ΤΕΙ Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Πτυχιακή Εργασία

Ανάπτυξη συστήματος ανάλυσης προτάσεων - ερωτήσεων της Ελληνικής γλώσσας, μετατροπή τους σε στόχους Prolog και επεξεργασία τους από σχεσιακή βάση δεδομένων.

Σπουδαστής: Βαρουξής Κωνσταντίνος
Επιβλέπων Καθηγητής: Μαρακάκης Εμμανουήλ

Ηράκλειο, Ιούλιος 2007

Ευχαριστίες

Ευχαριστώ την οικογένειά μου για τη στήριξή τους στην προσπάθεια μου.

Θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ και την ευγνωμοσύνη μου προς τον καθηγητή κ. Μανόλη Μαρακάκη για την καθοδήγηση, την οργάνωση και την υποστήριξη αυτής της εργασία.

Δε θα μπορούσα να παραλείψω τους πιο στενούς μου φίλους, οι οποίοι στάθηκαν δίπλα μου, μοιράστηκαν τις αγωνίες μου και με στήριζαν σε όλη αυτή τη δύσκολη διαδρομή.

Ευχαριστώ το συμφοιτητή μου Γιώργο Ανταλή για τη συνεργασία μας στην εκπόνηση της εργασίας αυτής.

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

1 ΕΙΣΑΓΩΓΗ

Εκατομμύρια άνθρωποι σε όλο τον κόσμο οι οποίοι διαχειρίζονται κείμενα και βάσεις δεδομένων, δεν έχουν την απαραίτητη γνώση ώστε να συμβαδίσουν με τα καινούργια δεδομένα που έχουν τεθεί στον τομέα της διαχείρισης γνώσης. Ένα απλό παράδειγμα αποτελεί η γραμματέας σε ένα γραφείο. Γράφοντας μια επίσημη επιστολή σε μια ξένη γλώσσα δεν είναι δυνατόν να λαμβάνει υπόψη όλους τους κανόνες γραμματικής και συντακτικού που απαιτούνται ώστε να είναι σωστό το κείμενο. Ακόμα, λέγεται πως το 99% της πληροφορίας που βρίσκεται στο internet είναι άχρηστη για το 99% των χρηστών. Αυτό σημαίνει πως οι χρήστες ενδιαφέρονται στην ουσία για ένα πολύ μικρό κομμάτι της διαθέσιμης πληροφορίας στο internet. Ο χρόνος όμως που δαπανούν για την εύρεση της ζητούμενης πληροφορίας είναι πολύς και σίγουρα σημαντικός. Όσοι επιθυμούν πληροφορίες από βάσεις δεδομένων θα πρέπει να είναι καλοί γνώστες των ειδικών γλωσσών διαμόρφωσης ερωτήσεων, π.χ.SQL, για να πάρουν τα αποτελέσματα που θέλουν.

Θα ήταν φθηνότερο και πιο αξιόπιστο να μπορούσαμε να διδάξουμε τον υπολογιστή να λύσει αυτά τα προβλήματα, παρά να εκπαιδεύεται κάθε γενιά χρηστών στο πώς να τα αντιμετωπίζει.

Τα συστήματα ερωτήσεων - απαντήσεων σε φυσική γλώσσα φιλοδοξούν να αποτελέσουν την επόμενη γενιά διεποικοινωνίας χρηστών με τον υπολογιστή. Επιτρέπουν στους χρήστες να θέτουν ερωτήματα σε φυσική γλώσσα (π.χ. "ποιο μάθημα διδάσκει ο καθηγητής Μαρακάκης", "ποιος καθηγητής διδάσκει το εργαστηριακό μάθημα Προγραμματισμός") και να λαμβάνουν απαντήσεις πάλι σε φυσική γλώσσα. **Στόχος** αυτής της πτυχιακής είναι να συνεισφέρει, στην διευκόλυνση των χρηστών οι οποίοι αλληλεπιδρούν με βάσεις γνώσεων.

Αντικείμενο της πτυχιακής είναι η ανάλυση και επεξεργασία ερωτήσεων, εκφρασμένες στην ελληνική, για ανάκτηση πληροφοριών από μια σχεσιακή

βάση δεδομένων.

Καθώς η ανάκτηση πληροφοριών από σχεσιακές βάσεις δεδομένων απαιτεί κατάλληλες γνώσεις συγκεκριμένων γλωσσών διαμόρφωσης ερωτήσεων (π.χ. SQL), είναι έκδηλη η ανάγκη δημιουργίας ενός συστήματος που θα επιτρέπει στον χρήστη να θέτει ερωτήσεις, όπως στην καθημερινή ζωή του, διατυπωμένες σε φυσική γλώσσα και να λαμβάνει απαντήσεις.

Αξίζει να σημειωθεί πως οι ανάγκες αυτές ήταν γνωστές από την δεκαετία του 1960. Το ISNLIS (LUNAR SCIENCE NATURAL LANGUAGE INFORMATION SYSTEM) ήταν ένα από τα πρώτα πειραματικά συστήματα ερωταποκρίσεων (Question Answering System), που επέτρεπε σε γεωλόγους να έχουν πρόσβαση, να αναλύουν και να συγκρίνουν δεδομένα που προέρχονταν από αναλύσεις πετρωμάτων που είχε λάβει η αποστολή Apollo στο φεγγάρι. Είναι εμφανές ότι η έλλειψη τέτοιου συστήματος, θα καθιστούσε επιβεβλημένη την ανάγκη εκμάθησης μιας ειδικής γλώσσας σχηματισμού ερωτήσεων στους γεωλόγους, γεγονός που θα είχε σοβαρές συνέπειες τόσο σε κόστος όσο και σε χρόνο.

Σκοπός της παρούσας εργασίας είναι η δημιουργία ενός συστήματος, στο οποίο ο χρήστης θα πληκτρολογεί ερωτήσεις σε φυσική γλώσσα (ελληνικά), οι οποίες θα αφορούν την εκπαιδευτική λειτουργία του τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων του Τ.Ε.Ι. Κρήτης. Συγκεκριμένα, η βάση δεδομένων(ΒΔ) του συστήματος αποτελείται από δεδομένα που αφορούν τους φοιτητές, τους καθηγητές, τα μαθήματα, τις πτυχιακές, τις αίθουσες διδασκαλίας και τις μεταξύ τους σχέσεις. Η ΒΔ του συστήματος έχει υλοποιηθεί σε Prolog. Αρχικά κατασκευάσαμε το E-R μοντέλο του προβλήματος. Αυτό το απεικονίσαμε στο σχεσιακό μοντέλο, κάθε σχέση του οποίου έχει υλοποιηθεί σαν ένα κατηγόρημα της Prolog. Το σύστημά μας θα επεξεργάζεται αυτές τις ερωτήσεις μετατρέποντάς τις σε στόχους (ερωτήσεις) της PROLOG. Οι ερωτήσεις(στόχοι), επεξεργάζονται από την Prolog επιστρέφοντας τα αποτελέσματα στον χρήστη. Η γενική προσπάθεια αφορά

την δημιουργία κανόνων συντακτικής, γραμματικής και σημασιολογικής ανάλυσης της πρότασης. Οι εισαγόμενες προτάσεις, θα υφίστανται επεξεργασία με βάση αυτούς τους κανόνες και θα δημιουργούνται ερωτήσεις-στόχοι σε γλώσσα Prolog. Οι στόχοι αυτοί, θα βρίσκουν την απάντηση για το ερώτημα του χρήστη μέσα από τη βάση δεδομένων και η απάντηση αυτή θα επιστρέφεται στον χρήστη σε μορφή λίστας δεδομένων. Ένα άλλο σύστημα το οποίο υλοποιήθηκε από τον συνάδελφο κ. Γ. Ανταλή παίρνει την έξοδο του συστήματός μου και με τη χρήση των βάσεων πληροφοριών συνθέτει την απάντηση σε φυσική γλώσσα στην Ελληνική

Οι πιο συνηθισμένες ερωτήσεις προς ένα σύστημα ερωταποκρίσεων μπορούν να χωριστούν σε 3 (τρεις) βασικές κατηγορίες με βάση την απάντηση που επιδέχονται:

- Ερωτήσεις με *καθορισμένη απάντηση (factual question)*, που περιλαμβάνουν με τη σειρά τους τις ακόλουθες πιο συνηθισμένες υποκατηγορίες:
  - τοποθεσίας: π.χ. «Πού διδάσκεται το μάθημα προγραμματισμός;»
  - χρόνου: π.χ. «Πότε ολοκλήρωσε το μάθημα προγραμματισμός ο φοιτητής Βαρουξής;»
  - ονόματος προσώπου: π.χ. «Ποιος καθηγητής διδάσκει το μάθημα τεχνητή νοημοσύνη;»
  - ποσότητας: π.χ. «Πόσα μαθήματα ολοκλήρωσε ο φοιτητής Ανταλής;»
  - ορισμού: π.χ. «Ποιο μάθημα διδάσκει ο καθηγητής Αϊβαλής;»
- Ερωτήσεις *γνώμης (opinion question)*
- Ερωτήσεις *περίληψης (summary question)*

Η εργασία αυτή ασχολείται μόνο με τις ***ερωτήσεις με καθορισμένη απάντηση*** και πιο συγκεκριμένα με ερωτήσεις ***ονόματος προσώπου*** και ερωτήσεις ***ορισμού***. Παρόλα αυτά, στο σχεδιασμό και στην αρχιτεκτονική του συστήματος δόθηκε ιδιαίτερη σημασία στο να μπορεί να δεχθεί περαιτέρω ανάπτυξη, με αποτέλεσμα να μπορούν να καλυφθούν οι υπόλοιπες

υποκατηγορίες με προσθήκη κώδικα Prolog.

Τέλος, επισημαίνεται ότι λόγω του μεγάλου όγκου των δεδομένων (π.χ.: μαθήματα, αίθουσες, ονόματα φοιτητών και καθηγητών, κ.α.), επιλέχθηκε μικρό δείγμα από κάθε οντότητα, δείγμα όμως επαρκές ώστε να αποδεικνύεται η σωστή λειτουργία του προγράμματος.

Τα ερωτήματα που δημιουργήθηκαν κατά την ανάλυση και τον σχεδιασμό του συστήματος, είχαν άμεση σχέση με την δυνατότητα υλοποίησης ενός τέτοιου προγράμματος. Σημαντικό ζήτημα αποτελούσε το να μπορέσει να δημιουργηθεί το συντακτικό, το οποίο αποτελεί τον βασικό κορμό της πτυχιακής αυτής. "Θα μπορέσει να δημιουργηθεί ένας "κορμός" που θα καλύπτει τις ερωτήσεις των χρηστών;", "θα γίνεται σωστή συντακτική ανάλυση των ερωτήσεων", ήταν μόνο μερικά από τα ερωτήματα τα οποία από την αρχή φάνταζαν απόμακρα. Εκτός από τα ερωτήματα που είχαν σχέση με τη ***σωστή*** υλοποίηση του συστήματος, τα ζητήματα προβληματισμού τα οποία θα αποσαφηνιστούν στο πλαίσιο της εργασίας είναι τα εξής: ποια είναι τα πλεονεκτήματα του συστήματος που θα αναπτυχθεί σε σχέση με τις "παραδοσιακές μεθόδους" ανάκτησης γνώσης από απλά συστήματα βάσεων δεδομένων και αν προσφέρει παραπάνω δυνατότητες.

Τέλος θα πρέπει να αναφέρω ότι σ’αυτή τη πτυχιακή εργασία συνεργάστηκα με το συνάδελφο κ. Γ. Ανταλή στα εξής τμήματα:

1. Ανάλυση, σχεδιασμός και υλοποίηση της βάσης δεδομένων.
2. Διαμόρφωση κοινής BNF σύνταξης για την Ελληνική γλώσσα, ώστε οι ερωτήσεις και οι απαντήσεις να αναγνωρίζονται από τον ίδιο συντακτικό αναλυτή.
3. Υλοποιήσαμε μαζί τον συντακτικό αναλυτή για την BNF σύνταξη της Ελληνικής.

Διάρθρωση της εργασίας

Η παρούσα μελέτη αποτελείται, πέραν της εισαγωγής, από 7 (εφτά) επιμέρους κεφάλαια:

Το **κεφάλαιο 2** πραγματεύεται την παρουσίαση του απαραίτητου θεωρητικού υποβάθρου για την δημιουργία ενός συστήματος ερωταποκρίσεων. Γίνεται εισαγωγή στην Επεξεργασία Φυσικής Γλώσσας, στα πλεονεκτήματα, μειονεκτήματα και εφαρμογές. Ακολουθεί το κριτήριο επιλογής της γλώσσας Prolog για την υλοποίηση του συστήματος, για να γίνει στην συνέχεια αναφορά στις βάσεις δεδομένων και την διεπικοινωνία του συστήματος με τον χρήστη. Στο **κεφάλαιο 3** περιγράφεται συνοπτικά η αρχιτεκτονική του συστήματος και των επιμέρους τμημάτων που αλληλεπιδρούν, για να υπάρξει εκτενής αναφορά στα μέρη αυτά στο **κεφάλαιο 4**. Ακολουθεί το **κεφάλαιο 5**, όπου παρουσιάζονται σενάρια λειτουργίας του συστήματος. Περιγράφονται επίσης τα βήματα για τη δημιουργία της απάντησης. Στο **κεφάλαιο 6** αναφέρονται παρόμοια συστήματα. Τέλος, στο **κεφάλαιο 7** γίνεται η αποτίμηση της εργασίας και τα συμπεράσματα στα οποία καταλήξαμε και στο **κεφάλαιο 8** προτείνονται επεκτάσεις του συστήματος.

2 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ)

Επεξεργασία Φυσικής γλώσσας (Natural Language Processing) είναι ο τομέας της Τεχνητής Νοημοσύνης και της Γλωσσολογίας ο οποίος διαπραγματεύεται την παραγωγή και κατανόηση της φυσικής γλώσσας.

Την επεξεργασία της φυσικής γλώσσας την διακρίνουμε σε επεξεργασία κειμένου και επεξεργασία φωνής, ανάλογα με το αντικείμενο της επεξεργασίας.

Αν η εισαγωγή των πληροφοριών γίνεται με τη μορφή ομιλίας απαιτούνται υψηλής τεχνολογίας μικρόφωνα, ψηφιακές τεχνικές κωδικοποίησης των εισαγομένων σημάτων και διατάξεις αναγνώρισής τους. Τα σημερινά ερευνητικά αποτελέσματα έχουν επιτύχει ως ένα σημαντικό βαθμό την αναγνώριση λέξεων και προτάσεων ενός ομιλητή. Παράλληλα, συνεχίζονται οι έρευνες για την αναγνώριση και κωδικοποίηση των χαρακτηριστικών του ομιλητή (χροιά, φύλο, κλπ.). Αν η εισαγωγή της πληροφορίας γίνεται με την μορφή ήδη τυπωμένου κειμένου χρησιμοποιούνται διατάξεις σάρωσης και οπτικής αναγνώρισης χαρακτήρων. Στη συνέχεια γίνεται κωδικοποίηση και αποθήκευση των χαρακτήρων με τους κλασικούς κώδικες παράστασης (character set codes). Τέλος, αν το κείμενο εισάγεται άμεσα με πληκτρολόγηση, αυτή γίνεται συνήθως στο περιβάλλον κάποιου λεκτικού επεξεργαστή (word processor) ή συντάκτη κειμένου (text editor). Ο μικροκώδικας του οδηγού του πληκτρολογίου (keyboard driver) παράγει τον αντίστοιχο κωδικό παράστασης του κάθε χαρακτήρα. Η επεξεργασία της φυσικής γλώσσας καλύπτει απλές διεργασίες, όπως για παράδειγμα την στατιστική γραμμάτων ή φθόγγων και μορφημάτων ενός κειμένου, όπου εφαρμόζεται κάποιο απλό μοντέλο και εξομοιώνεται από ένα σχετικά απλό

πρόγραμμα. Αναφέρεται όμως και σε σύνθετες απεικονίσεις γλωσσολογικής γνώσης, όπως είναι λόγου χάρη η απεικόνιση μιας παραγράφου που εισάγεται φωνητικά, με πολύπλοκες δομές και διαδικασίες σε ένα σύστημα λογισμικού (software system). Τα συστήματα που παριστάνουν και χειρίζονται την φυσική γλώσσα αξιοποιούν ένα ή περισσότερα από τα κλασικά επίπεδα της γλωσσολογικής ανάλυσης: **φωνολογία, μορφολογία, σύνταξη, σημασιολογία, πραγματολογία** *[Ανδρουτσόπουλος, 1991].*

Η **φωνολογία** αποτελεί επιστημονικό κλάδο της γλωσσολογίας που ασχολείται με τη μελέτη των φωνημάτων μίας συγκεκριμένης γλώσσας, δηλαδή με το ποιοι ήχοι έχουν διακριτή/διαφορετική λειτουργία για το νόημα του γλωσσικού σήματος [www.wikipedia.org].

Η **μορφολογική** ανάλυση αφορά τη κλίση, τη παραγωγή και τη σύνθεση των όρων των προτάσεων. Δηλαδή αποτελεί το μέρος της γραμματικής που εξετάζει τις μεταβολές των λέξεων *[www.wikipedia.org].* Για παράδειγμα, το ρήμα διδάσκ**ει**, διδάσκ**ουν**, κτλ.

Η **σύνταξη** είναι η μελέτη της δομής της πρότασης. Με όρους κανόνων προσπαθεί να περιγράψει τι ανήκει στη γραμματική δομή μιας ιδιαίτερης γλώσσας *[www.wikipedia.org]*. Για παράδειγμα, το επίθετο πρέπει να προηγείται του ουσιαστικού που προσδιορίζει, "το εργαστηριακό μάθημα" και όχι "το μάθημα εργαστηριακό".

Με τον όρο **σημασιολογία** εννοείται η μελέτη του νοήματος, της σημασίας των λέξεων. Συνδέεται με την περιγραφή της αναπαράστασης του νοήματος μιας λέξης στον νου μας και με το πώς χρησιμοποιούμε αυτή την αναπαράσταση - απεικόνιση για την κατασκευή προτάσεων *[www.wikipedia.org].* Για παράδειγμα, το γεγονός energeies_prosopon(1, καθηγητ, διδάσκ), περιγράφει μια ενέργεια("διδάσκουν") που αντιπροσωπεύει τους καθηγητές.

Με τον όρο **πραγματολογία** εννοείται η μελέτη της χρήσης της φυσικής γλώσσας σε πραγματικές συνθήκες ανθρώπινης επικοινωνίας. Πιο συγκεκριμένα, η μελέτη της επίδρασης που έχει το περιβάλλον, είτε είναι γλωσσικό (δηλ. τα συμφραζόμενα) είτε εξωγλωσσικό (π.χ. μορφασμοί, χειρονομίες, νεύματα, επιτονισμός, κ.λπ.) στην ερμηνεία μιας πρότασης, όπως αυτή εκφέρεται μέσα σε συγκεκριμένο χώρο και χρόνο *[www.wikipedia.org].*

Από τα παραπάνω επίπεδα γλωσσολογικής ανάλυσης, το σύστημα EROTISIS πραγματεύεται αυτά της *σύνταξης, της σημασιολογίας και της μορφολογίας*.

Ο τομέας της επεξεργασίας φυσικής γλώσσας με βάσεις δεδομένων, πραγματεύεται τη διεπικοινωνία χρήστη- υπολογιστή. Τα συστήματα αυτά δέχονται ερωτήσεις σε φυσική γλώσσα, περιορισμένου λεκτικού και συντακτικού ρεπερτορίου. Οι γραμματικοί κανόνες οι οποίοι είναι απαραίτητοι για το σχεδιασμό τέτοιων συστημάτων περιγράφονται συνήθως σε γλώσσες λογικού προγραμματισμού, όπως η Prolog.

Κύριο πλεονέκτημα της επεξεργασίας φυσικής γλώσσας είναι ότι ο χρήστης δεν είναι αναγκασμένος να μάθει κάποια γλώσσα προγραμματισμού. Αυτό σημαίνει πως είναι πιο εύχρηστο για περιστασιακούς χρήστες οι οποίοι δεν διαθέτουν χρόνο για να διδαχθούν την απαραίτητη γλώσσα προγραμματισμού.

Υπάρχουν τύποι ερωτήσεων, οι οποίοι είναι δύσκολο να απαντηθούν χωρίς την χρήση φυσικής γλώσσας. Τέτοιες ερωτήσεις μπορούν να περιέχουν άρνηση, όπως η ερώτηση "ποιος φοιτητής **δεν** ολοκλήρωσε το μάθημα προγραμματισμός". Ακόμα, ερωτήσεις οι οποίες εκφράζουν ενδείξεις ποσότητας, όπως "ποιος φοιτητής ολοκλήρωσε **όλα** τα μαθήματα του Στ' εξαμήνου". Οι ερωτήσεις αυτές, δεν έχουν πάντα απλό τρόπο αναπαράστασης στα συνηθισμένα συστήματα ερωτήσεων σε βάσεις δεδομένων. Τέτοιες ερωτήσεις απαιτούν εξειδικευμένες γνώσεις από τον χρήστη σε γλώσσες ανάκτησης δεδομένων από βάσεις δεδομένων.

Ακόμα ένα πλεονέκτημα είναι η χρήση *ελλειπτικών προτάσεων*, όπου κάποιοι όροι μπορούν να παραλειφθούν επειδή έχουν ήδη απαντηθεί. Ακολουθεί παράδειγμα από έναν διάλογο μεταξύ χρήστη και του Loqui:

> Who works on 3 projects?

B. Vandecapelle, C. Willems, D. Sedlock, J.L. Binot, L. Debille, ...

> Which of them are project leaders?

D. Sedlock, J.L. Binot

> Documents describing their projects?

Bim Loqui: "The Loqui Nlidb", "Bim Loqui"

Mmi2: "Technical Annex"

Στην τρίτη ερώτηση, το "their" αντιπροσωπεύει τα ονόματα τα οποία δόθηκαν στον χρήστη σαν απάντηση στην παραπάνω ερώτηση.

Τα συστήματα επεξεργασίας φυσικής γλώσσας έχουν περιορισμένο υπόβαθρο φυσικής γλώσσας, οπότε κάποια εκπαίδευση για την χρήση τους θεωρείται απαραίτητη. Σε μερικές περιπτώσεις μάλιστα, μπορεί να είναι πιο δύσκολο να κατανοήσει ο χρήστης τους περιορισμούς της φυσικής γλώσσας από το να διδαχθεί τη χρήση συστημάτων τα οποία χρησιμοποιούν γραφικό περιβάλλον για την ανάκτηση δεδομένων.

Είναι προφανές ότι ένα σύστημα ερωταποκρίσεων δεν μπορεί να καλύψει όλες τις δυνατές ερωτήσεις που μπορεί να κάνει ένας χρήστης για κάποιο θέμα. Η ελληνική γλώσσα, όπως και όλες οι γλώσσες παγκοσμίως, αναπτύσσονται καθημερινά με την δημιουργία νέων λέξεων και εννοιών. Λέξεις που έχουν συγκεκριμένη έννοια χρησιμοποιούνται σε πολλές διαφορετικές περιπτώσεις, εκφράζοντας διαφορετικά νοήματα κάθε φορά. Ένα απλό παράδειγμα αποτελεί η ερώτηση "πώς τα *πήγε* ο φοιτητής Βαρουξής;". το ρήμα της πρότασης "πήγε", προφανώς έχει εντελώς διαφορετική έννοια, παρόλα αυτά είναι μια ερώτηση αντιληπτή στον ανθρώπινο νου. Ακολουθεί ένα εξίσου απλό παράδειγμα με την ερώτηση "ποια διπλωματική *κάνει* ο φοιτητής Βαρουξής;". Στην πραγματικότητα, κανείς δεν "κάνει" διπλωματική. Το "εκπονεί" είναι το σωστό ρήμα. Η σημασιολογική ανάλυση απαιτεί από τον

χρήστη να πληκτρολογεί ερωτήσεις με όρους το νόημα των οποίων είναι το εκφραστικά ορθό.

Μειονέκτημα της φυσικής γλώσσας είναι ακόμα η ασάφεια που χαρακτηρίζει την φυσική γλώσσα, σε αντίθεση με τα γραφικά συστήματα διεπαφής, όπου οποιαδήποτε ερώτηση είναι δεδομένο πως θα δώσει απάντηση.

Συνηθισμένο φαινόμενο σε συστήματα επεξεργασίας φυσικής γλώσσας είναι ότι σε περίπτωση αποτυχίας της ερώτησης δεν είναι ξεκάθαρο στον χρήστη αν η ερώτηση δεν είναι συντακτικά σωστή ή αν το σύστημα έχει δώσει αρνητική απάντηση. Σαν αποτέλεσμα, μπορεί ο χρήστης να διατυπώνει την ερώτηση με διαφορετικό τρόπο, χωρίς να γνωρίζει πως το σύστημα αδυνατεί να απαντήσει λόγω περιορισμού του λεξικού. Τέτοια προβλήματα μπορούν να λυθούν με την χρήση μηνυμάτων προς τον χρήστη που να εξηγούν το λάθος (π.χ. "άγνωστη λέξη").

Ένα σύστημα Τεχνητής Νοημοσύνης(ΤΝ) έχει λογική και βγάζει συμπεράσματα χρησιμοποιώντας τη γνώση που του έχει δώσει ο σχεδιαστής του. Προφανώς, δε μπορούν να βγάλουν συμπεράσματα από γνώση η οποία δεν υπάρχει. Οι χρήστες συνήθως υπερεκτιμούν την έννοια της Τεχνητής Νοημοσύνης, υποθέτοντας πως το σύστημα μπορεί να καταλάβει οποιαδήποτε πρόταση εισάγουμε, καθώς στα συστήματα επεξεργασίας φυσικής γλώσσας η γνώση που περιέχεται στο σύστημα δεν είναι φανερή στον χρήστη. Αυτό το πρόβλημα δεν παρουσιάζεται στα συστήματα που δεν χρησιμοποιούν φυσική γλώσσα**.**

2.2 Prolog και Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ)

Η Επεξεργασία Φυσικής Γλώσσας κινείται στα πλαίσια της Τεχνητής Νοημοσύνης (AI). Είναι προφανές ότι η γλώσσα προγραμματισμού η οποία επρόκειτο να χρησιμοποιηθεί για την υλοποίηση ενός προγράμματος ερωταποκρίσεων, πρέπει να καλύπτει ευρύ φάσμα της Τεχνητής Νοημοσύνης. Υπάρχουν πολλές γλώσσες προγραμματισμού με διάφορα χαρακτηριστικά και σαφώς πολλοί και καλοί γνώστες των γλωσσών αυτών. Παρόλα αυτά, η υλοποίηση ενός τέτοιου συστήματος δεν πρέπει να αποτελέσει απλά μια πρόκληση για κάποιον καλό προγραμματιστή σε μια οποιαδήποτε γλώσσα προγραμματισμού. Κριτήριο για την επιλογή της κατάλληλης γλώσσας αποτελεί το κατά πόσο μια γλώσσα έχει χαρακτηριστικά τα οποία διευκολύνουν την υλοποίηση μιας εφαρμογής. Στην συγκεκριμένη περίπτωση ενδιαφερόμαστε για μια γλώσσα η οποία θα διαθέτει χαρακτηριστικά τα οποία διευκολύνουν την υλοποίηση ενός συστήματος επεξεργασίας φυσικής γλώσσας.

Η ανάλυση της σύνταξης των εισερχόμενων προτάσεων ονομάζεται **parsing**. Οι **λίστες διαφοράς (difference lists)** είναι πολύ ισχυρά εργαλεία για εφαρμογές parsing, στα οποία η είσοδος παρουσιάζεται με λίστες διαφοράς. Οι λίστες διαφοράς είναι ζευγάρια λιστών τα οποία χρησιμοποιούνται για αναπαράσταση της λίστας των στοιχείων τα οποία αναλύονται. Η ονομασία "λίστες διαφοράς" προκύπτει από την διαφορά μεταξύ της πρώτης λίστας με την δεύτερη. Οι λίστες διαφοράς είναι μια προγραμματιστική τεχνική η οποία δίνει τη δυνατότητα να προσαρτηθεί η λίστα Λ2 στην λίστα Λ1 σε σταθερό χρόνο, ο οποίος είναι ανεξάρτητος του μήκους των λιστών Λ1 και Λ2 [Μαρακάκης, 2006]. Έστω η πρόταση "ποιός, καθηγητής, διδάσκει, το, μάθημα, προγραμματισμός". Για τη συντακτική ανάλυση της πρότασης η Prolog χρησιμοποιεί λίστες διαφοράς για αναπαράσταση των συντακτικών τμημάτων της πρότασης. Η συντακτική ανάλυση γίνεται από τα αριστερά προς τα δεξιά. Αρχικά βρίσκει ότι το υποκείμενο της πρότασης είναι η λίστα διαφοράς

Λ1=[ποιός, καθηγητής| Χ]. Στη συνέχεια βρίσκει και προσαρτίζει στη λίστα Λ1 τη λίστα Χ, που είναι το ρήμα Χ=[διδάσκει| Υ]. Έτσι, φτιάχνει τη λίστα Λ2=[ποιός, καθηγητής, διδάσκει| Υ]. Τέλος, η συντακτική ανάλυση μας δίνει τη λίστα Υ, που παριστά το αντικείμενο της πρότασης Υ=[το, μάθημα, προγραμματισμός]. Η προσάρτιση της λίστας Υ στο τέλος της Λ2 μας δίνει τη συντακτική ανάλυση της τελικής πρότασης στη λίστα Λ3, δηλαδή Λ3=[ποιός, καθηγητής, διδάσκει, το, μάθημα, προγραμματισμός].

Σε εφαρμογές parsing, η πρώτη λίστα περιέχει το στοιχείο το οποίο θα αναλυθεί. Κατηγορήματα τα οποία κάνουν parsing βρίσκουν ό,τι ζητάνε στο μπροστινό μέρος της πρώτης λίστας και ενοποιούν τη δεύτερη λίστα με ότι έμεινε να αναλυθεί. Στο παραπάνω παράδειγμα, ένα κατηγόρημα ψάχνει για το ('ποιος καθηγητής') στην αρχή της πρότασης. Θα το βρει και θα επιστρέψει το υπόλοιπο της πρότασης ('διδάσκει το μάθημα προγραμματισμός'). Αυτό με τη σειρά του θα μπει σε ένα κατηγόρημα το οποίο θα αναζητήσει την ταυτοποίηση του ρήματος. Μόλις το βρει θα επιστρέψει ό,τι έχει απομείνει ('το μάθημα προγραμματισμός'). Αυτό συνεχίζεται μέχρι να μην έχει μείνει τίποτα άλλο να αναλυθεί. Με ποιο απλά λόγια, οι λίστες διαφοράς είναι οι πλέον κατάλληλες από άποψη αποτελεσματικής υλοποίησης της διαδικασία του parsing.

Ακολουθεί ένα απλό παράδειγμα συντακτικής ανάλυσης με χρήση λιστών διαφοράς, όπου καθένας από τους κανόνες subject/2, verb/2 και object/2, συνδέει λίστες. Ακολουθεί το *τερματικό σημείο*, όπου οι κεφαλές των λιστών διαφοράς ταυτοποιούνται με τα κατηγορήματα modifier/2, noun/2 και verb/2.

```
sentence(L1, L4):-
        subject(L1, L2),
         verb(L2, L3),
         object(L3, L4).


subject(L1, L3) :-
         modifier(L1, L2),
         noun(L2, L3).
```

```
object(L1, L3) :-
        modifier(L1, L2),
        noun(L2, L3).


modifier([the|X], X).
noun([cat|X], X).
noun([mouse|X], X).
noun([polar,bear|X], X).
verb([chases|X], X).
verb([eats|X], X).

?- sentence([the, mouse, chases, the, polar, bear], []).
yes

?- sentence([chases, mouse, the, cat], []).
no
```

Πρόγραμμα 2.1: Λίστες διαφοράς και επεξεργασία φυσικής γλώσσας

Η δημιουργία **Γραμματικής Ορισμένων Φράσεων (Definite Clause Grammar)** είναι πολύ σημαντικό εργαλείο της Prolog για επεξεργασία φυσικής γλώσσας. Η DCG είναι ένας βολικός τρόπος αναπαράστασης γραμματικών κανόνων. Η υλοποίησή τους σε Prolog γίνεται με την χρήση λιστών διαφοράς, σε επίπεδο όμως που δεν είναι ορατό στον χρήστη. Παίρνει δηλαδή τους γραμματικούς κανόνες και προσθέτει συνδεδεμένες λίστες στον στόχο.

Έτσι, το πρόγραμμα Πρόγραμμα 2.1 γράφεται σε DCG κανόνες όπως φαίνεται στο παράδειγμα κανόνων κανόνες του προγράμματος 2.2:

```
sentence -->
        subject,
        verb,
        object.

subject -->
        modifier,
        noun.
```

```
object -->
        modifier,
        noun.

modifier -->  [the].

noun -->  [cat].
noun -->  [mouse].
noun -->  [polar, bear].

verb -->  [chases].
verb -->  [eats].
```

Πρόγραμμα 2.2: Παράδειγμα κανόνων DCG

Είναι μια μορφή σαφώς πολύ φιλική προς τον χρήστη, όπου οι κανόνες είναι ξεκάθαροι και η δημιουργία νέων κανόνων είναι πολύ εύκολη. Ο τρόπος αναπαράστασης αυτός είναι πιο κοντά στην ανθρώπινη γνώση. Η προσθήκη νέων κανόνων γίνεται ανεξάρτητα από τους ήδη υπάρχοντες και οι υπάρχοντες κανόνες μπορούν να αλλάξουν ανεξάρτητα από τους υπόλοιπους. Επιπλέον, η περιγραφή των κανόνων είναι ανεξάρτητη από την υλοποίησή τους.

Άλλος ένας σημαντικός λόγος για τον οποίο επιλέξαμε την Prolog είναι ότι σε Prolog μπορεί να υλοποιηθεί εύκολα και αποτελεσματικά μια βάση δεδομένων που στηρίζεται στο μοντέλο οντότητα-σχέση. Η βάση του μοντέλου δεδομένων οντότητα-σχέση είναι η θεωρία συνόλων και η θεωρία σχέσεων. Κάθε κατηγόρημα με τα ορίσματα περιγράφει μια σχέση μεταξύ οντοτήτων. Για παράδειγμα, το κατηγόρημα *πατέρας(Γιάννης, Μαρία)* εκφράζει την σχέση ότι η οντότητα "Γιάννης" είναι πατέρας της οντότητας "Μαρία". Κάθε κατηγόρημα της Prolog αντιστοιχεί σε μια σχέση. Τα ορίσματα ενός κατηγορήματος παίρνουν τιμές από ένα σύνολο. Για το παράδειγμά μας, ο "Γιάννης" και η "Μαρία" είναι τιμές από το σύνολο "Άνθρωπος" .

2.3 Το Μοντέλο Οντότητα-Συσχέτιση(Entity-Relation Model)

Το **μοντέλο δεδομένων οντότητα-συσχέτιση** αναπτύχθηκε με στόχο να έχει τα πλεονεκτήματα των μέχρι τότε διαθέσιμων μοντέλων, δηλαδή του ***δικτύου(network model),*** του ***σχεσιακού(relational model)*** και του ***συνόλου οντότητας(data set model)***[Chen, 1976].

Το μοντέλο οντότητα-συσχέτιση υιοθετεί μια περισσότερο φυσική άποψη ότι ο πραγματικός κόσμος αποτελείται από οντότητες και τις μεταξύ τους συσχετίσεις. Αυτό το μοντέλο περικλείει σημασιολογικές πληροφορίες για τον πραγματικό κόσμο. Τέλος, το μοντέλο οντότητα-συσχέτιση επιτυγχάνει υψηλό βαθμό ανεξαρτησίας των δεδομένων και βασίζεται σε θεωρία συνόλων και σε θεωρία σχέσεων. Το μοντέλο οντότητα-συσχέτιση μπορεί να χρησιμοποιηθεί σαν βάση για ενοποίηση των άλλων μοντέλων δεδομένων. Προτείνονται τρόποι για παραγωγή άλλων μοντέλων από το μοντέλο Οντότητα-Συσχέτιση στο [Chen, 1976].

Η σύνταξη και η σημασιολογία των προτάσεων μιας φυσικής γλώσσας βασίζονται, μεταξύ άλλων, σε οντότητες του πραγματικού κόσμου και στις μεταξύ τους σχέσεις. Δηλαδή, η σημασιολογική πληροφορία για τον πραγματικό κόσμο η οποία περιέχεται στο μοντέλο δεδομένων οντότητα-συσχέτιση είναι σημαντική για την κατασκευή διεπικοινωνίας σε φυσική γλώσσα σε ένα σύστημα το οποίο διαχειρίζεται μια βάση δεδομένων.

Οι βάσεις δεδομένων είναι μεγάλες συλλογές δεδομένων. Αφορούν μοντέλα πραγματικών οργανισμών και καταχωρούν πληροφορίες για:

- οντότητες (π.χ. : φοιτητές, καθηγητές, μαθήματα) και
- συσχετίσεις (π.χ. : ο φοιτητής Χ ολοκλήρωσε το μάθημα Υ)

Ένα σύστημα διαχείρισης βάσης δεδομένων (ΣΔΒΔ) (database management system (DBMS)) αποτελείται από ένα σύνολο δεδομένων και προγράμματα πρόσβασης στα δεδομένα αυτά. Το σύνολο των δεδομένων καλείται βάση δεδομένων (database). Στόχος του ΣΔΒΔ είναι η εύκολη και γρήγορη χρήση και ανάκτηση των δεδομένων από τη βάση δεδομένων. Η διαχείριση των δεδομένων περιλαμβάνει τα εξής:

- Τον ορισμό δομών για την αποθήκευση των δεδομένων
- Τον ορισμό μεθόδων για τη διαχείριση των δεδομένων

Για την δημιουργία ενός μοντέλου οντότητα-συσχέτιση(**Entity-Relationship Model**), απαιτείται η καταγραφή των οντοτήτων.

Ορισμός 2.1: Οντότητα είναι ένα αντικείμενο του πραγματικού κόσμου διακριτό από άλλα αντικείμενα. Μια οντότητα περιγράφεται στη Βάση Δεδομένων (ΒΔ) χρησιμοποιώντας ένα σύνολο χαρακτηριστικών.

Ορισμός 2.2: Σχέση αδύνατης οντότητας ονομάζονται οι σχέσεις για τον προσδιορισμό των οποίων δεν χρησιμοποιούνται άλλες σχέσεις.

Ορισμός 2.3: Σχέση κανονικής οντότητας ονομάζονται οι σχέσεις οι οποίες δε χρησιμοποιούν άλλες σχέσεις για τον προσδιορισμό τους.

Σύνολο οντοτήτων είναι η συλλογή ομοειδών οντοτήτων (π.χ. : φοιτητές, καθηγητές). Όλες οι οντότητες σε ένα σύνολο οντοτήτων έχουν το ίδιο σύνολο χαρακτηριστικών. Κάθε σύνολο οντοτήτων έχει ένα **κλειδί**.

Ορισμός 2.4: Κλειδί είναι ο ελάχιστος αριθμός χαρακτηριστικών που προσδιορίζουν μονοσήμαντα μια οντότητα

Κάθε χαρακτηριστικό έχει ένα πεδίο ορισμού. Κλειδί στον παρακάτω πίνακα είναι το ΑΜ, αφού είναι το μοναδικό χαρακτηριστικό που καθορίζει

μονοσήμαντα την οντότητα ‘φοιτητές’.

Οντότητα **foitites / φοιτητές**
AM / αριθμός μητρώου
Epitheto_foititi / επίθετο φοιτητή
Onoma_foititi / όνομα φοιτητή
Sem_Eisagogis / εξάμηνο εισαγωγής
Tel / τηλέφωνο
Mail_foititi / mail φοιτητή

Πρ. κλειδί: AM
Δευτ. κλειδί: Mail_foititi

AM	Epitheto_foititi	Onoma_foititi	Sem_Eisagogis	Tel	Mail_foititi
642	Βαρουξής	Κωνσταντίνος	01-02 Ε	6937090404	kvarouxis@wall a.com
713	Ανταλής	Γεώργιος	01-02 Ε	2810319528	g_andalis@yaho o.gr
680	Αθανασιάδης	Αθανάσιος	01-02 Ε	6972687798	epp680@epp.tei her.gr

Οντότητα **mathimata / μαθήματα**
Kod_mathimatos / κωδικός μαθήματος
Onoma_mathimatos/ όνομα μαθήματος
Sem_mathimatos / εξάμηνο διδασκαλίας
Typos_mathimatos/ τύπος μαθήματος (**Υ**ποχρεωτικό ή **Ε**πιλογής)
Did_monades / διδακτικές μονάδες

Πρ. κλειδί: Kod_mathimatos
Δευτ. κλειδί: Onoma_mathimatos

Kod_mathimatos	Onoma_mathimatos	Sem_mathimatos	Tupos_mathimatos	Did_monades
ΤΠ4004	Δίκτυα δεδομένων	Δ	Υ	8
ΤΠ5003	Δίκτυα υπολογιστών	Ε	Υ	6
ΤΠ5007	Τεχνητή νοημοσύνη	Ε	Π	5

Πίνακας 2.1:Στιγμιότυπο οντοτήτων φοιτητές, μαθήματα

Η ονοματολογία είναι σε αγγλική γλώσσα καθώς γίνεται αντιστοίχιση των όρων της πρότασης με τα ονόματα των πεδίων των σχέσεων. Για παράδειγμα, αν η πρόταση περιέχει τον όρο "τηλέφωνο", γίνεται αντιστοιχία με το "Tel".

Εκτός από την καταγραφή των οντοτήτων, απαιτείται η καταγραφή των **συσχετίσεων**.

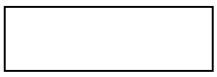
Ορισμός 2.5: Συσχέτιση είναι η σχέση δυο ή περισσότερων οντοτήτων.

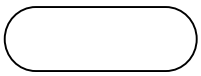
Ορισμός 2.6: ***Σχέση αδύναμης συσχέτισης***(weak relationship relation) ονομάζονται συσχετίσεις στις οποίες κάποιες οντότητες στη συσχέτιση προσδιορίζονται από άλλες συσχετίσεις.

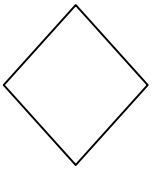
Ορισμός 2.7: ***Σχέσεις κανονικής συσχέτισης*** (regular relationship relation) ονομάζονται οι σχέσεις στις οποίες όλες οι οντότητες στη συσχέτιση προσδιορίζονται από τις τιμές των δικών τους ιδιοτήτων.

Π.χ. : ο καθηγητής Αιβαλής διδάσκει το μάθημα java. Προφανώς, σύνολο συσχετίσεων είναι η συλλογή ομοειδών συσχετίσεων.

Τα βασικά δομικά στοιχεία ενός ER μοντέλου είναι τα εξής:

Οντότητα 

Χαρακτηριστικό ή ισότητα 

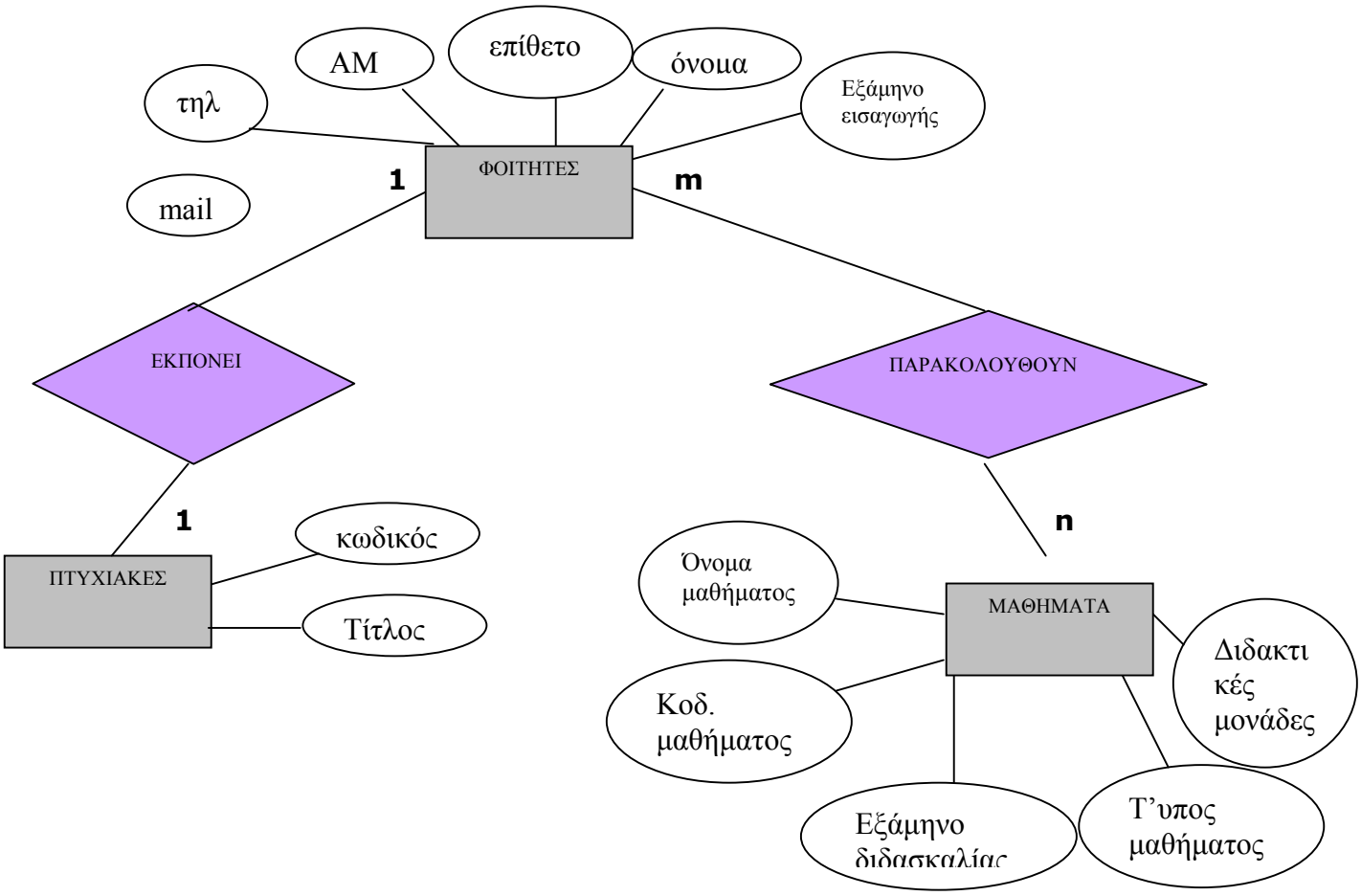
συσχέτιση 

Αδύναμη Οντότητα

Από οντότητα προς συσχέτιση.
Απεικονίζει *περιορισμό κλειδιού*

Σχήμα 2.1: Βασικά δομικά συστατικά για κατασκευή ενός E-R διαγράμματος

Ακολουθεί ένα δείγμα του ER διαγράμματος της εργασίας αυτής.



Σχήμα 2.2: Τμήμα από το E-R διάγραμμα του συστήματός μας

Οι συσχετίσεις ανάμεσα στις οντότητες ΦΟΙΤΗΤΕΣ και ΠΤΥΧΙΑΚΕΣ είναι 1:1, αφού κάθε φοιτητής εκπονεί μια μόνο πτυχιακή, ενώ ανάμεσα στους

ΦΟΙΤΗΤΕΣ και τα ΜΑΘΗΜΑΤΑ είναι m:n, καθώς m φοιτητές μπορούν να παρακολουθούν n μαθήματα.

2.4 Διεπικοινωνία σε Prolog

Το ***Jasper*** είναι η αμφίδρομη διεπαφή μεταξύ **Java** και **Sicstus**. Η μεριά της Java αποτελείται από ένα πακέτο java (se.sics.jasper) το οποίο περιέχει κλάσεις οι οποίες αναπαριστούν το runtime σύστημα της Prolog (SICStus,SPTerm etc.). Το κομμάτι της Prolog έχει σχεδιαστεί σαν μια αυτοτελής μονάδα (module) της βιβλιοθήκης (library (jasper)). Η αυτοτελής αυτή μονάδα της βιβλιοθήκης (library jasper) παρέχει λειτουργίες για φόρτωμα (load) και ξεφόρτωμα (unload) του JVM (Java Virtual Machine), λειτουργία για κλίση μεθόδων (jasper_call/4) και κατηγορήματα για διαχείριση αντικειμένων (Objects).

Το jasper μπορεί να χρησιμοποιηθεί με δύο τρόπους εφαρμογής ανάλογα σε ποιό σύστημα λειτουργεί ως ***κύρια εφαρμογή*** (parent application). Αν η Java είναι η κύρια εφαρμογή, η Sicstus θα φορτωθεί μέσα στο JVM χρησιμοποιώντας την μέθοδο System.loadlibrary. Αν η Sicstus είναι η κύρια εφαρμογή, η Java θα φορτωθεί με την χρήση του ερωτήματος/στόχος use_module(library(jasper)).

Κάλεσμα της Prolog από την Java γίνεται με χρήση του πακέτου se.sics.jasper. Αυτό το πακέτο περιέχει μία σειρά από κλάσεις Java οι οποίες χρησιμοποιούνται για δημιουργία και διαχείριση όρων, εκτέλεση ερωτήσεων/στόχων και ικανοποίηση μιας η περισσότερων λύσεων.

Ο παρακάτω κώδικας αναπαριστά την χρήση jasper σε single threaded mode [SICS, 2003]:

```
1       import se.sics.jasper.SICStus;
2       import se.sics.jasper.Query;
3       import java.util.HashMap;
4       public class Simple
5       {
6       public static void main(String argv[]) {
7               SICStus sp;
```

```
8               Query query;
9               HashMap WayMap = new HashMap();
10      try {
11              sp = new SICStus(argv,null);
12              s.restore("train.sav");
13              query = sp.openPrologQuery("connected(’¨Orebro’,
                ’Stockholm’,Way,Way).",WayMap);
14      try {
15              while (query.nextSolution()) {
16              System.out.println(WayMap);
17          }
18      } finally {
19              query.close();
20                  }
21      }
22      catch ( Exception e ) {
23              e.printStackTrace();
24                          }
25                    }
26      }
```

Πρόγραμμα 2.3: Παράδειγμα διεπικοινωνίας jasper με Prolog

Ας δούμε πως λειτουργελι ο κώδικας του Προγράμματος 2.3

Πριν την κλήση κατηγορημάτων το runtime system της Sicstus πρέπει να αρχικοποιηθεί. Αυτό επιτυγχάνεται δημιουργώντας στιγμιότυπο (instance) της κλάσης Sicstus. Κάθε **αντικείμενο** (Object) της Sicstus αντιστοιχεί σε ένα ανεξάρτητο αντίγραφο του runtime system της Sicstus. Σε αυτό το παράδειγμα έχουμε ορίσει την τιμή null σαν δεύτερο όρισμα στο αντικείμενο Sicstus (γραμμή 11). Αυτή η εντολή *διατάζει* την Sicstus να βρει το sprt.sav χρησιμοποιώντας τις δικές του εσωτερικές μεθόδους.

Ερωτήσεις/στόχοι γίνονται με κλήση της μεθόδου query (γραμμή 13). Τα ορίσματα σε αυτή την μέθοδο είναι μια συμβολοσειρά (String) καθορίζοντας τον στόχο Prolog και έναν πίνακα (Map) ο οποίος περιέχει την αντιστοίχιση των μεταβλητών με τις τιμές που ικανοποιούν τον στόχο. Αυτή η μέθοδος χρησιμοποιείται για την εύρεση μόνο μίας λύσης.

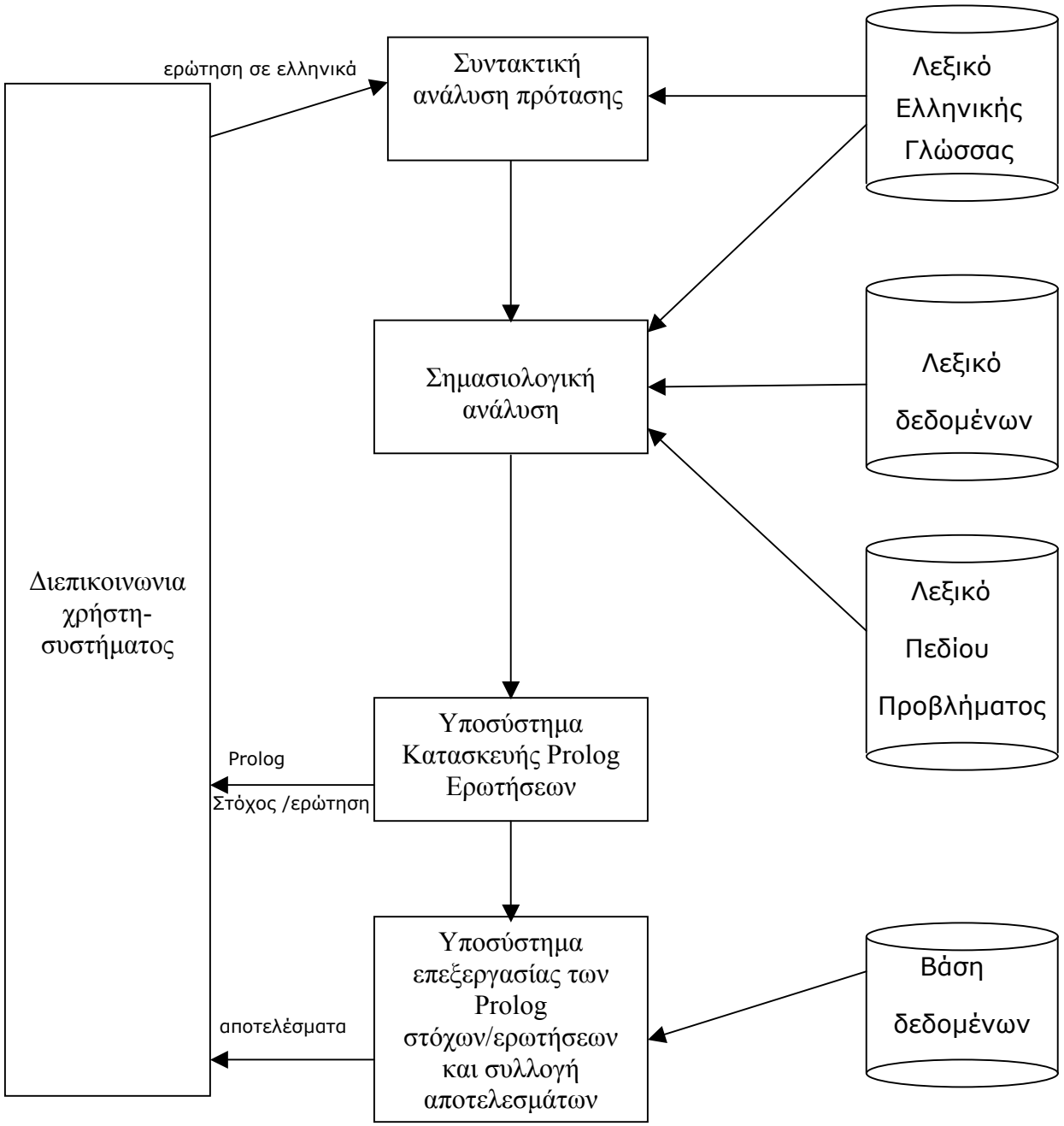
Επόμενο βήμα είναι να φορτώσουμε τον κώδικα Prolog. Αυτό επιτυγχάνεται καλώντας την μέθοδο restore (γραμμή 12). Αυτή η μέθοδος πρέπει να καλεστεί πριν από κάθε άλλη κλήση μεθόδων Sicstus.

Η μέθοδος openquery (γραμμή 13) επιστρέφει μία αναφορά του ερωτήματος στόχου. Για την ανάκτηση των υπόλοιπων λύσεων καλείται η μέθοδος nextSolution (γραμμή 15). Η μέθοδος αυτή καλείται χωρίς ορίσματα και επιστρέφει τιμή true όσο υπάρχουν λύσεις.

Η ερώτηση/στόχος πρέπει να κλείσει παρ' όλο που το nextSolution υποδεικνύει κάποια στιγμή ότι δεν υπάρχουν άλλες λύσεις. Αυτό επιτυγχάνεται με την μέθοδο queryclose() (γραμμή 19).

3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η αρχιτεκτονική του συστήματος αποτελείται από 5 (πέντε) τμήματα τα οποία συμβολίζονται με παραλληλόγραμμα και 4 (τέσσερις) βάσεις πληροφοριών οι οποίες συμβολίζονται με κυλίνδρους. Η σωστή λειτουργία και αλληλεπίδρασή τους καθορίζει την αποτελεσματικότητα του συστήματος.



Σχήμα 3.1: Αρχιτεκτονική Συστήματος EROTISIS

Το **πρώτο τμήμα** είναι η *Διεπικοινωνία χρήστη - συστήματος*. Ο χρήστης του συστήματος εισάγει την ερώτησή του σε ένα παραθυρικό περιβάλλον σχεδιασμένο σε Java.

Η πρόταση-ερώτηση του χρήστη περνά στο **δεύτερο τμήμα**, αυτό της *Συντακτικής Ανάλυσης Πρότασης*. Σε αυτό το τμήμα, οι γραμμικές ακολουθίες λέξεων μετατρέπονται σε δομές που απεικονίζουν τον τρόπο με τον οποίο συνδέονται η μια με την άλλη. Αν οι συντακτικοί κανόνες της γλώσσας, οι οποίοι ρυθμίζουν τον τρόπο με τον οποίο συνδυάζονται οι λέξεις (κανόνες DCG) τηρούνται στη δομή της πρότασης, η διαδικασία συνεχίζει. Σε αντίθετη περίπτωση, ο συντακτικός αναλυτής απορρίπτει την πρόταση. Για παράδειγμα, ο συντακτικός αναλυτής του συστήματος απορρίπτει την πρόταση "ποιο θεωρητικό μάθημα Βαρουξής παρακολουθεί φοιτητής ο;".

Οι συντακτικά σωστές προτάσεις περνάνε στο **τρίτο τμήμα**, στον *Σημασιολογικό Αναλυτή* όπου γίνεται σημασιολογική ανάλυση. Δυστυχώς όμως, οι λέξεις εκτός του ότι μπορεί να έχουν πολλαπλά νοήματα, μπορεί να έχουν και διαφορετικές χρήσεις. Ο Σημασιολογικός Αναλυτής αντλεί στοιχεία από το *Λεξικό Πεδίου Προβλήματος* και εξακριβώνει τη σωστή χρήση της κάθε λέξης. Χαρακτηριστικό παράδειγμα η ερώτηση "ποιο μάθημα διδάσκει η φοιτήτρια κατσαμάκη". Η πρόταση είναι συντακτικά σωστή, αλλά οι φοιτήτριες δεν διδάσκουν μαθήματα. Το *Λεξικό της Ελληνικής Γλώσσας* περιέχει γραμματικά στοιχεία των λέξεων. Καταχωρώντας το εξής στιγμιότυπο: ρήμα(3, διδάσκουν, διδάσκ), ορίζουμε ότι η ρίζα του ρήματος ‘διδάσκουν’ είναι ‘διδάσκ’. Το 3 είναι το κλειδί αυτής της εγγραφής, ώστε να είναι δυνατή η διάκρισή της από εγγραφές όπως: ρήμα(4, διδάσκει, διδάσκ). Η υλοποίηση σε Prolog του στιγμιότυπου ρήμα(3, διδάσκουν, διδάσκ), γίνεται από το εξής Prolog γεγονός: prot_rima(3, ‘διδάσκουν’, ‘διδάσκ’). Στο *Λεξικό Πεδίου Προβλήματος* υπάρχει το παρακάτω γεγονός το οποίο εκφράζει ότι η ενέργεια είναι η διδασκαλία μαθημάτων:

ενέργειες_προσώπων

Κλειδί	Οντότητα πεδίου προβλήματος	Ενέργεια
1	καθηγητ	διδάσκ

Αυτό έχει υλοποιηθεί σε Prolog από το εξής: energeies_prosopon(1, καθηγητ, διδάσκ). Για τα φυσικά πρόσωπα της Βάσης Δεδομένων (φοιτητές και καθηγητές), έχουν δηλωθεί με αντίστοιχο τρόπο οι ενέργειες που τους αντιστοιχούν. Για παράδειγμα, οι φοιτητές *παρακολουθούν, εκπονούν, ολοκλήρωσαν*. Επειδή δεν υπάρχει γεγονός που να δηλώνει ότι οι φοιτητές διδάσκουν, η πρόταση "ποιο μάθημα διδάσκει η φοιτήτρια κατσαμάκη;" αποτυγχάνει. Οι προτάσεις οι οποίες δε μπορούν να αναλυθούν από τον Σημασιολογικό Αναλυτή θα απορριφθούν.

Ακολουθεί το **τέταρτο τμήμα**, το *Υποσύστημα Κατασκευής της ερώτησης σε Prolog*, όπου η πρόταση η οποία έχει εισαχθεί μετατρέπεται σε στόχο της Prolog. Η ερώτηση "ποιο μάθημα διδάσκει ο καθηγητής μαρακάκης", μετατρέπεται στον εξής στόχο: *kathigites(A, μαρακάκης, B, C, D, E, F), didaskoun(A,G), (apoteleitai_apo(H,G,J); apoteleitai_apo(H,K,G)), mathimata(H,L,M,N,O), leptomereies_mathimaton(G,P,Q).*

Ο παραπάνω στόχος περνάει στο **πέμπτο τμήμα**, το οποίο είναι η *Επεξεργασία της Ερώτησης Prolog και Συλλογή των Ζητούμενων Αποτελεσμάτων*. Σε αυτό το τμήμα, ο στόχος σε Prolog βρίσκει τα αποτελέσματα από τη *Βάση Δεδομένων*. Η Βάση Δεδομένων περιέχει όλες τις πληροφορίες που εισάγαμε για τον σχεδιασμό του "κόσμου" μας. Για παράδειγμα, στη βάση δεδομένων έχουμε τους καθηγητές. Για κάθε καθηγητή υπάρχει μια εγγραφή(στιγμιότυπο) στη σχέση καθηγητές/kathigites της βάσης δεδομένων. Κάθε στιγμιότυπο υλοποιείται σε Prolog σαν ένα γεγονός, όπως το παρακάτω παράδειγμα: *kathigites(004, μαρακάκης, μανόλης, αρσ, πκ1, 2810379748, mail).* Η επεξεργασία των στόχων μπορεί να επιστρέψει και άλλα δεδομένα, τα οποία δε ζητούνται στην ερώτηση του χρήστη. Το παρόν

τμήμα επιλέγει μόνο εκείνα τα δεδομένα τα οποία ζητά ο χρήστης με την ερώτησή του.

4 ΛΕΠΤΟΜΕΡΗΣ ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ

4.1 Η Βάση Δεδομένων

Μια σωστά σχεδιασμένη βάση δεδομένων παρέχει πρόσβαση σε ενημερωμένες και ακριβείς πληροφορίες. Επειδή η σωστή σχεδίαση είναι ουσιαστικής σημασίας για την επίτευξη της πτυχιακής εργασίας, η επένδυση του χρόνου που απαιτήθηκε για την εκμάθηση των αρχών της καλής σχεδίασης ήταν σημαντική.

Υπάρχουν κάποιες βασικές αρχές που πρέπει να ακολουθηθούν για τη δημιουργία της βάσης δεδομένων. Πρώτα απ' όλα, δεν πρέπει να υπάρχουν άχρηστες πληροφορίες, γνωστά και ως *πλεονάζοντα δεδομένα*. Τα δεδομένα αυτά, αυξάνουν την πιθανότητα λάθους και σπαταλούν χώρο. Δεύτερη αρχή είναι ότι η ορθότητα και η πληρότητα των πληροφοριών είναι σημαντικές. Προφανώς, εάν η βάση δεδομένων περιέχει λανθασμένες πληροφορίες, τα αποτελέσματα που αντλούνται θα είναι και αυτά λανθασμένα.

Για τη σχεδίαση της Βάσης Δεδομένων, ακολουθήθηκαν τα εξής βήματα:

1. Κατασκευάσαμε το E-R μοντέλο του προβλήματός μας.
2. Απεικονίσαμε το E-R μοντέλο στο σχεσιακό μοντέλο δεδομένων.
3. Ελέγξαμε αν η σχεσιακή βάση δεδομένων που κατασκευάσαμε ικανοποιεί τους τρεις κανόνες κανονικοποίησης.

Πιο αναλυτικά, συγκεντρώσαμε τους τύπους των πληροφοριών που θέλαμε να εγγράψουμε στη βάση δεδομένων, όπως ονόματα φοιτητών και καθηγητών. Δηλαδή καταγραφή όλων των πιθανών τύπων πληροφοριών που θα μπορούσαν να αποτελέσουν πεδίο κάποιου πίνακα. Σημείο-κλειδί σε αυτή τη προσπάθεια είναι να σκεφτούμε τις ερωτήσεις που θέλουμε να απαντάει το σύστημα. Για παράδειγμα "ποιο μάθημα διδάσκει ο καθηγητής με τηλέφωνο 2810379748;". Αντιλαμβανόμαστε από την ερώτηση δυο τύπους πληροφορίας, το όνομα μαθήματος και το τηλέφωνο του καθηγητή. Στη

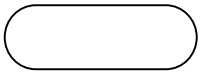
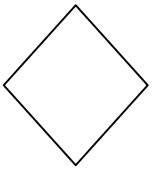
συνέχεια, οι πληροφορίες μετατρέπονται σε σχέσεις, για παράδειγμα φοιτητές/foitites, μαθήματα/mathimata. Έπειτα, τα στοιχεία μετατρέπονται σε πεδία. π.χ.: η σχέση φοιτητές/foitites έχει σαν πεδία τα "ΑΜ", "επίθετο", "όνομα", "γένος, "εξάμηνο εισαγωγής", "τηλέφωνο", "mail". Επιλέγεται το πρωτεύον κλειδί, το πεδίο δηλαδή που χρησιμοποιείται για το μοναδικό προσδιορισμό κάθε εγγραφής. Τέλος, γίνεται έλεγχος των τριών κανόνων κανονικοποίησης, για να διαπιστώσουμε αν η σχεσιακή βάση δεδομένων είναι σωστά σχεδιασμένη.

4.1.1 Κατασκευή Entity-Relationship Μοντέλου Προβλήματος

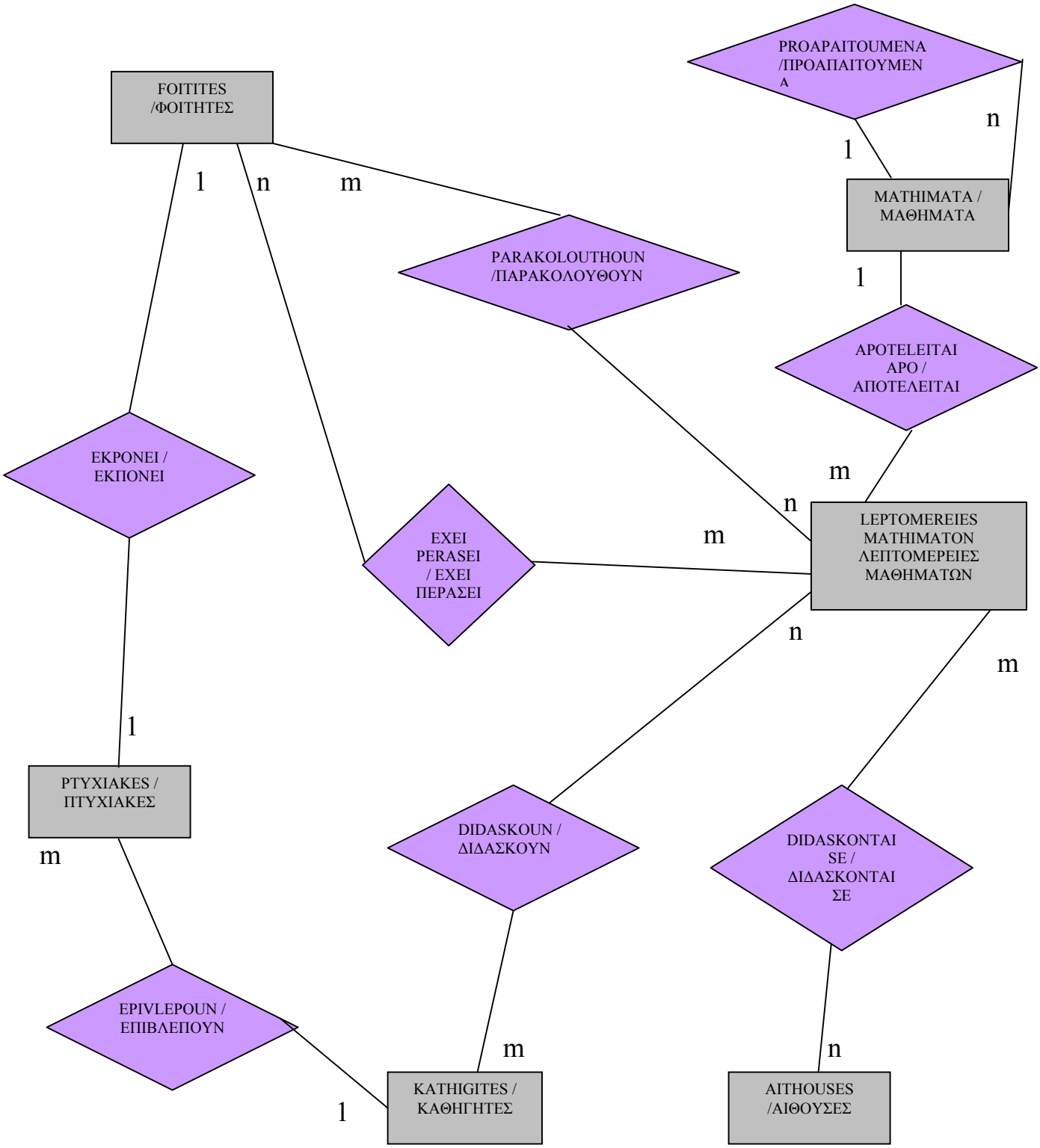
4.1.1.1 Entity-Relation Diagram/Διάγραμμα Συσχετίσεων-Οντοτήτων.

Ακολουθεί το ER μοντέλο, το οποίο είναι η φυσική απεικόνιση του "κόσμου" τον οποίο περιγράφουμε.

Τα βασικά δομικά στοιχεία του ER μοντέλου είναι τα εξής:

Οντότητα		
Χαρακτηριστικό ή ιδιότητα		
συσχέτιση		
m		Σύνδεση συσχέτισης με οντότητες. Το m παριστά τη πληθυκότητα εμφάνισης του κλειδιού της οντότητας στη συσχέτιση. Το m:n παριστά τη πληθυκότητα εμφάνισης των κλειδιών των οντοτήτων στη συσχέτιση.

Σχήμα 4.1: Βασικά δομικά συστατικά του E-R διαγράμματος του συστήματος EROTISIS.



Σχήμα 4.2: E-R διάγραμμα του συστήματος EROTISIS.

4.1.1.2 Ορισμός Συνόλου Τιμών

Σε αυτό το σημείο καθορίζεται το πεδίο ορισμού του κάθε πεδίου. Στην πρώτη στήλη βρίσκονται τα **πεδία των οντοτήτων και των συσχετίσεων** που περιγράφονται στη Βάση Δεδομένων. Στην δεύτερη στήλη βρίσκονται οι **αναπαραστάσεις** των πεδίων αυτών και στο τρίτο πεδίο οι **επιτρεπτές τιμές** που μπορούν να πάρουν.

Σύνολο τιμών	**Αναπαράσταση**	**Επιτρεπτές τιμές**
AM/Αριθμός_μητρώου	integer(5)	0-99999
Onoma_foititi/Όνομα_φοιτητή	character(20)	ALL
Epitheto_foititi/Επίθετο_φοιτητή	character(20)	ALL
Sem_Eisagogis/Εξ_εισαγ	string	Year/semester
Tel/Τηλ	string	ALL
Mail_foititi/Mail_φοιτητή	string	ALL
Kod_math_Th_E/Κωδ_αναλ_μαθ	alphanumeric(7)	ALL
Did_monades/Διδ_μονάδες	integer(2)	2 – 10
Onoma_mathimatos/Ονομα_μαθήμ	character(30)	ALL
Sem_mathimatos/Εξ_μαθήμ	character(2)	Α,Β,Γ,Δ,Ε,ΣΤ,Ζ
Kod_mathimatos/Κωδ_μαθήμ	alphanumeric(6)	ALL
Kod_proap_mathimatos/Κωδ_προαπ_μαθήμ	alphanumeric(6)	ALL
Kod_ptuxiakis/Κωδ_διπλωματ	alphanumeric(5)	ALL
Vathmos_ptyxiakis/Βαθμός_δπλωματ	real	5.00 – 10.00
Period_didaskal/Εξ_διδασκ	date	dd/mm/yy
Ores_mathimatos/Ωρες_μαθήμ	integer(1)	2 – 7
Typos_mathimatos/Τύπος_μαθήμ	character(5)	Επιλ, υποχρ
Hmer_eksetasis/Ημερ_εξέτ	string	dd/mm/yy, semester
Kod_aithousas/Κωδ_αίθους	alphanumeric(5)	ALL
Day/Μέτα	character(3)	Δευ, Τρ, Τετ, Πεμ, Παρ, Σαβ
Hour/Ωρα	real(2.2)	8.00 – 22.00
Kod_kathigiti/Κωδ_Καθηγητή	integer(4)	1 – 9999
Onoma_kathigiti/Ονομα_Καθηγητή	character(20)	ALL
Epitheto_kathigiti/Επίθετο_Καθηγητή	character(20)	ALL
Kod_Grafeiou_kathigiti/Κωδ_γραφ_Καθηγητή	alphanumeric(5)	ΠΚ1-5, ΝΚ1-5, ΣΤΕΦ1-5
Tel_grafeiou/Τηλ_γραφείου	integer(5)	100000 – 99999
Thema_ptyxiakis/Τίτλος_διπλωμ	character(200)	ALL

Onoma_aithousas/Ονομα_αιθους	alphanumeric(20)	ALL
Capacity/Χωρητικ	integer(2)	10 – 99
Vathmos_mathimatos/Βαθμός_μαθήμ	real(2.2)	5.00 – 10.00
Mail_kathigiti/Mail_καθηγητή	string	ALL
Ores_mathimatos/Ωρες_μαθήμ	integer(1)	2 – 6
Eidos_mathimatos/Τύπος_μαθήμ	character(1)	Θ, Ε
Per_parakolouthisis/Εξ_παρακολούθησης	string	Year/semester

Πίνακας 4.1: Ορισμός συνόλου τιμών πεδίων.

4.1.1.3 Σχέσεις Οντοτήτων και Συσχετίσεων

Οντότητα **foitites / φοιτητές**

AM	/	αριθμός μητρώου
Mail_foititi	/	mail φοιτητή
Onoma_foititi	/	όνομα φοιτητή
Epitheto_foititi	/	επίθετο φοιτητή
Sem_Eisagogis	/	εξάμηνο εισαγωγής
Tel	/	τηλέφωνο
Πρ. κλειδί:		AM
Δευτ. κλειδί:		Mail_foititi

Οντότητα **mathimata /μαθήματα**

Kod_mathimatos	/	κωδικός μαθήματος
Onoma_mathimatos/		όνομα μαθήματος
Did_monades	/	διδακτικές μονάδες
Sem_mathimatos	/	εξάμηνο διδασκαλίας
Typos_mathimatos/		τύπος μαθήματος (**Υ**ποχρεωτικό ή **Ε**πιλογής)
Πρ. κλειδί:		Kod_mathimatos
Δευτ. κλειδί:		Onoma_mathimatos

Οντότητα **Leptomereies mathimaton / λεπτομέρειες μαθημάτων**

Kod_math_Th_E	/	κωδικός αναλυτικών μαθημάτων
Ores_mathimatos	/	ώρες μαθήματος
Eidos_mathimatos	/	θεωρία ή εργαστήριο
Πρ. κλειδί:		Kod_math_Th_E

Οντότητα **aithouses / αίθουσες**

Kod_aithousas	/	κωδικός αίθουσας
Onoma_aithousas	/	όνομα αίθουσας
Capacity	/	χωρητικότητα
Πρ. κλειδί:		Kod_aithousas
Δευτ. κλειδί:		Onoma_aithousas

Οντότητα **ptyxiakes / πτυχιακές**

Kod_kathighti	/	κωδικός πτυχιακής
Thema_ptyxiakis	/	θέμα πτυχιακής
Πρ. κλειδί:		Kod_kathighti
Δευτ. κλειδί:		Thema_ptyxiakis

Οντότητα **kathightes** / **καθηγητές**

Kod_kathigiti	/	κωδικός καθηγητή
Onoma_kathigiti	/	όνομα καθηγητή
Epitheto_kathigiti	/	επίθετο καθηγητή
Tel_grafeiou	/	τηλ. Γραφείου
Kod_Grafeio_kathigiti/		γραφείο καθηγητή
Kod_kathigiti	/	mail καθηγητή
Πρ. κλειδί:		Kod_kathigiti
Δευτ. κλειδί:		Kod_kathigiti

Συσχέτιση **apoteleitai apo** / **αποτελείται από**

Kod_mathimatos	/	κωδικός μαθήματος
Kod_math_Th_E	/	κωδικός αναλυτικών μαθημάτων
Πρ. κλειδί:		Kod_mathimatos, Kod_math_T_E

Συσχέτιση **parakolouthoun** / **παρακολουθούν**

AM	/	αριθμός μητρώου
Kod_math_Th_E	/	κωδικός αναλυτικών μαθημάτων
Per_parakolouthisis/		περίοδος παρακολούθησης
Πρ. κλειδί:		AM, Kod_math_Th_E

Συσχέτιση **proapaitoumena** / **προαπαιτούμενα**

Kod_mathimatos	/	κωδικός μαθήματος
Kod_proap_mathihmatos	/	κωδικός προαπαιτούμενου μαθήματος
Πρ. κλειδί:		Kod_mathimatos, Kod_proap_mathihmatos

Συσχέτιση **exei perasei** / **έχει_ολοκληρώσει επιτυχώς**

AM	/	αριθμός μητρώου
Kod_math_Th_E	/	κωδικός αναλυτικών μαθημάτων
Vathmos_mathimatos	/	βαθμός μαθήματος
Hmer_eksetasis	/	ημερομηνία εξέτασης
Πρ. κλειδί:		AM, Kod_math_Th_E

Συσχέτιση **ekponei** / **εκπονεί**

AM	/	αριθμός μητρώου
Kod_ptuxiakis	/	κωδικός διπλωματικής
Vathmos_ptyxiakis/		βαθμός διπλωματικής
Πρ. κλειδί:		AM, Kod_ptuxiakis

Συσχέτιση **epivlepoun** / **επιβλέπουν**

Kod_ptuxiakis	/	κωδικός πτυχιακής
Kod_kathigiti	/	κωδικός καθηγητή
Πρ. κλειδί:		Kod_ptuxiakis, Kod_kathigiti

Συσχέτιση **didaskoun** / **διδάσκουν**

Kod_kathigiti	/	κωδικός καθηγητή
Kod_math_Th_E	/	κωδικός αναλυτικών μαθημάτων
Πρ. κλειδί:		Kod_kathigiti, Kod_math_Th_E

Συσχέτιση **didaskontai se** / **διδάσκονται_σε**

Kod_math_Th_E	/	κωδικός αναλυτικών μαθημάτων
Kod_aithousas	/	κωδικός αίθουσας
Period_didaskal	/	περίοδος διδασκαλίας
Day	/	μέρα
Hour	/	ώρα
Πρ. κλειδί:		Kod_math_Th_E, Kod_aithousas

Με *πλάγια και υπογραμμισμένα* είναι τα πρωτεύοντα κλειδιά, ενώ *πλάγια* μόνο είναι τα δευτερεύοντα κλειδιά

φοιτητές(*αριθμός_μητρώου, mail_φοιτητή*, ονομα_φοιτητή, επίθετο_φοιτητή, Εξάμηνο_εισαγωγής, τηλέφωνο)
foitites(*AM*, *Mail_foititi*, Name_foititi, Epitheto_foititi, Sem_Eisagogis , Tel)

παρακολουθούν (*αριθμός_μητρώου, Κωδικός_αναλυτικών_μαθημάτων*, Περίοδος_παρακολούθησης)
parakolouthoun*(AM, Kod_math_T_E,* Per_parakolouthisis*)*

μαθήματα(*κωδικός_μαθήματος, όνομα_μαθήματος*, Διδακτ_μονάδες, τύπος_μαθήματ)
mathimata(*Kod_mathimatos*, *Onoma_mathimatos*, Did_monades, Sem_mathimatos, Typos_mathimatos)

λεπτομέρειες μαθημάτων(*Κωδικός_αναλυτικών_μαθημάτων*, ωρες_μαθήματ, τύπος_μαθήματ)
leptomereies mathimaton(*Kod_math_Th_E*, Ores_mathimatos, Eidos_mathimatos)

αίθουσες(*κωδικός_αίθουσας, όνομα_αίθουσας*, χωρητικότητα)
aithouses(*Kod_aithousas*, *Onoma_aithousas,* Capacity)

πτυχιακές(κωδικός_διπλωματ, τίτλος_διπλωμ)
ptyxiakes(Kod_ptyxiakis, Thema_ptyxiakis)

καθηγητές(κωδικός_καθηγητή, mail_καθηγητή, όνομα_καθηγητή, επίθετο_καθηγητή, τηλ_γραφείου, κωδ_γραφείου_καθηγητή)
kathigites(Kod_kathigiti, Mail_kathigiti, Onoma_kathigiti,Epitheto_kathigiti, Tel_grafeiou, Kod_Grafeiou_kathigiti)

αποτελείται από(*κωδ_μαθήματος, κωδ_αναλυτικών_μαθημάτων*)
apoteleitai apo(*Kod_mathimatos*, *Kod_math_hT_E*)

προαπαιτούμενα(*κωδ_μαθήματος, κωδ_προαπ_μαθήματος*)
proapaitoumena(*Kod_mathimatos*, *Kod_proap_mathimatos*)

έχει ολοκληρώσει επιτυχώς(*αριθμός_μητρώου*, *κωδ_αναλυτικών_μαθημάτων*, βαθμός_μαθήμ, Ημερ_εξέτασης)
exei perasei(*AM*, *Kod_math_Th_E*, Vathmos_mathimatos, Hmer_eksetasis)

εκπονεί(*αριθμός_μητρώου, κωδ_διπλωματικής*, βαθμός διπλωματικής)
ekponei(*AM*, Kod_ptyxiakis, Vathmos_ptyxiakis, Vathmos)

επιβλέπουν(*κωδ_διπλωμ, κωδ_καθηγητή*)

epivlepoun(*Kod_ptyxiakis*, *Kod_kathigiti*)

διδάσκουν(*κωδ_καθηγητή, κωδ_αναλυτικών_μαθημάτων*)
didaskoun(*Kod_kathigiti, Kod_math_Th_E)*
Didaskontai se / διδάσκονται_σε(*Kod_math_Th_E*, *Kod_aithousas*, Period_didaskal, Day, Hour)

Πίνακας 4.2: Συμπαγής μορφή του σχήματος της Βάσης Δεδομένων

4.1.2 Παραγωγή Σχεσιακού Μοντέλου από το Μοντέλο Οντότητα-Συσχέτιση

Το τελικό βήμα στο σχεδιασμό της βάσης δεδομένων είναι η απεικόνιση των σχέσεων του E-R Μοντέλου στο σχεσιακό. Και τα δυο μοντέλα χρησιμοποιούν την ιδέα της σχέσης (μορφή πίνακα). ***Value sets/σύνολα τιμών*** του E-R μοντέλου αντιστοιχούν σε ***domains/πεδία*** του σχεσιακού μοντέλου. ***Attributes/ιδιότητες*** του E-R μοντέλου μεταφέρουν σημασιολογική έννοια, ενώ χαρακτηριστικά/ιδιότητες του σχεσιακού μοντέλου χρησιμοποιούνται για να ξεχωρίσουν/διακρίνουν πεδία με το ίδιο όνομα στην ίδια σχέση.

Για τον έλεγχο αν η σχεσιακή βάση δεδομένων είναι σωστά σχεδιασμένη, χρησιμοποιούμε τους ***κανόνες κανονικοποίησης***.

Πρώτη κανονική μορφή: Η πρώτη κανονική *μορφή* δηλώνει ότι σε κάθε διασταύρωση γραμμών και στηλών στον πίνακα, υπάρχει μία τιμή και ποτέ μια λίστα από τιμές. Για παράδειγμα, δεν είναι δυνατόν στο πεδίο που ονομάζεται "ΑΜ" να είναι τοποθετημένες περισσότερες από μία τιμές [http://office.microsoft.com/el-gr/access/HA012242471032.aspx].

Δεύτερη κανονική μορφή: Η δεύτερη κανονική *μορφή,* απαιτεί η σχέση να είναι στην πρώτη κανονική μορφή και επιπλέον κάθε στήλη χωρίς κλειδί να είναι πλήρως εξαρτημένη από ολόκληρο το πρωτεύον κλειδί και όχι απλώς από ένα μέρος του. Αυτός ο κανόνας εφαρμόζεται όταν το πρωτεύον κλειδί αποτελείται από περισσότερες από μία στήλες [http://office.microsoft.com/el-gr/access/HA012242471032.aspx].

Τρίτη κανονική μορφή: Η τρίτη κανονική *μορφή* απαιτεί να είναι στη δεύτερη κανονική μορφή και επιπλέον κάθε στήλη χωρίς κλειδί πρέπει όχι μόνον να εξαρτάται από ολόκληρο το πρωτεύον κλειδί, αλλά να είναι ανεξάρτητη από όλες τις άλλες στήλες [http://office.microsoft.com/el-gr/access/HA012242471032.aspx].

Επιπλέον, στο άρθρο [Chen, 1976], αποδεικνύεται ότι οι σχέσεις οντοτήτων και συσχετίσεων του E-R μοντέλου είναι όμοιες με τις σχέσεις του σχεσιακού μοντέλου που βρίσκονται στην $3^{η}$ κανονικοποιημένη μορφή. Επιπλέον, η σημασιολογία των σχέσεων του E-R μοντέλου είναι πιο ευκρινής. Απ' την παραπάνω συζήτηση προκύπτει ότι ο μετασχηματισμός των σχέσεων οντοτήτων και συσχετίσεων του E-R μοντέλου σε σχέσεις του σχεσιακού μοντέλου, είναι μια απεικόνιση 1:1. Ο πίνακας 4.1 παρουσιάζει το σχήμα της βάσης δεδομένων σε συμπαγή μορφή. Αυτό θα βοηθήσει τον αναγνώστη να έχει μια συνολική εικόνα και άποψη του σχήματος της Βάσης Δεδομένων.

4.2 Το Λεξικό της Ελληνικής γλώσσας

Το Λεξικό της ελληνικής γλώσσας περιέχει γραμματικά στοιχεία των λέξεων. Το λεξικό της Ελληνική γλώσσας του συστήματος EROTISIS, περιέχει λέξεις οι οποίες σχετίζονται με το πεδίο προβλήματός μας, που είναι το τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων του ΤΕΙ Ηρακλείου Κρήτης. Συνεπώς, τα **ουσιαστικά** τα οποία συναντάμε πρέπει να έχουν σχέση με την περιγραφή του "κόσμου" μας. Για παράδειγμα, εμφανίζονται το ουσιαστικό με το άρθρο τους, όπως "ο καθηγητής" ή "τα μαθήματα", ενώ δεν εμφανίζονται ουσιαστικά τα οποία δεν παρέχουν πληροφορίες για την περιγραφή του ΤΕΙ, όπως "ο ουρανός". Ακόμα, τα ουσιαστικά του λεξικού της Ελληνικής γλώσσας αντιστοιχούν και στα χαρακτηριστικά/ιδιότητες των οντοτήτων. Για παράδειγμα, τα ουσιαστικά "όνομα", "επίθετο" και "τηλέφωνο" είναι χαρακτηριστικά πεδία των οντοτήτων "φοιτητές" και "καθηγητές". Τα **ρήματα** που χρησιμοποιήσαμε στο λεξικό δεδομένων επιλέχθηκαν με αντίστοιχο κριτήριο. Πρέπει δηλαδή, να παρουσιάζονται σε προτάσεις οι οποίες αναφέρονται στο πεδίο προβλήματός μας. Μερικά από αυτά τα ρήματα είναι "διδάσκει", "παρακολουθεί", "επιβλέπει". Προφανώς, ρήματα όπως "κολυμπάω", "κοιτάζω" κ.α., δεν έχουν συμπεριληφθεί στο λεξικό της Ελληνικής γλώσσας. Οι **ερωτηματικές αντωνυμίες**, όπως "ποιός", "ποιοί", "ποιό" και πολλές άλλες οι οποίες ξεκινούν με τη ρίζα "ποι-" είναι οι λέξεις με τις οποίες ο χρήστης ξεκινάει τις ερωτήσεις του. **Επιθετικοί προσδιορισμοί** έχουν προστεθεί επίσης, ώστε να προσδιορίζονται ιδιότητες σε κάποια ουσιαστικά, όπως για παράδειγμα, "το *εργαστηριακό* μάθημα". Δε χρησιμοποιούνται επιθετικοί προσδιορισμοί για την περιγραφή προσώπων, όπως "ο *αυστηρός* καθηγητής" ή ο "ο *καλός* φοιτητής".

Έχει χρησιμοποιηθεί επίσης η πρόθεση "με", ώστε να μπορεί ο χρήστης να προσδώσει ένα χαρακτηριστικό σε μια οντότητα. Για παράδειγμα "Ποιό μάθημα διδάσκει ο καθηγητής *με* όνομα μανόλης".

4.2.1 Η Υλοποίηση του Λεξικού της Ελληνικής Γλώσσας: Λεξικό Κανόνων DCG

Το λεξικό κανόνων DCG περιέχει γραμματικά στοιχεία για τις λέξεις. Αυτά τα στοιχεία περιγράφονται σε DCG κανόνες. Η διαφορά ανάμεσα στα δυο αυτά επίπεδα είναι ότι στο λεξικό κανόνων DCG προστίθενται συνεχώς νέες λέξεις, σε αντίθεση σε το συντακτικό, όπου οι κανόνες είναι σταθεροί. Διαφορετικές περιοχές του συντακτικού, μπορούν να αλληλεπιδράσουν με διαφορετικά λεξικά, ενώ το λεξικό αλληλεπιδρά μόνο με ορισμένες λέξεις, όπως ουσιαστικό, ρήμα, επίθετο κ.α.. Για τους λόγους αυτούς διαχωρίζουμε το συντακτικό και το λεξικό κανόνων DCG σε δυο επίπεδα.

Στην πορεία της υλοποίησης του σχεσιακού μοντέλου, παρατηρήσαμε ότι χρειάζονταν και άλλες προσθήκες στο λεξικό κανόνων DCG, οι οποίες παρόλα αυτά δεν επηρεάζουν το φυσικό σχήμα της βάσης δεδομένων, αλλά είναι απαραίτητοι για την υλοποίηση του συστήματος. Πρέπει να αναφερθεί σε αυτό το σημείο, πως δεν έχουν αξιοποιηθεί όλα οι πρόσθετα γεγονότα, αλλά αποτελούν σημαντικό εργαλείο σε κάποια προσπάθεια επέκτασης του συστήματος.

Η χρήση του Λεξικού Ελληνικής Γλώσσας εφαρμόζεται καθ' όλη τη διαδικασία του parsing. Αντί ολόκληρης λέξης χρησιμοποιείται η ρίζα της. Για παράδειγμα στο επίπεδο του *Λεξικού Πεδίου Προβλήματος*, όπου αντιστοιχούνται οι οντότητες με τις ενέργειές τους, θα έπρεπε σε περίπτωση που δεν υπήρχε το Λεξικό Ελληνικής Γλώσσας να καταχωρηθούν όλοι οι συνδυασμοί οντοτήτων και ενεργειών, παρουσία ολόκληρων των λέξεων. Το παρόν επίπεδο αυτό δεν λειτουργεί ανεξάρτητα από τα υπόλοιπα, αλλά καλείται όπου κρίνεται απαραίτητο.

Είδαμε πως μπορούμε στο συντακτικό να προσθέσουμε επιπλέον όρους ώστε

να συμφωνούν τα στοιχεία σε αριθμό, γένος κ.α.. Αυτές οι *γραμματικές κατηγορίες* μπορούν να προστεθούν και στο λεξικό. Ενδεικτικά αναφέρουμε κάποιες γραμματικές κατηγορίες: γένος, αριθμός, χρόνος. Στην υλοποίηση του συστήματος, έχουν χρησιμοποιηθεί το γένος και ο αριθμός. Το λεξικό κανόνων DCG, δηλαδή, περιέχει εκτός από τη λέξη η οποία θα αντιστοιχηθεί με τη λέξη της πρότασης, πληροφορίες για το γένος και τον αριθμό.

```
ousiastiko_leksi(ousiastiko_leksi(καθηγητής), enikos, arseniko) -->
                    [καθηγητής].
ousiastiko_leksi(ousiastiko_leksi(καθηγητές), plithintikos, arseniko) -->
                    [καθηγητές].
ousiastiko_leksi(ousiastiko_leksi(καθηγήτρια), enikos, thiliko) -->
                     [καθηγήτρια].
```

Πρόγραμμα 4.1: Μέρος DCG κανόνων για ousiastiko/3.

Στα παραπάνω γεγονότα, περιγράφεται πως το ousiastiko_leksi αποτελείται από μια δομή του ousiastiko_leksi(καθηγητής), το οποίο είναι σε αριθμό ενικό και γένος αρσενικό. Στο δεξί μέρος ορίζεται η λέξη. Ομοίως στο δεύτερο γεγονός, βλέπουμε την ίδια λέξη να ορίζεται σε διαφορετικό αριθμό, αλλά στο ίδιο γένος. Για να περιγράψουμε το θηλυκό σε αριθμό ενικό, αρκεί να γράψουμε την τρίτη πρόταση. Στο λεξικό περιγράφονται ομοίως όλα τα γραμματικά μέρη του λόγου τα οποία είναι πιθανό να συναντήσουμε στη συντακτική ανάλυση μιας πρότασης. Κάτι το οποίο είναι φυσικό, τη στιγμή κατά την οποία για να είναι συντακτικά και γραμματικά σωστή η πρόταση, πρέπει ορισμένοι όροι να συμφωνούν μεταξύ τους στις γραμματικές κατηγορίες. Για παράδειγμα, το άρθρο με το ουσιαστικό το οποίο συνοδεύει δεν πρέπει να διαφέρουν στους γραμματικούς όρους. Σε αντίθεση όμως με το ρήμα που ίσως να ακολουθεί μετά, καθώς το ρήμα δεν έχει γένος αλλά μόνο αριθμό ως γραμματική κατηγορία. Η πρόταση "ποιά μαθήματα διδάσκει ο καθηγητής Μαρακάκης" έχει το ουσιαστικό "μαθήματα" σε πληθυντικό αριθμό, το ρήμα όμως είναι σε ενικό.

Έχουμε ήδη αναφέρει πως τα τερματικά σημεία του δένδρου ανάλυσης της

ερώτησης, οι λέξεις δηλαδή, αντιστοιχίζονται με τις αντίστοιχες σχέσεις στη βάση δεδομένων. Για παράδειγμα, για την ερώτηση "ποιος φοιτητής ολοκλήρωσε το μάθημα προγραμματισμός", για να προκύψει ο στόχος δημιουργούνται κάποια γεγονότα. Ένα από αυτά τα γεγονότα στην παραπάνω ερώτηση είναι foitites(A, B, C, D, E, F).

Για να επιτευχθεί κάτι τέτοιο, δημιουργήθηκε ο πίνακας greeklish_greek_rel_names ο οποίος δέχεται σαν ορίσματα το όνομα την σχέση και τη ρίζα της λέξης. Δηλαδή greeklish_greek_rel_names(foitites, 'φοιτητ'). Ο λόγος που επιλέξαμε το δεύτερο όρισμα να είναι η ρίζα της λέξης και όχι η λέξη αυτούσια, είναι ο εξής: στην περίπτωση που αντί της λέξης "φοιτητής" ζητούσαμε τον πληθυντικό αριθμό ("ποιοι φοιτητές ολοκλήρωσαν το μάθημα προγραμματισμός"), θα έπρεπε να υπάρχει ξεχωριστή καταχώρηση που να αντιστοιχεί τον πίνακα foitites με την λέξη "φοιτητές". Για να μην υπάρχουν στο σύστημα λοιπόν παραπάνω πληροφορίες, οι οποίες μάλιστα θα ήταν και *άχρηστες* εφόσον μπορούν να εξαχθούν από άλλες πληροφορίες, δημιουργήθηκε το Λεξικό της Ελληνικής Γλώσσας, το οποίο περιέχει τις λέξεις και τις ρίζες τους. Έτσι, ανεξάρτητα από τον αριθμό και το γένος μπορεί να δημιουργηθεί το κατάλληλο γεγονός που θα χρησιμοποιηθεί σαν στόχος της Prolog. Οι σχέσεις στο Λεξικό Ελληνικής Γλώσσας, εκτός από τα δυο ορίσματα που εκφράζουν την αντιστοιχία όνομα σχέσης και ρίζα λέξης, περιέχουν και έναν αύξοντα αριθμό, ο οποίος χρησιμοποιείται ως πρωτεύον κλειδί.

4.3 Λεξικό δεδομένων

Για να αντιστοιχηθούν οι λέξεις της πρότασης στις σχέσεις της βάσης δεδομένων, δημιουργήσαμε την σχέση greeklish_greek_rel_names/2, η οποία έχει δυο ορίσματα. Πρώτο όρισμα είναι το όνομα της σχέσης(π.χ.: foitites) και δεύτερο η ρίζα της αντίστοιχης λέξης στην πρόταση(π.χ.: φοιτητ). Γίνεται κατανοητό, πως για οποιαδήποτε τερματικό σημείο ελέγχεται αν υπάρχει η αντίστοιχη σχέση.
Δηλαδή: greeklish_greek_rel_names(foitites, 'φοιτητ').

Η σχέση greeklish_greek_attr_names/2 έχει και αυτή δυο πεδία. Το πρώτο πεδίο είναι το όνομα των πεδίων κάθε σχέσης, και το δεύτερο πεδίο είναι η ονομασία του πεδίου σε φυσική γλώσσα.
Δηλαδή: greeklish_greek_attr_names('Epitheto_foititi', 'επίθετο').

Η σχέση columns/3 περιέχει σαν πρώτο πεδίο τη σχέση, δεύτερο πεδίο το όνομα της ιδιότητας και τρίτο τον αριθμό της μέσα στη σχέση. Η σχέση αυτή χρησιμοποιείται στην εξόρυξη των σωστών δεδομένων.
Δηλαδή: columns(foitites, 'Epitheto_foititi', 2).

Για να εισάγουμε τα κλειδιά, δημιουργήθηκε η σχέση key. Τα πεδία της σχέσης είναι ο κωδικός κλειδιού, το όνομα της σχέσης, ο αριθμός πεδίου, το αν είναι πρωτεύον ή δευτερεύον κλειδί, το όνομα πεδίου. Έχουν προστεθεί και καποια επιπλέον πεδία, τα οποία δεν έχουν τιμές (έχουν οριστεί με αρχικές τιμές μεταβλητές). Σε περίπτωση που ο χρήστης επιθυμεί να προσθέσει πεδία, είναι δυνατή η αντικατάσταση των μεταβλητών αυτών με τις επιθυμητές τιμές.
Δηλαδή: key('K1', foitites, 1, primary, 'AM', '_', '_', '_').
Στη παραπάνω σχέση, αν ο χρήστης επιθυμεί, μπορεί να αφαιρέσει μια μεταβλητή και να την αντικαταστήσει με τη τιμή μου θέλει.

Η σχέση abbr_attr/2 δίνει τις συντομογραφίες. Το πρώτο πεδίο είναι η συντομογραφία και το δεύτερο η πλήρης ονομασία. Για παράδειγμα

abbr_attr('ΑΜ', 'Αριθμός Μητρώου').

abbr_attr('ΝΚ', 'Νέο κτίριο').

abbr_attr('Εργ', 'Εργαστήριο').

Η σχέση attr_vsets/4 δίνει τα πεδία τιμών των πεδίων της κάθε σχέσης. Πρώτο πεδίο/όρισμα είναι το όνομα του πεδίου, δεύτερο όρισμα/πεδίο ο αναμενόμενος τύπος δεδομένων και τρίτο όρισμα το πεδίο τιμών. Για παράδειγμα, attr_vsets('Tel', 'String', 'ALL'). Η στήλη έχει όνομα Tel, τα δεδομένα είναι τύπου String και είναι δεκτή οποιαδήποτε τιμή(ALL).
attr_vsets('tel_grafeiou', 'Integer(5)', '10000 - 99999'). Η σχέση έχει τιμή 'tel_grafeiou', τα δεδομένα είναι τύπου Integer με 5 στοιχεία και είναι δεκτά τα νούμερα από 10000 έως 99999.

Τέλος, η σχέση synonyms περιέχει τα συνώνυμα. Έχει τρία ορίσματα/πεδία, τα δύο πρώτα είναι τα συνώνυμα, και τρίτο όρισμα η λέξη σε φυσική γλώσσα. π.χ.: synonyms('kod_mathimatos', 'Onoma_mathimatos', 'μάθημα').
synonyms('kod_aithousas', 'Onoma_aithousas', 'αίθουσα').

4.4 Διεπικοινωνία Συστήματος

Όπως έχουμε αναφέρει στο κεφάλαιο 2.4 ο χρήστης του συστήματος EROTISIS επικοινωνεί με αυτό μέσω ενός παραθυρικού περιβάλλοντος (**G**raphical **U**ser **I**nterface) το οποίο έχει υλοποιηθεί σε γλώσσα Java. Η διεπικοινωνία του συστήματος εκτελεί τις παρακάτω λειτουργίες διεπικοινωνίας.

1. Εκκινεί την γλώσσα προγραμματισμού Prolog μέσω της Java.
2. Φορτώνει το σύστημα που υλοποιήθηκε σε Prolog.
3. Δίνει την δυνατότητα στον χρήστη να εισάγει την ερώτηση της οποίας επιθυμεί την απάντηση.
4. Μετατρέπει την ερώτηση του χρήστη, ερώτηση στην Ελληνική, σε στόχο/ερώτηση Prolog και την περνά στο τμήμα "Συντακτικής Ανάλυσης Πρότασης".
5. Εκτυπώνει στην οθόνη του χρήστη τα αποτελέσματα του υποσυστήματος "Κατασκευής Prolog Ερωτήσεων" και τα αποτελέσματα του υποσυστήματος "Επεξεργασία των Prolog στόχων/ερωτήσεων και συλλογή αποτελεσμάτων".
6. Παρέχει την δυνατότητα εισαγωγής νέας ερώτησης από τον χρήστη.
7. Τέλος δίνει την δυνατότητα στον χρήστη να εξέλθει από το πρόγραμμα.

Για την εκτέλεση της **πρώτης λειτουργίας**, αρχικά εισάγουμε στην Java την βιβλιοθήκη jasper. Η βιβλιοθήκη αυτή περιέχει τις απαραίτητες μεθόδους με τις οποίες η Java επικοινωνεί με την Prolog.

Δεύτερη λειτουργία είναι να φορτωθούν τα απαραίτητα αρχεία τα οποία απαρτίζουν το σύστημα EROTISIS. Τα αρχεία αυτά είναι σε μορφή .pl ενώ το κυρίως αρχείο (το οποίο περιλαμβάνει το κατηγόρημα που δημιουργείται μέσω

της Java) πρέπει να μετατραπεί σε αρχείο με κατάληξη .sav, δηλαδή σε αρχείο java. Για να επιτευχθεί αυτό χρησιμοποιούμε την εντολή του προγράμματος.

```
?- compile(final),save_program('final.sav').
```

Πρόγραμμα 4.2: Εντολή μετατροπής αρχείου .pl σε .sav

Το αρχείο final είναι το **κύριο αρχείο** του συστήματος μας.

Χρησιμοποιείται η εντολή **load** στο πρόγραμμα Πρόγραμμα 4.3 για να φορτωθούν τα αρχεία. Το όρισμα που δέχεται είναι η διαδρομή για τα απαραίτητα αρχεία.

```
sicstus.load("C:\\EROTISIS\\trexousa_protasi.pl");
sicstus.load("C:\\ROTISIS\\vasi2.pl");
sicstus.load("C:\\EROTISIS\\apantisi_write.txt");
sicstus.load("C:\\EROTISIS\\stoxos_write.txt");
sicstus.restore("C:\\EROTISIS\\final.sav");
```

Πρόγραμμα 4.3: Οι εντολές sicstus.load

Τρίτη λειτουργία είναι η δημιουργία της κλάσης Interface, η οποία αρχικά δημιουργεί το GUI. Δίνεται έτσι στον χρήστη η ικανότητα να εισάγει κάποια ερώτηση σε φυσική γλώσσα. Το GUI περιλαμβάνει τρία TextAreas. Στο πρώτο εισάγεται η ερώτηση, στο δεύτερο εμφανίζεται ο στόχος σε Prolog και στο τρίτο εμφανίζεται η απάντηση σε ελληνική γλώσσα. Περιέχει επίσης το κουμπί " Δημιουργία απάντησης " για την εμφάνιση των απαντήσεων(βλ. Παράρτημα Ζ).

Η **τέταρτη λειτουργία** μετατρέπει την ερώτηση του χρήστη, ερώτηση στην Ελληνική, σε στόχο/ερώτηση Prolog. Ο στόχος/ερώτηση περνάει στο τμήμα "Συντακτικής Ανάλυσης Πρότασης".

Η κύρια μέθοδος της λειτουργίας αυτής είναι η **executeProlog** η οποία εκτελεί τα παρακάτω

1. Δημιουργεί ένα instance του αντικειμένου Sicstus καλώντας την μέθοδο getInstance (γραμμή 4), την οποία περιγράψαμε παραπάνω.
2. φορτώνει τα αρχεία του αντικειμένου Sicstus καλώντας την μέθοδο loadFile (γραμμή 5) την οποία περιγράψαμε παραπάνω.
3. Παίρνει την είσοδο – ερώτηση του χρήστη και την αποθηκεύει σαν συμβολοσειρά (string) (γραμμή 6).
4. Δημιουργεί τον στόχο (γραμμή 7) ο οποίος θα εκτελεστεί στην Prolog. Ο στόχος αυτός έχει σαν όρισμα την παραπάνω συμβολοσειρά.
5. Μέσω του instance της Prolog μας παρέχεται μία μέθοδος query (γραμμή 8) από το jasper η οποία μας επιτρέπει να τρέξουμε τον παραπάνω στόχο.

Πρόγραμμα 4.4: Ψευδοκώδικας περιγραφής μεθόδου executeProlog()

```
1       private void executeProlog(Container c) {
2               BufferedReader reader = null;
3               try {
4                       sicstus = getInstance();
5                       loadFiles(sicstus);
6               String input = new String(goal.getText().getBytes(), "ISO-
8859-1");
7               String preparedGoal = "start('" + input + "').";
8           if ( sicstus.query(preparedGoal, new HashMap()) ) {
9               File file = new File(APANTISI);
10              File file2 = new File(APANTISI_STOXOS);
11              reader = new BufferedReader(new InputStreamReader(new
        FileInputStream(file), "ISO-8859-7"));
12              String line = reader.readLine();
13              String line2 = reader2.readLine();
14              if (line != null) {
15                      result.setText(line);
16              }else {
17                  logErrorMessage("The file that supposed to contain the
answer was found 16
        empty");
18                  System.exit(1);
19              }
20              }else {
21                              JOptionPane pane = new JOptionPane();
22                              pane.showMessageDialog(c, "control + shift + J");
23          }
24    } catch (Exception ex) {
25          showErrorDialaog(c);
```

```
26          logError(ex);
27          System.exit(1);
28       }finally {
29          if (reader != null) {
30             try {
31                reader.close();
32             } catch (Exception ex) {
33                logErrorMessage("Unable to close the file reader.");
34                showErrorDialaog(c);
35
36             }
37          }
38       }
39    }
```

Πρόγραμμα 4.5: Μέθοδος executeProlog()

Πέμπτη λειτουργία είναι η εκτύπωση του αποτελέσματος του υποσυστήματος "Επεξεργασία των Prolog στόχων/ερωτήσεων και συλλογή αποτελεσμάτων". Σύμφωνα με το Πρόγραμμα 4.5, τα αποτελέσματα της εκτέλεσης του στόχου αποθηκεύονται από την Prolog σε δύο ξεχωριστά αρχεία σε μορφή συμβολοσειράς (string). Οι συμβολοσειρές αυτές αποθηκεύονται σε μεταβλητές με ονόματα file και file2 μέσω των ακόλουθων εντολών "File file = new File(APANTISI);" (γραμμή 9) και "File file2 = new File(APANTISI_STOXOS);" (γραμμή 10). Με την χρήση των εντολών "result.setText(line);"(γραμμή 13) και " String line2 = reader2.readLine(); " (γραμμή 14), επιστρέφονται οι απαντήσεις στον χρήστη.

Η **έκτη λειτουργία** είναι η δημιουργία νέας ερώτησης σε φυσική γλώσσα. Το παραθυρικό περιβάλλον έχει ακόμα ένα κουμπί ("Νέα ερώτηση σε φυσική γλώσσα"), το οποίο δίνει στον χρήστη τη δυνατότητα να εισάγει μια νέα ερώτηση σε φυσική γλώσσα.

Τέλος, η **έβδομη λειτουργία**, δηλαδή η δυνατότητα του χρήστη να εξέρχεται από το πρόγραμμα, παρέχεται μέσω του MenuIBar Αρχείο το οποίο περιέχει ένα MenuItem *Έξοδος* . Το MenuItem τερματίζει το πρόγραμμα.

4.5 Συντακτική Ανάλυση Προτάσεων

Ορισμός 4.1: Ένα κείμενο/λόγος ολοκληρωμένος, που αποτελείται από μια ή περισσότερες προτάσεις και καταλήγει, όταν είναι γραπτός σε τελεία, ονομάζεται ***περίοδος*** [Συντακτικό Ελληνικής].

Ορισμός 4.2: Γραμματική ή τυπική γραμματική είναι ο ακριβής ορισμός για το ποιες ακολουθίες λέξεων ή συμβόλων ανήκουν σε κάποια γλώσσα.

Το συντακτικό επίπεδο προτάσεων ασχολείται με την σύνταξη των περιόδων. Όλες οι γραμματικές δίνουν *κανόνες φραστικής δομής (phrase structure rules).* Κανόνες δηλαδή για το πώς συντάσσεται μια περίοδος. Οι κανόνες αυτοί μπορεί να είναι συμφραστικά ανεξάρτητοι (context free rules). Έχουν όμως βρεθεί περιπτώσεις εκφράσεων της φυσικής γλώσσας που απαιτούν εξαρτημένους κανόνες για την παραγωγή τους. Αυτούς τους εξαρτημένους κανόνες πρέπει να τηρεί μια πρόταση φυσικής γλώσσας, τα στοιχεία της οποίας μετατρέπονται σε μια ιεραρχημένη δομή, η οποία ανταποκρίνεται στη διασύνδεση των δομικών στοιχείων της πρότασης. Η διαδικασία αυτή ονομάζεται *συντακτική ανάλυση* (parsing). Η ανάλυση μιας πρότασης μπορεί να δίνει πολλαπλά πιθανά αποτελέσματα.

Μια **γραμματική** ορίζεται σαν μια συλλογή από γραμματικούς κανόνες. Αυτοί ονομάζονται **rewrite κανόνες**, επειδή δείχνουν πως μπορούμε να ξαναγράψουμε ένα αντικείμενο του κανόνα σαν κάτι άλλο.

Σε μια τυπική πρόταση της ελληνικής, ένας κανόνας θα είχε την εξής μορφή:

Πρόταση --> υποκείμενο ρήμα αντικείμενο

Αυτός ο κανόνας μας δείχνει ότι μια **πρόταση** της Ελληνικής γλώσσας θα μπορούσε να κατασκευαστεί από ένα **υποκείμενο** το οποίο ακολουθείται από ένα **ρήμα**, το οποίο στη συνέχεια ακολουθείται από το **αντικείμενο**.

Μια απλή μορφή δομής είναι ένα *συντακτικό δέντρο* (parse tree).

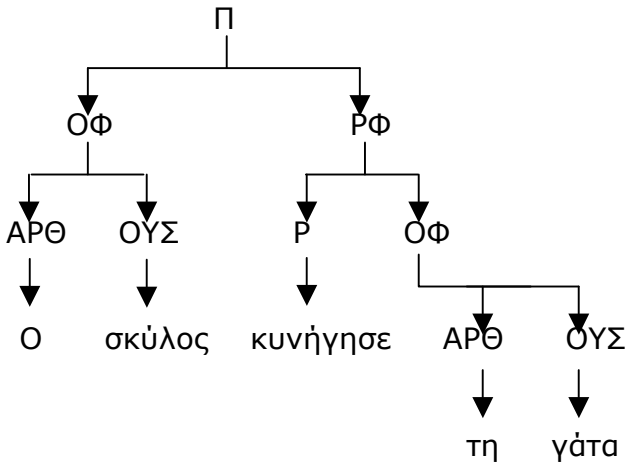
Ορισμός 4.3: ***Συντακτικό δένδρο (parse tree)*** είναι η σχηματική αναπαράσταση της συντακτικής ανάλυσης μιας πρότασης που προκύπτει από κάποιους κανόνες σύνταξης.

Για παράδειγμα, έστω η πρόταση, "ο σκύλος κυνήγησε τη γάτα". Η γραμματική Κανόνας 4.1 αναγνωρίζει συντακτικά αυτή την πρόταση.

Π → ΟΦ, ΡΦ.
ΟΦ → ΑΡΘ, ΟΥΣ.
ΡΦ → Ρ, ΟΦ.

Κανόνας 4.1: Απλοί DCG κανόνες της Ελληνικής γλώσσας.

Στους κανόνες Κανόνας 4.1, το **Π** σημάνει **Π**ρόταση, **ΟΦ** σημαίνει **Ο**νοματική **Φ**ράση, **ΡΦ** σημαίνει **Ρ**ηματική **Φ**ράση, **ΑΡΘ** σημαίνει **ΑΡΘ**ρο, **ΟΥΣ** σημαίνει **ΟΥΣ**ιαστικό και **Ρ** σημαίνει **Ρ**ήμα.



Σχήμα 4.3: Συντακτικό δένδρο για τη πρόταση "Ο σκύλος κυνήγησε τη γάτα"

Η Prolog διαθέτει μια συμβολική επέκταση η οποία ονομάζεται DCG (**D**efinite **C**lause **G**rammar), η οποία επιτρέπει την άμεση υλοποίηση τυπικών

γραμματικών.

Το σωστό συντακτικό, πρέπει να αναγνωρίζει τις σωστές προτάσεις ("ο σκύλος κυνήγησε τη γάτα"), να απορρίπτει τις λάθος ("ο κυνήγησε σκύλος τη γάτα") και να αποδίδει συντακτικές δομές στις προτάσεις [(ο σκύλος) (κυνήγησε τη γάτα)]. Η τεχνική κατασκευής γραμματικής που χρησιμοποιήθηκε είναι η **Γραμματική Ορισμένης Φράσεων (Definite Clause Grammar)**.

Το παράδειγμα κανόνων DCG 4.2 παρουσιάζει τη σύνταξη πρότασης σε κανόνες DCG οι οποίοι είναι αποδεκτοί από τη Prolog.

(οφ->ονοματική φράση, ρφ->ρηματική φράση, αρθ->άρθρο, ουσ->ουσιαστικό, ρ->ρήμα)

```
sentence--> οφ, ρφ.
οφ --> αρθ, ουσ.
ρφ --> ρ, οφ.
αρθ --> [το].
ουσ --> [παιδί] ; [μήλο].
ρ --> [έφαγε].
```

Πρόγραμμα 4.6: Κανόνες DCG για τη πρόταση "το παιδί έφαγε το μήλο"

Στο παραπάνω παράδειγμα, αναφέρουμε σε γλώσσα Prolog ότι η πρόταση (sentence), περιέχει την ονοματική φράση (οφ), η οποία ακολουθείται από την ρηματική φράση (ρφ) της πρότασης. Η ονοματική φράση (οφ) είναι ένα άρθρο (αρθ) μαζί με ένα ουσιαστικό (ουσ), η ρηματική φράση (ρφ) είναι ένα ρήμα που μαζί με μια ονοματική φράση (οφ). Ο τέταρτος κανόνας λέει ότι το άρθρο είναι η λέξη "το". Ο πέμπτος κανόνας ότι το ουσιαστικό είναι η λέξη "παιδί" ή η λέξη "μήλο". Τέλος, ο έκτος κανόνας λέει πως το ρήμα είναι η λέξη "έφαγε".
Οι παραπάνω δηλώσεις γίνονται με τη χρήση του τελεστή "**-->**" .

Για να εκτελέσουμε το συντακτικό του προγράμματος Πρόγραμμα 4.7, θα χρησιμοποιήσουμε την built-in συνάρτηση **phrase/2**. Η συνάρτηση αυτή έχει 2 ορίσματα. Το δεύτερο είναι η **πρόταση που θέλουμε να ελέγξουμε** και το **πρώτο είναι το όνομα της κατηγορίας που έχει η γραμματική που έχει δηλωθεί**.

Για παράδειγμα:

```
?- phrase(sentence, [το, παιδί, έφαγε, το, μήλο]).
yes

?- phrase(sentence, A).
A = [το, παιδί, έφαγε, το, παιδί] ;
A = [το, παιδί, έφαγε, το, μήλο] ;
A = [το, μήλο, έφαγε, το, παιδί] ;
A = [το, μήλο, έφαγε, το, μήλο]
yes
```

Οι προτάσεις οι οποίες δημιουργήθηκαν είναι συντακτικά σωστές. Η σημασιολογική τους ανάλυση προκαλεί προβληματισμό, αλλά δεν είναι αντικείμενο του παρόντος υποκεφαλαίου.

Είναι δυνατή η χρήση των αγκυλών {}, για να προσθέσουμε επιπλέον κώδικα σε Prolog αν κρίνεται απαραίτητο. Στους DCG κανόνες του συστήματός μας (βλ. Παράρτημα Δ.2), έχουμε δηλώσει τον εξής κανόνα:

```
protasi(prot(RIM_MEROS, ONOM_MEROS)) -->
                rimatiko_meros(RIM_MEROS,Arithmos),
                onomatiko_meros(ONOM_MEROS, Arithmos1).
```

Ο κανόνας αυτός μας λέει πως από την **πρόταση**(“protasi”) κατασκευάζεται το **ρηματικό**(“rimatiko_meros(RIM_MEROS,Arithmos), ”) και το **ονοματικό**

μέρος(“onomatiko_meros(ONOM_MEROS, Arithmos1).”). Αυτά τα μέρη αποτελούν την είσοδο στο σημασιολογικό επίπεδο και στο υποσύστημα κατασκευής Prolog ερωτήσεων. Αν στον παραπάνω κώδικα προσθέσουμε το:

```
{elegxos(RIM_MEROS, ONOM_MEROS), find_target(ONOM_MEROS,
RIM_MEROS)}.
```

θα έχουμε:

```
protasi(prot(RIM_MEROS, ONOM_MEROS)) -->
        rimatiko_meros(RIM_MEROS,Arithmos),
        onomatiko_meros(ONOM_MEROS, Arithmos1),
        {elegxos(RIM_MEROS, ONOM_MEROS),
        find_target(ONOM_MEROS, RIM_MEROS)}.
```

Ο παραπάνω κανόνας μας εξηγεί τη χρήση των αγκυλών. **Μόλις από την πρόταση κατασκευαστεί το ονοματικό και το ρηματικό μέρος, τα αποτελέσματα θα περάσουν ως ορίσματα πρώτα στο κατηγόρημα elegxos/2**(σημασιολογικό επίπεδο) **και αν η εκτέλεσή του είναι επιτυχής, στο κατηγόρημα find_target/2**(υποσύστημα κατασκευής Prolog ερωτήσεων).

Η χρήση των [] μαρτυρά την παρουσία τερματικού συμβόλου για το συντακτικό.

Για παράδειγμα, έστω ότι έχουμε τους κανόνες:

```
α --> β, α.
α --> [].
```

Στον πρώτο κανόνα η δομή **α** κατασκευάζεται/αποτελείται από τη δομή **β** και ακολουθεί η δομή **α**. Η δομή **α** στον δεύτερο κανόνα τερματίζει. Αυτό δηλώνεται με τη κενή λίστα [].

Είναι προφανές ότι όσοι περισσότεροι κανόνες υπάρχουν, τόσο περισσότερες προτάσεις θα μπορούν να απαντηθούν.

Στο συντακτικό επίπεδο αυτής της πτυχιακής εργασίας, δημιουργήθηκε το συντακτικό της Ελληνικής Γλώσσας, το οποίο καλύπτει τις πιθανές ερωτήσεις που θα υποβληθούν στο σύστημά μας. Το ονοματικό και το ρηματικό μέρος των προτάσεων μπορούν να έχουν διάφορες μορφές. Ας δούμε το εξής παράδειγμα: "ποιο μάθημα διδάσκει ο καθηγητής Μαρακάκης". Έχουμε δηλώσει στο πρόγραμμα Πρόγραμμα 4.7 με τη χρήση του τελεστή '**-->** ', ότι η ΡΦ μπορεί να περιέχει μια ΑΝΤΩΝΥΜΙΑ, την οποία θα ακολουθεί ένα ΑΝΤΙΚΕΙΜΕΝΟ και ένα ΡΗΜΑ. Το ΑΝΤΙΚΕΙΜΕΝΟ θα μπορούσε να περιέχει ένα ουσιαστικό, σε συνδυασμό ίσως με κάποιον επιθετικό προσδιορισμό. Δηλαδή:

```
ΡΦ --> ΑΝΤΩΝΥΜΙΑ, ΑΝΤΙΚΕΙΜΕΝΟ, ΡΗΜΑ.
ΑΝΤΙΚΕΙΜΕΝΟ --> ΕΠΙΘΕΤΙΚΟΣ_ΠΡΟΣΔΙΟΡΙΣΜΟΣ, ΟΥΣΙΑΣΤΙΚΟ.
ΑΝΤΙΚΕΙΜΕΝΟ --> ΟΥΣΙΑΣΤΙΚΟ.

ΑΝΤΩΝΥΜΙΑ --> [ποιό].
ΕΠΙΘΕΤΙΚΟΣ_ΠΡΟΣΔΙΟΡΙΣΜΟΣ --> [εργαστηριακό].
ΟΥΣΙΑΣΤΙΚΟ --> [μάθημα].
ΡΗΜΑ → [παρακολουθεί] ; [διδάσκει].
```

Πρόγραμμα 4.7: Μέρος κανόνων DCG για τη πρόταση "ποιό μάθημα διδάσκει ο καθηγητής μαρακάκης".

Στην ανάλυση της **ΡΦ** ("ποιο μάθημα διδάσκει"), η οποία ανάλυση εκτελείται σειριακά, η πρώτη λέξη που συναντάται είναι το "ποιο". Το DCG περιμένει η πρώτη λέξη να είναι ΑΝΤΩΝΥΜΙΑ, και ψάχνοντας στο λεξικό κάνει την αντιστοιχία, οπότε η ανάλυση συνεχίζεται κανονικά. Μετά συναντάει την λέξη "μάθημα". Μετά την αντωνυμία έχουμε δηλώσει πως ακολουθεί το ΑΝΤΙΚΕΙΜΕΝΟ. Στην ανάλυση του αντικειμένου, ο πρώτος όρος που συναντάται είναι ο ΕΠΙΘΕΤΙΚΟΣ_ΠΡΟΣΔΙΟΡΙΣΜΟΣ. Ο ΕΠΙΘΕΤΙΚΟΣ_ΠΡΟΣΔΙΟΡΙΣΜΟΣ έχει μόνο μια τιμή, την τιμή "εργαστηριακό", η οποία δε συμφωνεί με τη λέξη της πρότασης. Οπότε, η πρώτη περίπτωση του ΑΝΤΙΚΕΙΜΕΝΟ αποτυγχάνει. Έχουμε δηλώσει όμως και μια δεύτερη περίπτωση, όπου το αντικείμενο μπορεί να είναι ένα απλό ΟΥΣΙΑΣΤΙΚΟ. Αυτή

η περίπτωση επιτυγχάνει, αφού το ΟΥΣΙΑΣΤΙΚΟ όπως έχει δηλωθεί στο λεξικό έχει την τιμή "μάθημα". Τέλος έρχεται το ΡΗΜΑ της πρότασης το οποίο έχει τιμή "διδάσκει". Σαν τερματικά σύμβολα υπάρχουν δυο τιμές: "παρακολουθεί" ή "διδάσκει". Η πρώτη περίπτωση θα αποτύχει, για να ακολουθήσει η ταυτοποίηση με την δεύτερη. Η χρήση του διαζευκτικού "ή", έγινε με τη χρήση του τελεστή ';'.

Συνεπώς, χρειάζεται να υπάρχει μια σειρά από διαφορετικές περιπτώσεις, όπου σε περίπτωση που αποτυγχάνει κάποια, θα εξετάζεται η επόμενη, μέχρι να βρεθεί η σωστή. Υπεύθυνος για την εύρεση της σωστής συντακτικής ανάλυσης είναι η TOP-down μέθοδος parsing.

Ορισμός 4.4: Ξεκινώντας από ένα αρχικό γραμματικό σύμβολο, όπως "sentence", στο πρόγραμμα Πρόγραμμα 4.7, εφαρμόζουμε τους γραμματικούς κανόνες, π.χ. τους κανόνες τους προγράμματος Πρόγραμμα 4.7 προς τα εμπρός, έως ότου τα σύμβολα στα τερματικά σημεία του συντακτικού δένδρου αντιστοιχηθούν με τα κατάλληλα μέρη της πρότασης [Rich, Knight, 1991]. Αυτή η συντακτική ανάλυση ονομάζεται ***top-down συντακτική ανάλυση.***

Παρατηρούμε ότι η δημιουργία κανόνων DCG σε γλώσσα Prolog γίνεται με εύκολο τρόπο, καθώς για τη δημιουργία τους δεν απαιτείται χρήση πολύπλοκων τελεστών και εντολών.

Η έξοδος από το συντακτικό επίπεδο είναι μια ***συντακτικά δομημένη πρόταση***. Η πρόταση αυτή περιέχει τις συντακτικές και τις γραμματικές δομές οι οποίες συναντήθηκαν μέχρι την εύρεση του τερματικού σημείου στο δένδρο ανάλυσης. Για παράδειγμα, η ερώτηση του χρήστη "ποιό μάθημα διδάσκει ο καθηγητής Μαρακάκης", θα μας δώσει την εξής συντακτικά δομημένη πρόταση:

```
protasi(rimatiko_meros(antonimia(ποιό),
                antikeimeno(arthro(_4327),ousiastiko_leksi(μάθημα)),
                rima(διδάσκει)),
```

onomatiko_meros(ipokeimeno(arthro(**ο**),
ousiastiko_leksi(**καθηγητής**),stoixeio(onoma(**μαρακάκης**)))))

Πρόγραμμα 4.8: Συντακτικά δομημένη πρόταση της ερώτησης "ποιό μάθημα διδάσκει ο καθηγητής Μαρακάκης".

Στο πρόγραμμα Πρόγραμμα 4.8, φαίνονται όλες οι δομές που συναντήθηκαν κατά τη συντακτική ανάλυση της εισαγόμενης πρότασης του χρήστη. Πιο αναλυτικά, αναφέρεται πως η πρόταση η οποία αναλύθηκε (protasi), περιέχει το ονοματικό μέρος (onomatiko_meros) και το ρηματικό μέρος (rimatiko_meros). Το ονοματικό μέρος περιέχει ένα αντικείμενο (**μάθημα**) και το ρήμα (**διδάσκει**). Η διαδικασία αυτή συνεχίζεται μέχρι να εντοπιστούν τα τερματικά σημεία του συντακτικού δένδρου, δηλαδή οι λέξεις οι οποίες ταυτοποιούνται από το λεξικό δεδομένων.

Η χρήση της συντακτικά δομημένης πρότασης έγκειται στο ότι με χρήση κατάλληλων built-in κατηγορημάτων (έτοιμων από την Prolog), μπορούμε να συλλέξουμε οποιαδήποτε πληροφορία μας είναι απαραίτητη.

4.6 Σημασιολογικό Επίπεδο

4.6.1 Σημασιολογική Ανάλυση Πρότασης

Η Σημασιολογική Ανάλυση Πρότασης (semantic analysis) είναι το επίπεδο της αρχιτεκτονικής του συστήματος το οποίο ελέγχει το νόημα της πρότασης. Προτάσεις οι οποίες δε συμφωνούν με το γενικώς αποδεκτό νόημα, όπως αυτό εκφράζεται από τις βάσεις πληροφοριών(λεξικό δεδομένων, βάση δεδομένων, λεξικό πεδίου προβλήματος) απορρίπτονται. Οι προτάσεις οι οποίες εισάγονται σε αυτό το επίπεδο, είναι συντακτικά σωστές. Το σημασιολογικό επίπεδο προσδίδει νόημα στις προτάσεις αυτές. Παράλληλα, βρίσκει τις σχέσεις που θα σχηματίσουν την τελική ερώτηση/στόχο σε Prolog.

4.6.1.1 Λεξιλογική Επεξεργασία

Σκοπός της σημασιολογικής ανάλυσης είναι να ελέγχει το νόημα της εισαγόμενης πρότασης. Άλλος ένας σημαντικός ρόλος είναι **να προβάλλει περιορισμούς για το ποιες για το ποιες αναπαραστάσεις μπορούν να δημιουργηθούν λόγω των ενώσεων των δομών που υπάρχουν μέσα στο σημασιολογικό και συντακτικό επίπεδο** [Rich, Knight, 1991]. Αυτό επιτυγχάνεται στο επίπεδο ***Λεξιλογικής Επεξεργασίας***. Το πρώτο βήμα σε ένα τέτοιο σύστημα σημασιολογικής επεξεργασίας είναι να αναζητήσει τις λέξεις ενός λεξικού και να εξάγει τη σημασία τους. Αυτό όμως δεν είναι απλή διαδικασία, καθώς οι λέξεις μπορούν να έχουν πολλά νοήματα. Παράδειγμα είναι η λέξη "άλογα". Μια σημασία αυτής της λέξης είναι φυσικά τα ζώα και μια άλλη η ιπποδύναμη του αυτοκινήτου. Για να γίνει η σημασιολογική ανάλυση της πρότασης "το αυτοκίνητο έχει 76 άλογα", πρέπει το σύστημα να γνωρίζει πως τα άλογα δεν είναι μέσα στο αυτοκίνητο και πως *εννοείται* η ιπποδύναμη.

Στο σύστημα EROTISIS, υπάρχουν προτάσεις οι οποίες αναλύονται με επιτυχία στο συντακτικό επίπεδο και αποτυγχάνουν στο επίπεδο της λεξιλογικής επεξεργασίας. Η πρόταση "ποιο μάθημα διδάσκει η φοιτήτρια κατσαμάκη;" *συντακτικά* είναι σωστή. Παρόλα αυτά, οι φοιτητές δεν διδάσκουν μαθήματα. Δηλαδή *σημασιολογικά* είναι λάθος. Πρέπει λοιπόν με κάποιο τρόπο να κάνουμε το σύστημα να καταλάβει πως η ερώτηση είναι λανθασμένη.

Η απάντηση έρχεται από το ER διάγραμμα. Το διάγραμμα αυτό περιέχει τις οντότητες και συσχετίσεις. Οι ***οντότητες*** στο σύστημά μας είναι ***φυσικές οντότητες, η κάθε μια εκ των οποίων είναι διακριτή στη φύση***. Αντίθετα, οι ***συσχετίσεις*** είναι τα ***ρήματα τα οποία συνδέουν τις οντότητες***. Η ανάλυση καθοθηγείται από ένα σύνολο προβλέψεων που

στηρίζονται στο **κυρίως ρήμα** της πρότασης. Μπορούμε λοιπόν, να αντιστοιχίσουμε τις *οντότητες* με τις *ενέργειές* τους. Έτσι, δημιουργήθηκε η **εννοιολογική εξάρτηση**, η οποία περιέχει τις αντιστοιχίσεις αυτές.

σχέση **ενέργειες_προσώπων**

Κλειδί	**Όνομα οντότητας**	**Ενέργεια**
1	καθηγητ	διδάσκ
2	καθηγητ	επιβλ
3	φοιτητ	παρακολουθ
4	Φοιτητ	εκπον

Η υλοποίηση σε Prolog της σχέσης ενέργειες προσώπων γίνεται από το πρόγραμμα Πρόγραμμα 4.9.

```
energeies_prosopon(καθηγητ, διδάσκ).
energeies_prosopon(καθηγητ, επιβλέπ).
energeies_prosopon(φοιτητ, παρακολουθ).
energeies_prosopon(φοιτητ, εκπον).
```

Πρόγραμμα 4.9: Εννοιολογική εξάρτηση: Αντιστοιχίες οντοτήτων με τις ενέργειές τους.

Οι ενέργειες, δηλαδή τα ρήματα της πρότασης, όπως επίσης και οι οντότητες, παρουσιάζονται με τις ρίζες τους αντί τη χρήση ολόκληρων των λέξεων. Η ρίζα της λέξης είναι αρκετή για να προσδιοριστεί η αντίστοιχη ενέργεια. Έτσι, είτε η λέξη στην πρόταση είναι "φοιτητής" είτε είναι "φοιτητές", το σύστημα EROTISIS αποκτά τη ρίζα και βρίσκει την αντίστοιχη ενέργεια.

4.6.1.2 Σημασιολογική Ανάλυση

Από τη συντακτική ανάλυση της πρότασης έχουμε τα δομικά στοιχεία όπως υποκείμενο-οντότητα, ρήμα-συσχέτιση, αντικείμενο-οντότητα. Με αυτά τα δομικά στοιχεία και χρήση του λεξικού δηλώνουμε τους κανόνες ως εξής:

1. Βρίσκουμε για την οντότητα-αντικείμενο ποια σχέση της βάσης δεδομένων συνδέεται με την οντότητα.
2. Βρίσκουμε για το ρήμα-συσχέτιση ποια σχέση της βάσης δεδομένων συνδέεται με το ρήμα.
3. Βρίσκουμε για την οντότητα-υποκείμενο ποια σχέση της βάσης δεδομένων συνδέεται με το ρήμα.
4. Εμπλουτίζουμε τις σχέσεις των βημάτων 1,2 και 3 με επιπλέον δομικά στοιχεία τα οποία περνάμε από την ερώτηση.
5. Διακρίνουμε τις εξής περιπτώσεις οι οποίες προσθέτουν στη λίστα των σχέσεων που έχει σχηματιστεί μέχρι τώρα για τη δημιουργία του τελικού στόχου νέες σχέσεις. Αυτές οι νέες σχέσεις εμπλουτίζονται με δομικά στοιχεία εφόσον υπάρχουν στην ερώτηση. *<u>Σημείωση:</u> η σχέση apoteleitai_apo/2 δεν εμπλουτίζεται με βάση τη παρούσα υλοποίηση. Η σχέση αυτή έχει δυο ορίσματα, τον κωδικό μαθήματος και τον κωδικό αναλυτικού μαθήματος. Κανένα από αυτά τα πεδία δεν θεωρείται ότι μπορεί να μπει ως είσοδος στην ερώτηση του χρήστη. Δηλαδή δε περιμένουμε ερώτηση παρόμοια με "ποιός καθηγητής διδάσκει το μάθημα με κωδικό τπ4004α".*
    a. Αν στην ερώτηση υπάρχει η λέξη "μάθημα", τότε προστίθενται οι νέες σχέσεις apoteleitai_apo/3 και leptomereies_mathimaton/3.
    b. Αν στην ερώτηση υπάρχει η λέξη "διπλωματική", τότε προστίθεται η νέα σχέση ptyxiakes/2.

Παράδειγμα 1: έστω η ερώτηση "ποιό μάθημα ολοκλήρωσε ο φοιτητής

Βαρουξής". Από τη συντακτική ανάλυση προκύπτουν τα εξής:

a. υποκείμενο-οντότητα είναι ο "φοιτητής"
b. ρήμα-συσχέτιση είναι το "ολοκλήρωσε"
c. αντικείμενο- οντότητα είναι το "μάθημα"

Ακολουθώντας τα παραπάνω βήματα κατασκευάζουμε την ερώτηση/στόχο ως εξής:

1. Από το λεξικό δεδομένων βρίσκει το σύστημα ότι στην οντότητα "φοιτητής" αντιστοιχεί η σχέση foitites/7. foitites(AM, Epitheto-foititi, Onoma-foititi, Genos, Eksamino_eisag, Tel, Mail).
2. Από το λεξικό δεδομένων βρίσκει το σύστημα ότι στο ρήμα "ολοκλήρωσε" αντιστοιχεί η σχέση exei_perasei/4. exei_perasei(AM, Kod_mathimatos, Vathmos, Eksamino).
3. Από το λεξικό δεδομένων βρίσκει το σύστημα ότι στην οντότητα "μάθημα" αντιστοιχεί η σχέση mathimata/5. mathimata(Kod_mathimatos, Onoma_mathimatos, Sem_mathimatos, Tupos_mathimatos, Did_monades).
4. Εμπλουτίζεται η σχέση foitites/7 με το "Βαρουξής" ώστε προκύπτει η νέα σχέση:
   foitites(AM, Βαρουξής, Onoma-foititi, Genos, Eksamino_eisag, Tel, Mail).
5. Επειδή στην ερώτηση υπάρχει η λέξη "μάθημα" δημιουργούνται οι δυο νέες σχέσεις apoteleitai_apo/3 και leptomereies_mathimaton/3. apoteleitai_apo(Kod_mathimatos, Kod_anal_mathimatos1, Kod_anal_mathimatos2).
   leptomereies_mathimaton(Kod_anal_mathimatos1,Typos_mathimatos, Ores).
   leptomereies_mathimaton(Kod_anal_mathimatos2,Typos_mathimatos, Ores).

Συνεπώς, οι τελικές σχέσεις που επιλέχθηκαν για τη δημιουργία της τελικής ερώτησης/στόχου σε Prolog είναι οι εξής:

foitites(AM, Epitheto-foititi, Onoma-foititi, Genos, Eksamino_eisag, Tel, Mail).

exei_perasei(AM, Kod_mathimatos, Vathmos, Eksamino).

mathimata(Kod_mathimatos, Onoma_mathimatos, Sem_mathimatos, Tupos_mathimatos, Did_monades).

foitites(AM, Βαρουξής, Onoma-foititi, Genos, Eksamino_eisag, Tel, Mail).

apoteleitai_apo(Kod_mathimatos, Kod_anal_mathimatos1, Kod_anal_mathimatos2).

leptomereies_mathimaton(Kod_anal_mathimatos1,Typos_mathimatos,Ores).

leptomereies_mathimaton(Kod_anal_mathimatos2,Typos_mathimatos,Ores).

Παράδειγμα 2: έστω η ερώτηση "ποιό εργαστηριακό μάθημα διδάσκει ο καθηγητής Αιβαλής". Από τη συντακτική ανάλυση προκύπτουν τα εξής:

a. υποκείμενο-οντότητα είναι ο "καθηγητής"
b. ρήμα-συσχέτιση είναι το "διδάσκει"
c. αντικείμενο- οντότητα είναι το "εργαστηριακό μάθημα"

Ακολουθώντας τα παραπάνω βήματα κατασκευάζουμε την ερώτηση/στόχο ως εξής:

1. Από το λεξικό δεδομένων βρίσκει το σύστημα ότι στην οντότητα "καθηγητής" αντιστοιχεί η σχέση kathigites/6. kathigites(Kod_kathigiti, Epitheto_kathigiti, Onoma_kathigiti, Thl_grafeiou, Kod_Grafeio_kathigiti, Mail_ kathigiti).
2. Από το λεξικό δεδομένων βρίσκει το σύστημα ότι στο ρήμα "διδάσκει" αντιστοιχεί η σχέση didaskoun/2. didaskoun(Kod_kathigiti, Kod_anal_mathimatos).
3. Από το λεξικό δεδομένων βρίσκει το σύστημα ότι στην οντότητα

"μάθημα" αντιστοιχεί η σχέση mathimata/5.

mathimata(Kod_mathimatos, Onoma_mathimatos, Sem_mathimatos, Tupos_mathimatos, Did_monades).

4. Εμπλουτίζεται η σχέση mathimata/5 με το "εργαστηριακό" ώστε προκύπτει η νέα σχέση:

mathimata(Kod_mathimatos, Onoma_mathimatos, Sem_mathimatos, Tupos_mathimatos, Did_monades).

5. Επειδή στην ερώτηση υπάρχει η λέξη "μάθημα" δημιουργούνται οι δυο νέες σχέσεις apoteleitai_apo/3 και leptomereies_mathimaton/3.

apoteleitai_apo(Kod_mathimatos, Kod_anal_mathimatos1, Kod_anal_mathimatos2).

leptomereies_mathimaton(Kod_anal_mathimatos1,εργαστηριακό,Ores).

leptomereies_mathimaton(Kod_anal_mathimatos2,εργαστηριακό,Ores).

Συνεπώς, οι τελικές σχέσεις που επιλέχθηκαν για τη δημιουργία της τελικής ερώτησης/στόχου σε Prolog είναι οι εξής:

kathigites(Kod_kathigiti, Epitheto_kathigiti, Onoma_kathigiti, Thl_grafeiou, Kod_Grafeio_kathigiti, Mail_ kathigiti).

didaskoun(Kod_kathigiti, Kod_anal_mathimatos).

mathimata(Kod_mathimatos, Onoma_mathimatos, Sem_mathimatos, Tupos_mathimatos, Did_monades).

mathimata(Kod_mathimatos, Onoma_mathimatos, Sem_mathimatos, Tupos_mathimatos, Did_monades).

apoteleitai_apo(Kod_mathimatos, Kod_anal_mathimatos1, Kod_anal_mathimatos2).

leptomereies_mathimaton(Kod_anal_mathimatos1,εργαστηριακό,Ores).

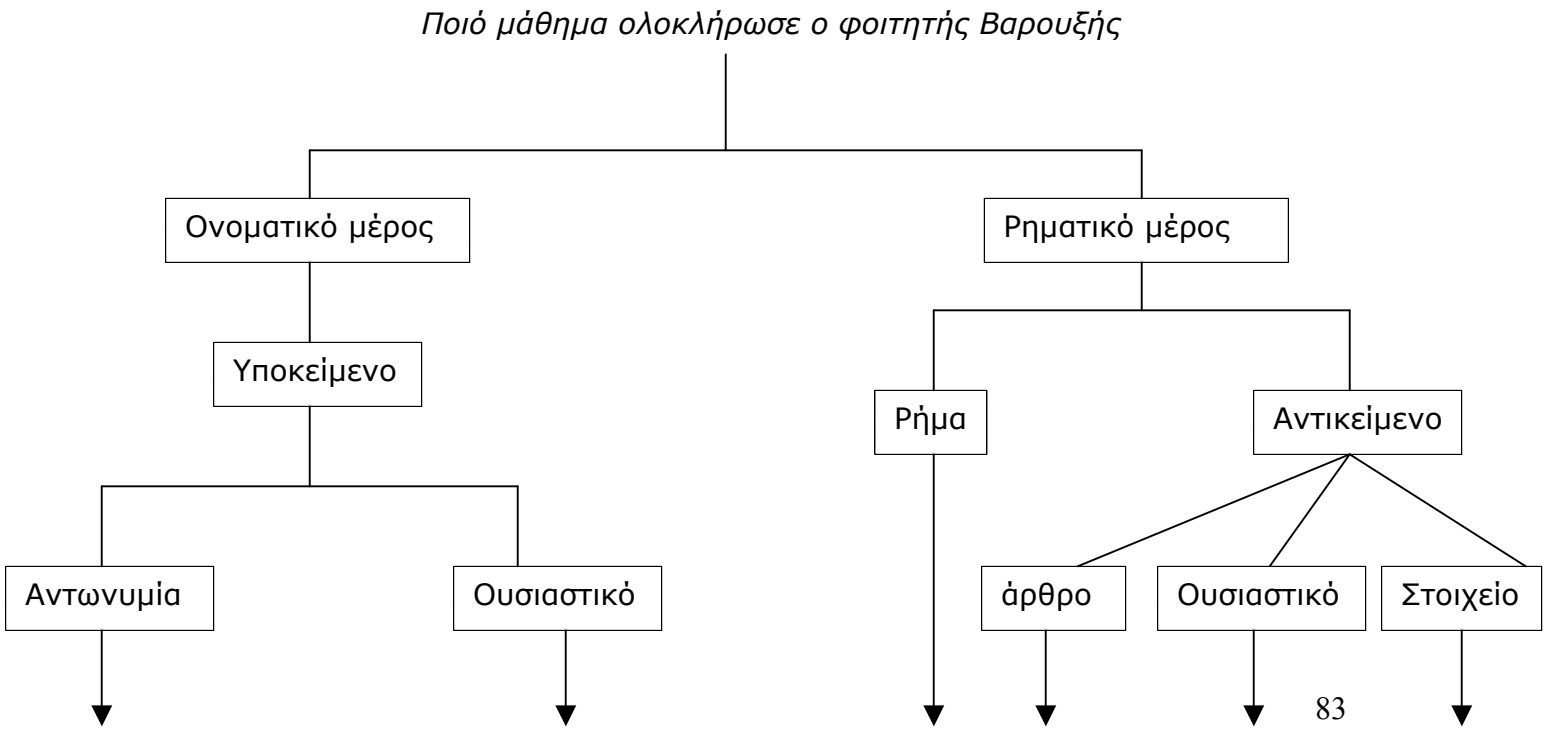
leptomereies_mathimaton(Kod_anal_mathimatos2,εργαστηριακό,Ores).

4.6.1.3 Σημασιολογικός Έλεγχος Συντακτικής Ανάλυσης Πρότασης

Η συντακτική ανάλυση μιας πρότασης μπορεί να μη γίνει αποδεκτή λόγω σημασιολογικού λάθους. Κατά την πρόταση "Ποιό μάθημα ολοκλήρωσε ο φοιτητής Βαρουξής", ο parser αναγνωρίζει τη δομή "μάθημα" σαν υποκείμενο το οποίο όμως είναι αντικείμενο. Αυτό ελέγχεται στο σημασιολογικό επίπεδο με έλεγχο στο λεξικό πεδίου προβλήματος. Από αυτό το λεξικό προκύπτει ότι η οντότητα "μάθημα" δε συνδέεται με την ενέργεια "διδάσκει". Η διόρθωση της συντακτικής ανάλυσης γίνεται με αποτυχία της συντακτικής ανάλυσης, οπισθοδρόμηση του parser και εύρεση εναλλακτικής λύσης. Η αποδεκτή λύση πρέπει να ικανοποιεί ότι το υποκείμενο συνδέεται με το ρήμα μέσω του λεξικού πεδίου προβλήματος.

Η αποδεκτή σημασιολογική ανάλυση για το παράδειγμά μας είναι ότι "ο φοιτητής" είναι το υποκείμενο και το "μάθημα" είναι το αντικείμενο.

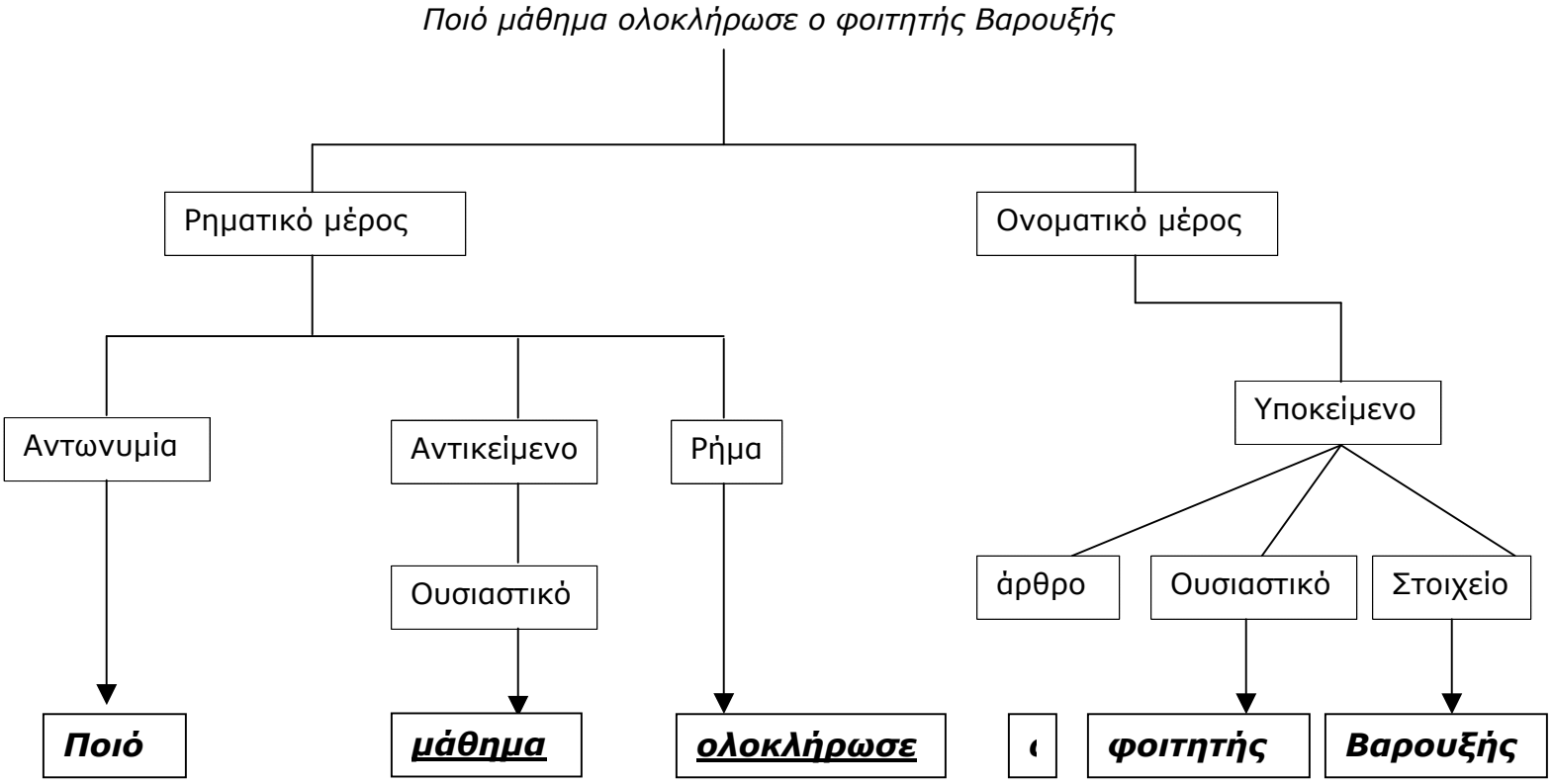
Παρακάτω εμφανίζεται σε απλή μορφή το συντακτικό δένδρο που θα απορριφθεί λόγω της σημασιολογικής ανάλυσης



Ποιό **Σχήμα 4.4:** Συντα ***μάθημα*** πριν τ ***ολοκλήρωσε*** λυ ***φοιτητής*** ***Βαρουξής***

Υπογραμμισμένες είναι οι λέξεις από τις οποίες γίνεται η σημασιολογική ανάλυση.

Παρακάτω παρουσιάζεται το νέο συντακτικό δένδρο που προκύπτει από την απόρριψη του παραπάνω σχήματος Σχήμα 4.4.



Σχήμα 4.5: Συντακτικό δένδρο μετά τη σημασιολογική ανάλυση.

Το Σχήμα 4.5 δείχνει τη σωστή συντακτική ανάλυση η οποία επιτυγχάνει στο σημασιολογικό επίπεδο.

4.7 Υποσύστημα Κατασκευής Prolog Ερωτήσεων

Το επίπεδο **κατασκευής ερωτήσεων Prolog** είναι υπεύθυνο για την για τη κατασκευή ενός στόχου/ερώτησης σε Prolog από τους απλούς στόχους που δημιουργήθηκαν από τη σημασιολογική ανάλυση. Η ανάκτηση των πληροφοριών από τη Βάση Γνώσεων, τις οποίες ζητά ο χρήστης με την ερώτησή του, γίνεται με τη δημιουργία μια ερώτησης στόχου σε Prolog. Αυτή η Prolog ερώτηση περιέχει πληροφορίες για τις οντότητες και τις συσχετίσεις, οι οποίες έχουν δοθεί από την ερώτηση του χρήστη του συστήματος EROTISIS. Αν δεν υπάρχουν μεταβλητές, η απάντηση ενός γεγονότος είναι yes/no. Σε περίπτωση που υπάρχουν, η απάντηση δίνει τιμή σε όλες τις μεταβλητές του στόχου. Για παράδειγμα:

?- foitites(642, βαρουξής, κωνσταντίνος, 01-02Εαρ, 6937090404,
kvarouxis@hotmail.com).
yes.

?- foitites(ΑΜ, βαρουξής, Onoma_foititi, 01-02Εαρ, 6937090404, Mail_foititi).
ΑΜ=642
Onoma_foititi = κωνσταντίνος
Mail_foititi = kvarouxis@hotmail.com

Η διαδικασία δημιουργίας απλών στόχων της ερώτησης του χρήστη, συνεχίζει μέχρι να ολοκληρωθεί η ανάλυση όλων των όρων. Οπότε, δημιουργείται ένα n-πλήθος απλών ερωτήσεων Prolog από τις οποίες δημιουργείται η τελική σύνθετη ερώτηση Prolog με λογική σύζευξή τους. Οι *υποερωτήσεις*, οι οποίες στο σύνολό τους απαρτίζουν την σύνθετη ερώτηση, χωρίζονται μεταξύ τους με κόμμα (','), που έχει την έννοια του λογικού "και".

Μερικοί απλοί στόχοι μπορούν να έχουν κάποια ίδια ορίσματα. Στη βάση

δεδομένων, η σχέση για τους φοιτητές foitites/7 έχει σαν πρωτεύον κλειδί το ΑΜ των φοιτητών. Η σχέση που αντιπροσωπεύει τα μαθήματα που έχει ολοκληρώσει κάθε φοιτητής exei_perasei/2 περιέχει δυο πεδία, το ΑΜ του φοιτητή που έχει ολοκληρώσει κάποιο μάθημα, και τον κωδικό του μαθήματος αυτού.

Δηλαδή έχουμε τους στόχους:

foitites(ΑΜ1, βαρουξής, C, D, E, F).

και

exei_perasei (ΑΜ2, G).

Με χρήση του λεξικού δεδομένων το σύστημα βρίσκει ότι τα πεδία ΑΜ1 και ΑΜ2 περιέχουν την ίδια οντότητα. Στη σύζευξη των δύο απλών στόχων αυτό θα εκφραστεί με την ίδια μεταβλητή. Π.χ.: ΑΜ, η οποία θα παριστά αυτή την ιδιότητα. Ο στόχος-σύζευξη των απλών στόχων που θα προκύψει θα είναι ο εξής:

foitites(**ΑΜ**, βαρουξής, C, D, E, F), exei_perasei (**ΑΜ**, G).

Με αυτόν τον τρόπο, μπορούμε να δούμε όλα τα μαθήματα τα οποία ολοκλήρωσε ένας ο φοιτητής Βαρουξής.

Η κοινή μεταβλητή Α στα κατηγορήματα foitites/7 και exei_perasei/2 παριστά την κοινή τιμή στο πεδίο ΑΜ.

Για να δημιουργηθεί ο τελικός στόχος σε Prolog, αρχικά δημιουργούνται δυο μικρότεροι στόχοι, ένας για το **ονοματικό μέρος** και ένας για το **ρηματικό μέρος**. Ο στόχος για το ονοματικό μέρος και ο αντίστοιχος για το ρηματικό μέρος, δημιουργούνται από τη **σύζευξη ή/και διάζευξη** των απλών στόχων. Για την καλύτερη διαχείριση των απλών στόχων που αφορούν το ονοματικό μέρος αποθηκεύονται όλοι οι απλοί στόχοι σε μια μεταβλητή, την STOXOS_ONOMATIKOY_MEROYS. Αντιστοίχως, οι απλοί στόχοι του ρηματικού μέρους αποθηκεύονται στη μεταβλητή STOXOS_RIMATIKOY_MEROYS. Στη συνέχεια, η σύζευξη των στόχων αυτών των δύο μεταβλητών δημιουργεί τον τελικό στόχο, ο οποίος καταχωρείται στη μεταβλητή STOXOS. Η σύνθεση των

απλών στόχων του ονοματικού και του ρηματικού μέρους, γίνεται ως εξής:

- Οι απλοί στόχοι του ρηματικού μέρους, ενώνονται με κόμμα(“,”)
- Οι απλοί στόχοι του ονοματικού μέρους, ενώνονται με κόμμα(“,”)
- Στην περίπτωση που υπάρχει στην εισαγόμενη πρόταση του χρήστη η λέξη “μάθημα”, οι υποστόχοι:

leptomereies_mathimaton(Kod_anal_mathimatos1,εργαστηριακό,Ores),
leptomereies_mathimaton(Kod_anal_mathimatos2,εργαστηριακό,Ores),
ενώνονται με χρήση διαζευκτικού ή (“;”). ***Σημείωση:*** *Η χρήση του διαζευκτικού γίνεται γιατί ο φοιτητής αρκεί να έχει ολοκληρώσει έναν από τους δυο τύπους μαθημάτων ώστε να επιστραφεί απάντηση στον χρήστη. Δηλαδή αρκεί να έχει ολοκληρώσει το θεωρητικό* ***ή*** *το εργαστηριακό μάθημα. Το διαζευκτικό “;” περιγράφει το “****ή****” στην παραπάνω πρόταση.*

Ακολουθεί ψευδοκώδικας σε Prolog που περιγράφει τη διαδικασία εύρεσης ερώτησης-στόχου σε Prolog.

1. Ένωσε όλες τις υποερωτήσεις, απλούς στόχους του **ονοματικού μέρους** μεταξύ τους με κόμμα(‘,’) και αποθήκευσε την ενιαία σύνθετη ερώτηση-Prolog στην μεταβλητή **STOXOS_ONOMATIKOY_MEROYS**.
2. Ένωσε όλες τις υποερωτήσεις, απλούς στόχους του **ρηματικού μέρους** μεταξύ τους με κόμμα(‘,’) και αποθήκευσε την ενιαία σύνθετη ερώτηση-Prolog στην μεταβλητή **STOXOS_RIMATIKOY_MEROYS**.
3. Ένωσε τις μεταβλητές **STOXOS_ONOMATIKOY_MEROYS** και **STOXOS_RIMATIKOY_MEROYS** με κόμμα (‘,’) και αποθήκευσε το αποτέλεσμα σε μια μεταβλητή με όνομα **STOXOS**.

Πρόγραμμα 4.10: Ψευδοκώδικας δημιουργίας ερώτησης-στόχου σε Prolog της εισαγόμενης ερώτησης του χρήστη.

Για παράδειγμα, έχουμε την ερώτηση του χρήστη στο σύστημα EROTISIS “ποιό εργαστηριακό μάθημα διδάσκει ο καθηγητής με επίθετο Βαρδιάμπασης” .

Ο στόχος που προκύπτει από το ονοματικό μέρος είναι ο **kathigites(A,βαρδιάμπασης,C,D,E,F,G)**

Από το ρηματικό μέρος προκύπτει

didaskoun(A,G),(apoteleitai_apo(L,K,M);apoteleitai_apo(L,N,K)),mathimata(L,O,P,Q,R),leptomereies_mathimaton(K,Y,T)

Ενώνοντας τους δυο παραπάνω στόχους σε μια μεταβλητή με όνομα STOXOS έχουμε την εξής τελική μορφή:

kathigites(A,βαρδιάμπασης,C,D,E,F,G),didaskoun(A,G),(apoteleitai_apo(L,K,M);apoteleitai_apo(L,N,K)),mathimata(L,O,P,Q,R),leptomereies_mathimaton(K,Y,T).

4.8 Συλλογή Ζητούμενων Αποτελεσμάτων

Η τελική σύνθετη ερώτηση που προκύπτει από το επίπεδο κατασκευής ερώτησης σε Prolog, εισάγεται στο επίπεδο **συλλογής ζητούμενων αποτελεσμάτων.**

Υπενθυμίζουμε πως η τελική ερώτηση αποτελείται από πολλούς απλούς στόχους, κάθε απλός στόχος αποτελείται από ένα κατηγόρημα. Οι απλοί αυτοί στόχοι συνδέονται μεταξύ τους με σύμβολα (',' ή /και ';'). Κάθε απλός στόχος περιέχει τιμές στα ορίσματα των κατηγορημάτων οι οποίες έχουν προέλθει από λέξεις της ερώτησης του χρήστη. Οι λέξεις αυτές μπορεί να είναι κάποιο όνομα, κάποιος κωδικός, τίτλος μαθήματος κ.ά.. Εκτός όμως από τα πεδία/ορίσματα των κατηγορημάτων τα οποία περιέχουν τιμές, στα υπόλοιπα ορίσματα των κατηγορημάτων της ερώτησης υπάρχουν μεταβλητές. Η απάντηση που ζητά ο χρήστης πρέπει να προέλθει από τις τιμές κάποιου από αυτά τα πεδία/ορίσματα. Η απάντηση που θα παρουσιάζεται στον χρήστη πρέπει να καλύπτει το αρχικό του ερώτημα.

Διακρίνουμε τις εξής περιπτώσεις:

1. **Η ερώτηση να αφορά πρόσωπο από το πεδίο του προβλήματος**

   Σε περίπτωση που ο χρήστης εισάγει **ερώτηση προσώπου**(ερώτηση που αφορά κάποιο πρόσωπο του πεδίου προβλήματος), η απάντηση θα είναι το επίθετο του καθηγητή ή του φοιτητή. Για να επιτευχθεί αυτό κατασκευάστηκε κώδικας σε Prolog ο οποίος οδηγεί το σύστημα στο να επιστρέφει το επίθετο του προσώπου. Θα επιστρέφει δηλαδή το επίθετο του καθηγητή ή του φοιτητής.

2. **Η ερώτηση να αφορά ορισμό.**

   Σε περίπτωση που η εισαγόμενη ερώτηση είναι **ερώτηση ορισμού**, η

επίλυση περιπλέκεται. Η πολυπλοκότητα προέρχεται από το γεγονός ότι ένα πεδίο δεν είναι αρκετό για να απαντήσει **ακριβώς** αυτό που ρώτησε ο χρήστης. Φανταστείτε την ερώτηση: "Ποιό μάθημα διδάσκει ο καθηγητής Μαρακάκης". Πριν βιαστείτε να απαντήσετε, σκεφτείτε το ποια είναι η ακριβή σημασία του μαθήματος. Δηλαδή όταν λέμε "μάθημα", τι ακριβώς εννοούμε; Εργαστηριακό ή θεωρητικό; Ή μήπως και εργαστηριακό και θεωρητικό; Η απάντηση σε αυτό είναι καθαρά υποκειμενική όταν φτάσουμε στην υλοποίηση του συστήματος. Μια πιθανή εκδοχή είναι ο καθηγητής να διδάσκει το θεωρητικό και το εργαστηριακό μάθημα. Σε αυτή την περίπτωση, η απάντηση θα δοθεί μόνο αν ο καθηγητής Μαρακάκης διδάσκει το θεωρητικό και το αντίστοιχο εργαστηριακό μάθημα. Αν ο καθηγητής Μαρακάκης διδάσκει μόνο τον ένα τύπο του μαθήματος(π.χ.: έμπειρα συστήματα εργαστηριακό), ο στόχος θα αποτύγχανε, καθώς για να πετύχει θα έπρεπε να διδάσκει και το αντίστοιχο εργαστήριο. Τι γίνεται όμως στην περίπτωση που κάποιο μάθημα δεν έχει έναν από τους δυο τύπους μαθημάτων; Το μάθημα "εισαγωγή στο marketing" έχει θεωρία, αλλά όχι εργαστήριο. Σε αυτή την περίπτωση, δε θα παίρναμε απάντηση γιατί το σύστημα θα έψαχνε να βρει αν ο καθηγητής ο οποίος διδάσκει το μάθημα αυτό, διδάσκει και τη θεωρία και το εργαστήριο. Καταλήξαμε, λοιπόν, στην εξής λύση: ***σε παρόμοιες ερωτήσεις ορισμού, το σύστημα να δίνει απαντήσεις ξεχωριστά για εργαστηριακά και θεωρητικά μαθήματα***. Παρόμοιο πρόβλημα παρουσιάζεται και στην ερώτηση όπου ο χρήστης ζητάει να μάθει ποια μαθήματα ολοκλήρωσε κάποιος φοιτητής. Στη πραγματικότητα, η χρήση του ρήματος "ολοκλήρωσε ένα μάθημα" έχει διάφορες ερμηνείες. Ολοκλήρωσε τη θεωρία και το εργαστήριο, ολοκλήρωσε μόνο τη θεωρία αλλά όχι το εργαστήριο και τέλος ολοκλήρωσε το εργαστήριο αλλά όχι την θεωρία. Και σε αυτή τη περίπτωση, απάντηση δίνεται ξεχωριστά για την θεωρία και το εργαστήριο. Δηλαδή ο φοιτητής δεν είναι ανάγκη να έχει περάσει τη θεωρία και το εργαστήριο για να μπορούμε να πούμε πως ολοκλήρωσε το μάθημα. Αρκεί το ένα από τα δύο. Έτσι, το σύστημα θα

επιστρέψει το όνομα του μαθήματος και τον τύπο του.

Η **συλλογή των αποτελεσμάτων** έρχεται από τη βάση δεδομένων. Η ερώτηση-στόχος σε Prolog εξάγει τα αποτελέσματα από εκεί. Η συλλογή των ζητούμενων αποτελεσμάτων γίνεται με τη χρήση του κατηγορήματος ***findall/3***.

ΠΕΡΙΓΡΑΦΗ ΚΑΤΗΓΟΡΗΜΑΤΟΣ findall/3

Το κατηγόρημα αυτό δέχεται τρία ορίσματα.

Περιγράφεται παρακάτω η λειτουργία του κατηγορήματος.

findall(Object,Goal,List).

Παράγει μια λίστα List με όλα τα αντικείμενα Objects τα οποία ικανοποιούν τον στόχο Goal.

```
happy(kostas).
happy(george).
happy(thanos).

?- findall(X, happy(X), L).
L = [kostas, george, thanos].
```

Πρόγραμμα 4.11: Λειτουργία κατηγορήματος findall/3.

ΧΡΗΣΗ ΚΑΤΗΓΟΡΗΜΑΤΟΣ findall/3 ΓΙΑ ΣΥΛΛΟΓΗ ΖΗΤΟΥΜΕΝΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Το κατηγόρημα findall/3 στο σύστημά μας, δέχεται σαν ορίσματα ***μια λίστα με όλες τις μεταβλητές από την ερώτηση-στόχο σε Prolog, την ερώτηση-στόχο σε Prolog και μια λίστα, στην οποία θα τοποθετηθούν οι τιμές που ικανοποιούν το ερώτηση-στόχο σε Prolog*.**

Δηλαδή, στην ερώτηση του χρήστη: "ποιό εργαστηριακό μάθημα διδάσκει ο

καθηγητής Βαρδιάμπασης", τα ορίσματα του findall/3 έχουν τιμές:

L_METAVLITES=[A,B,C,D,E,F,_7527,H,G,J,K,L,M,N,O,_8749,_8747,
P,_9144,_9142]

STOXOS=kathigites(A,βαρδιάμπασης,B,C,D,E,F),didaskoun(A,G),
(apoteleitai_apo(H,G,J);apoteleitai_apo(H,K,G)),
mathimata(H,L,M,N,O),
leptomereies_mathimaton(G,εργαστηριακό,P)

Πρόγραμμα 4.12: Απόδοση τιμών στο κατηγόρημα findall/3 για την εισαγόμενη ερώτηση "ποιό εργαστηριακό μάθημα διδάσκει ο καθηγητής Βαρδιάμπασης"

Στη λίστα APOTELESMATA_SE_LISTA αποθηκεύονται οι τιμές του L_METAVLITES, που ικανοποιούν την ερώτηση-στόχο σε Prolog η οποία υπάρχει στη μεταβλητή STOXOS.

?- findall(L_METAVLITES, STOXOS, APOTELESMATA_SE_LISTA),

Πρόγραμμα 4.13: Ερώτηση-στόχος σε Prolog για συλλογή αποτελεσμάτων.

Δηλαδή, τρέχοντας τον παραπάνω στόχο από το Πρόγραμμα 4.12, για τη μεταβλητή APOTELESMATA_SE_LISTA έχουμε:

APOTELESMATA_SE_LISTA = [[11,δημήτριος,αρσ,νκ5,-,-,_B,τπ2001,
τπ2001β |...], [11,δημήτριος,αρσ,νκ5,-,-,_I,τπ3001,τπ3001β|...]].

Η παρουσία δυο υπολιστών, δηλώνει την ύπαρξη δυο λύσεων.

Για το επίπεδο συλλογής των ζητούμενων αποτελεσμάτων, έχουμε τον παρακάτω ψευδοκώδικα:

1. Αν το ονοματικό μέρος **ξεκινάει με ερωτηματική αντωνυμία**("ποιός,ά,οί"κ.ά.) (δηλαδή ο χρήστης ζητάει το **επίθετο** του προσώπου),

δημιούργησε μεταβλητή με PEDIO=2.

2. Αν το ρηματικό μέρος **ξεκινάει με ερωτηματική αντωνυμία**("ποιο,ά,ός"κ.ά.) (δηλαδή ο χρήστης ζητάει το **τίτλο** του μαθήματος), τότε: **2.1** Αν το αντικείμενο έχει επιθετικό προσδιορισμό (δηλαδή "εργαστηριακό" ή "θεωρητικό"), δημιούργησε μεταβλητή με PEDIO=9. **2.2** Αν δεν έχει επιθετικό προσδιορισμό, δημιούργησε μεταβλητή με PEDIO=[12, 18].
3. Βάλε όλες τις **τιμές** των μεταβλητών που προκύπτουν από την ερώτηση-στόχο της Prolog σε μια λίστα με όνομα APOTELESMATA_SE_LISTA.
4. Από τη λίστα APOTELESMATA_SE_LISTA πάρε τις τιμές που βρίσκονται στις θέσεις που ορίζει η μεταβλητή PEDIO.

Πρόγραμμα 4.14: Ψευδοκώδικας εύρεσης των ζητούμενων αποτελεσμάτων από τη λίστα με τις απαντήσεις.

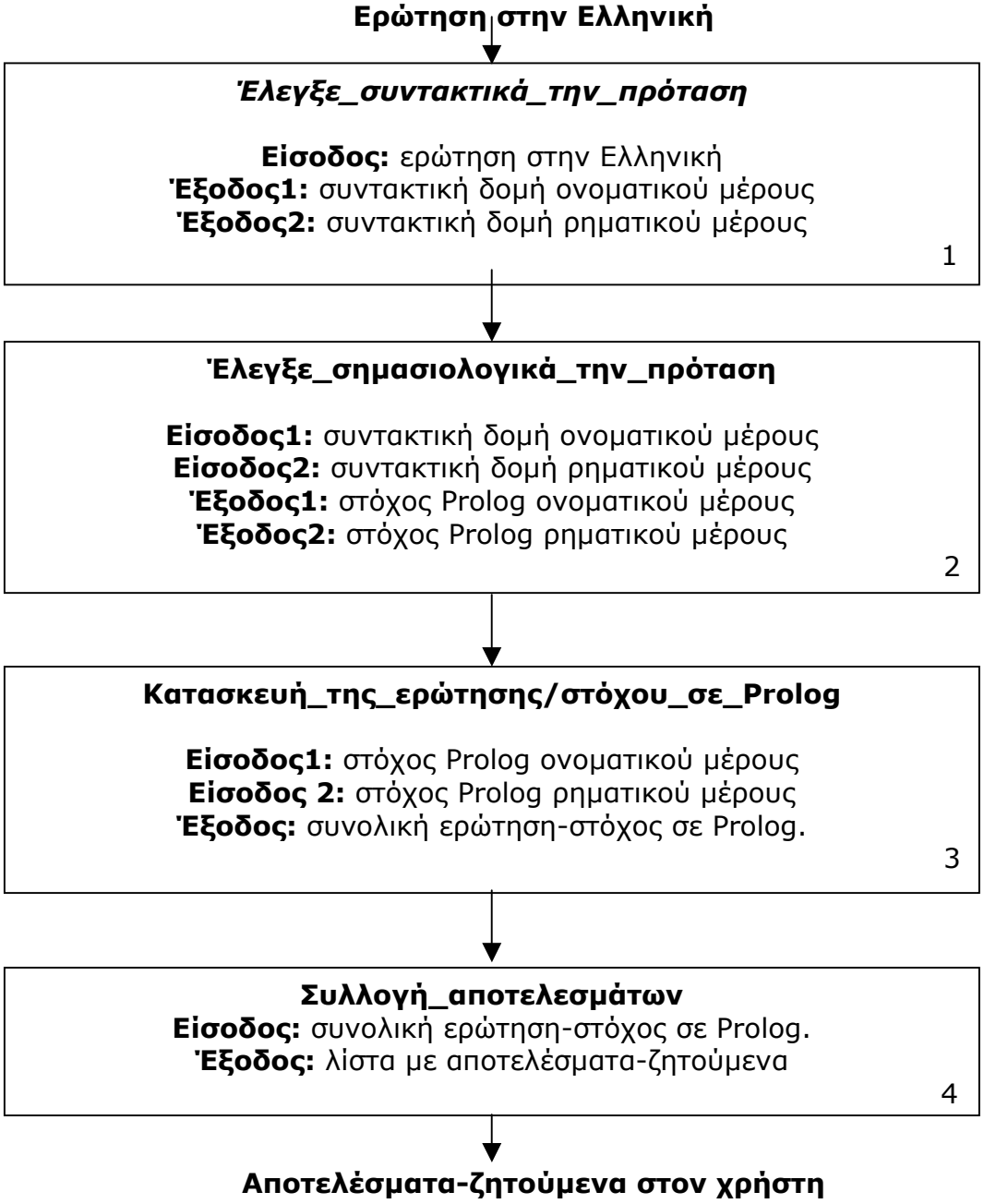
Η λίστα APOTELESMATA_SE_LISTA περιέχει, όπως περιγράψαμε στο πρόγραμμα Πρόγραμμα 4.13, όλες τις τιμές των μεταβλητών που προκύπτουν από την ερώτηση-στόχο της Prolog. Το σύστημα EROTISIS μπορεί να απαντήσει σε ερωτήσεις ορισμού και ερωτήσεις προσώπου. Αυτό σημαίνει πως τα πεδία που θα δοθούν στον χρήστη ως απάντηση είναι σε συγκεκριμένες θέσεις στην λίστα APOTELESMATA_SE_LISTA. Γι αυτό το λόγο αποθηκεύουμε στην μεταβλητή PEDIO συγκεκριμένη τιμή κάθε φορά.

Πιο αναλυτικά, το επίθετο του φοιτητή ή του καθηγητή, βρίσκεται στην δεύτερη θέση της λίστας APOTELESMATA_SE_LISTA. Είτε ζητείται το επίθετο του φοιτητή ή του καθηγητή είτε όχι, η λίστα APOTELESMATA_SE_LISTA περιέχει τη τιμή που επιστρέφεται από την εκτέλεση της ερώτησης-στόχος σε Prolog.

Με τη χρήση του προγράμματος Πρόγραμμα 4.13, ελέγχουμε τι ακριβώς θέλει να μάθει ο χρήστης, οπότε το σύστημα αποφασίζει ποιά τιμή θα δώσει στη μεταβλητή STILI_AGNOSTOU. Για παράδειγμα, αν ο χρήστης εισάγει ερώτηση προσώπου, η απάντηση θα έρθει από τη δεύτερη θέση της λίστας APOTELESMATA_SE_LISTA.

4.9 Αλγοριθμική Περιγραφή του Συστήματος

Στο **διάγραμμα ροής**, Σχήμα 4.6, παριστάνονται οι κύριες διαδικασίες του συστήματος EROTISIS για να απαντηθεί η εισαγόμενη ερώτηση του χρήστη.



Σχήμα 4.6: Διάγραμμα ροής πληροφοριών για την εύρεση αποτελέσματος

Θα παρουσιάσουμε τα τμήματα του σχήματος Σχήμα 4.6 ξεχωριστά και θα χρησιμοποιήσουμε το διάγραμμα ροής για την περιγραφή τους. Στο σχήμα δίνουμε αριθμούς οι οποίοι αντιστοιχούν στα τμήματα (modules) του συστήματος.

Για την περιγραφή των διαδικασιών χρησιμοποιούμε την συμβατική μορφή περιγραφής αλγορίθμων σε ψευδοκώδικα. Κάθε εντολή τελειώνει με το σύμβολο "***;***". Το σύμβολο "***:=***" είναι η εντολή καταχώρησης. Το σύμβολο "%" χρησιμοποιείται σαν σχόλια. Τα υπόλοιπα σύμβολα έχουν την ίδια σημασιολογία όπως χρησιμοποιούνται στα μαθηματικά και στις συμβολικές γλώσσες προγραμματισμού.

Τμήμα 1: Η πρώτη διαδικασία που τρέχει το πρόγραμμα όταν ξεκινήσουμε το σύστημα. Παίρνει σαν μοναδικό όρισμα την εισαγόμενη πρόταση του χρήστη από την διεπαφή. Επιπλέον, συντονίζει τις κλήσεις των απαραίτητων προγραμμάτων που απαιτούνται.

Στην διαδικασία αυτή εκτελείται ο παρακάτω αλγόριθμος σε ψευδοκώδικα:

Διαδικασία **start**

αρχή_διαδ
Φόρτωσε_όλα_τα_απαραίτητα_αρχεία;
% κάνει consult τα αρχεία

Μετέτρεψε_την_ερώτηση_του_χρήστη_σε_μορφή_λίστας;
% κλήση των κατηγορημάτων name/2 και tokenize/2

Έλεγξε_συντακτικά_την_πρόταση
Τέλος_διαδ

Διαδικασία **Μετέτρεψε_την_ερώτηση_του_χρήστη_σε_μορφή_λίστας**

Αρχή_διαδ
Μετέτρεψε_τους_χαρακτήρες_σε_αριθμούς_ASCII;
% χρήση κατηγορήματος name/2

```
        Χώρισε_ τις_λέξεις_με_κομμα;
        % χρήση κατηγορήματος tokenize/2
τέλος_διαδ
```

Διαδικασία **Έλεγξε_συντακτικά_την_πρόταση**

```
αρχή_διαδ
        για_κάθε Λέξη_εισαγώμενης_πρότασης κάνε
                εάν Ολες_οι_λέξεις_ταυτοποιούνται_με_λεξικά τότε
                        Έλεγξε_σημασιολογικά_την_πρόταση;
                τέλος_εάν
        τέλος_για_κάθε
τέλος_διαδ
```

Πρόγραμμα 4.15: Ψευδοκώδικας τμήματος συντακτικού ελέγχου της πρότασης

Τμήμα 2: Μετά τον επιτυχή έλεγχο του συντακτικού με τους γραμματικούς και του συντακτικούς κανόνες, σειρά έχει η σημασιολογική ανάλυση της εισαγόμενης ερώτησης. Γίνεται κλήση του κατηγορήματος elegxos/2. Τα ορίσματα του κατηγορήματος αυτού είναι το ονοματικό και ρηματικό μέρος, τα οποία έχουν ήδη αναλυθεί συντακτικά.

Διαδικασία **Έλεγξε_σημασιολογικά_την_πρόταση**

```
αρχή_διαδ
        Βρές_το_υποκείμενο_απο_τη_συντακτική_ανάλυση;
        % κλήση κατηγορήματος arg/3 στις πιθανές θέσεις του υποκειμένου

        Βρές_το_ρήμα_απο_τη_συντακτική_ανάλυση;
        % κλήση κατηγορήματος arg/3 στις πιθανές θέσεις του ρήματος

        Εάν Ταυτοποίηση_αποτελέσματος_με_εννοιολογική_εξάρτηση; τότε
        % από energies.pl έλεγξε τις ενέργειες των προσώπων

        Κατασκευή_της_ερώτησης/στόχου_σε_Prolog;
                αλλιώς Έλεγξε_συντακτικά_την_πρόταση
                % εύρεση άλλης λύσης από το πρώτο επίπεδο.
                τέλος_εάν
τέλος_διαδ
```

Πρόγραμμα 4.16: Ψευδοκώδικας τμήματος σημασιολογικού ελέγχου της πρότασης

Τμήμα 3: Η ερώτηση που εισήγαγε ο χρήστης είναι σωστή συντακτικά και σημασιολογικά, οπότε δημιουργείται η ερώτηση/στόχος σε Prolog.

Διαδικασία **Κατασκευή_της_ερώτησης/στόχου_σε_Prolog**

αρχή_διαδ
Φτιάξε_στόχο_ονοματικού_μέρους;
% κλήση κατηγορήματος arg/3 στις πιθανές θέσεις του υποκειμένου

Φτιάξε_στόχο_ρηματικού_μέρους;
% κλήση κατηγορήματος arg/3 στις πιθανές θέσεις του ρήματος

Ενοποίηση_των_στόχων_για_τον_σχηματισμό_του_τελικού;
τέλος_διαδ

Διαδικασία **Φτιάξε_στόχο_ονοματικού_μέρους**

αρχή_διαδ
Βρές_υποκείμενο_και_δημιούργησε_γεγονός;
%με χρήση κατηγορήματος arg/3 και τελεστή "=.."

Εισήγαγε_τιμές_από_εισαγώμενη_πρόταση_στον_στόχο_υποκειμένου;
%ταυτοποίηση τιμών από λεξικά

τέλος_διαδ

Διαδικασία **Φτιάξε_στόχο_ρηματικού_μέρους**

αρχή_διαδ
Βρές_ρήμα_και_δημιούργησε_γεγονός;
%με χρήση κατηγορήματος arg/3 και τελεστή "=.."

Εισήγαγε_τιμές_από_εισαγώμενη_πρόταση_στον_στόχο_ρήματος;
%ταυτοποίηση τιμών από λεξικά

Βρές_αντικείμενο_και_δημιούργησε_γεγονός;
%με χρήση κατηγορήματος arg/3 και τελεστή "=.."

Εισήγαγε_τιμές_από_εισαγώμενη_πρόταση_στον_στόχο_υποκειμένου;
%ταυτοποίηση τιμών από λεξικά

τέλος_διαδ

Διαδικασία
Ενοποίηση_των_δυο_στόχων_για_τον_σχηματισμό_του_τελικού;

αρχή_διαδ
Δημιουργία_ενιαίου_στόχου;
%με χρήση κατηγορήματος append/3.

Μετατροπή_μεταβλητών_σε_Αγγλικούς_χαρακτήρες;
%για ευκολία κατανόησης αποτελέσματος

τέλος_διαδ

Πρόγραμμα 4.17: Ψευδοκώδικας τμήματος δημιουργίας ερώτησης/στόχου Prolog

Τμήμα 4: Εκτέλεση ερώτησης/στόχου σε Prolog.

Διαδικασία **Συλλογή_αποτελεσμάτων**

αρχή_διαδ
Έυρεση_όλων_των_λύσεων;

Επιλογή_ζητούμενων_αποτελεσμάτων_βάση_ερώτησης;
%χρήση κατηγοτήματος evresi_sostou_apotelesmatos/4

τέλος_διαδ

Διαδικασία **Έυρεση_όλων_των_λύσεων**

αρχή_διαδ
Εισαγωγή_όλων_των_μεταβλητών_της_ερώτησης_στόχου_σε_λίστα;

Έυρεση_λύσεων_για_όλες_τις_μεταβλητές;
%χρήση κατηγορήματος findall/3.

τέλος_διαδ

Διαδικασία **Επιλογή_ζητούμενων_αποτελεσμάτων_βάση_ερώτησης**

αρχή_διαδ
Έλεγξε_τύπο_ερώτησης_και_πάρε_τα_κατάλληλα_δεδομένα
τέλος_διαδ

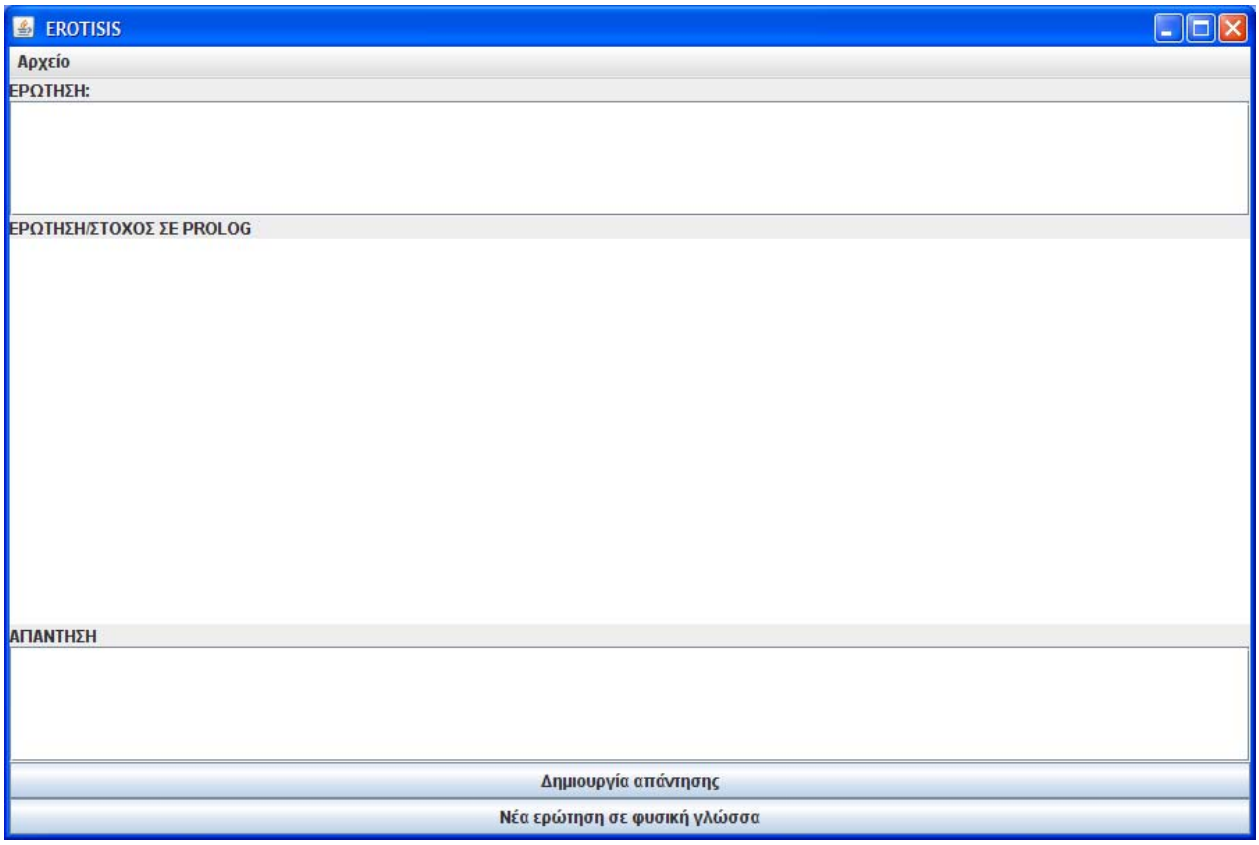
Πρόγραμμα 4.18: Ψευδοκώδικας τμήματος δημιουργίας απαντήσεων

5 ΔΕΙΓΜΑΤΑ ΣΕΝΑΡΙΩΝ- ΠΑΡΑΔΕΙΓΜΑΤΑ

Σε αυτό το κεφάλαιο, παρουσιάζονται ολοκληρωμένα σενάρια, από την είσοδο της ερώτησης μέχρι την παρουσίαση του αποτελέσματος στο χρήστη.

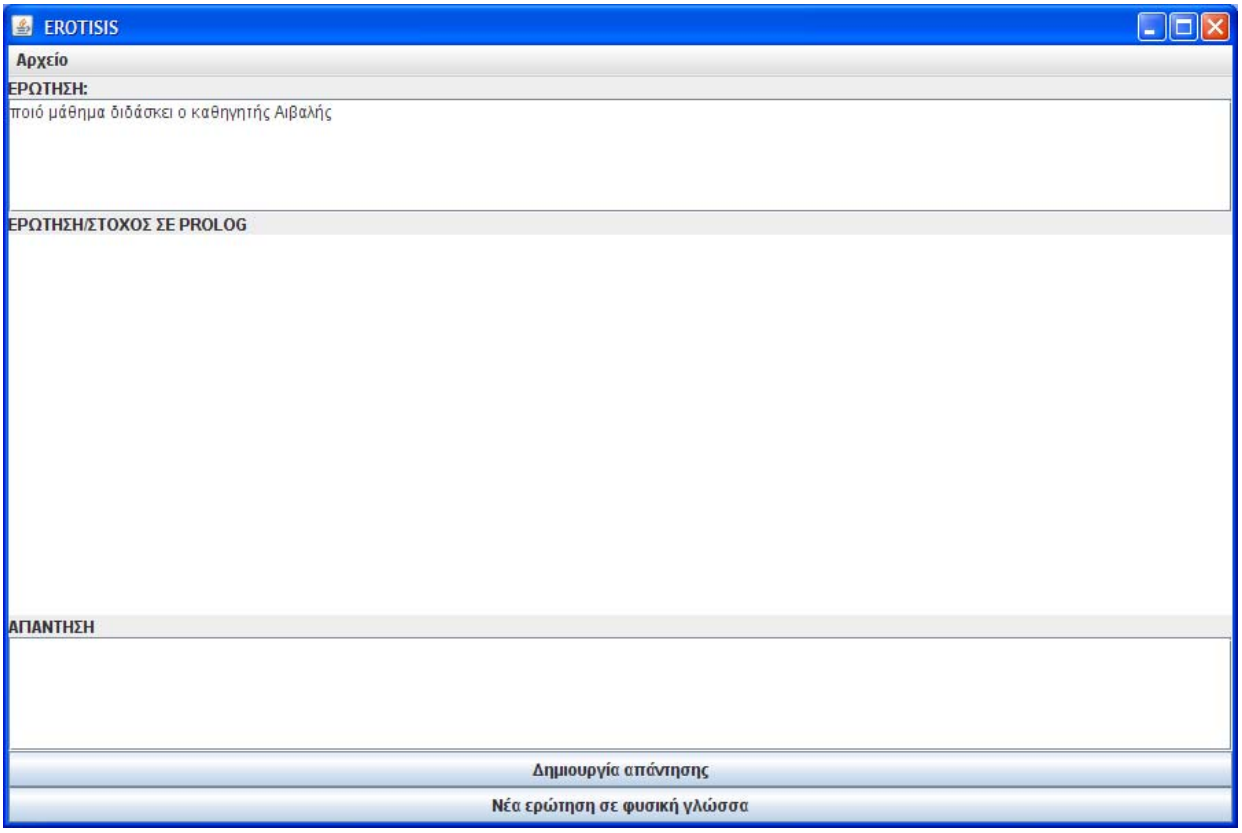
5.1 Σενάριο 1: Η εισαγόμενη πρόταση του χρήστη είναι συντακτικά και σημασιολογικά σωστή και η απάντηση βρίσκεται από τη Βάση Δεδομένων.

Εμφανίζεται ένα **παραθυρικό περιβάλλον**, με ένα παράθυρο στο οποίο ο χρήστης εισάγει την ερώτηση, άλλα δυο παράθυρα όπου εμφανίζεται η απάντηση του συστήματος και η ερώτηση/στόχος σε Prolog. Υπάρχει, ακόμα, ένα κουμπί το ("Δημιουργία απάντησης")το οποίο μόλις πατηθεί εμφανίζει τις απαντήσεις στην ερώτηση του χρήστη. Τέλος, υπάρχει άλλο ένα κουμπί ("Νέα ερώτηση σε φυσική γλώσσα"), με το οποίο ο χρήστης μπορεί να εισάγει νέα ερώτηση. Με την επιλογή File->Έξοδος γίνεται έξοδος από το σύστημα και με την επιλογή File->Νέο να δημιουργήσει νέα ερώτηση.



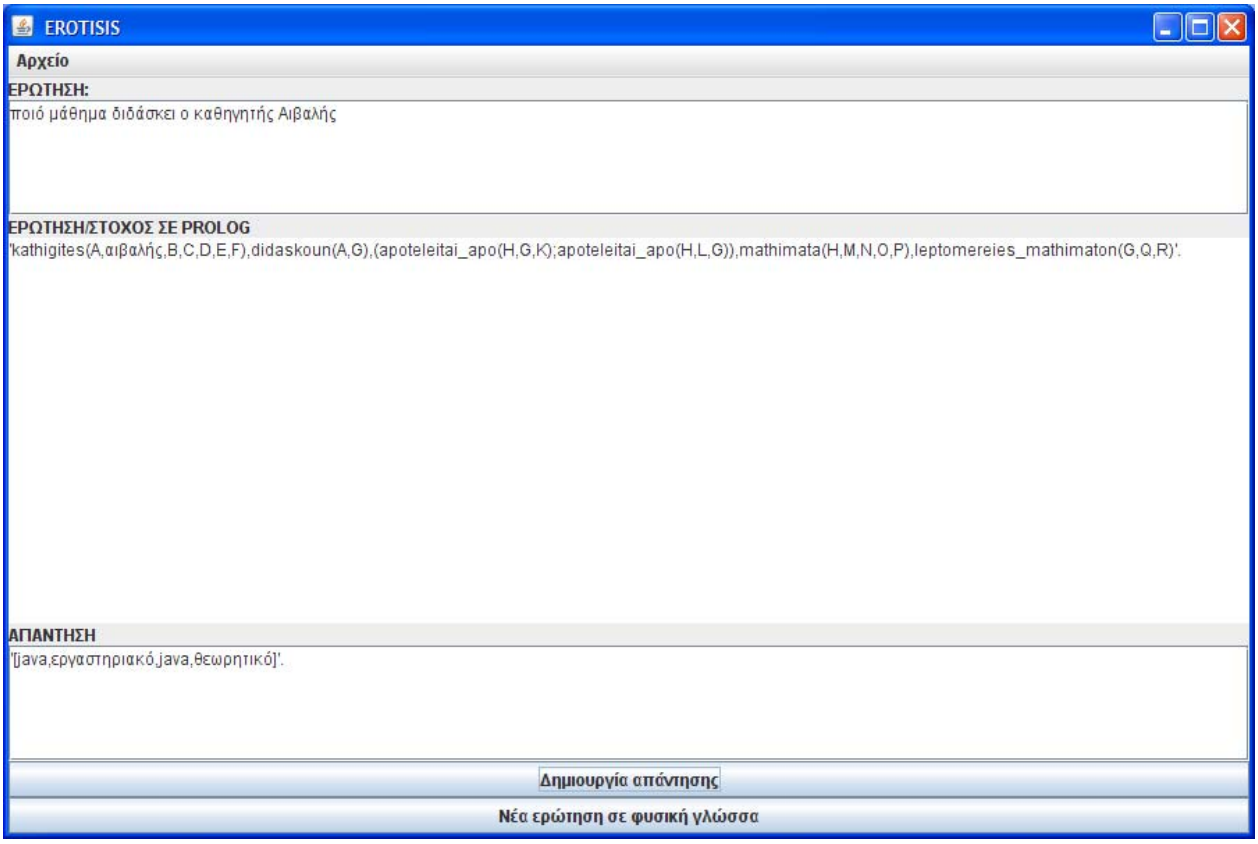
Εικόνα 5.1: Γραφικό περιβάλλον συστήματος EROTISIS

Εισάγουμε στο σύστημα την ερώτηση που θέλουμε σε φυσική γλώσσα (Ελληνικά).



Εικόνα 5.2: Εισαγωγή ερώτησης στο σύστημα EROTISIS (Παράδειγμα 1)

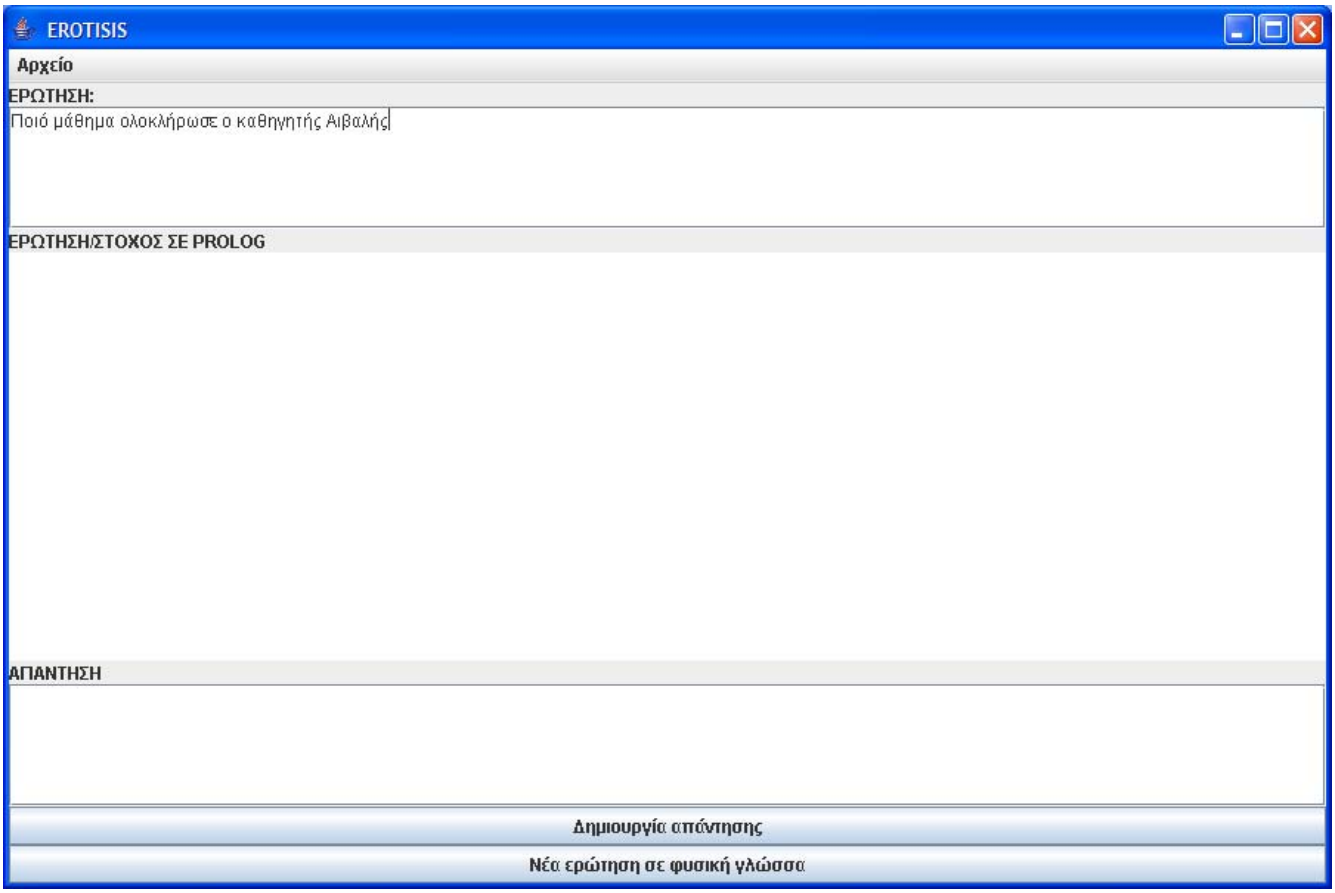
Πατάμε το κουμπί "Δημιουργία απάντησης". Η ερώτηση που εισάγαμε στο σύστημα μετατρέπεται σε στόχο Prolog. Ο στόχος αυτός εκτελείται. Εμφανίζεται το αποτέλεσμα στο παράθυρο "ΑΠΑΝΤΗΣΗ" η ερώτηση/στόχος και στο παράθυρο ΕΡΩΤΗΣΗ/ΣΤΟΧΟΣ ΣΕ PROLOG εμφανίζεται ο στόχος σε Prolog.



Εικόνα 5.3: Δημιουργία απάντησης στο σύστημα EROTISIS(Παράδειγμα 1)

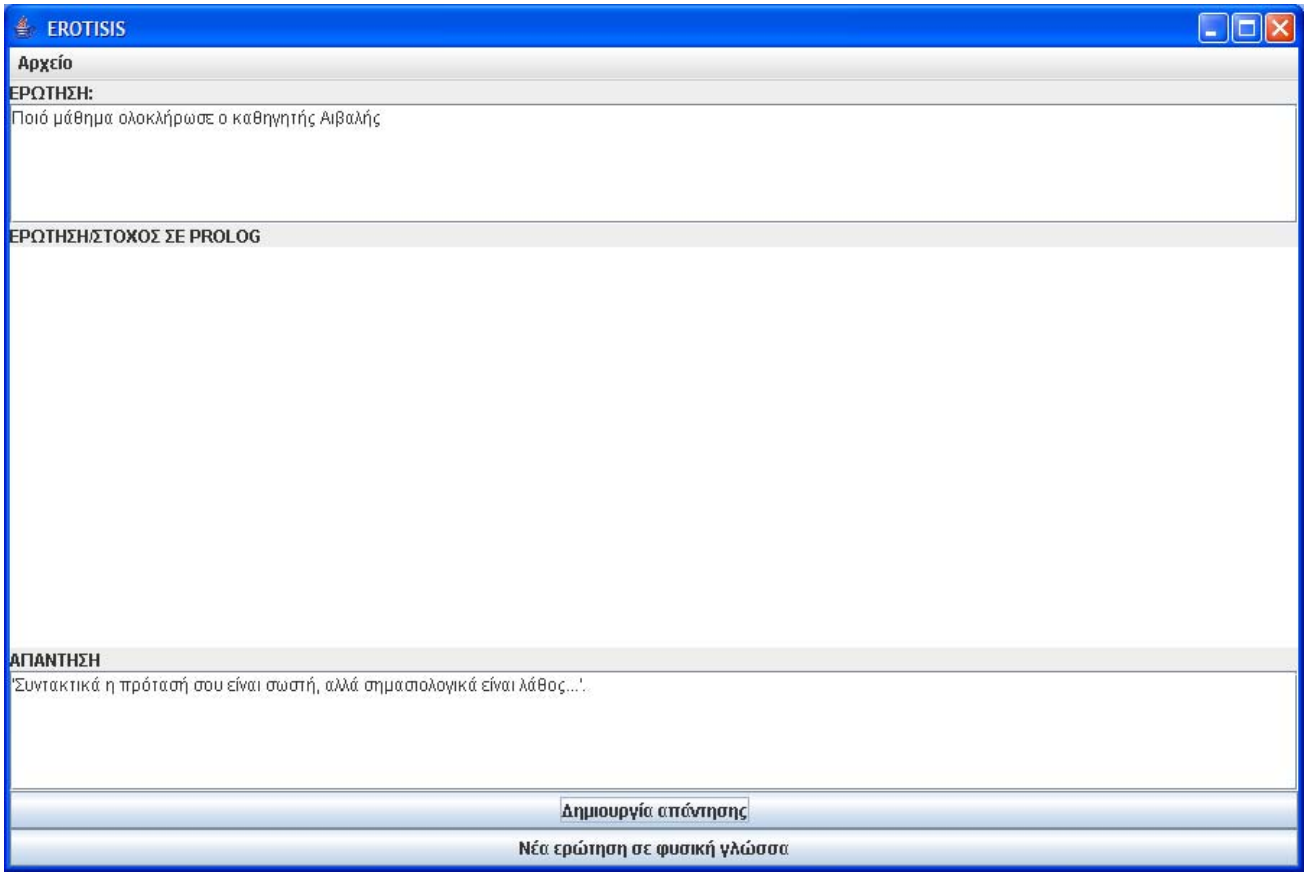
5.2 Σενάριο 2: Η εισαγόμενη πρόταση του χρήστη είναι συντακτικά σωστή αλλά σημασιολογικά λάθος.

Προσδίδουμε στο σύστημα την ερώτηση που θέλουμε σε φυσική γλώσσα(ελληνικά).



Εικόνα 5.4: Εισαγωγή ερώτησης στο σύστημα EROTISIS(Παράδειγμα 2)

Πατάμε το κουμπί "Δημιουργία απάντησης". Η ερώτηση που εισάγαμε στο σύστημα είναι συντακτικά σωστή, αλλά σημασιολογικά λάθος, καθώς οι καθηγητής δε διδάσκουν μαθήματα.

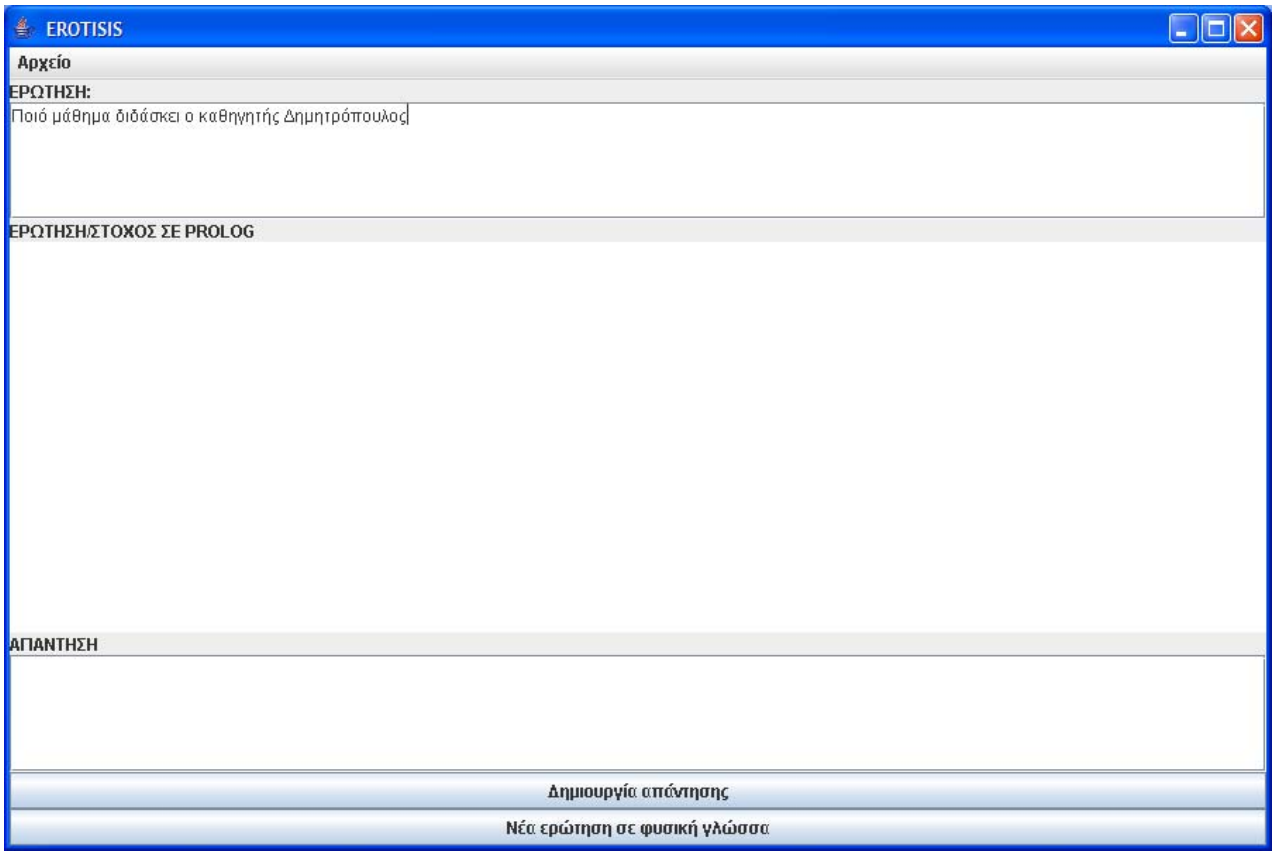


Εικόνα 5.5: Δημιουργία απάντησης στο σύστημα EROTISIS(Παράδειγμα 2)

Εμφανίζεται στο πεδίο της απάντησης μήνυμα ενημερώνοντας τον χρήστη πως η ερώτηση που εισήγαγε είναι σημασιολογικά λάθος. Το πεδίο ερώτησης/στόχου σε Prolog παραμένει κενό. Ο χρήστης μπορεί να δημιουργήσει νέα ερώτηση σε φυσική γλώσσα ή να εξέλθει από το πρόγραμμα.

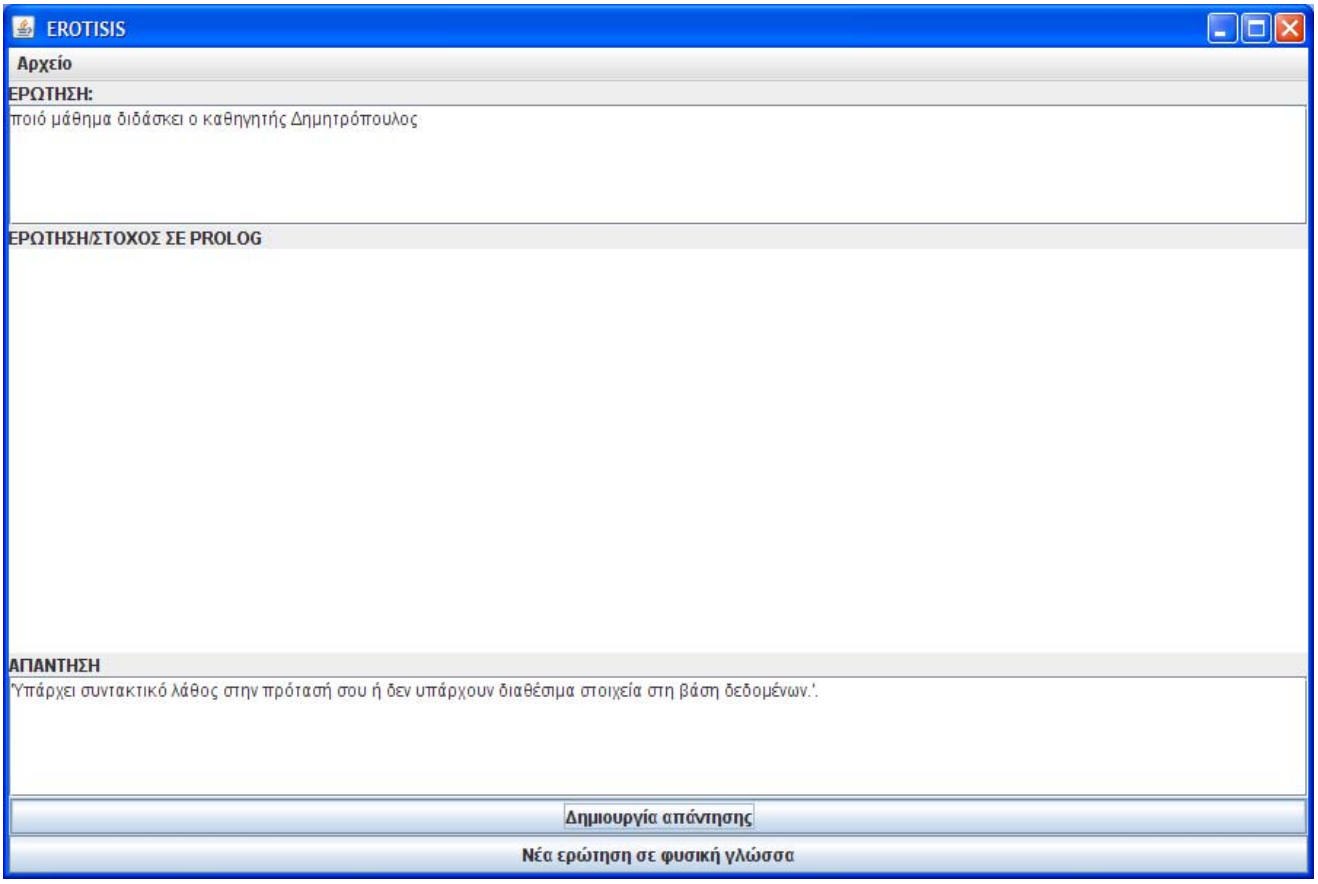
5.3 Σενάριο 3: Η εισαγόμενη πρόταση του χρήστη είναι σημασιολογικά σωστή, αλλά ή είναι συντακτικά λάθος ή δεν καλύπτεται η απάντηση από τη βάση δεδομένων

Προσδίδουμε στο σύστημα την ερώτηση που θέλουμε σε φυσική γλώσσα(ελληνικά).



Εικόνα 5.6: Εισαγωγή ερώτησης στο σύστημα EROTISIS(Παράδειγμα 3)

Πατάμε το κουμπί "Δημιουργία απάντησης". Η ερώτηση που εισάγαμε στο σύστημα είναι συντακτικά και σημασιολογικά σωστή, αλλά δεν υπάρχουν στη βάση δεδομένων μαθήματα που να διδάσκει ο καθηγητής Δημητρόπουλος.



Εικόνα 5.7: Δημιουργία απάντησης στο σύστημα EROTISIS(Παράδειγμα 3)

Εμφανίζεται στο πεδίο της απάντησης μήνυμα ενημερώνοντας τον χρήστη πως η πρόταση είναι συντακτικά λάθος ή δεν υπάρχουν διαθέσιμα στοιχεία στη βάση δεδομένων. Το πεδίο ερώτησης/στόχου σε Prolog παραμένει κενό. Ο χρήστης μπορεί να δημιουργήσει νέα ερώτηση σε φυσική γλώσσα ή να εξέλθει από το πρόγραμμα.

6 ΠΑΡΟΜΟΙΕΣ ΕΡΓΑΣΙΕΣ

Τα πρώτα συστήματα διεπικοινωνίας βάσεων δεδομένων σε φυσική γλώσσα εμφανίστηκαν τη δεκαετία του 70', καθώς η χρησιμοποίηση των βάσεων δεδομένων άρχισε να επεκτείνεται. Τα πιο γνωστά συστήματα ήταν το LUNAR [Woods, Kaplan, Webber, 1972], το οποίο παρουσιάστηκε το 72' και παρείχε διεπικοινωνία με βάση δεδομένων με πληροφορίες για πετρώματα που έφεραν οι αποστολές από το φεγγάρι. Βασιζόταν σε συντακτική ανάλυση και μπορούσε να δημιουργήσει πολλά δένδρα ανάλυσης για την ίδια ερώτηση. Αποδείχθηκε αναποτελεσματικό λόγω του ότι ήταν πολύ εξειδικευμένο. Το LADDER [Hendrix, Sacerdoti, Sagalowicz, Slocum, 1972], ήταν το πρώτο σύστημα που περιλάμβανε γραμματική ανάλυση, διεπικοινονώντας με μια βάση δεδομένων, περιέχοντας πληροφορίες για πλοία του αμερικάνικου ναυτικού. Το πιο γνωστό παράδειγμα συστήματος διεπικοινωνίας βάσης δεδομένων σε φυσική γλώσσα αποτελεί το CHAT-80 [Warren, Pereira, 1982]. Διεπικοινωνεί με μια βάση δεδομένων περιέχοντας γεωγραφικές πληροφορίες. Ο κώδικας του συστήματος αυτού είναι ελεύθερος και χρησιμοποιείται έως και σήμερα. Το CHAT-80 μετέτρεπε τις αγγλικές ερωτήσεις σε στόχους Prolog, οι οποίες έβρισκαν λύση μέσα από μια βάση δεδομένων. Η εξάπλωσή του ήταν μεγάλη και αποτέλεσε τη βάση για τη δημιουργία άλλων πειραματικών συστημάτων, όπως το MASQUE [Androutsopoulos, Ritchie, Thanisch, 1993]. Παρόμοια λειτουργεί το PRECISE, το οποίο σε περίπτωση που δε μπορεί να δώσει απάντηση, υποδεικνύει στον χρήστη ποιο μέρος της ερώτησης δεν κατάλαβε.

Αργότερα δημιουργήθηκε το RENDEVOUS [Codd, 1974] το οποίο ενέπλεκε τον χρήστη σε διάλογο για να τον βοηθήσει να σχηματίσει την ερώτησή του. Στα μέσα του 80', τα συστήματα διεποικοινωνίας με βάσεις δεδομένων αποτελούσαν αντικείμενο έρευνας και πολλά συστήματα είχαν δημιουργηθεί. Για παράδειγμα, το TEAM [Grosz, Appelt, Martin, Pereira, 1987] δημιουργήθηκε για διαχειριστές βάσεων δεδομένων χωρίς μεγάλη εμπειρία. Το ASK [B.H. Thompson, F.B. Thompson, 1985] επέτρεπε στους χρήστες να μαθαίνουν στο σύστημα νέες

λέξεις και έννοιες, καθ’ όλη τη διάρκεια του διαλόγου. Είχε τη δυνατότητα να επικοινωνήσει με εξωτερικές βάσεις δεδομένων, όπως προγράμματα διαχείρισης ηλεκτρονικού ταχυδρομείου και άλλες παρόμοιες εφαρμογές. Όλες οι εφαρμογές ήταν διαθέσιμες στον χρήστη μέσω επικοινωνίας σε φυσική γλώσσα.

Με την πάροδο του χρόνου, αναπτύχθηκαν παρόμοια συστήματα που χρησιμοποιούσαν άλλες γλώσσες πέραν της Αγγλικής, όπως Nchiql [Wang, Meng, Liu, 1999], το οποίο είναι ένα σύστημα ερωτήσεων σε βάσεις δεδομένων το οποίο δέχεται ερωτήσεις σε κινέζικη γλώσσα.

Στο άρθρο [Marakakis, Mori, Radhakrishnan, Castillo, 1983] αναλύονται θέματα τα οποία αφορούν την δημιουργία ομιλούμενων απαντήσεων από την επεξεργασία ερωτήσεων σε βάσεις δεδομένων. Παρουσιάζεται ο σχεδιασμός ενός συστήματος στο οποίο αλληλεπιδρούν οι εξής τρεις περιοχές, βάσεις δεδομένων, δημιουργία κειμένου στην Αγγλική και σύνθεση φωνής.

7 ΣΥΜΠΕΡΑΣΜΑΤΑ

Το σύστημα EROTISIS είναι ένα σύστημα επεξεργασίας φυσικής γλώσσας, το οποίο δέχεται σαν είσοδο ερωτήσεις γραμμένες σε ελληνικά οι οποίες αφορούν μια βάση δεδομένων η οποία περιέχει πληροφορίες για το τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων (ΕΠΠ) του Τ.Ε.Ι. Ηρακλείου Κρήτης. Οι απαντήσεις οι οποίες επιστρέφονται στον χρήστη είναι τα δεδομένα από τη Βάση Δεδομένων, τα οποία ζητούνται στο ερώτημά του.

Το σύστημα EROTISIS υποστηρίζει μια περιορισμένη μορφή ερωτήσεων, είναι όμως επεκτάσιμο. Ένα άλλο σημαντικό χαρακτηριστικό του συστήματος EROTISIS είναι η δυνατότητα που προσφέρει για συνεργασία με άλλο αντίστοιχο σύστημα το οποίο να δημιουργεί απαντήσεις στην Ελληνική από τα αποτελέσματα του EROTISIS.

Τέλος, από όσο γνωρίζουμε, δεν υπάρχει κάποιο ανάλογο σύστημα το οποίο να μπορεί να προσφέρει διεπικοινωνία με βάσεις δεδομένων στην Ελληνική γλώσσα.

8 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ

Οι προοπτικές που έχει η τεχνογνωσία του συστήματος EROTISIS, είναι πολλές και μπορεί να εφαρμοστεί σε διάφορους τομείς. Όσον αφορά στις τεχνικές που ακολουθήθηκαν, το σύστημα μπορεί να αναπτυχθεί κάνοντάς το να αναλύει και να δίνει αποτελέσματα για πιο σύνθετες προτάσεις. Αρωγό σε αυτή τη προσπάθεια, θα αποτελέσει η επεκτασιμότητα του συστήματος, καθώς δε δημιουργήθηκε αποκλειστικά για ορισμένους τύπους ερωτήσεων, αλλά με σκοπό να προκαλέσει και να προσκαλέσει ανθρώπους, χωρίς αναγκαστικά μεγάλη εμπειρία σε γλώσσα προγραμματισμού Prolog ή βάσεις δεδομένων, να το αναπτύξουν.

Τομέας έρευνας θα μπορούσαν να αποτελέσουν τα πιθανά ορθογραφικά λάθη του χρήστη. Σίγουρα θα ήταν πιο βολικό για τον χρήστη αντί να ξαναγράψει την ερώτηση λόγω ενός ορθογραφικού λάθους ή ενός αναγραμματισμού, το σύστημα να "καταλαβαίνει" το λάθος αυτό και να το προσπερνάει, καταλαβαίνοντας το τι ήθελε στη πραγματικότητα να γράψει ο χρήστης.

Μια άλλη κατεύθυνση και αντικείμενο πολλών μελετών είναι η κατανόηση "ελλειπτικών προτάσεων", οι οποίες εξάλλου αποτελούν πλεονέκτημα της επεξεργασίας φυσικής γλώσσας σε σχέση με άλλες μεθόδους ανάκτησης δεδομένων από βάσεις δεδομένων. Για παράδειγμα κάνοντας την ερώτηση "ποιος καθηγητής διδάσκει το εργαστηριακό μάθημα προγραμματισμός", το σύστημα θα δώσει ένα επίθετο Χ. Στη συνέχεια, ο χρήστης θα μπορούσε να ρωτήσει "ποια διπλωματική επιβλέπει;" κάνοντας το σύστημα να καταλάβει ότι η ερώτηση αναφέρεται στον καθηγητή με επίθετο Χ.

Ακόμα, το σύστημα θα μπορούσε να είναι στο Internet, ώστε οι ενδιαφερόμενοι να βρίσκουν on-line τις πληροφορίες που αναζητούν.

Τομέας ο οποίος ήδη αναπτύσσεται το σύστημα EROTISIS, αποτελεί η

πτυχιακή εργασία του Ανταλή Γεώργιου, στην οποία τα στοιχεία της βάσης δεδομένων τα οποία το σύστημα βρίσκει σαν απάντηση, με τη βοήθεια των ήδη συντακτικά και γραμματικά αναλυμένων μελών της ερώτησης, δημιουργούν την απάντηση σε ολοκληρωμένη πρόταση της ελληνικής γλώσσας. Η προσπάθεια αυτή, προσδίδει στο σύστημα την αντίληψη του διαλόγου.

Το σύστημα EROTISIS θα μπορούσε να βρει εφαρμογή για χρήση οπουδήποτε θεωρείται απαραίτητη η ανάκτηση των δεδομένων τα οποία περιέχει η βάση δεδομένων. Οι φοιτητές του Ε.Π.Π. θα μπορούσαν για παράδειγμα τα βρουν το όνομα γραφείου ενός καθηγητή που δε γνωρίζουν το επίθετό του, αλλά γνωρίζουν ότι διδάσκει κάποιο συγκεκριμένο μάθημα, ή να δουν ποια μαθήματα έχουν ολοκληρώσει μέχρι στιγμής.

Άλλη προέκταση του συστήματος θα ήταν η ενημέρωση της βάσης δεδομένων να γίνεται σε φυσική γλώσσα. Δηλαδή, η καταχώρηση, η αφαίρεση και η αλλαγή κάποιας εγγραφής, από τη βάση δεδομένων να γίνεται στην ελληνική. Για παράδειγμα, "καταχώρησε τη νέα διπλωματική με τίτλο «ενημέρωση της βάσης γνώσεων σε ένα σύστημα απόδειξης ορθότητας λογικών προγραμμάτων. Η διπλωματική εκπονείται από τον φοιτητή Μ.Χαρδαλά. Ο επιβλέπων καθηγητής είναι ο Μ.Μαρακάκης»".

9 ΒΙΒΛΙΟΓΡΑΦΙΑ

Ελληνόγλωσση Βιβλιογραφία

[Ανδρουτσόπουλος, 1991]
Ανδρουτσόπουλος Ι. **Κατασκευή Λεκτικού και Συντακτικού Αναλυτή της Νέας Ελληνικής Γλώσσας**, Διπλωματική Εργασία, Τομέας Πληροφορικής ΕΜΠ, Αθήνα 1991.

[Βλαχάβας κ.α., 2002]
Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Ι. Ρεφανίδης, Φ. Κόκκορας, Η. Σακελλαρίου, ***Τεχνητή Νοημοσύνη***, 2002, Εκδόσεις Γαρταγάνη.

[Καλυβιανάκης, 2004]
Μ. Καλυβιανάκης, **Έμπειρο σύστημα Διάγνωσης Παιδικής Επιλειψίας**. Πτυχιακή εργασία, Τμήμα Ηλεκτρολογίας, ΤΕΙ Κρήτης, Ηράκλειο 2004.

[Κουνάλη, 2005]
Κουνάλη Χαρά, ***Αυτόματο Σύστημα Μετασχηματισμού Λογικών Προγραμμάτων,*** Πτυχιακή εργασία, Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων, ΤΕΙ Κρήτης, Ηράκλειο 2005.

[Μαίστρος, Μαρκαντωνάτου]
Μαίστρος Γιάννης , Μαρκαντωνάτου Στέλλα, **Σημειώσεις μαθήματος Γλώσσες Προγραμματισμού** ΙΙ, Ε. Μ. Πολυτεχνείο, Τομέας Πληροφορικής, ιστοσελίδα.

[Μαρακάκης, 2003]
Μαρακάκης Εμμ., **Τεχνητή Νοημοσύνη**, Σημειώσεις Θεωρίας Τεχνητής Νοημοσύνης, ΤΕΙ Κρήτης, Ηράκλειο 2003.

[Μαρακάκης, 2003]

Μαρακάκης Εμμ., **Προγραμματισμός ΙΙ**, Σημειώσεις Θεωρίας Προγραμματισμού ΙΙ, Κατασκευή Λογισμικού, ΤΕΙ Κρήτης, Ηράκλειο 2003.

[Μαρακάκης, 2006]
Μαρακάκης Εμμ., **Prolog: Προγραμματισμός για Τεχνητή Νοημοσύνη**, Σημειώσεις εργαστηρίου Τεχνητής Νοημοσύνης, ΤΕΙ Κρήτης, Ηράκλειο Σεπτέμβριος 2006.

[Συντακτικό Ελληνικής]
Συντακτικό της Νέας Ελληνικής γλώσσας, Α, Β, και Γ Γυμνασίου. ΟΑΕΔΒ, 2005, ISBN 960-06-00236.

[Rich, Knight, 1991]
Rich E. και Knight K., **Artificial Intelligence**, Second Edition (E. Rich and K. Knight), McGraw-Hill Book Company, 1991, Κεφάλαιο15. Επεξεργασία Φυσικής Γλώσσας, Μετάφραση Ν. Φακωτάκης.

Ξενόγλωσση βιβλιογραφία

[Androutsopoulos, 1992]
I. Androutsopoulos, **Interfacing a Natural Language Front-End to a Relational Database.** MSc thesis, University of Edinburgh, 1992.

[Androutsopoulos, Ritchie, Thanisch, 1993]
I. Androutsopoulos, G. Ritchie, and P. Thanisch, **An Efficient and Portable Natural Language Query Interface for Relational Databases**. In P.W. Chung, G. Lovegrove, and M. Ali, editors, Proceedings of the 6th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh, U.K., pages 327–330. Gordon and Breach Publishers Inc., Langhorne, PA, U.S.A., June 1993.ISBN 2–88124–604–4

[Androutsopoulos, Ritchie, Thanisch, 1995]
Androutsopoulos, I. and G.D. Ritchie and P. Thanisch, **Natural Language Interfaces to Databases – An Introduction,** Journal of Language Engineering, 1995. pages 28-81

[Boyer, Moore, 1979]
Boyer S. Robert and Moore J. Strother, ***A Computational Logic***, Academic Pers pectives In Computing, 1979, page 408.

[Bratko, 2001]
Ivan Bratko, ***Prolog Programming for Artificial Intelligence***, Third Edition, 2001, Addison-Wesley.

[Chen, 1976]
P.Chen, **The Entity-Relationship Model-Toward a Unified View of Data**, ACM Transactions on Database Systems vol.I No 1, March 1976, pages 9-36.

[Codd, 1974]

E.F. Codd**, Seven Steps to RENDEZVOUS with the Casual User**. In J. Kimbie and K. Koffeman, editors, Data Base Management. North-Holland Publishers, 1974

[Cohen, 1985]
Cohen Jacques, ***Describing Prolog by its Interpretation and Compilation***, Communications of the ACM, December 1985. Pages: 1311 – 1324.

[Convington etal, 1997]
Michael A. Convington, Donald Nute, Andre Vellino, ***Prolog Programming in Depth***, 1997, Prentice – Hall.

[Grosz, Appelt, Martin, Pereira, 1987]
B.J. Grosz, D.E. Appelt, P.A. Martin, and F.C.N. Pereira, **TEAM**: **An Experiment in the Design of Transportable Natural-Language Interfaces**. Artificial Intelligence,32:173–243, 1987

[Hendrix, Sacerdoti, Sagalowicz, Slocum, 1978]
G. Hendrix, E. Sacerdoti, D. Sagalowicz, and J. Slocum, **Developing a Natural Language Interface to Complex Data**. ACM Transactions on Database Systems, 3(2):105–147,1978.

[H. Thompson, F. Thompson, 1985]
B.H. Thompson, F.B. Thompson**, ASK** is Transportable in Half a Dozen Ways ACM Transactions on Office Information Systems, 3(2):185–203, April 1985.

[Lloyd, 1987]
Lloyd J.W., **Foundations of Logic Programming**, Second, Extended Edition 1987.

[Nilsson, Maluszynski, 1996]

Ulf Nilsson, Jan Maluszynski, ***Logic Programming and Prolog***, Second Edition, 1996, John Wiley & Sons.

[Radhakrishnan, De Mori, Marakakis, Castillo, 1983]
T. Radhakrishnan, R. De Mori, E. Marakakis, R. Castillo, **Spoken Responses to Database Queries,** proceedings of the international Conference of Systems, Man and Cybernetics, Dec 30 1983 - Jan 1984, India Vol. II, pages 825-829, 1983 IEEE Catalog No 83CH 1962-0, ISBN 0-08382683-0.

[Rich, Knight, 1991]
E. Rich, K. Knight, ***Artificial Intelligence***, Second Edition, 1991, McGrawttill.

[SICS, 2003]
Swedish Institute of Computer Science, ***SICStus Prolog 3.11 User's Manual***, 2004.

[Sterling, Shapiro, 1999]
Leon Sterling, Ehud Shapiro, ***The Art of Prolog***, Second Edition, 1999, The MIT Press.

[Wang, Meng, Liu, 1999]
Shan Wang, Xiao Feng Meng, Shuang Liu, **Nchiql**: a Chinese natural language query system to databases. 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), pages 453-460.

[Warren, Pereira, 1982]

D. Warren and F. Pereira, **An Efficient Easily Adaptable System for Interpreting Natural Language Queries**. Computational Linguistics, 8(3-4):110–122, July-December 1982.

[Woods, Kaplan, Webber, 1972]
W.A. Woods, R.M. Kaplan, and B.N. Webber, **The Lunar Sciences Natural Language Information System**: Final Report. BBN Report 2378, Bolt Beranek and Newman Inc.,Cambridge, Massachusetts, 1972.

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α: Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Αναφέρονται ενδεικτικά λίγες εγγραφές από κάθε σχέση σύμφωνα με το σχήμα της Βάσης Δεδομένων όπως περιγράφεται στο κεφάλαιο 4.1 καθώς και η αντίστοιχη υλοποίησή τους σαν Prolog γεγονότα.

Α.1 Στιγμιότυπα Σχέσεων

foitites / φοιτητές

AM	Epitheto_foititi	Onoma_foititi	Sem_Eisagogis	Tel	Mail_foititi
642	Βαρουξής	Κωνσταντίνος	01-02Ε	6937090404	kvarouxis@walla.com
713	Ανταλής	Γεώργιος	01-02Ε	2810319528	g_andalis@yahoo.gr
680	Αθανασιάδης	Αθανάσιος	01-02Ε	6972687798	epp680@epp.teiher.gr

mathimata / μαθήματα

Kod_mathimatos	Onoma_mathimatos	Sem_mathimatos	Tupos_mathimatos	Did_monades
ΤΠ4004	Δίκτυα δεδομένων	Δ	Υ	8
ΤΠ5003	Δίκτυα υπολογιστών	Ε	Υ	6
ΤΠ5007	Τεχνητή νοημοσύνη	Ε	Π	5

leptomereies mathimaton / λεπτομέρειες μαθημάτων

Kod_math_T-E	Eidos_mathimatos	Ores_mathimatos
ΤΠ4004α	Θ	4
ΤΠ4004β	Ε	3
ΤΠ5003α	Θ	2
ΤΠ5003β	Ε	3

aithouses / αίθουσες

Kod_aithousas	Onoma_aithousas	Capacity
ΚΑ101	Αμφ Σμυρνάκη	99
ΚΑ102	ΠΚ2	30
ΚΑ103	ΝΚ1	50
ΚΑ104	Εργ10	30

ptyxiakes / πτυχιακές

Kod_ptyxiakis	Titlos_ptyxiakis
ΠΤ001	Δημιουργία συστήματος ερωταποκρίσεων
ΠΤ002	Δημιουργία προτάσεων με χρήση απαντήσεων από βάση δεδομένων

kathigites / καθηγητές

Kod_kathigiti	Epitheto_kathigiti	Onoma_kathigiti	Thl_grafeiou	Kod_Grafeio_kathigiti	Mail_ kathigiti
002	Αποστολάκης	Σπύρος	----	ΣΤΕΦ1	----
003	Σιδέρης	Αργύρης	----	ΣΤΕΦ2	----
004	Μαρακάκης	Μανόλης	379748	ΠΚ1	mmarak@teicrete.gr
005	Μπιτσάκη	Μαρίνα	----	ΝΚ2	----

apoteleitai apo / αποτελείται από

Kod_mathimatos	Kod_math_T_E
ΤΠ4004	ΤΠ4004α
ΤΠ4004	ΤΠ4004β
ΤΠ5003	ΤΠ5003α
ΤΠ5003	ΤΠ5003β

ΤΠ7003	ΤΠ7003α

parakolouthoun / παρακολουθούν

AM	Kod_math_T_E	Per_parakolouthisis
680	ΤΠ4004α	05-06Ε
713	ΤΠ5003α	05-06Ε
642	ΤΠ7003α	05-06Ε
713	ΤΠ7003α	05-06Ε
642	ΤΠ3001α	05-06Ε

proapaitoumena / προαπαιτούμενα

Kod_mathimatos	Kod_proap_mathimatos
ΤΠ5003	ΤΠ4004
ΤΠ3001	ΤΠ2001

exei perasei / έχει περάσει

AM	Kod_math_T_E	Vathmos_mathimatos	Hmer_eksetasis
642	ΤΠ4004α	6.00	04-05Χ
642	ΤΠ4004β	8.50	04-05Χ
642	ΤΠ5003α	5.50	05-06Χ
642	ΤΠ5003β	5.50	05-06Χ
931	ΤΠ7003α	8.50	04-05Χ

ekponei / εκπονεί

AM	Kod_ptuxiakis	Vathmos_ptyxiakis
642	ΠΤ001	----
713	ΤΠ002	---

epivlepoun / επιβλέπουν

Kod_ptyxiakis	Kod_kathigiti
ΠΤ001	004
ΠΤ002	004

didaskoun / διδάσκουν

Kod_kathigiti	Kod_math_T_E
001	ΤΠ4004α
002	ΤΠ4004β
003	ΤΠ5003β
005	ΤΠ5003α
004	ΤΠ5007α
004	ΤΠ5007β
006	ΤΠ7003α

didaskontai se / διδάσκονται_σε

Kod_math_Th_E	Kod_aithousas	Period_didaskal	Day	Hour
ΤΠ4004α	ΚΑ101	04-05Χ	Δευ	16:15
ΤΠ4004α	ΚΑ101	05-06Ε	Δευ	16:15
ΤΠ4004β	ΚΑ102	04-05Χ	Τρ	10:15
ΤΠ5003β	ΚΑ104	05-06^{Ε}	Δευ	08:15
ΤΠ5007α	ΚΑ106	05-06Χ	Τετ	10:15
ΤΠ6006α	ΚΑ105	05-06Χ	Παρ	16:15

Α.2 Υλοποίηση σε Prolog Βάσης Δεδομένων

foitites/7

foitites(642, βαρουξής, κωστής, αρσ, εαρ0102, 6937090404, mail).

foitites(713, ανταλής, γιώργος, αρσ, εαρ0102, 2810319528, mail).

mathimata/5

mathimata(τπ5007, 'τεχνητή νοημοσύνη', ε, επιλογής, 5).

mathimata(τπ2001, 'προγραμματισμός', β, υπο, 7).

leptomereies_mathimaton/3

leptomereies_mathimaton(τπ4004α, εργαστηριακό, 4).

leptomereies_mathimaton(τπ4004β, εργαστηριακό, 3).

leptomereies_mathimaton(τπ5003α, θεωρητικό, 2).

aithouses/3

aithouses(κα101, αμφ σμυρνάκη, 99).

aithouses(κα102, πκ2, 30).

aithouses(κα103, νκ1, 50).

ptyxiakes/2

ptyxiakes(πτ1001, 'σύστημα ερωταπαντήσεων').

ptyxiakes(πτ1002, 'δημιουργία απάντησης').

kathigites/7

kathigites(004, μαρακάκης, μανόλης, αρσ, πκ1, 2810379748, mail).

kathigites(005, μπιτσάκη, μαρίνα, θηλ, νκ2, -, -).

kathigites(006, κορτσιδάκη, αγνή, θηλ, νκ3, -, -).

apoteleitai_apo/3

apoteleitai_apo(τπ4004, τπ4004α, τπ4004β).

apoteleitai_apo(τπ6006, τπ6006α, τπ6006β).

apoteleitai_apo(τπ5003, τπ5003α, τπ5003β).

parakolouthoun/3

parakolouthoun(713, τπ5003α, εαρ0506).

parakolouthoun(642, τπ7003α, εαρ0506).

parakolouthoun(713, τπ7003α, εαρ0506).

parakolouthoun(642, τπ3001α, εαρ0506).

proapaitoumena/2

proapaitoumena(τπ5003, τπ4004).

proapaitoumena(τπ3001, τπ2001).

exei_perasei/4

exei_perasei(642, τπ4004α, 6.00, εαρ0405).

exei_perasei(642, τπ4004β, 8.50, εαρ0405).

exei_perasei(642, τπ5003α, 5.50, εαρ0506).

exei_perasei(713, τπ3001α, 5.50, εαρ0506).

exei_perasei(713, τπ4004α, 5.50, εαρ0506).

ekponei/3

ekponei(642, πτ1001, 'Vathmos_ΠΤΥ').

ekponei(713, πτ1002, 'Vathmos_ΠΤΥ').

Σημείωση: *Vathmos_ΠΤΥ είναι μια μεταβλητή η οποία δείχνει στην Prolog ότι ακόμα δε μπήκε βαθμός στην πτυχιακή*

epivlepoun/2

epivlepoun(004, πτ1001).

epivlepoun(004, πτ1002).

didaskoun/2

didaskoun(001, τπ4004α).
didaskoun(002, τπ4004β).
didaskoun(003, τπ5003β).

didaskontai_se/5

didaskontai_se(τπ4004α, κα101, εαρ0405, δευτέρα, '16:15').
didaskontai_se(τπ4004α, κα101, εαρ0506, τρίτη, '16:15').
didaskontai_se(τπ4004β, κα102, εαρ0506, τετάρτη, '10:15').
didaskontai_se(τπ5003β, κα104, εαρ0506, πέμπτη, '8:15').

ΠΑΡΑΡΤΗΜΑ Β: ΤΟ ΛΕΞΙΚΟ ΔΕΔΟΜΕΝΩΝ

Β.1 Στιγμιότυπα Λεξικού Δεδομένων

greeklish_greek_rel_names

Όνομα σχέσης	Ρίζα λέξης στην πρόταση
Foitites	φοιτητ
Mathimata	μαθημ
Kathigites	παρακολουθ

greeklish_greek_attr_names

Ονομα των πεδίων σχέσης	Ονομασία σε φυσική γλώσσα
Onoma_foititi	όνομα
Epitheto_foititi	επίθετο
Kod_kathigiti	κωδικός καθηγητής
Epitheto_kathigiti	επίθετο καθηγητή

columns/3

Όνομα σχέσης	Όνομα πεδίου	Αριθμός Στήλης
foitites	AM	1
foitites	Epitheto_foititi	2
mathimata	Onoma_Mathimatos	2
mathimata	Did_monades	3

Key

Κλειδί	Όνομα σχέσης	Αριθμός Στήλης	Είδος κλειδιού	Όνομα πεδίου	Κενη θεση 1	Κενη θεση 2	Κενη θεση 3
AM	Foitites	1	primary	AM	_	_	_
Epitheto_foititi	Foitites	2	Alternative	Mail_foititi	_	_	_
Onoma_Mathimatos	Mathimata	2	Primary	Kod_mathimatos	_	_	_
Did_monades	Mathimata	3	Alternative	Onoma_mathimatos	_	_	_

abbr_attr

Συντόμευση	Ορισμός συντόμευσης
tel	τηλέφωνο
ΑΜ	αριθμός μητρώου
Αμφ	αμφιθέατρο

Abbr_vsets

Όνομα πεδίου	Τύπος πληροφορίας	Πεδίο τιμών
AM	integer(5)	0-99999
Onoma_foititi	character(20)	ALL
Epitheto_foititi	character(20)	ALL
Sem_Eisagogis	string	Year/Semester
Tel	string	ALL

Synonyms

Συνώνυμο πεδίο	Συνώνυμο πεδίο	Συνώνυμο σε φυσική γλώσσα
AM	Onoma_foititi	φοιτητής
AM	Epitheto_foititi	φοιτητής
Kod_mathimatos	Onoma_mathimatos	μάθημα
Kod_mathimatos	Kod_mathimatos	μάθημα

Β.2. Υλοποίηση σε Prolog του Λεξικού Δεδομένων

greeklish_greek_rel_names/2

greeklish_greek_rel_names(foitites, 'φοιτητ').
greeklish_greek_rel_names(mathimata,'μάθημ').
greeklish_greek_rel_names(kathigites, 'καθηγητ').
greeklish_greek_rel_names(parakolouthoun, 'παρακολουθ').

greeklish_greek_attr_names/2

greeklish_greek_attr_names('Onoma_foititi', 'όνομα').
greeklish_greek_attr_names('Epitheto_foititi', 'επίθετο').
greeklish_greek_attr_names('Kod_kathigiti', 'κοδικό').
greeklish_greek_attr_names('Onoma_kathigiti', 'όνομα').
greeklish_greek_attr_names('Epitheto_kathigiti', 'επίθετο').

columns/3

columns(foitites, 'AM', 1).
columns(foitites, 'Epitheto_foititi', 2).
columns(mathimata, 'Onoma_mathimatos', 2).
columns(mathimata, 'Did_monades', 3).

key/8

key('K1', foitites, 1, primary, 'AM', '_', '_', '_').
key('K2', foitites, 1, alternative, 'Mail_foititi', '_', '_', '_').
key('K3', mathimata, 1, primary, 'Kod_mathimatos', '_', '_', '_').
key('K4', mathimata, 1, alternative, 'Onoma_mathimatos', '_', '_', '_').

abbr_attr/2

abbr_attr('Tel', 'Τηλέφωνο').

abbr_attr('AM', 'Αριθμός Μητρώου').
abbr_attr('Αμφ', 'Αμφιθέατρο').

attr_vsets/3

attr_vsets('AM', 'integer(5)', '0 - 99999').
attr_vsets('Onoma_foititi', 'character(20)', 'ALL').
attr_vsets('Epitheto_foititi', 'character(20)', 'ALL').
attr_vsets('Sem_Eisagogis', 'string', 'Year / Semester').
attr_vsets('Tel', 'string', 'ALL').

synonyms/3

synonyms('AM', 'Onoma_foititi', 'φοιτητής').
synonyms('AM', 'Eponymo_foititi', 'φοιτητής').
synonyms('kod_mathimatos', 'Onoma_mathimatos', 'μάθημα').
synonyms('kod_mathimatos', 'Kod_proap_mathimatos', 'μάθημα').

ΠΑΡΑΡΤΗΜΑ Γ: ΤΟ ΛΕΞΙΚΟ ΤΗΣ ΕΛΛΗΝΙΚΗΣ ΓΛΩΣΣΑΣ

Γ.1 Στιγμιότυπα Λεξικού της Ελληνινής Γλώσσας

Port_rima

Όνομα συσχέτισης	Ρίζα ρήματος
ολοκλήρωσε	ολοκλήρω
διδάσκουν	Διδάσκ
εκπονεί	Εκπον

prot_erotimatiki_antonimia

Ερωτηματική αντωνυμία	Ρίζα ερωτηματικής Αντωνυμίας
ολοκλήρωσε	ολοκλήρω
διδάσκουν	διδάσκ
Εκπονεί	εκπον

prot_arthro

Κλειδί	Άρθρο	Αριθμός	γένος
1	Ο	ενικός	αρσενικό
5	των	πληθυντικός	αρσενικό
8	της	ενικός	θηλυκό

prot_ousiastiko

κλειδί	ουσιαστικό	Ρίζα ουσιαστικού
1	φοιτητής	φοιτητ
5	καθηγητής	καθηγητ
14	εκπον	μαθήμ

prot_epitheto

κλειδί	Επίθετο	Ρίζα επιθέτου
1	εργαστηριακό	εργαστηριακ
2	εαρινό	εαριν
7	θεωρητικό	θεωρητικ

Γ.2 Υλοποίηση Λεξικού της Ελληνικής Γλώσσας σε Prolog

```
prot_rima(1, 'ολοκλήρωσε', 'ολοκλήρωσ').
prot_rima(2, 'ολοκλήρωσαν', 'ολοκλήρωσ').
prot_rima(3, 'διδάσκει', 'διδάσκ').
prot_rima(4, 'διδάσκουν', 'διδάσκ').
prot_rima(5, 'εκπονεί', 'εκπον').
prot_rima(6, 'εκπονούν', 'εκπον').
prot_rima(7, 'επιβλέπει', 'επιβλέπ').
prot_rima(8, 'επιβλέπουν', 'επιβλέπ').
prot_rima(9, 'παρακολουθούν', 'παρακολουθ').
prot_rima(10, 'παρακολουθεί', 'παρακολουθ').
prot_rima(11, 'διδάσκονται', 'διδάσκ').
prot_rima(12, 'διδάσκεται', 'διδάσκε').

prot_erotimatiki_antonimia(1, 'ποιός', 'ποι').
prot_erotimatiki_antonimia(2, 'ποιά', 'ποι').
prot_erotimatiki_antonimia(3, 'ποιό', 'ποι').
prot_erotimatiki_antonimia(4, 'ποιοί', 'ποι').
prot_erotimatiki_antonimia(5, 'ποιές', 'ποι').
prot_erotimatiki_antonimia(6, 'ποιά', 'ποι').
prot_erotimatiki_antonimia(7, 'πού', 'πού').

prot_arthro(1, 'ο', enikos, arseniko).
prot_arthro(2, 'του', enikos, arseniko).
prot_arthro(3, 'τον', enikos, arseniko).
prot_arthro(4, 'οι', plithintikos, arseniko).
prot_arthro(5, 'των', plithintikos, arseniko).
prot_arthro(6, 'τους', plithintikos, arseniko).
prot_arthro(7, 'η', enikos, thiliko).
prot_arthro(8, 'της', enikos, thiliko).
prot_arthro(9, 'την', enikos, thiliko).
```

```
prot_arthro(10, 'τις', plithintikos, thiliko).
prot_arthro(11, 'το', enikos, oudetero).
prot_arthro(12, 'τα', plithintikos, oudetero).
prot_arthro(13, 'οι', plithintikos, thiliko).


prot_ousiastiko(1, 'φοιτητής', 'φοιτητ').
prot_ousiastiko(2, 'φοιτητές', 'φοιτητ').
prot_ousiastiko(3, 'φοιτήτρια', 'φοιτήτ').
prot_ousiastiko(4, 'φοιτήτριες', 'φοιτήτ').
prot_ousiastiko(5, 'καθηγητής', 'καθηγητ').
prot_ousiastiko(6, 'καθηγητές', 'καθηγητ').
prot_ousiastiko(7, 'καθηγήτρια', 'καθηγήτ').
prot_ousiastiko(8, 'καθηγήτριες', 'καθηγήτ').
prot_ousiastiko(9, 'πτυχιακή', 'πτυχιακ').
prot_ousiastiko(10, 'διπλωματική', 'διπλωματικ').
prot_ousiastiko(11, 'αίθουσα', 'αίθουσ').
prot_ousiastiko(12, 'αίθουσες', 'αίθουσ').
prot_ousiastiko(13, 'μάθημα', 'μάθημ').
prot_ousiastiko(14, 'μαθήματα', 'μάθημ').
prot_ousiastiko(15, 'εξάμηνο', 'εξάμην').
prot_ousiastiko(16, 'εξάμηνα', 'εξάμην').
prot_ousiastiko(17, 'έτος', 'έτ').
prot_ousiastiko(18, 'έτη', 'έτ').
prot_ousiastiko(19, 'βαθμό', 'βαθμ').
prot_ousiastiko(20, 'τηλέφωνο', 'τηλέφων').
prot_ousiastiko(21, 'mail', 'mai').
prot_ousiastiko(22, 'επίθετο', 'επίθετ').
prot_ousiastiko(23, 'τηλέφωνο', 'τηλέφων').
prot_ousiastiko(24, 'όνομα', 'όνομ').
prot_ousiastiko(25, 'διπλωματική', 'διπλωματικ').
prot_ousiastiko(26, 'βαθμό', 'βαθμ').
```

prot_epitheto(1, 'εργαστηριακό', 'εργαστηριακ').
prot_epitheto(2, 'εργαστηριακά', 'εργαστηριακ').
prot_epitheto(3, 'μεγαλύτερο', 'μεγαλύτερ').
prot_epitheto(4, 'μικρότερο', 'μικρότερ').
prot_epitheto(5, 'εαρινό', 'εαριν').
prot_epitheto(6, 'εαρινά', 'εαριν').
prot_epitheto(7, 'θεωρητικό', 'θεωρητικ').

ΠΑΡΑΡΤΗΜΑ Δ: ΚΑΝΟΝΕΣ ΣΥΝΤΑΚΤΙΚΗΣ ΑΝΑΛΥΣΗΣ

Δ.1 Κανόνες BNF

Πρόταση → <Ρηματικό_Μέρος, Ονοματικό_Μέρος> | <Ονοματικό_Μέρος, Ρηματικό_Μέρος>.

Ρηματικό_Μέρος → <Αντωνυμία, Αντικείμενο, Ρήμα> | <Αντωνυμία, Ρήμα> | <Ρήμα, Αντικείμενο, Προσδιορισμός>.

Ονοματικό_Μέρος → <Υποκείμενο, Προσδιορισμός> | <Υποκείμενο> | <Υποκείμενο, Προσδιορισμός, Προσδιορισμός >.

Αντικείμενο → <Προσδιορισμός, Ουσιαστικό_λέξη> | <Προσδιορισμός, Ουσιαστικό_λέξη, Στοιχείο> | <Άρθρο, Ουσιαστικό_λέξη> | <Άρθρο, Ουσιαστικό_λέξη, Στοιχείο> | < Άρθρο, Ουσιαστικό_λέξη, Προσδιορισμός> | < Άρθρο, Προσδιορισμός, Ουσιαστικό_λέξη>.

Υποκείμενο → <Άρθρο, Ουσιαστικό_λέξη> | <Άρθρο, Ουσιαστικό_λέξη, Στοιχείο> | <Άρθρο, Προσδιορισμός, Ουσιαστικό_λέξη, Στοιχείο> | <Προσδιορισμός, Ουσιαστικό_λέξη> | <Αντωνυμία, Ουσιαστικό_λέξη>.

Προσδιορισμός→<Επιθετικός_Προσδιορισμός>|<Εμπρόθετος_Προσδιορισμός> | <Επιρρηματικός_Προσδιορισμός_Χρόνου >.

Επιθετικός_Προσδιορισμός → <Επίθετο> | <Επίθετο, Σύνδεσμος, Επίθετο>

Εμπρόθετος_Προσδιορισμός → <Πρόθεση, Ουσιαστικό_λέξη, Στοιχείο>

Επιρρηματικός_Προσδιορισμός_Χρόνου → <Άρθρο, Επίθετο, Ουσιαστικό_λέξη, Στοιχείο>.

Αντωνυμία → <ποιός> | <ποιοί> | <ποιό> | <ποιά> | <ποιές> | <πού>.

Επίθετο → <εργαστηριακό> | < θεωρητικό> | <εργαστηριακά> | <θεωρητικά>.

Ουστιαστικό_λέξη → <καθηγητής> | <καθηγητές> | <καθηγήτρια> | <καθηγήτριες> | <φοιτητής> | <φοιτητές> | <φοιτήτρια> | <φοιτήτριες> | <μάθημα> | <μαθήματα> | <τηλέφωνο> | <επίθετο> | <όνομα> | <τίτλο> | <mail> | <αμ> | <κωδικό > | <διπλωματική> | <βαθμό> | <εξάμηνο>.

Άρθρο → <τα> | <το> | <ο> | <οι> | <η> | <οι> | <>.

Σύνδεσμος → <και> | <ή>.

Πρόθεση → <με>.

Ρήμα → <διδάσκει> | <διδάσκουν> | <εκπονεί> | <παρακολουθεί> | <διδάσκουν> | <παρακολουθούν> | <επιβλέπει> | <ολοκλήρωσε> | <διδάσκεται>.

Αμ → <642> | <680> | <713>.

Όνομα → <βαρουξής> | <ανταλής> | <αθανασιάδης> | <παπαδογιάννη>.

Μάθημα → <προγραμματισμός> | <java> | <δίκτυα δεδομένων> | <εισαγωγή στο marketing> | <τεχνητή νοημοσύνη> | <δίκτυα υπολογιστών>.

Κοδικός_Μαθήματος → <τπ4004> | <τπ4004α> |<τπ4004β> | <τπ5007> |<τπ5007α> | <τπ5007β> | <τπ2001> | <τπ2001α> | <τπ2001β>.

ΒΑΘΜΟΣ → <5> | <6> | <7> | <8> | <9>.

Εξάμηνο → <0102εαρ> | <0405εαρ>

Όνομα → <κωνσταντίνος> | <γιώργος> | <θανάσης> | <δέσποινα>

Τηλέφωνο → <6937090404> | <2810319528> | <6972687798> | <2810379748>.

Mail → < kvarouxis@hotmail.com > | < g_andalis@yahoo.gr >| < epp680@epp.teiher.gr >.

Κωδικός → <001> | <002> | <003> | <004> | <005> | <006> | <007> <008> | <009> | <010> | <010> | <012> | <013>.

Όνομα → <αποστολάκης> | <σιδέρης> | <μαρακάκης> | <μπιτσάκη> | <κορτσιδάκη> | <μαστοράκης> | <στρατάκης> | <ξεζωνάκης> | <ασκοξυλάκης> | <βαρδιάμπασης> | <πάλλης> | <αιβαλής> | <βασίλης> | <σπύρος> | <αργύρης> | <μανόλης> | <μαρίνα> | <αγνή> | <γιώργος> | <δημήτριος> | <γιάννης> | <ευάγγελος> | <κοστής>.

Δ.2 Κανόνες DCG

Αυτοί οι DCG κανόνες αντιστοιχούν στους BNF κανόνες του παραρτήματος Δ.1, όπως ακριβώς έχουν υλοποιηθεί σε Prolog.

```
protasi(prot(RIM_MEROS, ONOM_MEROS)) -->
        rimatiko_meros(RIM_MEROS,
        Arithmos), onomatiko_meros(ONOM_MEROS, Arithmos1).

protasi(prot(ONOM_MEROS, RIM_MEROS)) -->
        onomatiko_meros(ONOM_MEROS, Arithmos1),
        rimatiko_meros(RIM_MEROS, Arithmos).

rimatiko_meros(rimatiko_meros(ANTONIMIA, ANTIKEIMENO, RIMA),
Arithmos) -->
         antonimia(ANTONIMIA, Arithmos, Genos),
        antikeimeno(ANTIKEIMENO, Arithmos, Genos), rima(RIMA,
        Arithmos1).

rimatiko_meros(rimatiko_meros(ANTONIMIA, RIMA), Arithmos) -->
        antonimia(ANTONIMIA, Arithmos, Genos), rima(RIMA,
         Arithmos1).

rimatiko_meros(rimatiko_meros(RIMA, ANTIKEIMENO), Arithmos) -->
        rima(RIMA, Arithmos1), antikeimeno(ANTIKEIMENO, Arithmos,
        Genos).

rimatiko_meros(rimatiko_meros(RIMA, ANTIKEIMENO, PROSDIORISMOS),
Arithmos) -->
        rima(RIMA, Arithmos1), antikeimeno(ANTIKEIMENO, Arithmos,
        Genos), prosdiorismos(PROSDIORISMOS, Arithmos1, _).
```

```
%rimatiko_meros(rimatiko_meros(RIMA, ARTHRO, PROSDIORISMOS,
ANTIKEIMENO), Arithmos) -->
                rima(RIMA, Arithmos1), arthro(ARTHRO, Arithmos, Genos),
                antikeimeno(ANTIKEIMENO, Arithmos, Genos).

onomatiko_meros(onomatiko_meros(IPOKEIMENO, PROSDIORISMOS),
Arithmos1) -->
                 ipokeimeno(IPOKEIMENO, Arithmos1),
                prosdiorismos(PROSDIORISMOS, Arithmos1, _).

onomatiko_meros(onomatiko_meros(IPOKEIMENO), Arithmos1) -->
                 ipokeimeno(IPOKEIMENO, Arithmos1).

onomatiko_meros(onomatiko_meros(IPOKEIMENO, PROSDIORISMOS,
PROSDIORISMOS1), Arithmos1) -->
                 ipokeimeno(IPOKEIMENO, Arithmos1),
                prosdiorismos(PROSDIORISMOS, Arithmos1, _),
                prosdiorismos(PROSDIORISMOS1, Arithmos1, _).

antikeimeno(antikeimeno(PROSDIORISMOS, OUSIASTIKO_LEKSI), Arithmos,
Genos) -->
                 prosdiorismos(PROSDIORISMOS, Arithmos, Genos),
                ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos).

antikeimeno(antikeimeno(ARTHRO, PROSDIORISMOS, OUSIASTIKO_LEKSI,
STOIXEIO), Arithmos, Genos) -->
                arthro(ARTHRO, Arithmos, Genos),
                prosdiorismos(PROSDIORISMOS, Arithmos, Genos),
                ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos),
                stoixeio(STOIXEIO, _, _).

antikeimeno(antikeimeno(ARTHRO,OUSIASTIKO_LEKSI),Arithmos,Genos)-->
```

```
        arthro(ARTHRO, Arithmos, Genos),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos).

antikeimeno(antikeimeno(ARTHRO, OUSIASTIKO_LEKSI, STOIXEIO),
Arithmos, Genos) -->
        arthro(ARTHRO, Arithmos, Genos),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos),
        stoixeio(STOIXEIO, _, _).

antikeimeno(antikeimeno(ARTHRO, OUSIASTIKO_LEKSI, PROSDIORISMOS),
Arithmos, Genos) -->  arthro(ARTHRO, Arithmos, Genos),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos),
        prosdiorismos(PROSDIORISMOS, Arithmos1, _).

antikeimeno(antikeimeno(ARTHRO, PROSDIORISMOS, OUSIASTIKO_LEKSI),
Arithmos, Genos) -->
        arthro(ARTHRO, Arithmos, Genos),
        prosdiorismos(PROSDIORISMOS, Arithmos, Genos),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos).

%antikeimeno(antikeimeno(ARTHRO, PROSDIORISMOS,
OUSIASTIKO_LEKSI, PROSDIORISMOS), Arithmos,Genos)-->
        arthro(ARTHRO, Arithmos, Genos),
        prosdiorismos(PROSDIORISMOS1, Arithmos, _),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos),
        prosdiorismos(PROSDIORISMOS, Arithmos1, _).

ipokeimeno(ipokeimeno(ARTHRO, OUSIASTIKO_LEKSI), Arithmos1) -->
        arthro(ARTHRO, Arithmos1, Genos),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos1, Genos).

ipokeimeno(ipokeimeno(ARTHRO, OUSIASTIKO_LEKSI, STOIXEIO),
```

```
Arithmos1) -->
        arthro(ARTHRO, Arithmos1, Genos),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos1, Genos),
        stoixeio(STOIXEIO, _, _).

ipokeimeno(ipokeimeno(ARTHRO, PROSDIORISMOS, OUSIASTIKO_LEKSI,
STOIXEIO), Arithmos) -->
        arthro(ARTHRO, Arithmos, Genos),
        prosdiorismos(PROSDIORISMOS, Arithmos, _),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos),
        stoixeio(STOIXEIO, _, _).

ipokeimeno(ipokeimeno(ARTHRO, PROSDIORISMOS, OUSIASTIKO_LEKSI),
Arithmos) -->
        arthro(ARTHRO, Arithmos, Genos),
        prosdiorismos(PROSDIORISMOS, Arithmos, _),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos).


ipokeimeno(ipokeimeno(ANTONIMIA, OUSIASTIKO_LEKSI), Arithmos) -->
        antonimia(ANTONIMIA, Arithmos, Genos),
        ousiastiko_leksi(OUSIASTIKO_LEKSI, Arithmos, Genos).

prosdiorismos(prosdiorismos(EPITHETIKOS_PROSD),Arithmos,Genos)-->
        epithetikos_prosdiorismos(EPITHETIKOS_PROSD, Arithmos,
        Genos).

prosdiorismos(prosdiorismos(EMPROTHETOS_PROSD), Arithmos, Genos) -->
        emprothetos_prosdiorismos(EMPROTHETOS_PROSD, Arithmos,
        Genos).

prosdiorismos(prosdiorismos(EPIRRIM_PROSD_XRON),Arithmos,Genos) -->
```

```
 epirrimat_prosdiorismos_xronou(EPIRRIM_PROSD_XRONOY, Arithmos,
Genos).

epithetikos_prosdiorismos(epithetikos_prosdiorismos(EPITHETO), Arithmos,
Genos) -->
        epitheto(EPITHETO, Arithmos, Genos).

epithetikos_prosdiorismos(epithetikos_prosdiorismos(EPITHETO,
SYNDESMOS, EPITHETO1), Arithmos, Genos) -->
        epitheto(EPITHETO, Arithmos, Genos),
        syndesmos(SYNDESMOS), epitheto(EPITHETO1, Arithmos,
        Genos).

emprothetos_prosdiorismos(emprothetos_prosdiorismos(PROTHESI,
OUSIASTIKO_LEKSI, STOIXEIO), Arithmos, Genos) -->
        prothesi(PROTHESI), ousiastiko_leksi(OUSIASTIKO_LEKSI,
        Arithmos, Genos), stoixeio(STOIXEIO, _, _).

epirrimat_prosdiorismos_xronou(epirrimat_prosdiorismos_xronou(ARTHRO,
EPITHETO, OUSIASTIKO_LEKSI, STOIXEIO), Arithmos, Genos) -->
        arthro(ARTHRO, Arithmos, Genos), epitheto(EPITHETO,
        Arithmos, Genos), ousiastiko_leksi(OUSIASTIKO_LEKSI,
        Arithmos, Genos), stoixeio(STOIXEIO, _, _).



stoixeio(stoixeio(ONOMA), _, _) -->  onoma(ONOMA).
stoixeio(stoixeio(TEL), _, _) -->  tilefono(TEL).
stoixeio(stoixeio(MAIL), _, _) -->  mail(MAIL).
stoixeio(stoixeio(AM), _, _) -->  am(AM).
stoixeio(stoixeio(ΚΟΔΙΚΟ), _, _) -->  code(ΚΟΔΙΚΟ).
stoixeio(stoixeio(ΒΑΘΜΟ), _, _) -->  grade(ΒΑΘΜΟ).
```

stoixeio(stoixeio(MATHIMA), _, _) **-->** mathima(MATHIMA).
stoixeio(stoixeio(KOD_MATHIMA), _, _) **-->** kod_mathima(KOD_MATHIMA).
stoixeio(stoixeio(DATE), _, _) **-->** semester(DATE).

antonimia(antonimia(ποιός), enikos, arseniko) **-->** [ποιός].
antonimia(antonimia(ποιοί), plithintikos, arseniko) **-->** [ποιοί].
antonimia(antonimia(ποιό), enikos, oudetero) **-->** [ποιό].
antonimia(antonimia(ποιά), plithintikos, oudetero) **-->** [ποιά].
antonimia(antonimia(ποιά), enikos, thiliko) **-->** [ποιά].
antonimia(antonimia(ποιά), plithintikos, thiliko) **-->** [ποιές].
antonimia(antonimia(πού), _, _) **-->** [πού].

epitheto(epitheto(εργαστηριακό), enikos, _) **-->** [εργαστηριακό].
epitheto(epitheto(θεωρητικό), enikos, _) **-->** [θεωρητικό].
epitheto(epitheto(εργαστηριακά), plithintikos, oudetero) **-->** [εργαστηριακά].
epitheto(epitheto(θεωρητικά), plithintikos, oudetero) **-->** [θεωρητικά].
epitheto(epitheto(εαρινό), enikos, _) **-->** [εαρινό].

ousiastiko_leksi(ousiastiko_leksi(καθηγητής), enikos, arseniko) **-->** [καθηγητής].
ousiastiko_leksi(ousiastiko_leksi(καθηγητές), plithintikos, arseniko) **-->** [καθηγητές].
ousiastiko_leksi(ousiastiko_leksi(καθηγήτρια), enikos, thiliko) **-->** [καθηγήτρια].
ousiastiko_leksi(ousiastiko_leksi(καθηγήτριες), plithintikos, thiliko) **-->** [καθηγήτριες].
ousiastiko_leksi(ousiastiko_leksi(φοιτητής), enikos, arseniko) **-->** [φοιτητής].
ousiastiko_leksi(ousiastiko_leksi(φοιτητές), plithintikos, arseniko) **-->** [φοιτητές].
ousiastiko_leksi(ousiastiko_leksi(φοιτήτρια), enikos, thiliko) **-->** [φοιτήτρια].
ousiastiko_leksi(ousiastiko_leksi(φοιτήτριες), plithintikos, thiliko) **-->** [φοιτήτριες].
ousiastiko_leksi(ousiastiko_leksi(μάθημα), enikos, oudetero) **-->** [μάθημα].
ousiastiko_leksi(ousiastiko_leksi(μαθήματα), plithintikos, oudetero) **-->** [μαθήματα].
ousiastiko_leksi(ousiastiko_leksi(τηλέφωνο), _, _) **-->** [τηλέφωνο].
ousiastiko_leksi(ousiastiko_leksi(επίθετο), _, _) **-->** [επίθετο].
ousiastiko_leksi(ousiastiko_leksi(όνομα), _, _) **-->** [όνομα].
ousiastiko_leksi(ousiastiko_leksi(τίτλο), _, _) **-->** [τίτλο].

ousiastiko_leksi(ousiastiko_leksi(mail), _, _) **-->** [mail].
ousiastiko_leksi(ousiastiko_leksi(αμ), _, _) **-->** [αμ].
ousiastiko_leksi(ousiastiko_leksi(κωδικό), _, _) **-->** [κωδικό].
ousiastiko_leksi(ousiastiko_leksi(διπλομματική), _, _) **-->** [διπλωματική].
ousiastiko_leksi(ousiastiko_leksi(βαθμό), _, _) **-->** [βαθμό].
ousiastiko_leksi(ousiastiko_leksi(εξάμηνο), _, _) **-->** [εξάμηνο].

arthro(arthro(το), enikos, oudetero) **-->** [το].
arthro(arthro(τα), plithintikos, oudetero) **-->** [τα].
arthro(arthro(ο), enikos, arseniko) **-->** [ο].
arthro(arthro(οι), plithintikos, arseniko) **-->** [οι].
arthro(arthro(η), enikos, thiliko) **-->** [η].
arthro(arthro(οι), plithintikos, thiliko) **-->** [οι].
arthro(arthro(_), _, _) **-->** [].

syndesmos(syndesmos(και)) **-->** [και].
syndesmos(syndesmos(η)) **-->** [η].

prothesi(prothesi(με)) **-->** [με].

rima(rima(διδάσκει), enikos) **-->** [διδάσκει].
rima(rima(διδάσκει), plithintikos) **-->** [διδάσκουν].
rima(rima(εκπονεί), enikos) **-->** [εκπονεί].
rima(rima(παρακολουθεί), enikos) **-->** [παρακολουθεί].
rima(rima(διδάσκουν), plithintikos) **-->** [διδάσκουν].
rima(rima(παρακολουθούν), plithintikos) **-->** [παρακολουθούν].
rima(rima(επιβλέπει), ενικός) **-->** [επιβλέπει].
rima(rima(ολοκλήρωσε), ενικός) **-->** [ολοκλήρωσε].
rima(rima(διδάσκεται), ενικός) **-->** [διδάσκεται].

am(am(642)) **-->** [642].
am(am(713)) **-->** [713].

am(am(680)) **-->** [680].
am(am(999)) **-->** [999].

onoma(onoma(βαρουξής)) **-->** [βαρουξής].
onoma(onoma(ανταλής)) **-->** [ανταλής].
onoma(onoma(αθανασιάδης)) **-->** [αθανασιάδης].
onoma(onoma(παπαδογιάννη)) **-->** [παπαδογιάννη].

mathima(mathima(προγραμματισμός)) **-->** [προγραμματισμός].
mathima(mathima(java)) **-->** [java].
mathima(mathima('δίκτυα δεδομένων')) **-->** [δίκτυα], [δεδομένων].
mathima(mathima('εισαγωγή στο marketing')) **-->** [εισαγωγή], [στο], [marketing].
mathima(mathima('έμπειρα συστήματα ')) **-->** [έμπειρα], [συστήματα].
mathima(mathima('δίκτυα υπολογιστών')) **-->** [δίκτυα], [υπολογιστών].

kod_mathima(kod_mathima(τπ4004)) **-->** [τπ4004].
kod_mathima(kod_mathima(τπ4004α)) **-->** [τπ4004α].
kod_mathima(kod_mathima(τπ4004β)) **-->** [τπ4004β].
kod_mathima(kod_mathima(τπ5007)) **-->** [τπ5007].
kod_mathima(kod_mathima(τπ5007α)) **-->** [τπ5007α].
kod_mathima(kod_mathima(τπ5007β)) **-->** [τπ5007β].
kod_mathima(kod_mathima(τπ2001)) **-->** [τπ2001].
kod_mathima(kod_mathima(τπ2001α)) **-->** [τπ2001α].
kod_mathima(kod_mathima(τπ2001β)) **-->** [τπ2001β].

grade(grade(5)) **-->** [5].
grade(grade(6)) **-->** [6].
grade(grade(7)) **-->** [7].
grade(grade(8)) **-->** [8].
grade(grade(9)) **-->** [9].
grade(grade(10)) **-->** [10].

semester(semester(0405)) **-->** [0405].

onoma(onoma(κωστής)) **-->** [κωστής].
onoma(onoma(γιώργος)) **-->** [γιώργος].
onoma(onoma(θανάσης)) **-->** [θανάσης].
onoma(onoma(δέσποινα)) **-->** [δέσποινα].

tilefono(tilefono(6937090404)) **-->** [6937090404].
tilefono(tilefono(2810319528)) **-->** [2810319528].
tilefono(tilefono(6972687798)) **-->** [6972687798].
tilefono(tilefono(9999999)) **-->** [9999999].
tilefono(tilefono(2810379748)) **-->** 2810379748].

mail(mail('kvarouxis@hotmail.com')) **-->** ['kvarouxis@hotmail.com'].
mail(mail('g_andalis@yahoo.gr')) **-->** ['g_andalis@yahoo.gr'].
mail(mail('epp680@epp.teiher.gr')) **-->** ['epp680@epp.teiher.gr'].
mail(mail('epp999@epp.teiher.gr')) **-->** ['epp999@epp.teiher.gr'].

code(code(001)) **-->** [001].
code(code(002)) **-->** [002].
code(code(003)) **-->** [003].
code(code(004)) **-->** [004].
code(code(005)) **-->** [005].
code(code(006)) **-->** [006].
code(code(007)) **-->** [007].
code(code(008)) **-->** [008].
code(code(009)) **-->** [009].
code(code(010)) **-->** [010].
code(code(011)) **-->** [011].
code(code(012)) **-->** [012].
code(code(013)) **-->** [013].

onoma(onoma(αποστολάκης)) **-->** [αποστολάκης].
onoma(onoma(σιδέρης)) **-->** [σιδέρης].
onoma(onoma(μαρακάκης)) **-->** [μαρακάκης].
onoma(onoma(μπιτσάκη)) **-->** [μπιτσάκη].
onoma(onoma(κορτσιδάκη)) **-->** [κορτσιδάκη].
onoma(onoma(μαστοράκης)) **-->** [μαστοράκης].
onoma(onoma(στρατάκης)) **-->** [στρατάκης].
onoma(onoma(ξεζωνάκης)) **-->** [ξεζωνάκης].
onoma(onoma(ασκοξυλάκης)) **-->** [ασκοξυλάκης].
onoma(onoma(βαρδιάμπασης)) **-->** [βαρδιάμπασης].
onoma(onoma(πάλλης)) **-->** [πάλλης].
onoma(onoma(αιβαλής)) **-->** [αιβαλής].
onoma(onoma(βασίλης)) **-->** [βασίλης].
onoma(onoma(σπύρος)) **-->** [σπύρος].
onoma(onoma(αργύρης)) **-->** [αργύρης].
onoma(onoma(μανόλης)) **-->** [μανόλης].
onoma(onoma(μαρίνα)) **-->** [μαρίνα].
onoma(onoma(αγνή)) **-->** [αγνή].
onoma(onoma(γιώργος)) **-->** [γιώργος].
onoma(onoma(δημήτριος)) **-->** [δημήτριος].
onoma(onoma(γιάννης)) **-->** [γιάννης].
onoma(onoma(ευάγγελος)) **-->** [ευάγγελος].
onoma(onoma(κωστής)) **-->** [κωστής].

ΠΑΡΑΡΤΗΜΑ Ε: ΛΕΞΙΚΟ ΠΕΔΙΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Ε. 1 Στιγμιότυπο Λεξικού Πεδίου Προβλήματος

Energeies_prosopon/ενέργειες_προσώπων

κλειδί	**Όνομα οντότητας**	**Ενέργεια**
1	καθηγητ	διδάσκ
2	καθηγητ	επιβλ
3	φοιτητ	παρακολουθ
4	φοιτητ	εκπον

Ε. 2 Υλοποίηση Λεξικού Πεδίου Προβλήματος σε Prolog

```
energeies_prosopon(1, καθηγητ, διδάσκ).
energeies_prosopon(2, καθηγητ, επιβλέπ).
energeies_prosopon(3, φοιτητ, παρακολουθ).
energeies_prosopon(4, φοιτητ, εκπον).
energeies_prosopon(5, φοιτητ, ολοκλήροσ).
energeies_prosopon(6, μάθημ, διδάσκε).
energeies_prosopon(7, καθηγήτ, διδάσκ).
energeies_prosopon(8, καθηγήτ, επιβλέπ).
energeies_prosopon(9, φοιτήτ, παρακολουθ).
energeies_prosopon(10, φοιτήτ, εκπον).
energeies_prosopon(11, φοιτήτ, ολοκλήροσ).
```

Σημείωση: *"φοιτητ" είναι ρίζα της λέξης "φοιτητής", "φοιτήτ" είναι ρίζα της λέξης "φοιτήτρια".*

ΠΑΡΑΡΤΗΜΑ ΣΤ: ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ-ΑΠΑΝΤΗΣΕΩΝ

Στην παρουσίαση των παρακάτω παραδειγμάτων, **Ε**, **Α** και **Στ** σημαίνουν **Ε**ρώτηση, **Α**πάντηση και **Στ**όχος αντίστοιχα.

Ε:*'Ποιό μάθημα διδάσκει ο καθηγητής με επίθετο Μαρακάκης'.*

Α:'[έμπειρα συστήματα, θεωρητικό, έμπειρα συστήματα, εργαστηριακό]'.

Στ:'kathigites(A,μαρακάκης,B,C,D,E,F),didaskoun(A,G),
(apoteleitai_apo(H,G,K);apoteleitai_apo(H,L,G)),mathimata(H,M,N,O,P),
leptomereies_mathimaton(G,Q,R)'.

Ε:*'Ποιό μάθημα διδάσκει ο καθηγητής με κωδικό 7'.*

Α:'[δίκτυα δεδομένων, εργαστηριακό]'.

Στ:'kathigites(7,A,B,C,D,E,F),didaskoun(7,G),(apoteleitai_apo(H,G,K);apotel
eitai_apo(H,L,G)),mathimata(H,M,N,O,P),leptomereies_mathimaton(G,Q,
R)'.

Ε:*'Ποιό μάθημα διδάσκει η καθηγήτρια με όνομα Αγνή'.*

Α:'[εισαγωγή στο marketing,θεωρητικό]'.

Στ:'kathigites(A,B,αγνή,C,D,E,F),didaskoun(A,G),(apoteleitai_apo(H,G,K);apo
teleitai_apo(H,L,G)),mathimata(H,M,N,O,P),leptomereies_mathimaton(G,
Q,R)'.

Ε:*'Ποιό μάθημα διδάσκει ο καθηγητής με τηλέφωνο 2810379748'.*

Α:'[έμπειρα συστήματα,θεωρητικό, έμπειρα συστήματα,εργαστηριακό]'.

Στ:'kathigites(A,B,C,D,E,2810379748,F),didaskoun(A,G),(apoteleitai_apo(H,
G,K);apoteleitai_apo(H,L,G)),mathimata(H,M,N,O,P),leptomereies_mathi
maton(G,Q,R)'.

Ε:*'Ποιά μαθήματα διδάσκει ο καθηγητής με όνομα Σπύρος'.*

Α:'[δίκτυα δεδομένων,εργαστηριακό]'.

Στ:'kathigites(A,B,σπύρος,C,D,E,F),didaskoun(A,G),(apoteleitai_apo(H,G,K);apoteleitai_apo(H,L,G)),mathimata(H,M,N,O,P),leptomereies_mathimaton(G,Q,R)'.

Ε:*'Ποιό μάθημα ολοκλήρωσε ο φοιτητής με επίθετο Βαρουξής'.*

Α:'[java,εργαστηριακό,δίκτυα υπολογιστών,θεωρητικό,δίκτυα υπολογιστών,εργαστηριακό,έμπειρα συστήματα,θεωρητικό,έμπειρα συστήματα,εργαστηριακό,προγραμματισμός,θεωρητικό,προγραμματισμός,εργαστηριακό]'.

Στ:'foitites(A,βαρουξής,B,C,D,E,F),exei_perasei(A,G,H,K),(apoteleitai_apo(L,G,M);apoteleitai_apo(L,N,G)),mathimata(L,O,P,Q,R),leptomereies_mathimaton(G,S,T)'.

Ε:*'Ποιό μάθημα ολοκλήρωσε ο φοιτητής με ΑΜ 713'.*

Α:'[δίκτυα υπολογιστών,θεωρητικό,εισαγωγή στο marketing, θεωρητικό, προγραμματισμός,εργαστηριακό]'.

Στ:'foitites(713,A,B,C,D,E,F),parakolouthoun(713,G,H),(apoteleitai_apo(K,G,L);apoteleitai_apo(K,M,G)),mathimata(K,N,O,P,Q),leptomereies_mathimaton(G,R,S)'.

Ε:*'Ποιά μαθήματα παρακολουθεί ο φοιτητής με ΑΜ 713 με τηλέφωνο* 2810319528'.

Α:'[δίκτυα υπολογιστών,θεωρητικό,εισαγωγή στο marketing,θεωρητικό,προγραμματισμός,εργαστηριακό]'.

Στ:'foitites(713,A,B,C,D,2810319528,E),parakolouthoun(713,F,G),(apoteleitai_apo(H,F,K);apoteleitai_apo(H,L,F)),mathimata(H,M,N,O,P),leptomereies_mathimaton(F,Q,R)'.

Ε:*'Ποιός καθηγητής διδάσκει το μάθημα Προγραμματισμός'.*

Α:'[ξεζωνάκης,εργαστηριακό,ξεζωνάκης,θεωρητικό,βαρδιάμπασης,εργαστηριακό]'.

Στ:'kathigites(A,B,C,D,E,F,G),didaskoun(A,H),(apoteleitai_apo(K,H,L);apotele

itai_apo(K,M,H)),mathimata(K,προγραμματισμός,N,O,P),leptomereies_mathimaton(H,Q,R)'.

Ε:*'Ποιός φοιτητής παρακολουθεί το μάθημα Java'.*

Α:'[αθανασιάδης,εργαστηριακό,αθανασιάδης,θεωρητικό,παπαδογιάννη,θεωρητικό]'.

Στ:'foitites(A,B,C,D,E,F,G),parakolouthoun(A,H,K),(apoteleitai_apo(L,H,M);apoteleitai_apo(L,N,H)),mathimata(L,java,O,P,Q),leptomereies_mathimaton(H,R,S)'.

Ε:*'Ποιά διπλωματική εκπονεί ο φοιτητής με επίθετο Βαρουξής'.*

Α:'[δημιουργία συστήματος ερωταποκρίσεων]'.

Στ:'foitites(A,βαρουξής,B,C,D,E,F),ekponei(A,G,H),ptyxiakes(G,K)'.

ΠΑΡΑΡΤΗΜΑ Ζ: ΤΜΗΜΑ ΚΩΔΙΚΑ ΣΕ JAVA.

Κλάση Interface

```
public Interface() {
        super(" EROTISIS ");
        c = getContentPane();
        c.setLayout(new BorderLayout());
        question = new JLabel("ΕΡΩΤΗΣΗ:");
        answer = new JLabel("ΑΠΑΝΤΗΣΗ");
        stoxos = new JLabel("ΕΡΩΤΗΣΗ/ΣΤΟΧΟΣ ΣΕ PROLOG");
        bar = new JMenuBar();
        setJMenuBar(bar);
        file = new JMenu("Αρχείο");
        bar.add(file);
        Exit = new JMenuItem("'Εξοδος");
        file.add(Exit);

        panel1 = new JPanel();
        panel1.setLayout(new BorderLayout());
        panel2 = new JPanel();
        panel2.setLayout(new BorderLayout());
        panel3 = new JPanel();
        panel3.setLayout(new BorderLayout());
        goal = new JTextArea(5,20);
        result = new JTextArea(5,20);
        result_stoxos = new JTextArea(5,20);
        scroll1 = new JScrollPane(goal);
        scroll2 = new JScrollPane(result);
        button = new JButton("Δημιουργία απάντησης");
        button2 = new JButton("Νέα ερώτηση σε φυσική ");
        button.addActionListener(
```

```
        new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        executeProlog(c);
    }
});

 button2.addActionListener(
        new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        goal.setText("");
        result.setText("");
        result_stoxos.setText("");
    }
});

Exit.addActionListener(
        new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});

panel1.add(question,BorderLayout.NORTH);
panel1.add(scroll1,BorderLayout.CENTER);
panel2.add(stoxos,BorderLayout.NORTH);
panel2.add(result_stoxos,BorderLayout.CENTER);
panel2.add(answer,BorderLayout.SOUTH);
panel3.add(scroll2,BorderLayout.NORTH);
panel3.add(button,BorderLayout.CENTER);
panel3.add(button2,BorderLayout.SOUTH);
c.add(panel1,BorderLayout.NORTH);
c.add(panel2,BorderLayout.CENTER);
```

```
    c.add(panel3,BorderLayout.SOUTH);

    setSize(900,600);
    show();
}
```

ΕΥΡΕΤΗΡΙΟ

Αγγλικής Γλώσσας

Ελληνικής Γλώσσας