενη ενότητα και όλους τους αντίστοιχους περιορισμούς.
- 2) Ένα σύνολο με όλα τα ονόματα των τύπων των μεταβλητών και των ορισμάτων για κάθε κατηγορία ή συνάρτηση, που εμφανίζεται στην εφαρμογή μας.
- 3) Ένα σύνολο με το όνομα του τύπου κάθε μεταβλητής η οποία αποτελεί και όρισμα στον ορισμό της αντίστοιχης δοσοληψίας.

Πιο αναλυτικά, για το (1) έχουμε ένα αρχείο εισόδου το οποίο έχει την δομή που θα περιγράψουμε αμέσως τώρα. Το συντακτικό της γλώσσας φαίνεται στο σχήμα που ακολουθεί. Αποτελείται από δηλώσεις δοσοληψιών της μορφής:

Όνομα δοσοληψίας(Σύνολο παραμέτρων)

Precondition: Μία έκφραση με παρουσία ποσοδεικτών.

Postcondition: Μία έκφραση με παρουσία ποσοδεικτών.

Invariants: Μία έκφραση με παρουσία ποσοδεικτών.

Ο ορισμός της δοσοληψίας γίνεται σύμφωνα με την γλώσσα που επινοήσαμε. Στα πεδία Precondition, Postcondition και Invariants, οι εκφράσεις που απαιτούνται είναι παρόμοιες με τις κανονικές εκφράσεις οποιασδήποτε άλλης γραμματικής. Η διαφορά της με τις υπόλοιπες είναι στην εισαγωγή του χρονικού στοιχείου στα κατηγορήματα. Έτσι, αν το κατηγορημα αναφέρεται στην κατάσταση που έπεται της ενημέρωσης εισάγουμε το σύμβολο @ για να δηλώσουμε την επόμενη κατάσταση. Η απουσία του από το κατηγορημα δηλώνει ότι το τελευταίο αναφέρεται στην παρούσα κατάσταση. Επίσης, δεν γίνεται αποδεκτό το σύμβολο @, στην έκφραση του πεδίου Precondition, γιατί εξορισμού το πεδίο αυτό αναφέρεται στην παρούσα κατάσταση. Κατά τα υπόλοιπα, οι εκφράσεις που χρησιμοποιεί η γλώσσα μας είναι απλές κανονικές εκφράσεις.

Για τον ορισμό αυτό, πρώτα ορίζουμε την ίδια την δοσοληψία, δίνοντάς της ένα όνομα και ορισμένες χρήσιμες παραμέτρους. Είναι συνήθως ένα όνομα, παρόμοια με την περίπτωση δήλωσης συναρτήσεων, με την διαφορά ότι το όνομα αρχίζει από κεφαλαίο γράμμα. Επειδή πάντα θα πρέπει να γνωρίζουμε και την κατηγορία(δηλαδή την κλάση ή την οντότητα, στο πεδίο των Βάσεων Δεδομένων), στην οποία ανήκει το κάθε γνώρισμα. Αυτή την πληροφορία μας την δίνει ο ενδιάμεσος μηχανισμός που προαναφέραμε και για τον λόγο αυτό μας είναι πολύ χρήσιμος.

Στο πεδίο Precondition, περιγράφονται με την βοήθεια μιας λογική έκφρασης, όλες οι προσυνθήκες της εν λόγω δοσοληψίας. Την έκφραση αυτή την επεξεργάζεται ο γραμματικός και εν συνεχεία ο συντακτικός αναλυτής μας, παράγοντας μας ένα σύνολο από δομές, χρήσιμες για την επεξεργασία των δεδομένων αυτών. Έτσι, όταν η έκφραση περάσει από λεξικογραφική και συντακτική ανάλυση, περνάει από ένα σύνολο επεξεργασιών. Πιο αναλυτικά, σε πρώτη φάση, γίνεται απαλοιφή των ποσοδεικτών, δηλαδή σε όποιο βάθος και αν συναντήσουμε κάποιο ποσοδείκτη, τον απαλείφουμε, τοποθετώντας τον στην αρχή της έκφρασης. Στην συνέχεια αφαιρούμε οποιαδήποτε μεταβλητή δεν χρησιμοποιείται αλλά προήλθε από κάποιο λάθος του χρήστη, οδηγώντας το σύστημα στην βέλτιστη και οικονομικότερη, από άποψη απαιτούμενου χώρου αποθήκευσης, μορφή. Μόλις τελειώσει η φάση αυτή, εκτελούμε τον αλγόριθμο μετατροπής σε ΔΚΜ, όπως αναφέραμε και προηγουμένως. Στο τέλος και αυτής της φάσης έχουμε, πλέον, όλη την απαραίτητη πληροφορία και αρχίζουμε την αποθήκευση των κατηγορημάτων και των συναρτήσεων στις κατάλληλες δομές.

Στο πεδίο Postcondition, περιγράφονται με την βοήθεια μιας λογική έκφρασης, όλες οι μετασυνθήκες της εν λόγω δοσοληψίας. Η διαδικασία επεξεργασίας είναι παρόμοια με προηγουμένως με την διαφορά ότι μόλις τελειώσουν όλες οι παραπάνω λειτουργίες, ακολουθεί η υλοποίηση του αλγορίθμου που περιγράψαμε στην ενότητα 4.2. Δημιουργούμε δηλαδή, σταδιακά, όλα τα αξιώματα της μορφής (3) και (4)

Τέλος, στο πεδίο Invariants, περιγράφονται, πάλι με την βοήθεια μιας λογική έκφρασης, όλες οι απαραίτητες αναλλοίωτες συνθήκες, που πρέπει να ικανοποιηθούν καθ' όλη την διάρκεια ζωής της εφαρμογής μας. Η επεξεργασία της αντίστοιχης έκφρασης είναι όμοια με προηγουμένως αλλά, στο τέλος, όλες οι συνθήκες αυτές συγκεντρώνονται σε μία λίστα, προκειμένου να υλοποιήσουμε το τελευταίο στάδιο της μεθόδου. Δηλαδή, μετά την δημιουργία των αξιωμάτων της μορφής (3) και (4), να δημιουργήσουμε τα τελικά αξιώματα της μορφής (9).

4.4 Περιγραφή των δεδομένων εξόδου

Τέλος, όπως έχουμε αναφέρει, σκοπός της εργασίας αυτής είναι η παραγωγή αξιωμάτων της μορφής (9). Δηλαδή, αξιωμάτων επόμενης κατάστασης, τα οποία θα αποτελούνται από ένα σύνολο δηλώσεων μεταβλητών, που εισάγονται από ποσοδείκτες και ανήκουν σε μία συγκεκριμένη κατηγορία όπως Time, Integer, String κ.λ.π. Όταν ένα γραμματικό ή συντακτικό λάθος συμβεί κατά την δήλωση μίας δοσοληψίας, το πρόγραμμά μας παράγει ένα μήνυμα λάθους και σταματάει την διαδικασία μετάφρασης,

Επίσης, το σύνολο αυτών των αξιωμάτων θα αποτελέσει είσοδο του τέταρτου και τελευταίου τμήματος του συστήματος μας (Μηχανισμός Έλεγχου Ορθότητας Περιορισμών Ακεραιότητας) και θα αποφασίζει πότε και εάν, ένας περιορισμός ακεραιότητας παραβιάζεται, ώστε να προειδοποιεί τον χρήστη σχεδιαστή να αλλάξει τον τρόπο που έχει δηλώσει κάποιες δοσοληψίες. Πιο συγκεκριμένα, για να αποφασίσουμε πότε ένας περιορισμός ακεραιότητας παραβιάζεται, βασιζόμαστε στο Πόρισμα 2 (βλέπε §3.1.1).

4.5 Περιγραφή του αλγορίθμου μετατροπής σε Δ.Κ.Μ.

Στην παράγραφο αυτή θα παρουσιάσουμε την υλοποίηση του αλγορίθμου μετατροπής μίας κανονικής έκφρασης σε Διαζευκτική Κανονική Μορφή (ή ΔΚΜ). Μία έκφραση λέμε ότι βρίσκεται σε ΔΚΜ όταν και μόνο όταν οι όροι της αποτελούνται από διαζεύξεις συζευγμένων όρων. Όταν, δηλαδή, είναι της μορφής $((A \vee B) \wedge (C \wedge D))$. Ο αλγόριθμος αυτός απασχόλησε πολλούς ερευνητές στον χώρο των Βάσεων Δεδομένων, αλλά, σύμφωνα με την έρευνα που κάναμε, σε παγκόσμια κλίμακα, δεν υπάρχει ένα έτοιμο εργαλείο, που να εκτελεί τον αλγόριθμο αυτό και να είναι ευέλικτο στις τροποποιήσεις, που θα επιθυμούσαμε να κάνουμε, για να το εντάξουμε στο εργαλείο μας. Για τον λόγο αυτό αποφασίσαμε να υλοποιήσουμε ένα δικό μας αλγόριθμο, βασιζόμενοι σε γενικές αρχές γύρω από την θεωρία της Λογικής Κατηγορημάτων. Στην περιοχή αυτή έχει ήδη γίνει μία θεωρητική πρόταση για τον εν λόγω αλγόριθμο.

4.5 Περιγραφή του αλγορίθμου μετατροπής σε Δ.Κ.Μ.

Πιο αναλυτικά βασιστήκαμε στις ακόλουθες ιδιότητες του Λογικής Κατηγορημάτων:

1. $(A \wedge B \equiv B \wedge A)$ και $(A \vee B \equiv B \vee A)$ (μεταθετικότητα του \wedge και \vee αντίστοιχα).
2. $((A \wedge B) \wedge \Gamma \equiv A \wedge (B \wedge \Gamma))$ και $((A \vee B) \vee \Gamma \equiv A \vee (B \vee \Gamma))$
(προσεταιριστικότητα του \wedge και \vee αντίστοιχα)
3. $(A \wedge A \equiv A)$ και $(A \vee A \equiv A)$ (αυτοπάθεια του \wedge και \vee αντίστοιχα).
4. $(A \wedge (B \vee \Gamma) \equiv (A \wedge B) \vee (A \wedge \Gamma))$ και $(A \vee (B \wedge \Gamma) \equiv (A \vee B) \wedge (A \vee \Gamma))$
(επιμερισμός του \wedge και \vee αντίστοιχα).
5. $(A \wedge (A \vee B) \equiv A)$ και $(A \vee (A \wedge B) \equiv A)$ (απορρόφηση του \wedge και \vee αντίστοιχα).

Επίσης, ένα άλλο σύνολο ιδιοτήτων που χρησιμοποιήσαμε από τον Προτασιακό Λογισμό είναι οι:

1. $x/X \neg A(x) \equiv \neg \exists x/X A(x)$
2. $\exists x/X \neg A(x) \equiv \neg (\forall x/X A(x))$
3. $\forall x/X A(x) \wedge (\vee) \forall y/Y C(x) \equiv \forall x/X, \forall y/Y (A(x) \wedge (\vee) C(y))$
4. $\forall x/X A(x) \wedge (\vee) C \equiv \forall x/X (A(x) \wedge (\vee) C)$, όπου C είναι μία σταθερά
5. Τέλος, χρησιμοποιήσαμε και την αναπαράσταση μίας έκφρασης με **Κανονική Μορφή Skolem**[Μητ86].

Για να γίνει κατανοητή η διαδικασία του αλγορίθμου ακολουθεί ένα αντιπροσωπευτικό παράδειγμα. Στο εργαλείο μας η ανάγνωση των δεδομένων εισόδου που μας χρειάζονται για την εκτέλεση του αλγορίθμου γίνεται σειριακά, με την βοήθεια ενός συντακτικού αναλυτή, διαβάζοντας έναν ένα τους όρους, όχι όλους μαζί, όπως απαιτούν οι ήδη διαθέσιμες μέθοδοι. Αυτή είναι η πρώτη καινοτομία του αλγορίθμου. Ας υποθέσουμε την έκφραση:

$$(\forall x/X B(x)) \wedge ((\forall z/Z, \forall y/Y (A(z) \wedge (A(z) \vee B(z) \vee C(y))) \vee \exists u/U C(u)).$$

- Διαβάζω την πρώτη έκφραση $(\forall x/X B(x))$, η οποία δεν έχει ανάγκη από καμία τροποποίηση, οπότε η αρχική έκφραση παραμένει $(\forall x/X B(x)) \wedge ((\forall z/Z, \forall y/Y (A(z) \wedge (A(z) \vee B(z) \vee C(y))) \vee \exists u/U C(u))$.
- Διαβάζω την έκφραση $(A(z) \wedge (A(z) \vee B(z) \vee C(y)))$ και κάνω τις πράξεις, μετασχηματίζοντάς την στην $((A(z) \wedge A(z)) \vee (A(z) \wedge B(z)) \vee (A(z) \wedge C(y)))$

- Με την βοήθεια μίας διαδικασίας που υλοποιήσαμε (την `prone_conj`), απαλείφουμε τον όρο $A(z)$ από τον όρο $(A(z) \wedge A(z))$. Έτσι, η νέα έκφραση γίνεται $A(z) \vee (A(z) \wedge B(z)) \vee (A(z) \wedge C(y))$. Η αρχική έκφραση γίνεται: $(\forall x/X B(x)) \wedge ((\forall z/Z, \forall y/Y (A(z) \vee (A(z) \wedge B(z)) \vee (A(z) \wedge C(y))) \vee \exists u/U C(u))$.
- Μετατρέπω το $(\exists u/U C(u))$ σε Skolem Κανονική Μορφή, ως C_0 , ώστε η τελική μας πρόταση να περιέχει μόνο καθολικούς ποσοδείκτες. Έτσι, η νέα έκφραση γίνεται $(\forall x/X B(x)) \wedge (\forall y/Y, \forall z/Z (A(z) \vee (A(z) \wedge B(x)) \vee (A(z) \wedge C(y)))) \vee C_0$.
- Μεταφέρω όλες τις μεταβλητές x, y, z στην αρχή της έκφρασης και θα έχω: $\forall x/X, \forall y/Y, \forall z/Z (B(x) \wedge (A(z) \vee (A(z) \wedge B(z)) \vee (A(z) \wedge C(y))) \vee C_0)$.

Κάνοντας τις πράξεις, έχουμε την τελική μορφή:

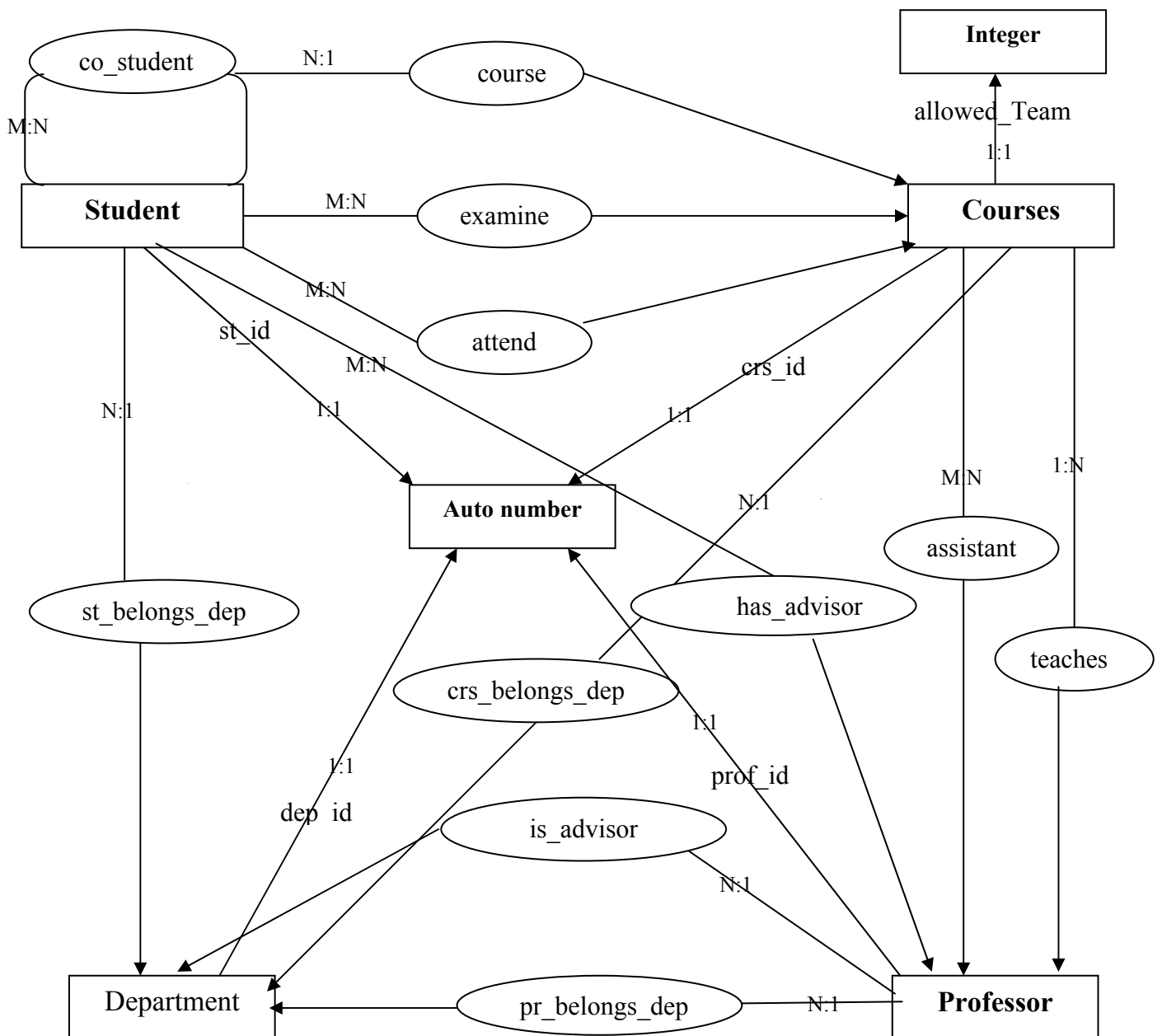
$$\forall x/X, \forall y/Y, \forall z/Z ((B(x) \wedge A(z)) \vee (B(x) \wedge (A(z) \wedge B(z))) \vee (B(x) \wedge (A(z) \wedge C(y))) \vee (B(x) \wedge C_0)) \Leftrightarrow$$

$$\forall x/X, \forall y/Y, \forall z/Z ((B(x) \wedge A(z)) \vee (B(x) \wedge A(z) \wedge B(z)) \vee (B(x) \wedge A(z) \wedge C(y)) \vee (B(x) \wedge C_0)), \text{ το οποίο είναι σε ΔΚΜ.}$$

Στον αλγόριθμο μας, φροντίσαμε ώστε να μετατρέπουμε κάθε έκφραση σε ΔΚΜ, χωρίς την ύπαρξη υπαρξιακών ποσοδεικτών, ώστε να είναι ευκολότερη η αποτίμησή τους από το σύστημα. Επίσης, στην περίπτωση της έκφρασης $\forall x/X (B(x) \wedge (\forall x/X A(x)))$, υπάρχει ο κίνδυνος να μπερδευτεί το σύστημα με την ύπαρξη της ίδιας μεταβλητής x , σε δύο διαφορετικές περιπτώσεις, απλά έχοντας το ίδιο όνομα και να τις θεωρήσει ως μία, παράγοντας την έκφραση $\forall x/X (B(x) \wedge A(x))$, χάνοντας, έτσι, την δεύτερη μεταβλητή. Για να αποφύγουμε αυτή την δυσάρεστη εξέλιξη, όταν το σύστημα αναγνωρίσει το δεύτερο x , που εισάγεται από το κατηγορημα A , θα το μετονομάσει σε x_1 . Έτσι, η έκφραση θα γίνει $\forall x/X, \forall x_0/X (B(x) \wedge A(x_0))$.

Παρουσίαση αποτελεσμάτων – Παράδειγμα

Στο κεφάλαιο αυτό, θα παρουσιάσουμε ένα παράδειγμα εκτέλεσης του εργαλείου μας. Αν υποθέσουμε ότι για έχουμε μία Σχεσιακή Βάση Δεδομένων για την αναπαράσταση της λειτουργίας του Μεταπτυχιακού Τμήματος Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης. Το σχήμα της Βάσης φαίνεται στο σχήμα που ακολουθεί.



Αναλυτικότερα, στο σχήμα φαίνονται τα εξής:

Οντότητες(Entities):

- **Student**(st_id). Είναι η οντότητα που αναπαριστά τους φοιτητές του τμήματος. Για την εφαρμογή μας, μόνο το πρωτεύον κλειδί st_id μας είναι απαραίτητο.
- **Professor**(prof_id, is_adv). Ομοίως, η οντότητα αυτή αναπαριστά τους καθηγητές. Το πρωτεύον κλειδί είναι το prof_id, ενώ το γνώρισμα is_adv είναι τύπου Boolean και δηλώνει εάν ο εν λόγω καθηγητής έχει αναλάβει τον ρόλο του ακαδημαϊκού συμβούλου σε κάποιους φοιτητές ή όχι. Μας είναι χρήσιμο για την περίπτωση που θέλουμε να ορίσουμε ακαδημαϊκό σύμβουλο σε ένα φοιτητή.
- **Courses**(crs_id). Παρόμοια για την οντότητα μαθήματα, το πρωτεύον κλειδί είναι το crs_id.
- **Department**(dep_id). Τέλος, παρόμοια με τις προηγούμενες, η οντότητα μαθήματα έχει πρωτεύον κλειδί το dep_id.

Ακολουθεί μία λίστα με όλες τις χρήσιμες συσχετίσεις, που στο πρόγραμμά μας τις χρησιμοποιούμε για την παραγωγή των τιμών των συναρτήσεων, αλλά και των κατηγορημάτων.

Συσχετίσεις(relationships- predicates)

- *examine*(st, crs). Δηλώνει ότι ο φοιτητής st έχει εγγραφεί και θα εξεταστεί στο μάθημα crs.
- *attend*(st, crs). Δηλώνει ότι ο φοιτητής st έχει εγγραφεί, αλλά δεν θα εξεταστεί στο μάθημα crs. Απλά θα το παρακολουθεί χωρίς καμία υποχρέωση.
- *pr_belongs_dep*(prof, dep). Δηλώνει ότι ο καθηγητής prof είναι επίσημο μέλος του τμήματος dep.

- $st_belongs_dep(st, dep)$. Δηλώνει ότι ο φοιτητής st είναι επίσημα μέλος του τμήματος dep . Έχει, δηλαδή, επίσημα εγγραφεί στο τμήμα.
- $crs_belongs_dep(crs, dep, limit)$. Δηλώνει ότι το μάθημα crs είναι επίσημα μάθημα του τμήματος dep , ενώ ο μέγιστος επιτρεπόμενος αριθμός φοιτητών που το παρακολουθούν είναι ο $limit$.
- $teaches(prof, crs)$. Δηλώνει ότι ο καθηγητής $prof$ είναι επίσημος διδάσκοντας του μαθήματος crs .
- $assistant(prof, crs)$. Δηλώνει ότι ο καθηγητής $prof$ είναι βοηθός του επίσημου διδάσκοντα του μαθήματος crs .
- $co_student(st1, st2, crs)$. Δηλώνει ότι ο φοιτητής $st1$ είναι στη ίδια ομάδα εργασίας, σε όσα μαθήματα επιτρέπονται συνεργασίες, με τον φοιτητή $st2$, στο μάθημα crs .
- $allowed_Team(crs)$. Είναι τύπου Boolean και δηλώνει εάν επιτρέπονται ή όχι συνεργασίες στο μάθημα crs .
- $postgraduate(st, dep)$. Είναι τύπου Boolean και δηλώνει εάν ο φοιτητής st είναι μέλος του μεταπτυχιακού προγράμματος ή αν απλά είναι προπτυχιακός φοιτητής. Το γνώρισμα αυτό μας είναι χρήσιμο για την επέκταση του συστήματος σε εφαρμογές του αντίστοιχου Προπτυχιακού Τμήματος.
- $has_advisor(st, prof, dep)$. Δηλώνει ότι ο μεταπτυχιακός φοιτητής st έχει ακαδημαϊκό σύμβουλο τον καθηγητή $prof$, ο οποίος είναι επίσημος διδάσκοντας του τμήματος dep .
- $is_advisor(prof, dep)$. Δηλώνει ότι ο καθηγητής $prof$, ο οποίος είναι επίσημος διδάσκοντας του τμήματος dep , είναι ακαδημαϊκός σύμβουλος για κάποιους φοιτητές.

Συναρτήσεις(functions)

- $num_prof(dep)$. Δηλώνει τον αριθμό των διδασκόντων στο τμήμα dep.
- $num_st(dep)$. Δηλώνει τον αριθμό των εγγεγραμμένων φοιτητών στο τμήμα dep.
- $size_crs(crs)$. Δηλώνει τον αριθμό των εγγεγραμμένων φοιτητών στο μάθημα crs.
- $team_size(crs)$. Δηλώνει τον μέγιστο επιτρεπόμενο αριθμό φοιτητών που αποτελούν μία ομάδα εργασίας στο μάθημα crs, εάν επιτρέπεται συνεργασία. Σε περίπτωση που δεν επιτρέπεται, ο αριθμός αυτός είναι 0.

Στο σημείο αυτό θα προσπαθήσουμε να περιγράψουμε τις διάφορες λειτουργίες που επιτρέπονται σε ένα τέτοιο σύστημα. Θα περιγράψουμε, δηλαδή, τις διάφορες δοσοληψίες, καθώς και τους περιορισμούς ακεραιότητας μαζί με τις αναλλοίωτες συνθήκες που το σύστημά μας καλείται να διατηρήσει. Αναλυτικότερα έχουμε τις δοσοληψίες:

EnrolledInCourse(st, crs)

Precondition: not **examine**(st, crs)

Postcondition: @**examine**(st, crs) and (@**size_crs**(crs) = **size_crs**(crs) + 1)

Invariant: **size_crs**(crs) > 0 and exist limit/Integer(
size_crs(crs) < limit and
(examine(st, crs) =>(exist dep/Department exist prof/ Professor(
st_belongs_dep(st, dep) and
crs_belongs_dep(crs, dep, limit) and
pr_belongs_dep(prof, dep) and
teaches(prof, crs))))

Η δοσοληψία αυτή δηλώνει την διαδικασία εγγραφής ενός φοιτητή st, για εξέταση σε ένα μάθημα crs. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι

θέτουμε σαν προϋπόθεση, ο φοιτητής να μην είναι ήδη εγγεγραμμένος στο ίδιο μάθημα. Επίσης, μόλις τελειώσει η εγγραφή του, στο μάθημα, θα πρέπει να αυξήσουμε τον μετρητή των εγγεγραμμένων φοιτητών κατά ένα και να τον αποθηκεύσουμε στον πίνακα **examine**. Η αναλλοίωτη συνθήκη είναι να υπάρχει διδάσκοντας για το μάθημα αυτό και να ανήκει στο ίδιο τμήμα με τον φοιτητή και το μάθημα. Επίσης θα πρέπει ο αριθμός των εγγεγραμμένων φοιτητών να μην υπερβαίνει τον μέγιστο επιτρεπόμενο του μαθήματος.

DropCourse(st, crs)

Precondition: **examine**(st, crs)

Postcondition: not **@examine**(st, crs) and (**@size_crs**(crs) = **size_crs**(crs) - 1)

Invariant: **size_crs**(crs) > 0 and **examine**(st, crs) =>
 (exist dep/Department exist limit/Integer exist prof/ Professor(
 st_belongs_dep(st, dep) and
 crs_belongs_dep(crs, dep, limit) and
 pr_belongs_dep(prof, dep) and **teaches**(prof, crs)))

Η δοσοληψία αυτή δηλώνει την διαδικασία διαγραφής ενός φοιτητή st, από την εξέταση ενός μαθήματος crs. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, ότι ο φοιτητής θα πρέπει να είναι ήδη εγγεγραμμένος στο μάθημα. Επίσης, μόλις τελειώσει η διαγραφή του, από το μάθημα, θα πρέπει να μειώσουμε τον μετρητή των εγγεγραμμένων φοιτητών κατά ένα και να τον διαγράψουμε από τον πίνακα **examine**. Η αναλλοίωτης συνθήκη είναι να υπάρχει διδάσκοντας για το μάθημα αυτό και να ανήκει στο ίδιο τμήμα με τον φοιτητή και το μάθημα. Επίσης θα πρέπει ο αριθμός των εγγεγραμμένων φοιτητών να μην είναι μικρότερος του ενός, γιατί τότε το μάθημα ακυρώνεται.

AttendCourse(st, crs)

Precondition: not **attend**(st, crs)

Postcondition: **@attend**(st, crs)

Invariant: (exist dep/Department, exist limit exist prof/ Professor(
 st_belongs_dep(st, dep) and
 crs_belongs_dep(crs, dep, limit) and

pr_belongs_dep(prof, dep) and
teaches(prof, crs)))

Η δοσοληψία αυτή δηλώνει την διαδικασία εγγραφής ενός φοιτητή st, για απλή παρακολούθηση σε ένα μάθημα crs. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, να μην είναι ήδη εγγεγραμμένος στο ίδιο μάθημα. Επίσης, μόλις τελειώσει η εγγραφή του, στο μάθημα, θα πρέπει να τον αποθηκεύσουμε στον πίνακα **attend**. Η αναλλοίωτη συνθήκη είναι να υπάρχει διδάσκοντας για το μάθημα αυτό και να ανήκει στο ίδιο τμήμα με τον φοιτητή και το μάθημα. Θα πρέπει να τονίσουμε ότι δεν υπάρχει συνθήκη για τον μέγιστο αριθμό εγγεγραμμένων φοιτητών στο μάθημα.

DropAttend(st, crs)

Precondition: **attend** (st, crs)

Postcondition: not **@attend** (st, crs)

Invariant: (exist dep/Department, exist limit exist prof/ Professor (
st_belongs_dep(st, dep) and
crs_belongs_dep(crs, dep, limit) and
pr_belongs_dep(prof, dep) and
teaches (prof, crs))))

Η δοσοληψία αυτή δηλώνει την διαδικασία διαγραφής ενός φοιτητή st, από την παρακολούθηση ενός μαθήματος crs. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, να είναι ήδη εγγεγραμμένος για παρακολούθηση, στο μάθημα. Επίσης, μόλις τελειώσει η διαγραφή του, από το μάθημα, θα πρέπει να τον διαγράψουμε από τον πίνακα **attend**. Η αναλλοίωτης συνθήκη είναι να υπάρχει διδάσκοντας για το μάθημα αυτό και να ανήκει στο ίδιο τμήμα με τον φοιτητή και το μάθημα. Επίσης θα πρέπει ο αριθμός των εγγεγραμμένων φοιτητών να μην είναι αρνητικός.

InsertCrsDep(crs, dep, limit)

Precondition: not exist limit2/Integer not **crs_belongs_dep**(crs, dep, limit2)

Postcondition: **@crs_belongs_dep**(crs, dep, limit) and

$(@num_crs(dep) = num_crs(dep) + 1)$

Invariant: $num_crs(dep) > 0$ and
 (not exist dep_1, dep_2 / Department not exist limit / Integer(
 $crs_belongs_dep(crs, dep_1, limit)$ and
 $crs_belongs_dep(crs, dep_2, limit)$
 and $(dep_1 \neq dep_2)$))

Η δοσοληψία αυτή δηλώνει την διαδικασία εισαγωγής ενός μαθήματος crs , σε ένα τμήμα dep , με ανώτερο επιτρεπόμενο αριθμό εγγεγραμμένων φοιτητών ίσο με $limit$. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, το μάθημα να μην είναι ήδη εγγεγραμμένο ούτε στο ίδιο τμήμα, αλλά ούτε και σε διαφορετικό. Δηλαδή θα πρέπει ο ορισμός ενός μαθήματος να είναι μοναδικός. Επίσης, μόλις τελειώσει η εγγραφή του μαθήματος στο τμήμα, θα πρέπει να αυξήσουμε τον μετρητή των μαθημάτων κατά ένα και να το αποθηκεύσουμε στον πίνακα **$crs_belongs_dep$** . Η αναλλοίωτης συνθήκη είναι να μην υπάρχει δήλωση του ίδιου μαθήματος σε άλλο πανεπιστημιακό τμήμα.

DropCrsDep(crs, dep)

Precondition: exist limit / Integer $crs_belongs_dep(crs, dep, limit)$

Postcondition: not exist limit / Integer not

$@crs_belongs_dep(crs, dep, limit)$ and
 $(@num_crs(dep) = num_crs(dep) - 1)$

Invariant: $num_crs(dep) > 0$ and
 (not exist dep_1, dep_2 / Department not exist limit / Integer (
 $crs_belongs_dep(crs, dep_1, limit)$ and
 $crs_belongs_dep(crs, dep_2, limit)$ and
 $(dep_1 \neq dep_2)$))

Η δοσοληψία αυτή δηλώνει την διαδικασία διαγραφής ενός μαθήματος crs , από ένα τμήμα dep . Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, το μάθημα να είναι ήδη εγγεγραμμένο μόνο στο τμήμα αυτό. Δηλαδή θα πρέπει ο ορισμός ενός μαθήματος να είναι μοναδικός. Επίσης, μόλις τελειώσει η εγγραφή του, στο τμήμα, θα πρέπει να μειώσουμε τον μετρητή των

μαθημάτων κατά ένα και να το διαγράψουμε από τον πίνακα **crs_belongs_dep**. Η αναλλοίωτη συνθήκη είναι να μην υπάρχει δήλωση του ίδιου μαθήματος σε περισσότερα του ενός πανεπιστημιακά τμήματα.

InsertProfDep(prof, dep)

Precondition: not **pr_belongs_dep**(prof, dep)

Postcondition: **@pr_belongs_dep**(prof, dep) and
(**@num_prof**(dep) = **num_prof**(dep) + 1)

Invariant: **num_prof**(dep) > 0 and
(not exist dep₁, dep₂/Department(
 pr_belongs_dep(prof, dep₁) and
 pr_belongs_dep(prof, dep₂) and
 (dep₁ != dep₂)))

Η δοσοληψία αυτή δηλώνει την διαδικασία εισαγωγής ενός καθηγητή prof, σε ένα τμήμα dep. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, ο καθηγητής να μην είναι ήδη εγγεγραμμένος ούτε στο ίδιο τμήμα, αλλά ούτε και σε διαφορετικό. Δηλαδή θα πρέπει ο ορισμός ενός καθηγητή να είναι μοναδικός. Επίσης, μόλις τελειώσει η εγγραφή του, στο τμήμα, θα πρέπει να τον αποθηκεύσουμε στον πίνακα **prof_belongs_dep**. Η αναλλοίωτη συνθήκη είναι να μην υπάρχει δήλωση του ίδιου καθηγητή σε άλλο πανεπιστημιακό τμήμα.

DropProfDep(prof, dep)

Precondition: **pr_belongs_dep**(prof, dep)

Postcondition: not **@pr_belongs_dep**(prof, dep) and
(**@num_prof**(dep) = **num_prof**(dep) - 1)

Invariant: **num_prof**(dep) > 0 and
(not exist dep₁, dep₂/Department(
 pr_belongs_dep(prof, dep₁) and
 pr_belongs_dep(prof, dep₂) and
 (dep₁ != dep₂)))

Η δοσοληψία αυτή δηλώνει την διαδικασία διαγραφής ενός καθηγητή prof, από ένα τμήμα dep. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, ο καθηγητής να είναι ήδη εγγεγραμμένος μόνο στο τμήμα αυτό. Δηλαδή θα πρέπει ο ορισμός ενός καθηγητή να είναι μοναδικός. Επίσης, μόλις τελειώσει η διαγραφή του, από το τμήμα, θα πρέπει να τον διαγράψουμε από τον πίνακα **prof_belongs_dep**. Η αναλλοίωτης συνθήκη είναι να μην υπάρχει δήλωση του ίδιου καθηγητή σε περισσότερα του ενός πανεπιστημιακά τμήματα.

InsertStudDep(st, dep)

Precondition: not **st_belongs_dep**(st, dep)

Postcondition: **@st_belongs_dep**(st, dep) and (**@num_st**(dep) = **num_st**(dep) + 1)

Invariant: **num_prof**(dep) >0 and
 (not exist dep₁, dep₂/Department(
 st_belongs_dep(st, dep₁) and
 st_belongs_dep(st, dep₂) and
 (dep₁ != dep₂)))

Η δοσοληψία αυτή δηλώνει την διαδικασία εισαγωγής ενός φοιτητή st, σε ένα τμήμα dep. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, ο φοιτητής να μην είναι ήδη εγγεγραμμένος ούτε στο ίδιο τμήμα, αλλά ούτε και σε διαφορετικό. Δηλαδή θα πρέπει ο ορισμός ενός φοιτητή να είναι μοναδικός. Επίσης, μόλις τελειώσει η εγγραφή του, στο τμήμα, θα πρέπει να τον αποθηκεύσουμε στον πίνακα **st_belongs_dep**. Η αναλλοίωτης συνθήκη είναι να μην υπάρχει δήλωση του ίδιου φοιτητή σε άλλο πανεπιστημιακό τμήμα.

DropStudDep(st, dep)

Precondition: **st_belongs_dep**(st, dep)

Postcondition: not **@st_belongs_dep**(st, dep) and (**@num_st**(dep) = **num_st**(dep)- 1)

Invariant: **num_prof**(dep) >0 and
 (not exist dep₁, dep₂/Department(
 st_belongs_dep(st, dep₁) and
 st_belongs_dep(st, dep₂) and
 (dep₁ != dep₂)))

Η δοσοληψία αυτή δηλώνει την διαδικασία διαγραφής ενός φοιτητή *st*, από ένα τμήμα *dep*. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, ο φοιτητή να είναι ήδη εγγεγραμμένος μόνο στο τμήμα αυτό. Δηλαδή θα πρέπει ο ορισμός ενός φοιτητή να είναι μοναδικός. Επίσης, μόλις τελειώσει η διαγραφή του, από το τμήμα, θα πρέπει να τον διαγράψουμε από τον πίνακα **st_belongs_dep**. Η αναλλοίωτης συνθήκη είναι να μην υπάρχει δήλωση του ίδιου φοιτητή σε περισσότερα του ενός πανεπιστημιακά τμήματα.

InsertTeacher(prof, crs)

Precondition: not **teaches**(prof, crs) and
 exist dep/Department(
 pr_belongs_dep(prof, dep) and
 crs_belongs_dep(crs, dep))

Postcondition: @**teaches**(prof, crs)

Invariant: not exist prof₁, prof₂/Professor(
 teaches(prof₁, crs) and
 teaches(prof₂, crs) and
 (prof₁ != prof₂) and
 (**num_st**(crs) > 0))

Η δοσοληψία αυτή δηλώνει την διαδικασία εισαγωγής ενός καθηγητή *prof*, να διδάξει ένα μάθημα *crs*. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, ο καθηγητής να μην είναι ήδη διδάσκοντας στο ίδιο μάθημα. Δηλαδή θα πρέπει ο ορισμός του διδάσκοντα καθηγητή σε ένα μάθημα να είναι μοναδικός. Επίσης, μόλις τελειώσει η εγγραφή του, στο μάθημα, θα πρέπει να τον αποθηκεύσουμε στον πίνακα **teaches**. Η αναλλοίωτης συνθήκη είναι να μην υπάρχει δεύτερος διδάσκοντας καθηγητή του ίδιου μαθήματος.

DropTeacher(prof, crs)

Precondition: **teaches** (prof, crs) and (not exist prof₂/Professor **teaches** (prof₂, crs))

Postcondition: not @**teaches**(prof, crs)

Invariant: not exist prof₁, prof₂/Professor(
 teaches(prof₁, crs) and

teaches(prof₂, crs) and
 (prof₁ != prof₂) and
 (**num_st**(crs) > 0))

Η δοσοληψία αυτή δηλώνει την διαδικασία διαγραφής ενός καθηγητή prof, από την διδασκαλία ενός μαθήματος crs. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, ο καθηγητής να είναι ήδη διδάσκοντας στο μάθημα αυτό. Δηλαδή θα πρέπει ο ορισμός ενός διδάσκοντα σε ένα μάθημα να είναι μοναδικός. Επίσης, μόλις τελειώσει η διαγραφή του καθηγητή από το μάθημα, θα πρέπει να τον διαγράψουμε από τον πίνακα **teaches**. Η αναλλοίωτης συνθήκη είναι να υπάρχει μόνο ένας διδάσκοντας καθηγητής σε κάθε μάθημα.

ProfIsAssist(prof, crs)

Precondition: not **assistant**(prof, crs) and not **teaches**(prof, crs) and
 exist dep/Department exist prof₁/Professor (
 pr_belongs_dep(prof, dep) and
 crs_belongs_dep(crs, dep) and
 teaches(prof₁, crs) and
 (prof₁ != prof))

Postcondition: **@assistant**(prof, crs)

Invariant: not exist prof₁, prof₂/Professor(
 teaches(prof₁, crs) and
 teaches(prof₂, crs) and
 (prof₁ != prof₂) and
 (**num_st**(crs) > 0))

Η δοσοληψία αυτή δηλώνει την διαδικασία εισαγωγής ενός καθηγητή prof, για βοηθού διδασκαλίας σε ένα μάθημα crs. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, να υπάρχει διδάσκον καθηγητής και ο καθηγητής prof να μην είναι ήδη βοηθός στο ίδιο μάθημα. Επίσης θα πρέπει ο διδάσκοντας καθηγητής σε ένα μάθημα και ο βοηθός του να ανήκουν στο ίδιο τμήμα. Επίσης, μόλις τελειώσει η εγγραφή του, σαν βοηθός στο μάθημα, θα πρέπει να τον

αποθηκεύσουμε στον πίνακα **assistant**. Η αναλλοίωτης συνθήκη είναι να μην υπάρχει δεύτερος διδάσκοντας καθηγητή του ίδιου μαθήματος.

DropAssist(prof, crs)

Precondition: **assistant**(prof, crs) and (exist prof₁ /Professor **teaches**(prof₁, crs))

Postcondition: not **@assistant**(prof, crs)

Invariant: (not exist prof₁, prof₂/Professor(
teaches(prof₁, crs) and
teaches(prof₂, crs) and
(prof₁ != prof₂) and
(**num_st**(crs) > 0))

Η δοσοληψία αυτή δηλώνει την διαδικασία διαγραφής ενός καθηγητή prof, από βοηθός στην διδασκαλία ενός μαθήματος crs. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, ο καθηγητής να είναι ήδη βοηθός του διδάσκοντας στο μάθημα αυτό. Επίσης, μόλις τελειώσει η διαγραφή του βοηθού καθηγητή από το μάθημα, θα πρέπει να τον διαγράψουμε από τον πίνακα **assistant**. Η αναλλοίωτης συνθήκη είναι να μην υπάρχει δεύτερος διδάσκοντας καθηγητή του ίδιου μαθήματος.

MakeTeam(st₁, st₂, crs)

Precondition: **examine**(st₁, crs) and **examine**(st₂, crs) and not **co_student**(st₁, st₂) and
(allowed_Team(crs) > 0)

Postcondition: **@co_student**(st₁, st₂) and (**@team_size**(st₁) = **team_size**(st₁) + 1) and
(@team_size(st₂) = **team_size**(st₂) + 1)

Invariant (for all st₁, st₂/Student, exist crs/Courses,
exist limit/Integer exist dep/Department(
(co_student(st₁, st₂, crs) =>
(crs_belongs_dep(crs, dep, limit) and
st_belongs_dep(st₁, dep) and
st_belongs_dep(st₂, dep) and
(team_size(st₁) = **team_size**(st₂)) and

(team_size(st₁) < allowed_Team(crs)))

Η δοσοληψία αυτή δηλώνει την διαδικασία δημιουργίας μία ομάδας εργασίας μεταξύ δύο φοιτητών, του φοιτητή st₁ με τον φοιτητή st₂. Η δημιουργία της ομάδας, φυσικά, επιτρέπεται μόνο εφόσον το μάθημα το επιτρέπει(εάν δηλαδή **allowed_Team(crs)> 0**, που σημαίνει ότι οι επιτρεπόμενες ομάδες έχουν μέγιστο αριθμό μελών **allowed_Team**) και εφόσον δεν ανήκουν ήδη στην ίδια ομάδα. Επίσης, μόλις τελειώσει η εγγραφή των φοιτητών στην ομάδα, θα πρέπει να αυξήσουμε τον αριθμό των μαλών της ομάδας κάθε φοιτητή για το συγκεκριμένο μάθημα κατά μία μονάδα, ενώ θα πρέπει και να τους αποθηκεύσουμε στον πίνακα **co_student**. Η αναλλοίωτη συνθήκη είναι οι δύο φοιτητές να είναι εγγεγραμμένοι για εξέταση στο εν λόγω μάθημα και να ανήκουν και οι δύο στο τμήμα που ανήκει το μάθημα αυτό.

BecomeAdvisor(prof, st, dep)

Precondition: **postgraduate**(st, dep) and
not **has_advisor**(st, prof, dep)

Postcondition: **@has_advisor**(st, prof, dep) and **@is_advisor**(prof, dep)

Invariant **@is_advisor**(prof, dep) or
is_advisor(prof, dep)

Η δοσοληψία αυτή δηλώνει την διαδικασία εισαγωγής ενός καθηγητή prof, ως ακαδημαϊκού συμβούλου ενός φοιτητή st. Στον ορισμό της δοσοληψίας αυτής φαίνεται καθαρά ότι θέτουμε σαν προϋπόθεση, ο καθηγητής να μην είναι ήδη ακαδημαϊκός σύμβουλος του φοιτητή st. Επίσης, μόλις τελειώσει η εγγραφή του καθηγητή σαν ακαδημαϊκού συμβούλου, θα πρέπει να τον αποθηκεύσουμε στον πίνακα **has_advisor**. Η αναλλοίωτη συνθήκη είναι, κάθε καθηγητής ενός τμήματος, είτε να μην είναι ακαδημαϊκός σύμβουλος είτε μετά την δοσοληψία **BecomeAdvisor** να γίνει.

Με αυτά τα δεδομένα λοιπόν, η έξοδος του συστήματός μας είναι το εξής σύνολο αξιωμάτων αποτελεσμάτων:

- for all prof/Professor for all st/Student for all dep/Department

not **@has_advisor**(st, prof) or
has_advisor(st, prof) or
BecomeAdvisor(prof, st, dep)

Το αξίωμα αυτό δηλώνει ότι αν η αρχική τιμή του **has_advisor**(st, prof) είναι ψευδής, μόνο μετά την εκτέλεση της δοσοληψίας **BecomeAdvisor**(prof, st, dep), θα μπορέσει το κατηγορήμα αυτό να αλλάξει τιμή και να γίνει αληθές.

- for all prof/Professor for all st/Student for all dep/Department
not **@is_advisor**(prof, dep) or
is_advisor(prof, dep) or
BecomeAdvisor(prof, st, dep)

Το αξίωμα αυτό δηλώνει ότι αν η αρχική τιμή του **is_advisor**(prof, dep) είναι ψευδής, μόνο αν εκτελεστεί η δοσοληψία **BecomeAdvisor**(prof, st, dep), θα μπορέσει το κατηγορήμα αυτό να αλλάξει τιμή και να γίνει αληθές.

- for all prof/Professor for all dep/Department
not **is_advisor**(prof, dep) or
@is_advisor(prof, dep) or
DropProfDep(prof, dep)

Το αξίωμα αυτό δηλώνει ότι αν η αρχική τιμή του **is_advisor**(prof, dep) είναι αληθής, μόνο αν εκτελεστεί η δοσοληψία **DropProfDep**(prof, dep), θα μπορέσει το κατηγορήμα αυτό να αλλάξει τιμή και να γίνει ψευδές.

- for all prof/Professor for all dep/Department
not **@prof_bel_dep**(prof, dep) or
prof_bel_dep(prof, dep) or
InsertProfDep(prof, dep)

Το αξίωμα αυτό δηλώνει ότι αν η αρχική τιμή του **prof_bel_dep**(prof, dep) είναι ψευδής, μόνο μετά την εκτέλεση της δοσοληψίας **InsertProfDep**(prof, dep), θα μπορέσει το κατηγορήμα αυτό να αλλάξει τιμή και να γίνει αληθές.

- for all prof/Professor for all dep/Department
not **prof_bel_dep**(prof, dep) or
@prof_bel_dep(prof, dep) or
DropProfDep(prof, dep)

Το αξίωμα αυτό δηλώνει ότι αν η αρχική τιμή του **prof_bel_dep** (prof, dep) είναι αληθής, μόνο αν εκτελεστεί η δοσοληψία **InsertProfDep**(prof, dep), θα μπορέσει το κατηγορήμα αυτό να αλλάξει τιμή και να γίνει ψευδές.

- for all prof/Professor for all dep/Department
@num_prof(dep) = **num_prof**(dep) or
DropProfDep(prof, dep) or
InsertProfDep(prof, dep)

Το αξίωμα αυτό δηλώνει ότι για να αλλάξει αριθμητική τιμή το **num_prof**(prof, dep), θα πρέπει να εκτελεστεί είτε η δοσοληψία **InsertProfDep**(prof, dep) είτε η δοσοληψία **DropProfDep**(prof, dep).

Επίσης, παράγεται και το ακόλουθο αξίωμα επόμενης κατάστασης:

for all prof/Professor for all st/Student for all dep/Department
DropProfDep(prof, dep) and **is_advisor**(prof, dep) or
not **BecomeAdvisor**(prof, st, dep)

Το αξίωμα αυτό βασίζεται στη αναλλοίωτη συνθήκη για την δοσοληψία **BecomeAdvisor**(prof, st, dep) και δηλώνει ότι σε περίπτωση που εκτελεστεί η δοσοληψία αυτή για ένα καθηγητή, τότε αυτός αυτομάτως θα γίνει και ακαδημαϊκός σύμβουλος(**is_advisor**(prof, dep)).

Για να γίνει πλήρως κατανοητή η χρησιμότητα του εργαλείου μας, ας θεωρήσουμε ένα παράδειγμα περιορισμού που παραβιάζεται. Επιλέξαμε το ακόλουθο παράδειγμα επειδή πάνω σε αυτό είναι εύκολο να φανούν ορισμένες λεπτομέρειες

πάνω στην χρήση και την λειτουργία του εργαλείου μας. Ας υποθέσουμε την ακόλουθη περιγραφή της δοσοληψίας:

EnrolledInCourse(st, crs)

Precondition: $\neg \text{examine}(\text{st}, \text{crs})$

Postcondition: $@\text{examine}(\text{st}, \text{crs}) \wedge (@\text{size_crs}(\text{crs}) = \text{size_crs}(\text{crs}) + 1)$

Invariant: $\text{size_crs}(\text{crs}) \leq \text{limit}$

Η εργασία μας κατασκεύασε ένα μηχανισμό, ο οποίος προτείνει στον χρήστη ορισμένες νέες συνθήκες, τις οποίες θα μπορεί να ενσωματώσει, με την μορφή σύζευξης, στις μετασυνθήκες των δοσοληψιών, προκειμένου να υπάρχει η εξασφάλιση ότι, μετά την εκτέλεση των δοσοληψιών αυτών, οι αντίστοιχοι περιορισμοί δεν θα παραβιάζονται. Στο παράδειγμά μας, υποθέτουμε ότι η εκτέλεση της δοσοληψίας **EnrolledInCourse**(st, crs) πάνω στην Βάση, έχει σαν αποτέλεσμα την παραβίαση του περιορισμού $\text{size_crs}(\text{crs}) \leq \text{limit}$. Τότε, ο μηχανισμός που προαναφέραμε θα προτείνει στον χρήστη μία συνθήκη Cond_1 , που θα πρέπει να προσθέσει στην μετασυνθήκη post_T , για να μην παραβιάζεται ο περιορισμός, οπότε θα έχουμε ότι $\text{post}_T \equiv \text{post}_T \wedge \text{Cond}_1$.

Για την θεωρητική τεκμηρίωση του ισχυρισμού μας, υπάρχει ένα θεώρημα [Ple96], που αναφέρει ότι:

Θεώρημα 5.1.

Ας Υποθέσουμε δοσοληψία T, με προσυνθήκη P, μετασυνθήκη Q, και περιορισμό ακεραιότητας I. Εάν N είναι μία επίπτωση (βλ. §3.1), πάνω στο $(Q \wedge \neg @I)$, δηλαδή πάνω στη σύζευξη του Q με το συμπλήρωμα του περιορισμού I, που προκύπτει από την κατάσταση που έπεται της εκτέλεσης της T, τότε η $\neg N$ μπορεί να συζευχθεί με την μετασυνθήκη Q, εξασφαλίζοντας την διατήρηση του περιορισμού I και μετά την εκτέλεση της δοσοληψίας T, σαν αποτέλεσμα της διατήρησης των μετασυνθηκών.

Η απόδειξη σε αυτό το πολύ ενδιαφέρον και χρήσιμο θεώρημα είναι διατυπωμένη στην εργασία [Ple96]. Ας επανέλθουμε, όμως, στο παράδειγμά μας. Στο παράδειγμα αυτό έχουμε $T \equiv \text{EnrolledInCourse}(\text{st}, \text{crs})$, $P \equiv \neg \text{examine}(\text{st}, \text{crs})$, $Q \equiv$

$(@examine(st, crs) \wedge (@size_crs(crs) = size_crs(crs) + 1)), I \equiv (size_crs(crs) \leq limit).$

Σύμφωνα με τους τύπους αυτούς έχουμε

$Q \wedge \neg @I \equiv (@examine(st, crs) \wedge (@size_crs(crs) = size_crs(crs) + 1))$

$\wedge \neg @ (size_crs(crs) \leq limit) \Leftrightarrow$

$Q \wedge \neg @I \equiv (@examine(st, crs) \wedge (@size_crs(crs) = size_crs(crs) + 1)) \wedge \neg (@size_crs(crs) \leq limit). (I)$

Σύμφωνα με την μετασυνθήκη Q, στην κατάσταση του συστήματος, που έπεται της εκτέλεσης της δοσοληψίας **EnrolledInCourse**(st, crs), θα ισχύει ότι $(@size_crs(crs) = size_crs(crs) + 1)$, ενώ το **examine**(st, crs) θα είναι αληθές από τον ορισμό της μετασυνθήκης. Έτσι, αν αντικαταστήσουμε το **examine**(st, crs) και το $@size_crs(crs)$ με τις τιμές True και $(size_crs(crs) + 1)$, αντίστοιχα, θα πάρουμε την ζητούμενη επίπτωση N.

1. Δηλαδή, αντικαθιστώντας το **examine**(st, crs) με True και το $@size_crs(crs)$ με $size_crs(crs) + 1$, παίρνω το N, ως: $N \equiv (True \wedge (size_crs(crs) + 1 = size_crs(crs) + 1)) \wedge \neg (size_crs(crs) + 1 \leq limit) \Rightarrow$
2. Αντικαθιστώντας το $(size_crs(crs) + 1 = size_crs(crs) + 1)$ με True παίρνω: $N \equiv (True \wedge True) \wedge \neg (size_crs(crs) + 1 \leq limit) \Rightarrow$
3. $N \equiv \neg (size_crs(crs) + 1 \leq limit) \Leftrightarrow$
4. $\neg N \equiv size_crs(crs) + 1 \leq limit$

Μετά το Βήμα 4, γνωρίζω, πλέον, την συνθήκη $\neg N$. Προσθέτοντάς την, λοιπόν, στην μετασυνθήκη Q, η δοσοληψία **EnrolledInCourse**(st, crs), μετασχηματίζεται στην:

EnrolledInCourse(st, crs)

Precondition: $\neg examine(st, crs)$

Postcondition: $@examine(st, crs) \wedge$
 $(@size_crs(crs) = size_crs(crs) + 1) \wedge (size_crs(crs) + 1 \leq limit)$

Invariant: $size_crs(crs) \leq limit$

Τότε και μόνο τότε, θα υπάρχει πλέον η εγγύηση ότι ο περιορισμός ακεραιότητας ($\text{size_crs}(\text{crs}) \leq \text{limit}$) δεν θα παραβιαστεί από την εκτέλεση της δοσοληψίας **EnrolledInCourse**(st, crs).

Στο Παράρτημα A, ακολουθεί μία επίδειξη της πραγματικής εισόδου και εξόδου του συστήματος για το παράδειγμα που περιγράψαμε στην αρχή του κεφαλαίου.

Κεφάλαιο 6^ο

Επεκτάσεις, Βελτιώσεις

6.1 Επεκτάσεις του συστήματος

Όπως είδαμε και σε προηγούμενη ενότητα, το εργαλείο, που υλοποιήσαμε, είναι ένα τμήμα ενός ολοκληρωμένου συστήματος ανάπτυξης και σχεδίασης σε ένα σύστημα δοσοληψιών πάνω σε Βάσεις Δεδομένων. Έτσι, λοιπόν, θα μπορούσαμε, κατά πρώτον, να επεκτείνουμε το εργαλείο μας, πλαισιώνοντάς το με τα υπόλοιπα τρία τμήματα που περιγράψαμε σε προηγούμενη ενότητα (βλέπε §4.1). Θα πρέπει, δηλαδή, να υλοποιηθούν και τα εναπομείναντα κομμάτια του συστήματος, το οποίο κληθήκαμε να σχεδιάσουμε, αλλά στα πλαίσια της μεταπτυχιακής αυτής εργασίας δεν υπήρξε ο απαιτούμενος χρόνος για μία τέτοια υλοποίηση. Ωστόσο, η υλοποίηση των τμημάτων αυτών παρουσιάζει ορισμένες δυσκολίες, τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο και δεν θεωρείται ως μία τετριμμένη διαδικασία.

Επίσης, όπως αναφέραμε στην αρχή της εργασίας μας, το σύστημά μας είναι κατασκευασμένο, εξ αρχής, για την υποστήριξη μίας σχεσιακής Βάσης Δεδομένων. Όμως, όπως φάνηκε κατά τη διάρκεια της παρουσιάσής μας, η υλοποίηση του εργαλείου μας δεν λαμβάνει καθόλου υπόψη την συγκεκριμένη υλοποίηση της Βάσης, αλλά το μόνο που απαιτεί είναι η ύπαρξη μίας πηγής γνώσης, από όπου θα αντλεί τις πληροφορίες που χρειάζεται. Για τον λόγο αυτό, γίνεται κατανοητό ότι θα μπορούσαμε να εφαρμόσουμε το σύστημά μας σε οποιοδήποτε είδος Βάσης Δεδομένων. Έτσι, με την κατάλληλη διεπιφάνεια χρήσης θα μπορούσαμε να εξασφαλίσουμε συνεργασία μεταξύ διαφορετικών ειδών Βάσεων Δεδομένων.

Με την κατάλληλα διαμορφωμένη διεπιφάνεια χρήσης, ο χρήστης θα μπορούσε να επικοινωνήσει τόσο με μία αντικειμενοστραφής όσο και με μία σχεσιακή Βάση Δεδομένων. Την υλοποίηση αυτής της διεπιφάνειας την αναλαμβάνει το δεύτερο τμήμα που ονομάσαμε Μηχανισμό Ανακάλυψης και Δημιουργίας Δεδομένων Επεξεργασίας. Για αυτό τον λόγο, είναι πολύ σημαντική μία πολύ προσαρμόσιμη, όσο και επεκτάσιμη υλοποίηση του τμήματος της διεπιφάνειας, ώστε

ακόμα και αν το έχουμε υλοποιήσει για να εφαρμοστεί σε μία σχεσιακή Βάση, να μπορεί εύκολα να χρησιμοποιηθεί και για μία αντικειμενοστραφής Βάση Δεδομένων.

Επιπλέον, γίνεται ολοένα και περισσότερο επιτακτική η ανάγκη, τα τελευταία χρόνια, για ανάπτυξη Συστημάτων Διαχείρισης Χωρικών Βάσεων Δεδομένων, εξαιτίας της ραγδαίας εξάπλωσης των Γεωγραφικών Πληροφοριακών Συστημάτων (Geographical Information Systems- G.I.S.). Έτσι, λοιπόν γίνεται κατανοητή η ανάγκη υποστήριξης- από ένα σύστημα, όπως το δικό μας, για δοσοληψίες, πάνω σε χωρικά δεδομένα, εκτός από τις δοσοληψίες που περιγράψαμε πάνω στα ήδη υπάρχοντα χρονικά δεδομένα.

Παράλληλα, απαραίτητη θεωρείται μία τροποποίηση της διεπιφάνειας χρήσης του συστήματος, ώστε να παρέχεται ευκολότερη πρόσβαση στα δεδομένα από τον χρήστη σε ένα πιο φιλικό για αυτόν περιβάλλον. Η βελτίωση της διεπιφάνειας είναι απαραίτητη και επιτακτική ανάγκη, γιατί κάθε σύστημα που προσβλέπει στην καθιέρωσή του στην αγορά, πρέπει να είναι φιλικό προς τον εκάστοτε χρήστη, ανεξαρτήτως του μορφωτικού και κοινωνικού επιπέδου του χρήστη.

Τέλος, μία εξίσου σημαντική επέκταση του συστήματός μας, είναι η δημιουργία μηχανισμών για υποστήριξη και άλλων κλάσεων δοσοληψιών. Με την έννοια ότι το σύστημά μας θα πρέπει να υποστηρίζει και δοσοληψίες που αναφέρονται σε μη Σχεσιακές Βάσεις, με διαφορετική σύνταξη και ερμηνεία, όπως για παράδειγμα δοσοληψίες που αναφέρονται σε αντικειμενοστραφείς Βάσεις Δεδομένων.

6.2 Βελτιώσεις του συστήματος

Το εργαλείο που υλοποιήσαμε, επιδέχεται κάποιες προσθήκες για καλύτερη απόδοση και βελτιστοποίηση της λειτουργίας του. Ο κώδικας, ο οποίος είναι γραμμένος για το εργαλείο μας, επιδέχεται αρκετές προσθήκες για βελτιστοποίηση. Οι απαιτήσεις του εργαλείου σε μνήμη δεν είναι μεγάλες, αλλά μπορούμε να τις μειώσουμε ακόμα περισσότερο. Σε ορισμένα σημεία του κώδικα θα μπορούσαμε να εισάγουμε συστήματα με πολλούς επεξεργαστές, ώστε να μειωθεί το κόστος και ο χρόνος επεξεργασίας των ανεξάρτητων δεδομένων. Αν αναλογιστούμε ότι σε μεγάλο βαθμό ο κώδικας επιδέχεται παράλληλη επεξεργασία, το κέρδος για ένα σύστημα που

υποστηρίζει το εργαλείο μας θα είναι μεγάλο. Το κέρδος αυτό θα είναι ακόμα μεγαλύτερο αν σκεφτούμε ότι τα περισσότερα συστήματα, που χρειάζονται ένα εργαλείο επεξεργασίας δοσοληπιών, αποτελούνται από δεκάδες, αν όχι εκατοντάδες, χιλιάδες εγγραφές, οπότε και ο χρόνος που θα χρειάζονταν ένα τέτοιο σύστημα με ένα επεξεργαστή θα ήταν, ενδεχομένως, τάξεις μεγέθους μεγαλύτερος.

Επίσης, στην παρούσα υλοποίηση του εργαλείου μας, κληθήκαμε να υλοποιήσουμε τον αλγόριθμο της μετατροπής μίας έκφρασης σε Δ.Κ.Μ.(βλέπε §4.2). Η συγκεκριμένη υλοποίηση, όμως, δεν είναι η βέλτιστη, οπότε, μία ταχύτερη και πιο αποδοτική υλοποίησή της, θα βοηθούσε πολύ στην αύξηση της ταχύτητας απόκρισης του συστήματος. Ωστόσο, τέτοιοι αλγόριθμοι δεν είναι προφανείς και δεν μπορέσαμε να ανακαλύψουμε κανένα στο διαδίκτυο ή σε άλλες πηγές που απευθυνθήκαμε.

Τέλος, ο κώδικας του εργαλείου μας είναι γραμμένος στην γλώσσα προγραμματισμού C. Επειδή, όμως, στις μέρες μας οι αντικειμενοστραφείς γλώσσες προγραμματισμού τείνουν να καθιερωθούν εις βάρος των υπολοίπων, λόγω των πολλών και γνωστών πλεονεκτημάτων τους, θα ήταν καλό να μετατρέψουμε τον κώδικα του εργαλείου μας σε μία τέτοια γλώσσα, όπως, για παράδειγμα, στην γλώσσα προγραμματισμού C++.

Συμπεράσματα

Στην εργασία μας αυτή, βασικό μέλημα ήταν να παρουσιάσουμε μία τεχνική, που μπορεί να υιοθετηθεί στην ανάπτυξη εργαλείων για Συστήματα Διαχείρισης Βάσεων Δεδομένων, με κύριο σκοπό της απόδειξη των περιορισμών ακεραιότητας με αποδεκτή απόδοση. Επίσης ορίσαμε τις απαραίτητες έννοιες για την απόδειξη της ικανοποίησης των περιορισμών αυτών.

Πιο αναλυτικά, προσπαθήσαμε να υλοποιήσουμε μία μεθόδου διατήρησης των περιορισμών, χρησιμοποιώντας διαδικασίες τροποποίησης των δοσοληψιών. Αρχίσαμε με την διαπίστωση της χρησιμότητας, της επίλυσης των προβλημάτων του πλαισίου και των επιπτώσεων, πάνω σε ένα σύστημα Βάσεων Δεδομένων. Η διαδικασία επίλυσης των προαναφερθέντων προβλημάτων, μπορεί να μας παράσχει πολύτιμη βοήθεια, κατά την διάρκεια της φάσης της σχεδίασης των δοσοληψιών, για αναδρομική σχεδίαση. Η διατύπωση, της σημασίας των προβλημάτων αυτών, βρίσκεται σε μία εργασία που προαναφέραμε [BMR95].

Πιο συγκεκριμένα, παρουσιάσαμε μία προσαρμοσμένη έκδοση των ιδεών της Τεχνητής Νοημοσύνης στο πρόβλημα της διατήρησης των περιορισμών σε μία Βάση Γνώσεων. Για την συντακτική περιγραφή των δοσοληψιών, επινοήσαμε μία γλώσσα προσδιορισμού χρονικών δοσοληψιών, η οποία και παρουσιάστηκε στο κεφάλαιο 4.

Επίσης, επεκτείναμε, μία συστηματική λύση στο πρόβλημα του πλαισίου και των επιπτώσεων για να επιτύχουμε- κατά την διάρκεια της μετάφρασης, της εφαρμογής- την διαδικασία απόδειξης, της αλήθειας της πρότασης ότι η εκτέλεση μίας ακολουθίας δοσοληψιών ενημέρωσης δεν παραβιάζει τους περιορισμούς ακεραιότητας μίας Βάσης Δεδομένων. Αυτό το εγχείρημα γίνεται δυνατό από την συστηματική δημιουργία αξιωμάτων επόμενης κατάστασης, τα οποία, στην περίπτωση των τελικών προσδιορισμών των δοσοληψιών, χαρακτηρίζουν ολοκληρωτικά τις συνθήκες κάτω από τις οποίες τα κατηγορήματα ή οι συναρτήσεις του πεδίου εφαρμογής, μπορούν να αλλάξουν τιμή αληθείας, σαν αποτέλεσμα εκτέλεσης των δοσοληψιών. Για την δημιουργία αξιωμάτων αυτών, βελτιώσαμε και διορθώσαμε, όπου αυτό κρίθηκε απαραίτητο, ένα αλγόριθμο που παρουσιάστηκε στην εργασία [Ple96], καθώς, επίσης, δημιουργήσαμε μία νέα υλοποίηση του αλγορίθμου

μετατροπής μίας έκφρασης σε Διαζευκτική Κανονική Μορφή. Ο αλγόριθμος, που έχει καθιερωθεί διεθνώς και περιγράφει την διαδικασία αυτή, δεν ήταν κατάλληλος για την δική μας εφαρμογή, για διάφορους λόγους, τους οποίους και αναφέραμε λεπτομερώς στο κεφάλαιο 4.

Η τεχνική αυτή μπορεί να οδηγήσει στην δημιουργία ενός εργαλείου, το οποίο σαν μέρος ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων(Σ.Δ.Β.Δ.), θα είναι σε θέση να προτείνει προσθήκες για τον ορισμό των μετασυνθηκών, όπως ορίσαμε και σε προηγούμενη ενότητα, των δοσοληψιών των οποίων τα αποτελέσματα θα είναι η διατήρηση των αναλλοίωτων συνθηκών.

Το σύστημα, που υλοποιήσαμε, με ορισμένες επεκτάσεις και τροποποιήσεις όπως περιγράψαμε και στο προηγούμενο κεφάλαιο, μπορεί να παράσχει πολύτιμη βοήθεια στον σχεδιαστή των δοσοληψιών, με μηχανισμούς ανάδρασης, είτε με την μορφή μίας προτεινόμενης, από το σύστημα, ενίσχυσης των περιγραφών των δοσοληψιών, είτε με την μορφή εντοπισμού ασυνεπειών στην Βάση Δεδομένων. Επιπλέον, με την υλοποίηση της νέας αυτής μεθόδου διατήρησης των περιορισμών ακεραιότητας, σε μεγάλες Βάσεις Δεδομένων, το σύστημα επιτυγχάνει ικανοποιητική απόδοση, κατά την διάρκεια της εκτέλεσης.

Με άλλα λόγια, το εργαλείο που παρουσιάζουμε, θα βοηθήσει ένα σχεδιαστή δοσοληψιών στον καθορισμό προδιαγραφών σωστών δοσοληψιών. Λαμβάνοντας σαν είσοδο ένα σύνολο δοσοληψιών, σύμφωνα με ένα συγκεκριμένο τρόπο αναπαράστασης, το εργαλείο, που υλοποιήσαμε, θα παράγει σαν έξοδο ένα σύνολο νέων κανόνων και προσδιορισμών, οι οποίοι αποτελούν αναγκαίες συνθήκες για την εξασφάλιση της ορθότητας των περιορισμών. Οι κανόνες αυτοί, μετά από κατάλληλη επεξεργασία, μπορούν να αποτελέσουν και ικανές συνθήκες, για την εξασφάλιση της ορθότητας των περιορισμών, όπως προαναφέραμε.

Το εργαλείο μας, αυτό, θα βοηθήσει στην διαδικασία σχεδίασης μίας Βάσης Γνώσεων, παρέχοντας την απαραίτητη πληροφορία ανάδρασης στον σχεδιαστή των δοσοληψιών και αυτοματοποιώντας την διαδικασία επαλήθευσης της ασφάλειας τους. Ο έλεγχος για την ικανοποίηση των περιορισμών, μετά από την εκτέλεση κάθε δοσοληψίας, δεν είναι πλέον απαραίτητος, για τους περιορισμούς οι οποίοι παραμένουν προφανώς αληθείς. Παρόμοια, για αυτούς που προφανώς παραβιάζονται, το κόστος της ακύρωσης, ώστε να επανέλθει η Βάση στην προηγούμενη κατάσταση, αποφεύγεται. Σε κάθε άλλη περίπτωση, παράγεται μία απλούστερη συνθήκη, που είναι αρκετή για την εξασφάλιση ότι οι αντίστοιχοι περιορισμοί δεν παραβιάζονται.

Τέλος, η προτεινόμενη αυτή μέθοδος, είναι μία πολλά υποσχόμενη ερευνητική κατεύθυνση για την επίτευξη αποτελεσματικών μηχανισμών διατήρησης των περιορισμών, κατά την διάρκεια της εκτέλεσης μίας εφαρμογής ενός Σ.Δ.Β.Δ.

Παράρτημα Α

Η πραγματική είσοδος είναι η εξής:

EnrolledInCourse(st, crs)

Precondition: not **Examine**(st, crs)

Postcondition: @**Examine**(st, crs) and (@**size_crs**(crs) = **size_crs**(crs) +1)

Invariant: exist limit/Integer (
(**size_crs**(crs) >0) and (**size_crs**(crs) < limit) and
(**Examine**(st, crs) =>(exist dep/Department exist prof/Professor(
 St_belongs_dep(st, dep) and
 Crs_belong_dep(crs, dep, limit) and
 Pr_belongs_dep(prof, dep) and
 Teaches(prof, crs))))))

DropCourse(st, crs)

Precondition: **Examine**(st, crs)

Postcondition: not @**Examine**(st, crs) and (@**size_crs**(crs) = **size_crs**(crs) -1)

Invariant: exist limit/Integer(
(**size_crs**(crs) >0) and
(**Examine**(st, crs) =>(exist dep/Department exist prof/Professor(
 St_belongs_dep(st, dep) and
 Crs_belong_dep(crs, dep, limit) and
 Pr_belongs_dep(prof, dep) and
 Teaches(prof, crs))))))

AttendCourse(st, crs)

Precondition: not **Attend**(st, crs)

Postcondition: @**Attend**(st, crs)

Invariant: (exist limit/Integer exist dep/Department exist prof/Professor(
 St_belongs_dep(st, dep) and

Crs_belong_dep(crs, dep, limit) and
Pr_belongs_dep(prof, dep) and
Teaches(prof, crs))

DropAttend(st, crs)

Precondition: **Attend**(st, crs)

Postcondition: not **@Attend**(st, crs)

Invariant: (exist limit/Integer exist dep/Department exist prof/Professor(
St_belongs_dep(st, dep) and
Crs_belong_dep(crs, dep, limit) and
Pr_belongs_dep(prof, dep) and
Teaches(prof, crs)))

InsertCrsDep(crs, dep, limit)

Precondition: not exist limit2/Integer **Crs_belong_dep**(crs,dep,limit2)

Postcondition: **@Crs_belong_dep**(crs, dep, limit) and
(@num_crs(dep) = **num_crs**(dep) +1)

Invariant: **(@num_crs**(dep) > 0) and
(not exist dep1,dep2/Department not exist limit/Integer(
Crs_belong_dep(crs, dep, limit) and
Crs_belong_dep(crs,dep2,limit) and
(dep1 != dep2)))

DropCrsDep(crs, dep, limit)

Precondition: **Crs_belong_dep**(crs, dep, limit)

Postcondition: not **@Crs_belong_dep**(crs, dep, limit) and
(@num_crs(dep) = **num_crs**(dep) -1)

Invariant: **(@num_crs**(dep) > 0) and
(not exist dep1,dep2/Department not exist limit/Integer(
Crs_belong_dep(crs, dep, limit) and
Crs_belong_dep(crs,dep2,limit) and
(dep1 != dep2)))

InsertProfDep(prof, dep)

Precondition: not **Prof_belong_dep**(prof, dep)

Postcondition: **@Prof_belong_dep**(prof, dep) and
 (**@num_prof**(dep) = **num_prof**(dep) + 1)

Invariant: (**@num_prof**(dep) > 0) and
 (not exist dep1,dep2/Department(
 Prof_belong_dep(prof, dep) and
 Prof_belong_dep(prof,dep2) and
 (dep1 != dep2)))

DropProfDep(prof, dep)

Precondition: **Prof_belong_dep**(prof, dep)

Postcondition: not **@Prof_belong_dep**(prof, dep) and
 not **@Is_advisor**(prof, dep) and
 (**@num_prof**(dep) = **num_prof**(dep) - 1)

Invariant: (**@num_prof**(dep) > 0) and
 (not exist dep1,dep2/Department(
 Prof_belong_dep(prof, dep) and
 Prof_belong_dep(prof,dep2) and
 (dep1 != dep2)))

InsertStudDep(st, dep)

Precondition: not **St_belongs_dep**(st, dep)

Postcondition: **@St_belongs_dep**(st, dep) and
 (**@num_st**(dep) = **num_st**(dep) + 1)

Invariant: (**@num_st**(dep) > 0) and
 (not exist dep1,dep2/Department(
 St_belongs_dep(st, dep) and
 St_belongs_dep(st,dep2) and
 (dep1 != dep2)))

DropStudDep(st, dep)

Precondition: **St_belongs_dep**(st, dep)

Postcondition: not **@St_belongs_dep**(st, dep) and

$(@num_st(dep) = num_st(dep) - 1)$
Invariant: $(@num_st(dep) > 0)$ and
 (not exist dep1,dep2/Department(
 St_belongs_dep(st, dep) and
 St_belongs_dep(st,dep2) and
 (dep1 != dep2)))

InsertTeacher(prof, crs)

Precondition: not **Teaches**(prof, crs)

Postcondition: **@Teaches**(prof, crs)

Invariant: (not exist prof1,prof2/Professor(
 Teaches(prof1,crs) and **Teaches**(prof2,crs) and
 (**num_st**(crs) >0) and
 (prof1 != prof2)))

DropTeacher(prof, crs)

Precondition: **Teaches**(prof, crs)

Postcondition: not **@Teaches**(prof, crs)

Invariant: (not exist prof1,prof2/Professor(
 Teaches(prof1,crs) and **Teaches**(prof2,crs) and
 (**num_st**(crs) >0) and
 (prof1 != prof2)))

ProfAssist(prof, crs)

Precondition: not **Assistant**(prof, crs) and
 exist prof1/Professor **Teaches**(prof1,crs)

Postcondition: **@Assistant**(prof, crs)

Invariant: True

DropAssist(prof, crs)

Precondition: **Assistant**(prof, crs) and
 exist prof1/Professor **Teaches**(prof1,crs)

Postcondition: not **@Assistant**(prof, crs)

Invariant: True

MakeTeam(st1,st2,crs)

Precondition: **Examine**(st1,crs) and **Examine**(st2,crs) and
not **Co_student**(st1,st2,crs) and
(**allowed_Team**(crs) > 0)

Postcondition: **@Co_student**(st1,st2,crs) and
(**@team_size**(st1,crs) = **team_size**(st1,crs) + 1) and
(**@team_size**(st2,crs) = **team_size**(st2,crs) + 1)

Invariant: (exist st1,st2/Student exist crs/Course exist dep/Department
exist limit/Integer(
Co_student(st1,st2,crs) =>(
Crs_belong_dep(crs, dep, limit) and
St_belongs_dep(st1,dep) and
St_belongs_dep(st2,dep) and
(**team_size**(st1,crs) == **team_size**(st2,crs))))))

BecomeAdvisor(prof, st, dep)

Precondition: **Postgraduate**(st, dep) and not **Has_advisor**(st, prof)

Postcondition: **@Has_advisor**(st, prof) and **@Is_advisor**(prof, dep)

Invariant: **Is_advisor**(prof, dep) or **@Is_advisor**(prof, dep)

Τότε, η έξοδος είναι η εξής:

Η T_ef έχει ως εξής:

Has_advisor

Positive axiom:

for all Pr_0/Professor for all St_0/Student for all Dep_0/Department for all t/Time
not **Has_advisor**(St_0,Pr_0,next(t)) or
Has_advisor(St_0,Pr_0,t) or
BecomeAdvisor(Pr_0,St_0,Dep_0,t)

Co_student

Positive axiom:

for all St_0/Student for all St_1/Student for all Crs_0/Course for all t/Time
 not **Co_student**(St_0,St_1,Crs_0,next(t)) or
Co_student(St_0,St_1,Crs_0,t) or
MakeTeam(St_0,St_1,Crs_0,t)
 team_size

Positive axiom:

for all St_1/Student for all St_0/Student for all Crs_0/Course for all t/Time
team_size(St_0,Crs_0,next(t)) = **team_size**(St_0,Crs_0,t) or
MakeTeam(St_0,St_1,Crs_0,t) or
MakeTeam(St_1,St_0,Crs_0,t)

Assistant

Teaches

St_belongs_dep

Positive axiom:

for all St_0/Student for all Dep_0/Department for all t/Time
 not **St_belongs_dep**(St_0,Dep_0,next(t)) or
St_belongs_dep(St_0,Dep_0,t) or
InsertStudDep(St_0,Dep_0,t)

Negative axiom:

for all St_0/Student for all Dep_0/Department for all t/Time
 not **St_belongs_dep**(St_0,Dep_0,t) or
St_belongs_dep(St_0,Dep_0,next(t)) or
DropStudDep(St_0,Dep_0,t)

num_st

Positive axiom:

for all St_0/Student for all Dep_0/Department for all t/Time
num_st(Dep_0,next(t)) = **num_st**(Dep_0,t) or
DropStudDep(St_0,Dep_0,t) or
InsertStudDep(St_0,Dep_0,t)

Is_advisor*Positive axiom:*

for all Pr_0/Professor for all St_0/Student for all Dep_0/Department for all t/Time
 not **Is_advisor**(Pr_0,Dep_0,next(t)) or
Is_advisor(Pr_0,Dep_0,t) or
BecomeAdvisor(Pr_0,St_0,Dep_0,t)

Negative axiom:

for all Pr_0/Professor for all Dep_0/Department for all t/Time
 not **Is_advisor**(Pr_0,Dep_0,t) or
Is_advisor(Pr_0,Dep_0,next(t)) or
DropProfDep(Pr_0,Dep_0,t)

Successor State Axiom: for all prof/Professor for all st/Student for all dep/Department
 for all t/Time

DropProfDep(prof, dep, t) and **Is_advisor**(prof, dep, t) or
 not **BecomeAdvisor**(prof, st, dep, t)

Prof_belong_dep*Positive axiom:*

for all Pr_0/Professor for all Dep_0/Department for all t/Time
 not **Prof_belong_dep**(Pr_0,Dep_0,next(t)) or
Prof_belong_dep(Pr_0,Dep_0,t) or
InsertProfDep(Pr_0,Dep_0,t)

Negative axiom:

for all Pr_0/Professor for all Dep_0/Department for all t/Time
 not **Prof_belong_dep**(Pr_0,Dep_0,t) or
Prof_belong_dep(Pr_0,Dep_0,next(t)) or
DropProfDep(Pr_0,Dep_0,t)

num_prof*Positive axiom:*

for all Pr_0/Professor for all Dep_0/Department for all t/Time

num_prof(Dep_0,next(t)) = **num_prof**(Dep_0,t) or
DropProfDep(Pr_0,Dep_0,t) or
InsertProfDep(Pr_0,Dep_0,t)

Crs_belong_dep

Positive axiom:

for all Crs_0/Course for all Dep_0/Department for all In_0/Integer for all t/Time
not **Crs_belong_dep**(Crs_0,Dep_0,In_0,next(t)) or
Crs_belong_dep(Crs_0,Dep_0,In_0,t) or
InsertCrsDep(Crs_0,Dep_0,In_0,t)

Negative axiom:

for all Crs_0/Course for all Dep_0/Department for all In_0/Integer for all t/Time
not **Crs_belong_dep**(Crs_0,Dep_0,In_0,t) or
Crs_belong_dep(Crs_0,Dep_0,In_0,next(t)) or
DropCrsDep(Crs_0,Dep_0,In_0,t)

num_crs

Positive axiom:

for all Crs_0/Course for all Dep_0/Department for all In_0/Integer for all t/Time
num_crs(Dep_0,next(t)) = **num_crs**(Dep_0,t) or
DropCrsDep(Crs_0,Dep_0,In_0,t) or
InsertCrsDep(Crs_0,Dep_0,In_0,t)

Attend

Examine

Positive axiom:

for all St_0/Student for all Crs_0/Course for all t/Time
not **Examine**(St_0,Crs_0,next(t)) or
Examine(St_0,Crs_0,t) or
EnrolledInCourse(St_0,Crs_0,t)

Negative axiom:

for all St_0/Student for all Crs_0/Course for all t/Time

not **Examine**(St_0,Crs_0,t) or

Examine(St_0,Crs_0,next(t)) or

DropCourse(St_0,Crs_0,t)

size_crs

Positive axiom:

for all St_0/Student for all Crs_0/Course for all t/Time

size_crs(Crs_0,next(t)) = **size_crs**(Crs_0,t) or

DropCourse(St_0,Crs_0,t) or **EnrolledInCourse**(St_0,Crs_0,t)

Βιβλιογραφία

- [BMJ90] A. Borgida, J. Mylopoulos, M. Jarke, M. Koubarakis. Telos: Representing knowledge about Information Systems. In *ACM TOIS*, 8(4): 325-362, 1990
- [BMR93] A. Borgida, J. Mylopoulos, R. Reiter. And nothing else change: The Frame Problem in Procedure Specifications. In *Proceedings of the 15th International Conference on Software Engineering*. 1993
- [BMR95] A. Borgida, J. Mylopoulos, R. Reiter. The Frame Problem in Procedure Specifications. In *IEEE Transaction on Software Engineering*. 1995
- [BDM88] F. Bry, H. Decker, R. Manthey. A Uniform Approach to Constraint Satisfaction and Constraints Satisfiability in Deductive Databases. . In *Proceedings of International Conference on Extending Data Base Technology*. p.p. 488 – 577, 1988
- [BMSU86] F. Bancilhon, D. Maier, Y. Sagiv, J. Ullman. Magic Sets and Other Strange Ways to Implement Logic Programs. In *ACM SIGACT-SIGMOD-SIGMART Symposium on Principles of Database Systems*, 1986.
- [Cas89] Y. Caseau. A Formal System for Producing Demons from Rules in an Object-Oriented Database. In *Proceedings of the 1st International Conference on Deductive and O-O Databases*, p.p. 188-204, Kyoto, Japan, 1989.

- [Cho92a] J. Chomicki. History-less Checking of Dynamic Integrity Constraints. In *8th International Conference on Data Engineering*, p.p. 557-564, Phoenix, AZ, 1992.
- [CRZNM88] L. Chung, D. Rios-Zertuche, B. Nixon and J. Mylopoulos. Process Management and Assertion Enforcement for a Semantic Data Model. In *Proceedings, International Conference on Extended Database Technology*, p.p. 469-487, Venice, Italy, 1988.
- [CT94] J. Chomicki, D. Toman. Implementing Integrity Constraints Using an Active DBMS. In *Proceedings of RIDE-94: Active Database System*, p.p. 87-95, 1994.
- [CT95] J. Chomicki, D. Toman. Implementing Temporal Integrity Constraints Using an Active DBMS. In *IEEE Transaction on Knowledge and Data Engineering*, p.p. 566-581, 1995.
- [CW90] S. Ceri, J. Widom. Deriving Production Rules for Constraints Maintenance. In *VLDB-90*, p.p. 566-577, 1990.
- [Des90] S. Dessoach. Enforcing Integrity in the KBMS KRISYS. In *Proceedings 2nd International Workshop on Foundation of Models and Languages for Data and Objects*, Austria, 1990.
- [ELG84] H. Ehrich, U. Lipeck. M. Gogolla. Specification, Semantics and Enforcement of Dynamic Database Constraints. In *VLDB-84*, p.p. 301-308, 1984.
- [F88] J. Finger. Exploiting Constraints in Design Synthesis. Technical Report STAN-CS-88-1204, Stanford University, 1974.

- [FCT88] A. Furtado, M. Casanova, L. Tucheran. Transforming Constraints into Logic Programs: a Case Study. In R. Meersman and A. Sernadas, editors, *Data and Knowledge (DS-2)*, p.p. 137-149. Elsevier Science Publishers, 1988.
- [GL93] M. Gertz, U. Lipeck. Deriving Integrity Maintenance Triggers from Transaction Graphs. In *9th International Conference on Data Engineering*, p.p. 22-29, 1993.
- [JJ91] M. Jeusfeld, M. Jarke. From Relational to Object Oriented Integrity Simplification. In *Proceedings of DOOD-91*, p.p. 460-477, 1991.
- [Li90] U. Lipeck. Transformation on Dynamic Integrity Constraints into Transaction Specifications. In *Theoretical Computer Science*, 76:115-142, 1990.
- [LF88] U. Lipeck, D. Feng. Construction of Deterministic Transaction Graphs from Integrity Constraints. In *Workshop on Graph Theoretic Concepts in Computer Science*, p.p. 166-179, Springer-Verlag, 1988.
- [LLPS91] G. Lohman, B. Lindsay, H. Pirahesh, B. Schiefer. Extension to Starburst: Objects, Types, Functions and Rules. In *Communication of the ACM*, 34(10), p.p. 94-109, 1991.
- [LR94] F. Lin, R. Reiter. State Constraints Revisited. In *Journal of Logic and Computation-Special Issues on Actions and Processes*, 1994.
- [LS87] U. Lipeck, G. Saake. Monitoring Dynamic Integrity Constraints Based on Temporal Logic. In *Information Systems*, 12(3), p.p. 255-269, 1987.

- [LTW93] M. Lawley, R. Topor, M. Wallace. Using Weakest Preconditions to Simplify Integrity Constraints Checking. In *Proceedings of the Australian Database Conference*, p.p. 161-170, 1993.
- [McCarthy69] J. McCarthy. Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence 4*, Melzter, B. Michie, D.(eds.), 463-502, Edinburgh University Press, 1969.
- [Mat91] N. Mattos. An Approach To Knowledge Base Management. *Springer-Verlag*, 1991.
- [Μητ86] Γ. Μητακίδης. Από την λογική στον Λογικό Προγραμματισμό και την Prolog. 1986.
- [PM96] D. Plexousakis, J. Mylopoulos. Accommodating Integrity Constraints During Database Design. In *Proceedings of the International Conference on Extended Database Technology*, 1996.
- [Pi94] J. Pinto. Temporal Reasoning in the Situation Calculus. Ph.D. thesis, Department of Computer Science, University of Toronto, 1994.
- [Ple93] D. Plexousakis. Integrity Constraints and Rule Management in Temporal Deductive Knowledge Bases. In *Proceedings of the International Conference on Very Large Databases*, p.p.146-157, 1993.
- [Ple93b] D. Plexousakis. Semantical and Ontological Considerations in Telos: a Language for Knowledge Representation. In *Computational Intelligence*, 9(1): 41-72, 1993.

- [Ple95] D. Plexousakis. Compilation and Simplification of Temporal Integrity Constraints. In *Proceedings of the 2nd International Workshop on Rules in Database Systems*, p.p. 260-274, Athens, GR, September 1995.
- [Ple96] D. Plexousakis. On the Efficient Maintenance of Temporal Integrity in Knowledge Bases. Department of Computer Science in the University of Toronto. 1996.
- [Rei88] R. Reiter. On Integrity Constraints. In *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning About Knowledge*, p.p. 97-111, Pacific Grove, CA, March 1988.
- [Rei91] R. Reiter. The Frame Problem in the Situation Calculus: A Simple Solution (Sometimes) and a Completeness Model for Goal Regression. In *V. Lifschitz, editor, Artificial Intelligence and the Mathematical Theory of Computation: Papers in Honor of John McCarthy*, p.p. 359-380, Academic Press, 1991.
- [Rei95] R. Reiter. On Specifying Database Update. In *Journal of Logic Programming*, 25(1), p.p. 53-91, 1995.
- [SMS87] D. Stemple, S. Mazumdar, and T. Sheard. On the Modes and Meaning of Feedback to Transaction Designers. In *Proceedings of ACM-SIGMOD Int. Conference on the Management of Data*, p. p 374-386, San Francisco, CA, 1987.
- [SS89] T. Sheard, D. Stemple. Automatic Update Verification of Database Transaction Safety. In *ACM Transaction on Database Systems*, 14(3), p.p. 322-368, 1989.
- [SS92] S. Schwiderski, G. Saake. Monitoring Temporal Permissions using Partially Evaluated Transaction Graphs. In *Proceedings, 4th International Workshop on Foundation of Models and Languages for Data and Objects*, 1992.

- [Sto75] M. Stonebraker. Implementation of Integrity Constraints and Views by Query Modification. In *Proceedings of ACM-SIGMOD International Conference on the Management of Data*, p.p. 65-78, 1975.
- [STSW92] K-D. Schewe, B. Thalheim, J. Schmidt, I Wetzal. Integrity Enforcement in Object-Oriented Databases. In *Proceedings of the Workshop on Foundations of Models and Languages*, p.p. 174-195, 1992.
- [STW93] K-D. Schewe, B. Thalheim, I Wetzal. Integrity Preserving Updates in Object-Oriented Databases. In *Proceedings of the Australian Database Conference*, p.p. 171-185, 1993.
- [SW95a] A. Siska, O. Wolfson. Temporal Triggers in Active Databases. In *IEEE Transactions of Knowledge and Data Engineering*, Volume 7, p.p. 471-486, 1995.
- [SW95b] A. Siska, O. Wolfson. Temporal Conditions and Integrity Constraints in Active Database Systems. In *Proceedings of ACM SIGMOD*, p.p. 269-280, 1995.
- [Wal91] M. Wallace. Compiling Integrity Checking into Update Procedures. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, p.p. 903-908, 1991.