

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΤΜΗΜΑ ΜΕΘΟΔΟΛΟΓΙΑΣ ΙΣΤΟΡΙΑΣ ΚΑΙ ΘΕΩΡΙΑΣ ΤΗΣ
ΕΠΙΣΤΗΜΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΒΑΣΙΚΗ ΚΑΙ ΕΦΑΡΜΟΣΜΕΝΗ ΓΝΩΣΤΙΚΗ ΕΠΙΣΤΗΜΗ»

Υπολογιστικό Μεταγνωστικό Σύστημα Απάντησης Ερωτήσεων

εμπνευσμένο από Μεταγνωστικές Λειτουργίες

Διδακτορική Διατριβή

Ιωσήφ Αρμάου

Αθήνα, Ιούνιος 2010

ΕΥΧΑΡΙΣΤΙΕΣ

Με το πέρας αυτής της διδακτορικής διατριβής θα ήθελα να εκφράσω τις ευχαριστίες μου σε όλους όσους συνέβαλαν στην πραγματοποίησή του. Συγκεκριμένα, θα ήθελα να ευχαριστήσω:

- τον επιβλέποντα καθηγητή μου Ιωάννη Κόντο για τις τόσο πολύτιμες γνώσεις και εμπειρίες που μου προσέφερε απλόχερα καθώς και για τον σημαντικό από τον πολύτιμο χρόνο του που διέθεσε για να με στηρίξει επιστημονικά. Επίσης για την εμπιστοσύνη που μου δείχνει, πράγμα που με τιμά.
- την καθηγήτρια Στέλλα Βοσνιάδου για την παρότρυνσή της να ξεκινήσω την ερευνητική μου προσπάθεια στο χώρο της Τεχνητής Νοημοσύνης καθώς και για τη σχετική βιβλιογραφία που με παρέπεμψε, η οποία συνέβαλε στη θεωρητική ενίσχυση τις διδακτορικής μου διατριβής.
- την καθηγήτρια Μαρία Γρηγοριάδου για τις εύστοχες υποδείξεις και εποικοδομητικές επισημάνσεις της κατά τη συγγραφή της τρέχουσας διατριβής οι οποίες συνέβαλαν αποφασιστικά στην ολοκλήρωσή της.

Επίσης θα ήθελα να ευχαριστήσω θερμά και ιδιαίτερα:

- τη φίλη και συνάδελφο Αφροδίτη Παπαϊωάννου για τις εποικοδομητικές της παρατηρήσεις και την αδιάλειπτη συμπαράστασή της.
- τη φιλόλογο και συνάδελφό μου στο σχολείο Ζωή Αλεξανδρή για την επιμέλεια του κειμένου.
- τους κοντινούς μου φίλους Κωνσταντίνο Κούτρα και Paul Aleksiewicz για την αμέριστη συμπαράστασή τους.

I.A.

Αθήνα, 2010

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1	σελ.	6
1.1 Η συμβολή	σελ.	6
1.2 Θεωρητικό πλαίσιο	σελ.	8
1.3 Δομή της διατριβής	σελ.	9
ΚΕΦΑΛΑΙΟ 2 Συστήματα Απάντησης Ερωτήσεων	σελ.	11
2.1 Γενικοί ορισμοί	σελ.	11
2.2 Συστήματα Απάντησης Ερωτήσεων	σελ.	12
2.2.1 Ορισμός – Χαρακτηριστικά – Σκοπός	σελ.	12
2.2.2 Κατασκευή συστημάτων απάντησης ερωτήσεων: Ιστορική αναδρομή	σελ.	16
ΚΕΦΑΛΑΙΟ 3 Η Μεταγνωσιακή λειτουργία στη Ψυχολογία.	σελ.	23
3.1 Ορισμός της μεταγνωσιακής λειτουργίας (metacognition)	σελ.	23
3.2 Το θεωρητικό πλαίσιο του Flavell (1978)	σελ.	26
3.3 Νεώτερες σχετικές εξελίξεις στον τομέα της Γνωσιακής Επιστήμης και Ψυχολογίας	σελ.	30
ΚΕΦΑΛΑΙΟ 4 Συστήματα Τεχνητής Νοημοσύνης εμπνευσμένα από Μεταγνωσιακές Διαδικασίες	σελ.	35
4.1 Γενικά περί ‘μεταγνωσιακών’ διαδικασιών	σελ.	35
4.2 Τεχνητή Νοημοσύνη (TN), «μετα-συλλογισμός» και ενδοσκόπηση	σελ.	35
4.3 Εφαρμογές συστημάτων QA σε φυσική γλώσσα που επιδεικνύουν «ρητή αυτοεπίγνωση»	σελ.	38
ΚΕΦΑΛΑΙΟ 5 Ιδέες και μέθοδοι των επιμέρους Επιστημονικών κλάδων στους οποίους στηρίχτηκε η έρευνα στην εφαρμογή του συστήματος « ΑΜΥΝΤΑΣ »	σελ.	43
5.1 Η αποδεικτική διαδικασία στη Γεωμετρία από τους αρχαίους Έλληνες Μαθηματικούς των Κλασικών Χρόνων	σελ.	43
5.1.1 Γενικά	σελ.	43

5.1.2	Τα γεωμετρικά σχήματα στη διαδικασία της απόδειξης	σελ.	45
5.1.3	Από το παραστατικό στο ιδεατό – σύμμετρα και ασύμμετρα μεγέθη	σελ.	46
5.1.4	Η ασυμμετρία και ο αλγόριθμος της ανθυφαίρεσης	σελ.	47
5.1.5	Το άπειρο της ανθυφαίρεσης και το κριτήριο ασυμμετρίας δύο μεγεθών – γραμμική ασυμμετρία	σελ.	48
5.1.6	Η έμμεσος αποδεικτική διαδικασία (η εις άτοπον απαγωγή)	σελ.	49
5.1.7	Η αποδεικτική διαδικασία στην πρώτη Πρόταση των Στοιχείων του Ευκλείδη	σελ.	50
5.2	Το φαινόμενο της διαδικασίας βρόχου ή ανατροφοδότησης των πρωτεϊνών p53 και mdm2 στο πεδίο της Βιοϊατρικής	σελ.	51
ΚΕΦΑΛΑΙΟ 6	Γενική περιγραφή και παραδείγματα εφαρμογής του υλοποιηθέντος συστήματος	σελ.	53
6.1	Γενική περιγραφή του υλοποιηθέντος συστήματος Απάντησης Ερωτήσεων « ΑΜΥΝΤΑΣ » – Αυτόματο Μ εταγνωστικών Υ πολογιστικών Ν Τροποποιήσιμο Α παντητικό Σ ύστημα	σελ.	53
6.2	Περιγραφή παραδειγμάτων εφαρμογής του συστήματος	σελ.	54
6.2.1	Παράδειγμα 1 ^ο : Παραγωγή απάντησης όταν αυτή είναι ρητή δήλωση σε κείμενο Γεωμετρίας	σελ.	60
6.2.2	Παράδειγμα 2 ^ο : Παραγωγή απάντησης όταν αυτή δεν δηλώνεται ρητά σε κείμενο Γεωμετρίας	σελ.	66
6.2.3	Παράδειγμα 3 ^ο : Παραγωγή απάντησης όταν αυτή δεν δηλώνεται ρητά σε Βιοϊατρικά κείμενα	σελ.	68
ΚΕΦΑΛΑΙΟ 7	Αναλυτική περιγραφή των υποσυστημάτων	σελ.	78
7.1	Υποσύστημα επεξεργασίας της ερώτησης	σελ.	78
7.2	Υποσύστημα ανάλυσης κειμένου	σελ.	82
7.3	Υποσύστημα αυτόματης εξαγωγής γνώσης	σελ.	98
7.4	Υποσύστημα απάντησης ερωτήσεων	σελ.	104
7.5	Υποσύστημα σχεδίασης σχημάτων	σελ.	129
ΚΕΦΑΛΑΙΟ 8	Συμπεράσματα	σελ.	137
ΠΑΡΑΡΤΗΜΑ		σελ.	139
1.	Πρόγραμμα Ανάλυσης Κειμένου	σελ.	139

2. Πρόγραμμα Εξαγωγής Γνώσης	σελ.	144
3. Πρόγραμμα Απάντησης Ερωτήσεων	σελ.	145
4. Πρόγραμμα της σχεδίασης του σχήματος της πρώτης πρότασης των <i>Στοιχείων</i>	σελ.	160
5. Κατάλογος Πινάκων και Σχημάτων	σελ.	164
Βιβλιογραφία	σελ.	166

ΚΕΦΑΛΑΙΟ 1

1.1 Η συμβολή

Η **συμβολή** αυτής της διατριβής αφορά στο πεδίο έρευνας για τη δημιουργία Συστημάτων Απάντησης Ερωτήσεων (ΣΑΕ) τα οποία κατά το δυνατόν διευκολύνουν περισσότερο το χρήστη. Στη διατριβή προτείνεται μια νέα αρχιτεκτονική ΣΑΕ που διαθέτει λειτουργίες εμπνευσμένες από τις «μεταγνωσιακές λειτουργίες» (metacognition), όπως ορίστηκαν για πρώτη φορά από τον Flavell το 1976, ο οποίος περιγράφει από το πεδίο της Ψυχολογίας κάτι πολύ πιο σύνθετο από αυτό που η πληροφορική μπορεί σήμερα να παρουσιάσει.

Εξ όσων γνωρίζουμε, τα μόνα Συστήματα Απάντησης Ερωτήσεων που βρίσκονται σε ανάπτυξη και επιδεικνύουν λειτουργίες που μπορεί να θεωρηθούν ότι είναι εμπνευσμένες από τις μεταγνωσιακές λειτουργίες που μελετούν οι ψυχολόγοι, οι οποίοι ισχυρίζονται ότι προσδίδουν στα συστήματά τους ‘self-awareness’ (αυτοεπίγνωση) ή και ‘metacognition’ (μεταγνωσιακή λειτουργία), είναι το σύστημα CASSIE (Shapiro et al., 2007) και το σύστημα EPILOG (Morbini & Schubert, 2009). Και τα δύο συστήματα χρησιμοποιούν για τις εξαγόμενες απαντήσεις πληροφοριών αποκλειστικά μόνον τυπικές βάσεις γνώσεων και δεν έχουν καμία εφαρμογή στο χώρο της Γεωμετρίας και της Βιοϊατρικής. Επιπλέον η εφαρμογή του συστήματός μας στους τομείς της Γεωμετρίας και της Βιοϊατρικής επιβεβαιώνει την προσαρμοστικότητά του.

Η κύρια συμβολή της διατριβής είναι η λειτουργία ενός ΣΑΕ, του οποίου η αρχιτεκτονική του στηρίζεται στη μεταγνωστική διαχείριση στρατηγικών για την αντιμετώπιση των εισαγόμενων ερωτήσεων. Επιπλέον διαθέτει ισχυρή προσαρμοστικότητα λόγω της απευθείας ανάλυσης του κειμένου χωρίς προηγούμενη μετάφραση σε τυπική γλώσσα με εφαρμογή σε πολύ διαφορετικά πεδία γνώσης, όπως της Γεωμετρίας και της Βιοϊατρικής, δεδομένου ότι απαντά ερωτήσεις με την εξαγωγή γνώσης απευθείας από κείμενα δίχως προηγούμενη μετατροπή τους σε τυπική αναπαράσταση. Ο μηχανισμός εξαγωγής γνώσης του συστήματος χρησιμοποιείται και για την αυτόματη δημιουργία «οντολογικών» γνώσεων (βασικών εννοιών που συγκροτούν το πεδίο έρευνας στο οποίο εφαρμόζεται το σύστημα). *Το σύστημα επιτελεί «μεταγνωστική» (metagnostic) απάντηση ερωτήσεων από κείμενα σε οποιονδήποτε γνωστικό τομέα.*

Συγκεκριμένα στο χώρο της Γεωμετρίας το σύστημα μπορεί να απαντήσει σε ερωτήσεις με βάση τα κείμενα των αποδείξεων θεωρημάτων, με την παροχή εξηγήσεων, οι οποίες επιδεικνύουν κάποιο είδος «αυτοεπίγνωσης» (self-awareness). Αυτό επιτυγχάνεται μέσω της αναπαράστασης και επεξεργασίας της κατάστασης του συστήματος καθώς και της αυτόματης ανάλυσης της ερώτησης και του κειμένου. Οι παραγόμενες εξηγήσεις του συστήματος προσφέρονται για να εφοδιάσουν το χρήστη με την τεκμηρίωση της απάντησης που δίνεται στις ερωτήσεις. Το σύστημα είναι μια εφαρμογή σε γλώσσα Prolog και χρησιμοποιεί μια γραμματική ερώτησης που συνδυάστηκε με μια γραμματική κειμένου. Αυτές οι δύο γραμματικές χρησιμοποιούν κανόνες, λεξικά, οντολογίες και την «ιστορία» της κατάστασης του συστήματος για να απαντήσει **αυτό** στις ερωτήσεις εισαγωγής. Να σημειώσουμε ότι λέγοντας «ιστορία» της κατάστασης του συστήματος εννοούμε το εξής: η εσωτερική καταγραφή των διαδοχικών καταστάσεων του συστήματος περιέχει πληροφορίες σχετικές με την ακολουθούμενη στρατηγική και την πρόοδο του παραγωγικού συλλογισμού, που εκτελείται. Στη συνέχεια οι ερωτήσεις αναλύονται αρχικά στα συστατικά τους χρησιμοποιώντας τη γραμματική της ερώτησης. Στο χώρο της Ευκλείδειας Γεωμετρίας, το σύστημά μας επίσης μπορεί ταυτόχρονα με τις παραγόμενες εξηγήσεις να υποστηρίζεται και από γεωμετρικά σχήματα. Για το λόγο αυτό το σύστημά μας θα μπορούσε να εφαρμοστεί και στο χώρο της εκπαίδευσης ενισχύοντας αποτελεσματικά την εκπαιδευτική διαδικασία.

Όσον αφορά στο χώρο της Βιοϊατρικής, το σύστημά μας μπορεί να απαντήσει ερωτήσεις από βιοϊατρικά κείμενα με την παροχή εξηγήσεων που επιδεικνύουν κάποιο είδος «αυτοεπίγνωσης» (self-awareness). Αυτό επιτυγχάνεται ομοιοτρόπως με την εφαρμογή του συστήματος στην Ευκλείδεια Γεωμετρία. Το σύστημά μας μπορεί να εφαρμοστεί είτε στη βιοϊατρική εκπαίδευση, με ιδιότητα εμπνεόμενη από μεταγνώση είτε στην απάντηση ερωτήσεων από βιοϊατρικά κείμενα για τη χρήση των ερευνητών καθώς επίσης για χρήση του βιοϊατρικού προσωπικού.

Η κατάσταση πρόόδου των συστημάτων απάντησης ερωτήσεων - και συγκεκριμένα των βιοϊατρικών συστημάτων απάντησης ερωτήσεων - στερούνται μέχρι στιγμής της ικανότητας παραγωγής εξηγήσεων που να επιδεικνύουν 'self-awareness', (Maybury, 2003; Rinaldi, 2008), η οποία ικανότητα χαρακτηρίζει το σύστημα που κατασκευάστηκε και θα παρουσιαστεί στη συνέχεια.

1.2 Θεωρητικό πλαίσιο

Οι Marvin Minsky και John McCarthy δημιούργησαν το θεωρητικό πλαίσιο για την ανάπτυξη δυνατοτήτων της μηχανικής «αυτό-γνώσης» (self-knowledge) και της «ενδοσκόπησης» (introspection) των υπολογιστικών συστημάτων Τεχνητής Νοημοσύνης ήδη από τα μέσα της δεκαετίας του '50. Η βασική ιδέα είναι ότι ένα υπολογιστικό σύστημα μπορεί να αποτελείται από δύο τμήματα π.χ. A και B, όπου το A παρακολουθεί την λειτουργία του τμήματος B, την καταγράφει, την αξιολογεί και την ελέγχει επεμβαίνοντας όποτε χρειάζεται. Ο Minsky (1968,1985) ισχυρίστηκε ότι μια μηχανή για να μπορεί να δίνει επαρκείς απαντήσεις σε ερωτήσεις που αφορούν τον κόσμο και τον εαυτό της, θα πρέπει να έχει ένα εκτελέσιμο μοντέλο και για τον ίδιο της τον εαυτό. Ένα Σύστημα Τεχνητής Νοημοσύνης (ΣΤΝ) κατά τον Minsky πρέπει να έχει ένα σύνολο πληροφοριών αποθηκευμένων σε υπολογιστή, που περιγράφουν ένα δυναμικό σύστημα (υπολογιστικό μοντέλο) ενός τμήματος του εξωτερικού κόσμου. Το ΣΤΝ πρέπει να μπορεί να απαντά σε ερωτήσεις για ενέργειες που συμβαίνουν στον πραγματικό κόσμο. Επιπλέον πρέπει να διαθέτει ένα σύνολο πληροφοριών του εαυτού του για να απαντά σε ερωτήσεις που να εξηγούν την συμπεριφορά του.

Ο McCarthy (1959) ισχυρίστηκε ότι ένα από τα χαρακτηριστικά ενός ΣΤΝ είναι να εκφράζει με απλό τρόπο τις αλλαγές στη συμπεριφορά του. Επίσης ο McCarthy (1995) υποστήριξε ότι ενώ υπάρχουν προγράμματα που κάνουν συλλογισμό μέσω λογικών προτάσεων, δεν υπάρχουν προγράμματα με 'επίγνωση' (conscious) των διαδικασιών τους.

Δεδομένης της ευρύτητας του θεωρητικού πλαισίου η υλοποίηση ενός συστήματος σε συγκεκριμένο χρονικό διάστημα με βάση το ανωτέρω πλαίσιο, απαιτεί την επιλογή τμήματος αυτού. Στη συγκεκριμένη περίπτωση της παρούσης διατριβής και λόγω των εφαρμογών που επιλέχθηκαν, δόθηκε έμφαση στο metareasoning περιορίζοντας τον εξωτερικό κόσμο στα εισαγόμενα κείμενα του συστήματος της εκάστοτε εφαρμογής.

Το θεωρητικό αυτό πλαίσιο εξελίχτηκε και εξειδικεύτηκε στη συνέχεια λόγω νέων τεχνολογικών αναγκών (π.χ. διαδίκτυο, ρομποτική) από νεώτερους ερευνητές που εμπνεύστηκαν κατ' αναλογία και από τις εξελίξεις της Ψυχολογίας όπως περιγράφεται στα κεφάλαια που ακολουθούν.

1.3 Δομή της διατριβής

Στο 1^ο Κεφάλαιο εκτίθεται η συμβολή της παρούσας διατριβής, η οποία συνίσταται στην δημιουργία και στην αξιολόγηση ενός μεταγνωσιακού συστήματος αυτόματης ανάλυσης κειμένων με υπολογιστή σε φυσική γλώσσα από διαφορετικά πεδία γνώσης και απάντησης ερωτήσεων από αυτά μετά από συμπερασματική διαδικασία. Επίσης εκτίθεται και το θεωρητικό πλαίσιο της διατριβής.

Στο 2^ο Κεφάλαιο παρουσιάζονται οι ορισμοί των ερευνητικών πεδίων της Γνωσιακής Επιστήμης, της Τεχνητής Νοημοσύνης και της Λογομηχανικής καθώς και η μεταξύ τους σχέση. Επίσης παρουσιάζονται τα πεδία έρευνας της Απάντησης Ερωτήσεων και της Εξαγωγής Πληροφοριών. Στο τέλος του κεφαλαίου δίνεται μια ιστορική αναδρομή κατασκευής των πρώτων συστημάτων απάντησης ερωτήσεων.

Στο 3^ο Κεφάλαιο εκτίθεται το θεωρητικό μοντέλο (model) των *‘μεταγνωσιακών’ διαδικασιών* του ανθρώπου που πρότεινε ο Flavell (Metacognition and Cognitive Monitoring, 1979) και το οποίο αποτέλεσε πηγή έμπνευσης και της δικιάς μας εργασίας. Ειδικότερα πηγή έμπνευσής μας απετέλεσε η θεωρητική προσέγγιση και η υπολογιστική εφαρμογή του υλοποιηθέντος συστήματος περί *στρατηγικών* για την παραγωγή απαντήσεων αυτών των συστημάτων. Επίσης για την περαιτέρω ανασκόπηση των νεωτέρων εξελίξεων της μελέτης του ψυχολογικού φαινομένου «metacognition», ενδεικτικά επιλέξαμε τις πρόσφατες σχετικές προσεγγίσεις που παρουσιάζονται στα άρθρα των Zion et al., Vosniadou & Kyriakopoulou, Chi Min & VanLehn.

Στο 4^ο Κεφάλαιο παρουσιάζεται η επίδραση του τομέα των *‘μεταγνωσιακών λειτουργιών’* της Ψυχολογίας και άλλων τομέων σκέψης στην Τεχνητή Νοημοσύνη με την αναφορά και τον σχολιασμό ορισμένων σχετικών άρθρων.

Στο 5^ο Κεφάλαιο εισάγονται ορισμένες βασικές έννοιες από τους επιστημονικούς τομείς της Ευκλείδειας Γεωμετρίας και της αλληλεπίδρασης πρωτεϊνών. Στις έννοιες αυτές βασιστήκαμε για την δημιουργία παραδειγμάτων που χρησιμοποιήθηκαν στην αξιολόγηση του συστήματος *‘AMYNTΑΣ’* της παρούσας διατριβής.

Στο 6^ο Κεφάλαιο δίδονται αφενός μεν μια περιγραφή της δομής και της λειτουργίας του συστήματος *‘AMYNTΑΣ’*, αφετέρου δε η αξιολόγησή του με εφαρμογή σε τρία παραδείγματα εκ των οποίων τα δύο αφορούν τον χώρο της Ευκλείδειας Γεωμετρίας και το τρίτο αφορά τον χώρο της Βιοϊατρικής και συγκεκριμένα την αλληλεπίδραση των πρωτεϊνών p53 και mdm2.

Στο 7^ο Κεφάλαιο δίνεται αναλυτική περιγραφή όλων των προγραμμάτων και των αρχείων με τα οποία υλοποιούνται τα πέντε υποσυστήματα.

Στο 8^ο Κεφάλαιο εκτίθενται Συμπεράσματα και μελλοντικοί στόχοι της παρούσας διατριβής.

Στο 9^ο Κεφάλαιο περιέχονται τα επιμέρους προγράμματα του Συστήματος ΑΜΥΝΤΑΣ.

Στο 1^ο αυτό κεφάλαιο αναφέρθηκε η συμβολή της παρούσας διατριβής μας, η οποία συνίσταται στην δημιουργία και στην αξιολόγηση ενός μεταγνωσιακού συστήματος αυτόματης ανάλυσης κειμένων με υπολογιστή σε φυσική γλώσσα από διαφορετικά πεδία γνώσης και απάντησης ερωτήσεων από αυτά έπειτα από συμπερασματική διαδικασία. Η έρευνα στηρίχτηκε σε ιδέες και μεθόδους που προέρχονται από την Τεχνητή Νοημοσύνη και των επιμέρους επιστημών εφαρμογής, όπως της Γεωμετρίας και της Βιοϊατρικής (Heath, 1956; Kontos, 1992; Kontos&Malagardi, 1999; Kontos, Malagardi & Peros, 2003). Η έμπνευση για τις μεταγνωστικές ιδιότητες προήλθε από θεωρίες της Ψυχολογίας για τις μεταγνωσιακές διαδικασίες του ανθρώπου.

Στο επόμενο κεφάλαιο παρουσιάζονται οι ορισμοί των ερευνητικών πεδίων της Γνωσιακής Επιστήμης, της Τεχνητής Νοημοσύνης και της Λογομηχανικής καθώς και η μεταξύ τους σχέση. Επίσης παρουσιάζονται τα πεδία έρευνας της Απάντησης Ερωτήσεων και της Εξαγωγής Πληροφοριών. Στο τέλος του κεφαλαίου δίνεται μια ιστορική αναδρομή κατασκευής των πρώτων συστημάτων απάντησης ερωτήσεων.

ΚΕΦΑΛΑΙΟ 2

Συστήματα Απάντησης Ερωτήσεων

(Question Answering / QA) - Επεξεργασία Ερωτήσεων και Εξαγωγής Πληροφοριών (Information Extraction / I.E.)

2.1 Γενικοί ορισμοί

Σύμφωνα με τον Κόντο (2004) : «η Επεξεργασία Ερωτήσεων είναι ο κλάδος της Λογομηχανικής που έχει στόχο τη δημιουργία υπολογιστικών συστημάτων, που αναλύουν ερωτήσεις γραμμένες σε φυσική γλώσσα και παράγουν τις απαραίτητες πληροφορίες για την απάντησή τους» (Βοσνιάδου, Κόντος κ.α., 2004, σ.93).

Ο Κόντος (1996) προτείνει τον όρο Λογομηχανική αντί του γνωστού από την διεθνή βιβλιογραφία Natural Language Processing (NLP), δηλαδή Επεξεργασία Φυσικής Γλώσσας. Το πεδίο μελέτης της Λογομηχανικής αφορά την τεχνολογία ανάπτυξης συστημάτων προγραμματισμού H/Y για την επεξεργασία γλωσσών, οι οποίες ομοιάζουν, ως ένα βαθμό, με την ανθρώπινη γλώσσα. Τα συστήματα αυτά μπορούν να εφαρμοστούν είτε στον ευκολότερο προγραμματισμό των H/Y, είτε στη μετάφραση από τη μία φυσική γλώσσα σε μια άλλη, είτε στην επεξεργασία πληροφοριών και γνώσεων διατυπωμένων σε φυσική γλώσσα. Άλλοι βασικοί κλάδοι της Λογομηχανικής είναι η Επεξεργασία Αφηγηματικών Κειμένων, η Επεξεργασία Γνωσιακών Κειμένων και η Μηχανική Μετάφραση (Κόντος, 1996, σ.67).

Η Επεξεργασία Φυσικής Γλώσσας με υπολογιστή ή Λογομηχανική αποτελεί κεντρικό θέμα της Τεχνητής Νοημοσύνης (T.N), η οποία στηρίζεται σε μεγάλο βαθμό στην επεξεργασία της γνώσης, η δε ανθρώπινη γνώση είναι εκφρασμένη κυρίως σε φυσική γλώσσα (Κόντος, 1996, σ.21). Οι ερευνητές της Τεχνητής Νοημοσύνης προσπαθούν να επιτύχουν με τον προγραμματισμό των υπολογιστών την αυτόματη εκτέλεση λειτουργιών, όπως αυτόν της άντλησης και ανάλυσης πληροφοριών από κείμενα καθώς επίσης και της απάντησης σε ερωτήσεις διατυπωμένες σε φυσική γλώσσα (Κόντος, 1996, σ.4).

Ονομάζεται δε κατά τον Κόντο (2004) Τεχνητή Νοημοσύνη ή Artificial Intelligence και για συντομία T.N. ή A.I., η επιστήμη και η τεχνολογία που επιδιώκει την ανάπτυξη υπολογιστικών συστημάτων που εκτελούν εργασίες επεξεργασίας πληροφοριών υψηλής πολυπλοκότητας (Βοσνιάδου, Κόντος κ.α., 2004, σ.38).

Ο επιστημονικός χώρος της Τεχνητής Νοημοσύνης, ως κλάδος της Πληροφορικής δημιουργήθηκε το 1956 (Gardner, 1987, σ.138) και προέκυψε από τις προσπάθειες να

κατασκευαστούν υπολογιστικά συστήματα, τα οποία σχεδιάζονται με έμπνευση από τον τρόπο σκέψης του ανθρώπου. Υποστηρίζει ο Gardner (1987) ότι η Τεχνητή Νοημοσύνη είναι η επιστήμη που χτίζεται γύρω από την προσομοίωση με υπολογιστές και η οποία εξετάζεται από πολλούς ως κεντρικός επιστημονικός κλάδος της Γνωσιακής Επιστήμης (Gardner, 1987, σ.40).

Η Γνωσιακή Επιστήμη κατά την Βοσνιάδου (2004) : *«ξεκίνησε πριν από περίπου 30 χρόνια σαν μια συναρπαστική ιδέα για το πώς θα μπορούσαμε να μελετήσουμε επιστημονικά τα νοητικά φαινόμενα»* (Βοσνιάδου κ.α., 2004, σ.19), με κύριο χαρακτηριστικό της τη διεπιστημονικότητα, η οποία στα πρώτα στάδιά της *«χαρακτηριζόταν κυρίως από τη συνεργασία ανάμεσα στην πληροφορική και τη γνωστική ψυχολογία»* (Βοσνιάδου ό.π. σ.36). Επίσης, κατά τον Edelman (1996), η Γνωσιακή Επιστήμη *«είναι μια διεπιστημονική προσπάθεια που αφορά την ψυχολογία, την πληροφορική, την τεχνητή νοημοσύνη, τη νευροβιολογία, τη γλωσσολογία και τη φιλοσοφία»* (Edelman, 1996, σελ.44).

Η Demner-Fushman (2006), ισχυρίζεται ότι γνώσεις από την Ψυχολογία, την Γλωσσολογία, την Εκπαίδευση και την επιστήμη της Πληροφορικής, αξιοποιούνται για το πεδίο έρευνας της Απάντησης Ερωτήσεων. Ως συνέπεια υποστηρίζει ότι οι μελέτες της Τ.Ν. για την Επεξεργασία της Φυσικής Γλώσσας και το σκέλος του συστήματος Απάντησης Ερωτήσεων που αφορά την ανάκτηση των πληροφοριών θα ωφεληθεί από γνώσεις που προέρχονται από τους παραπάνω επιστημονικούς κλάδους.

2.2 Συστήματα Απάντησης Ερωτήσεων

2.2.1 Ορισμός – Χαρακτηριστικά - Σκοπός

Συστήματα Απάντησης Ερωτήσεων ή Question Answering και για συντομία ΣΑΕ ή QA ονομάζονται τα υπολογιστικά συστήματα, τα οποία επεξεργάζονται την ανθρώπινη γλώσσα και έχουν τη δυνατότητα να αναλύουν ερωτήσεις που είναι γραμμένες σε φυσική γλώσσα και στη συνέχεια να παράγουν τις απαραίτητες πληροφορίες, από βάσεις δεδομένων, κειμένων ή εικόνων, για να απαντήσουν κατά το πλείστον στη φυσική γλώσσα. Τα συστήματα αυτά μπορούν να απαντούν σε ερωτήσεις αναζητώντας την απάντηση σε περισσότερες από μία πηγές δεδομένων ή γνώσης και επιπλέον να παρέχουν μια φιλική διεπαφή στους χρήστες, οι οποίοι δεν είναι έμπειροι με τους υπολογιστές. Ο κύριος στόχος αυτών των συστημάτων είναι να παραχθεί μια σύντομη απάντηση σε μια ερώτηση και όχι ένας κατάλογος από

έγγραφα σχετικά με την ερώτηση. Επειδή στο διαδίκτυο αυτό γίνεται όλο και πιο δύσκολο μέσω των κλασικών μηχανών αναζήτησης, η χρήση της τεχνολογία των συστημάτων QA τείνει να γίνει όλο και πιο επιτακτική. (Kontos, J., and Malagardi, I., 2006).

Σκέλος του Συστήματος Απάντησης Ερωτήσεων (ΣΑΕ), που αφορά στην ανάκτηση πληροφοριών από κείμενα τα οποία εκφράζουν γεγονότα (facts) σε φυσική γλώσσα καθώς και την ανάλυση των κειμένων αυτών, αποτελεί το πεδίο έρευνας της Εξαγωγής Πληροφοριών ή Information Extraction, για συντομία I.E.

Κατά τον Fei – Yu Xu (2008), ο στόχος της I.E. είναι να εντοπίζει και να κατασκευάζει δομημένες πληροφορίες από μη δομημένα κείμενα γραμμένα σε φυσική γλώσσα και να αναζητά τις σχέσεις μεταξύ τους. Σύμφωνα με τον Tang (2007), η I.E. προσδιορίζει και εξάγει ένα υποσύνολο μιας ακολουθίας περιστάσεων, το οποίο αναπαριστά την πληροφορία που μας ενδιαφέρει. Αποτελεί δε σημαντική δραστηριότητα με πολλές πρακτικές εφαρμογές που έχουν σχέση με κείμενα ή το διαδίκτυο όπως π.χ. στην Απάντηση Ερωτήσεων ή στην εύρεση πρωτεϊνών που αναφέρονται σε άρθρα βιοϊατρικών περιοδικών (Tang, 2007).

Ο Fei – Yu Xu (ό.π.), υποστηρίζει ότι η πραγματοποίηση της πλήρους ανάλυσης του κειμένου με ηλεκτρονικό υπολογιστή αργεί όσο αυτή καθοδηγείται μόνο από την γλωσσική ανάλυση. Η I.E. σε αντίθεση ενδιαφέρεται μόνον για την ερμηνεία τμημάτων κειμένου και των δομών του που σχετίζονται με εφαρμογές. Επιπλέον θεωρεί ότι στον σταθερά αυξανόμενο ψηφιοποιημένο κόσμο εκείνοι που θα κερδίσουν στην παγκόσμια κοινωνία των πληροφοριών θα είναι τα άτομα ή οι οργανισμοί οι οποίοι θα μπορούν με τον καλύτερο τρόπο να εκμεταλλεύονται πιο γρήγορα, να κατανοούν και να έχουν ορθή πρόσβαση στις ψηφιακές πληροφορίες για την επεξεργασία των αποφάσεών τους από ό,τι οι ανταγωνιστές τους (Fei – Yu Xu, 2008, σελ.1-2). Αναφέρει επίσης ότι η ιδέα να εξάγονται δομημένες πληροφορίες από κείμενα φυσικής γλώσσας εντοπίζεται για πρώτη φορά το 1987 σε μια εφαρμογή ενός συστήματος διαχείρισης ιατρικών κειμένων από την Naomi Sager (Sager et al., 1987). Όμως η IE ως ένας αναγνωρισμένος ερευνητικός τομέας καθιερώθηκε αρκετά χρόνια αργότερα από τα συνέδρια MUC (Message Understanding Conferences) (Grishman & Sundhim, 1996), και το ACE (Automatic Extraction program) (Doddington et al., 2004) (Fei – Yu Xu, 2008, σελ.12).

Στα MUC υποβάλλονταν τα ερευνητικά αποτελέσματα των συμμετεχόντων στο διαγωνισμό των συνεδρίων και αξιολογούνταν σε σύγκριση με τα αποτελέσματα από

ανθρώπους. Όπως αναφέρει ο Fei – Yu Xu (ό.π.) το πρώτο στη σειρά MUC-1 έγινε το 1987 με θέμα τα μηνύματα επιχειρήσεων του Ναυτικού, ενώ το έκτο MUC-6 έγινε το 1995 με θέμα τα άρθρα των ειδήσεων που αφορούσαν μετακινήσεις διοικητικών στελεχών. Με το τελευταίο MUC-7 που έγινε το 1998 με θέμα τις αναφορές για τις εκτοξεύσεις των δορυφόρων (Satellite launch reports) καθορίστηκαν και οι στόχοι έρευνας της I.E. ως εξής:

1. Αναγνώριση επιλεγμένης οντότητας, που στοχεύει στα ονόματα των οντοτήτων (επιχειρήσεων, προσώπων), ονόματα τοποθεσιών, χρονικών εκφράσεων και κάποιων μορφών αριθμητικών εκφράσεων.
2. Συγγώνευση μορφών, που προσδιορίζουν διαφορετικά μέρη του κειμένου, τα οποία παρέχουν πληροφορίες για την ίδια οντότητα ή το ίδιο συμβάν.
3. Εξαγωγή φόρμας στοιχείων με την οποία συμπληρώνονται οι κενές θέσεις σε μικρότερες φόρμες συγκεκριμένων κατηγοριών οντοτήτων με τα χαρακτηριστικά τους.
4. Φόρμα σχέσης κατά την οποία συμπληρώνονται δύο κενές θέσεις της, που προσδιορίζουν με δείκτες τα στοιχεία που συνδέονται με μια σχέση και κάθε ένα από αυτά ανήκει σε μια από τις μικρότερες φόρμες της προηγούμενης εργασίας (εξαγωγή φόρμας στοιχείων).
5. Φόρμα σεναρίου κατά την οποία συμπληρώνεται μια δομή φόρμας με πληροφορίες που περιλαμβάνουν διάφορες σχέσεις ή ενδιαφέροντα περιστατικά, όπως ο προσδιορισμός των συνεργατών (Fei – Yu Xu, 2008, σελ.14-15).

Η I.E. διασπάστηκε σε διάφορους τομείς έρευνας όπως η αναγνώριση επιλεγμένης οντότητας (named entity), η συμπλήρωση πίνακα στοιχείων (template) και η εξαγωγή φόρμας σεναρίου.

Κατόπιν αναπτύχθηκε ένα νέο πρόγραμμα το ACE που στοχεύει στην ανάπτυξη τεχνολογιών για τμηματική ανάλυση κειμένων, συμπεριλαμβανομένης της ανίχνευσης και ταξινόμησης των βασικών οντοτήτων, των σχέσεων και των περιστατικών που περιγράφονται (Appelt, 1999) (Fei – Yu Xu, 2008, σελ.16).

Ο Fei – Yu Xu (ό.π.), αναφέρει ότι η I.E. είναι μια εφαρμογή των τεχνολογιών της Επεξεργασίας της Φυσικής Γλώσσας (Natural Language Processing ή NLP). Υπάρχουν εργαλεία της NLP που χρησιμοποιούνται για τον προσδιορισμό γλωσσικών δομών που μπορεί να είναι από λέξεις ή σημεία στίξης έως και φράσεις.

Η πολυπλοκότητα του έργου της I.E. απαιτεί την εις βάθος γλωσσική ανάλυση. Στην περίπτωση εξαγωγής μιας επιλεγμένης οντότητας, διαδικασίες όπως η «tokenization»

(χωρισμός σε λέξεις, σημεία στίξεις και named entities), η μορφολογική ανάλυση, η επισημείωση και η αναγνώριση φράσεων είναι αρκετές για να δώσουν τις δομές που χρειάζεται η οντότητα. Η απαιτούμενη οργάνωση εξαγωγής σχέσεων και συμβάντων είναι εκείνη για την οποία ένα σύστημα NLP είναι ικανό να παρέχει πληροφορίες για τις εξαρτήσεις μεταξύ των γλωσσικών τμημάτων. Τα συστήματα I.E. είναι εξειδικευμένα σε γνωστικές περιοχές και λειτουργούν διαφορετικά σε κάθε περιοχή. Τα συστήματα NLP που είναι σχεδιασμένα για να δίνουν βαθιές δομές καλούνται *deep* NLP συστήματα (Uszkoreit, 2002). Παρόλο που αυτά τα *deep* NLP συστήματα έχουν σημειώσει πρόοδο, στην περίπτωση που αντιμετωπίζεται μεγάλος όγκος κειμένων προτιμούνται τα επιφανειακά (shallow) συστήματα NLP διότι είναι ταχύτερα και πιο αποτελεσματικά.

Ο σκεπτικισμός ως προς την χρήση των *deep* NLP συστημάτων προκύπτει από την πολυπλοκότητα των προτάσεων και την δυσκολία να αντιμετωπίσουν την πολυσημία. Η πρόσφατη ανάπτυξη των συστημάτων αυτών αυξάνει την ανάγκη για περισσότερη ακρίβεια στην I.E. όσον αφορά την εξαγωγή των σχέσεων και των συμβάντων, κάτι που απαιτεί μια βαθύτερη ανάλυση των κειμένων της φυσική γλώσσας. Οι περισσότερες από τις σύνθετες προσεγγίσεις των δύο συστημάτων NLP κάνουν πρώτα την ρηχή ανάλυση σε «chunks» (ένα ή περισσότερα tokens) και μετά κάνουν τη βαθιά ανάλυση του κάθε «chunk» χωριστά (Fei – Yu Xu, 2008, σελ.20-21).

Στο χώρο της I.E. από τους Kontos και Malagardi (1999), παρουσιάστηκε ένα σύστημα που χρησιμοποιεί την εξαγωγή πληροφοριών από κείμενα. Το σύστημα έχει ως απώτερο στόχο την δημιουργία ευπροσάρμοστων εργαλείων εξαγωγής πληροφοριών ικανών να δέχονται ερωτήσεις σε φυσική γλώσσα και να παράγουν απαντήσεις που να περιέχουν πληροφορίες που εξήχθησαν είτε από τα κείμενα είτε μέσω εφαρμογής παραγωγικού συμπερασμού.

Το σύστημα αποτελείται από δύο υποσυστήματα. Το ένα επιτυγχάνει την εξαγωγή της γνώσης από μεμονωμένες προτάσεις σύμφωνα με την καθιερωμένη εξαγωγή πληροφοριών των Cowie και Lehnert (1996), και Grishman (1996). Το άλλο χρησιμοποιεί μια διαδικασία συλλογισμού που συνδυάζει τη γνώση η οποία έχει εξαχθεί από το προηγούμενο υποσύστημα για την απάντηση ερωτήσεων δίχως τη χρήση πίνακα αναπαράστασης (template) (Kontos, J., and Malagardi, I., 2006).

Τον όρο «μηχανισμός απάντησης ερωτήσεων» χρησιμοποιεί ο Simmons (1965), για να συμπεριλάβει τους γλωσσικούς επεξεργαστές που ασχολούνται με δηλώσεις ή και ερωτήσεις σε φυσική αγγλική γλώσσα. Αναφέρει δε ότι η έρευνα για συστήματα

απάντησης ερωτήσεων φυσικής γλώσσας χρονολογείται από το 1959. Ως πρόδρομους αυτών των συστημάτων θεωρεί τα προγράμματα ‘The Conversation Machine’ (Green, Berkeley & Gotlieb, 1959) και ‘The Oracle’ (Phillips, 1960) που πραγματοποιήθηκε στα πλαίσια της εκπόνησης της διατριβής του για το Master υπό την επίβλεψη του John McCarthy εμπνευστή του όρου «*artificial intelligence*» κατά τον Gardner (ό.π. σελ.139). Συγκεκριμένα, το πρώτο πρόγραμμα επιτρέπει σε έναν υπολογιστή να κάνει μια φαινομενικά ευφυή συνομιλία για τον καιρό, ενώ το δεύτερο απαντά σε ερωτήσεις απλών αγγλικών προτάσεων (Simmons, 1965).

2.2.2 Κατασκευή συστημάτων απάντησης ερωτήσεων: Ιστορική αναδρομή

Τα συστήματα που κατασκευάστηκαν πρώτα, περιγράφονται από τον Simmons (1965) κατά ομάδες σύμφωνα με κάποιο κοινό χαρακτηριστικό τους γνώρισμα. Στην ομάδα συστημάτων, των οποίων η πηγή των πληροφοριών είναι δομημένη, αναφέρει το σύστημα ‘SAD SAM’ (Appraiser and Diagrammer and Semantic Analyzing Machine) (Lindsay, 1963) καθώς επίσης και το ‘Baseball’ (Green, Wolf, Chomsky & Laughery, 1961).

Το πρώτο σύστημα ‘SAD SAM’ διαιρείται σε δύο τμήματα: το τμήμα της ανάλυσης και το τμήμα του χειρισμού των οντοτήτων και των σχέσεων μεταξύ τους. Είναι σχεδιασμένο να δέχεται απλές προτάσεις στην αγγλική γλώσσα, το λεξιλόγιο των οποίων έχει περιορισμένο αριθμό λέξεων με μία απλή γραμματική. Οι προτάσεις αφορούν οικογενειακές σχέσεις και το πρόγραμμα αναπαριστά σε βάση δεδομένων το οικογενειακό δένδρο (Simmons, ό.π.).

Όπως αναφέρουν οι Kontos & Malagardi (2006), το δεύτερο σύστημα ‘Baseball’ ήταν ένα από τα πρώτα βήματα υλοποίησης συστήματος QA που αναπτύχθηκαν κατά την δεκαετία 1960-1970 και εφαρμόστηκε στο εργαστήριο Lincoln (Lincoln Laboratory). Ήταν το πρώτο QA σύστημα που αναφέρθηκε στη βιβλιογραφία σύμφωνα με τις αναφορές που υπάρχουν στο πρώτο βιβλίο με συλλογές άρθρων Τεχνητής Νοημοσύνης (Feigenbaum και Feldman, 1963, σελ.207). Οι απαντήσεις και οι ερωτήσεις που δίνονταν στο σύστημα ήταν στα αγγλικά και αφορούσαν παιχνίδια ομάδων baseball. Το σύστημα μετασχημάτιζε τις προτάσεις σε μορφή που επέτρεπε την αναζήτηση των απαντήσεων σε βάση δεδομένων. Τα στοιχεία και το λεξικό ήταν δομές δεδομένων, ενώ οι ερωτήσεις γίνονταν με μία μόνο πρόταση (Kontos, J., and Malagardi, I., 2006).

Άλλες ομάδες συστημάτων που περιγράφει ο Simmons (1965), είναι εκείνες που οι βάσεις δεδομένων των συστημάτων τους αποτελούνται από γραφικά καθώς και εκείνες που τα συστήματά τους βασίζονται σε κείμενα δηλαδή προσπαθούν να βρουν απαντήσεις από αγγλικά κείμενα (Simmons, 1965, σελ.58-62).

Στο τέλος της ίδιας δεκαετίας αναπτύχθηκε το σύστημα Απάντησης Ερωτήσεων DELFI (Kontos and Papakonstantinou, 1970; Κόντος, 2004), που είχε ως βάση την αναπαράσταση ερωτήσεων με τη μέθοδο της διαδικαστικής σημασιολογίας (procedural semantics) σύμφωνα με την οποία μια πρόταση μεταφράζεται σε ένα συνδυασμό διαδικασιών. Το σύστημα απαντούσε σε ερωτήσεις φυσικής γλώσσας για ένα σύνολο αντικειμένων, τα οποία συνδέονται μεταξύ τους με σχέσεις χώρου και έχουν ιδιότητες που αφορούν στο σχήμα τους. Οι ερωτήσεις μεταφράζονταν αυτόματα σε διαδικασίες ανάκτησης, οι οποίες είχαν διαδικαστικά συστατικά γενικού σκοπού με τη δυνατότητα να ανακτούν πληροφορίες από βάση δεδομένων με ιδιότητες των αντικειμένων και των χωρικών τους σχέσεων (Kontos, J., and Malagardi, I., 2006; Βοσνιάδου, Κόντος κ.α., 2004, σελ.93-98).

Ακολούθησε και δεύτερη έκδοση του DELFI, το DELFI II (Kontos και Kossidas, 1971), με αυξημένες δυνατότητες. Με τη νέα έκδοση αναπτύχθηκαν δύο νέα είδη εφαρμογών, επιπλέον της εφαρμογής με βάση εικόνων που έγινε με την πρώτη έκδοση. Οι δύο νέες εφαρμογές αφορούσαν τον χειρισμό βάσης δεδομένων ενός οργανισμού και την απάντηση ερωτήσεων έπειτα από συμπερασμό (deductive question answering). Η λειτουργία του συστήματος DELFI II είναι αυτή ενός έμπειρου συστήματος και οι κανόνες συμπερασμού του αποτελούν αυτό που σήμερα ονομάζεται «βάση γνώσης» (Knowledge base) του έμπειρου συστήματος (Βοσνιάδου, Κόντος κ.α., 2004, σελ.98-99).

Το σύστημα DELFI όπως αναφέρει ο Κόντος, (1996) χαρακτηρίζεται από την ύπαρξη μιας Ενδιάμεσης Τυπικής Γλώσσας και αποτελείται από τρία τμήματα. Στο πρώτο τμήμα του συστήματος γίνεται η λεκτική ανάλυση των ερωτήσεων και η ανάκτηση των λεκτικών χαρακτηριστικών. Στο δεύτερο τμήμα του παράγεται αυτόματη μετάφραση των ερωτήσεων σε προγράμματα γραμμένα στην Ενδιάμεση διαδικαστική Τυπική Γλώσσα του συστήματος. Στο τρίτο τμήμα του εκτελούνται τα προγράμματα του δεύτερου τμήματος, τα οποία είτε ενημερώνουν τη βάση δεδομένων (Database) του συστήματος, είτε αντλούν στοιχεία για την παραγωγή απάντησης στην ερώτηση που υποβλήθηκε (Κόντος, 1996, σελ.76).

Στη συνέχεια ακολούθησε και άλλη ανάπτυξη εφαρμογής αυτού του τύπου, με το σύστημα SHRDLU, το όνομα του οποίου προήλθε από τα διαδοχικά πλήκτρα της δεύτερης σειράς του πληκτρολογίου της εποχής κατά την οποία αναπτύχθηκε το σύστημα. Το σύστημα SHRDLU δημοσιεύτηκε από τον T. Winograd (1972), και ήταν μέρος της διατριβής του στο MIT με εντυπωσιακά για την εποχή του αποτελέσματα (Morhini, 2006, σελ.35).

Κατά τους Kontos & Malagardi (2006), η γλώσσα 4ης γενιάς SQL (Standard Query Language) ως Ενδιάμεση Τυπική Γλώσσα των συστημάτων QA, έδωσε τη δυνατότητα αφενός μεν της επεξεργασίας των ερωτήσεων σε φυσική γλώσσα, αφετέρου δε της απάντησής των, κάνοντας χρήση οποιουδήποτε εμπορικού συστήματος βάσης δεδομένων.

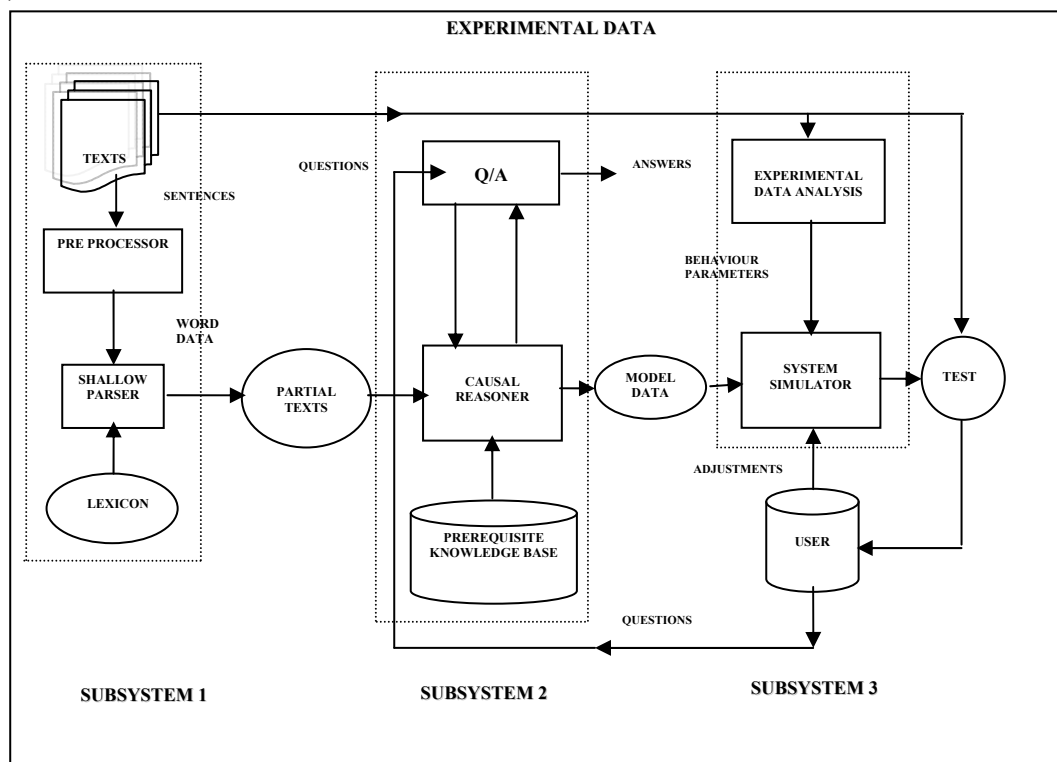
Στη συνέχεια δημιουργήθηκαν συστήματα QA τα οποία μπορούν να απαντούν σε απλές ερωτήσεις ‘factoid’ (είναι ερωτήσεις που αφορούν γεγονότα ή δηλώσεις) και αναζητούν μια πληροφορία που περιλαμβάνεται σε ένα γεγονός (fact) που έχει δηλωθεί. Επιπλέον αναφέρει, στο πλαίσιο της κινητικότητας στο χώρο των συστημάτων QA σε κείμενα, τις πρόσφατες δημοσιεύσεις των : Diekema (2003), Doan-Nguyen και Kosseim (2004), Harabagiu *et al* (2003), Kosseim *et al* (2003), Nyberg *et al* (2002), Plamondon και Kosseim (2002), Ramakrishnan (2004), Roussinof και Robles-Flores (2004) καθώς επίσης και του Waldinger *et al* (2003), τις μελλοντικές κατευθύνσεις, μερικές από τις οποίες προτείνονται από τον Maybury (2003), καθώς επίσης και το διεθνή διαγωνισμό μεταξύ συστημάτων QA που οργανώθηκε από το NIST (National Institute of Standards and Technology), (Voorhees, 2001) (Kontos, J., and Malagardi, I., 2006).

Ομοίως τα συστήματα QA, ARISTA (Kontos, 1992), και AROMA (Kontos et al., 2003) τα οποία εξάγουν πληροφορίες από επιστημονικά και τεχνικά κείμενα, που μπορούν να χρησιμοποιηθούν για την απάντηση σύνθετων ερωτήσεων της συμπεριφοράς αιτιακών μοντέλων με χρήση κατάλληλων γλωσσικών και παραγωγικών μηχανισμών. Το πρώτο σύστημα QA ARISTA (Automatic Representation Independent Syllogistic Text Analysis) (Κόντος, 1996, σελ. 162) απαντά σε ερωτήσεις για την ανάκτηση γνώσης από κείμενα φυσικής γλώσσας. Το σύστημα βασίστηκε στην αναπαράσταση μιας ανεξάρτητης πρωτοποριακής μεθόδου που καλείται επίσης ARISTA. Με τη μέθοδο αυτή η επεξεργασία γνώσεων γίνεται από κείμενο χωρίς προηγούμενη μετάφραση του κειμένου σε τυπική γλώσσα (Κόντος, 1996). Το πλεονέκτημα αυτής της μεθόδου είναι ότι τα κείμενα δεν

μεταφράζονται σε κάποιο αναπαραστασιακό φορμαλισμό. Με αυτόν τον τρόπο αποφεύγεται η μετάφραση κάθε νέας προαπαιτούμενης γλωσσικής γνώσης των κειμένων, η οποία προκύπτει λόγω των εξελίξεων του κλάδου της επιστήμης τον οποίον αφορούν τα κείμενα, απαραίτητα για την απάντηση των ερωτήσεων.

Η μέθοδος ARISTA αναζητά τις κατάλληλες αιτιακές προτάσεις από το κείμενο και τις συνδέει, με τη βοήθεια λειτουργιών του συστήματος, σε διαδοχικές αιτιακές συνδέσεις με σκοπό την ανακάλυψη μιας αιτιακής διαδοχής. Με τον τρόπο αυτό επιτυγχάνεται η εξαγωγή αιτιακής γνώσης για την απάντηση της ερώτησης, κάνοντας χρήση ενός παραγωγικού συλλογισμού. Ταυτόχρονα είναι μια εναλλακτική μέθοδος στην παραδοσιακή μέθοδο της μετάφρασης των κειμένων σε μια τυπική αναπαράσταση πριν τη χρήση του περιεχομένου τους για παραγωγική απάντηση ερώτησης από τα κείμενα. Για τους λόγους αυτούς το σύστημα QA ARISTA έχει τη δυνατότητα απάντησης ερωτήσεων απευθείας από κείμενα χρησιμοποιώντας συμπερασμό (Κόντος, 1996, σελ.162) και μπορεί να παράγει απαντήσεις που δεν ήταν ρητά δηλωμένες στα κείμενα εισαγωγής του συστήματος (Kontos, J., and Malagardi, I., 2006).

Το δεύτερο σύστημα QA AROMA (ARista Oriented Model Adaptation) απαντά σε ερωτήσεις της μορφής: «What if» (Kontos et al., 2002) (Kontos, J., and Malagardi, I., 2006).



The AROMA system architecture (Kontos *et al*, 2003)

Στο σύστημα αυτό η διαδικασία ανακάλυψης της γνώσης στηρίζεται στην αναζήτηση αιτιακών διαδοχικών σχέσεων. Αυτές στη συνέχεια αναζητούν τις προτάσεις που περιέχουν τις αναγκαίες για την επίτευξη αυτών των σχέσεων, εκφράσεις σε φυσική γλώσσα. Στην αναζήτηση των κατάλληλων προτάσεων για την επίτευξη της διαδοχικότητας, χρησιμοποιήθηκε ο αλγόριθμος που περιγράφεται από τους Kontos and Malagardi (2001), με σκοπό να επιταχυνθεί η διαδικασία απόκτησης γνώσης. Η γενική αρχιτεκτονική του συστήματος AROMA δείχνεται στο σχήμα και αποτελείται από τρία υποσυστήματα :

- A το υποσύστημα Εξαγωγής Γνώσης,
- B το υποσύστημα Αιτιακού Συλλογισμού και
- Γ το υποσύστημα Προσομοίωσης.

Αναλυτικότερα το υποσύστημα Εξαγωγής Γνώσης ενοποιεί την επιμέρους εξαγόμενη αιτιακή γνώση από διάφορα κείμενα. Η εξαγόμενη γνώση εκφράζεται σε φυσική γλώσσα με χρήση αιτιακών ρημάτων όπως «ρυθμίζω», «αυξάνω», «εμποδίζω». Αυτά τα ρήματα συνήθως λαμβάνουν ορίσματα οντοτήτων (δηλ. ονόματα οντοτήτων ή ονόματα διαδικασιών) που απαντώνται στα κείμενα των εφαρμογών. Έτσι οι αιτιακές σχέσεις εκφράζουν πλέον σχέσεις μεταξύ οντοτήτων, μεταξύ διαδικασιών και μεταξύ ζευγών διαδικασίας και οντότητας.

Τα κείμενα εισαγωγής στο σύστημα υποβάλλονται πρώτα από το υποσύστημα σε μια προ-επεξεργασία με τη βοήθεια της διαδικασίας «tokenization» κατά την οποία γίνεται κατάτμηση σε μονάδες κειμένου, όπου η κάθε μονάδα αποτελούμενη από ένα σύνολο χαρακτήρων χωρίζεται από το υπόλοιπο κείμενο με κενές θέσεις στην αρχή και το τέλος της. Με την προ-επεξεργασία γίνεται αυτόματη μετατροπή της κάθε πρότασης σε μια μορφή που να παρουσιάζει τις λέξεις, σαν στοιχεία με αριθμητικές πληροφορίες ως προς τον προσδιορισμό της πρότασης που περιέχει τη λέξη και τη θέση της στη συγκεκριμένη πρόταση. Αυτή η μετατροπή δεν έχει να κάνει με τη λογική αναπαράσταση του περιεχομένου της κάθε πρότασης, απλά είναι μια επισημείωση που αφορά την θέση της κάθε λέξης μέσα στο κείμενο. Οι προτάσεις τώρα με αυτή τη μορφή, αναλύονται και παράγονται επιμέρους κείμενα με αιτιακή γνώση.

Στη συνέχεια το υποσύστημα Αιτιακού Συλλογισμού, η λειτουργία του οποίου βασίζεται στη μέθοδο ARISTA, δέχεται ως είσοδο το αποτέλεσμα που παράγει το υποσύστημα Εξαγωγής Γνώσης και συνδυάζει την αιτιακή γνώση που είναι σε μορφή φυσικής γλώσσας για να επιτύχει μέσω παραγωγικής διαδικασίας να δώσει τις

απαντήσεις και τα δεδομένα που δεν αναφέρονται ρητά στο κείμενο εισαγωγής. Τα τμήματα των προτάσεων που περιέχουν αιτιακή γνώση αναλύονται και έτσι αναγνωρίζονται τα ζεύγη διαδικασίας και οντότητας. Οι ερωτήσεις των χρηστών υποβάλλονται σε επεξεργασία από το πρόγραμμα επεξεργασίας ερωτήσεων, η οποία εξάγει τους στόχους των προγραμμάτων δημιουργίας της απάντησης. Όλα αυτά γίνονται με τη διαδοχική σύνδεση αιτιακών σχέσεων κάνοντας χρήση της προ-απαιτούμενης γνώσης της οντολογίας στην υποστήριξη της διαδικασίας του συλλογισμού.

Η αρχιτεκτονική του συστήματος AROMA δείχνει τις μελλοντικές τάσεις στο χώρο των συστημάτων QA, όπως αυτή στην επεξεργασία των ερωτήσεων του τύπου «*What if*». Οι «*What if*» ερωτήσεις αναμένουν απαντήσεις για το τι πρόκειται να συμβεί σε ένα σύστημα το οποίο αφορούν οι ερωτήσεις, όταν αυτό είναι υπό ορισμένες προϋποθέσεις (Maybury 2003).

Άλλη εφαρμογή συστήματος QA με δυνατότητα απάντησης από ένα πλήθος πηγών πληροφόρησης αποτελεί η εργασία των Hobbs et al. (2004). Και εδώ έχουμε ερωτήσεις στη φυσική γλώσσα, οι δε απαντήσεις δίνονται αφού ληφθούν συμβουλές από διάφορες πηγές πληροφόρησης. Υπάρχει δε, ένα αυτόματο σύστημα παραγωγής, εφοδιασμένο με μια αξιωματική θεωρία για τον μικρόκοσμο, το οποίο συντονίζει τη διαδικασία. Οι πηγές πληροφόρησης είναι βάσεις δεδομένων, ιστοσελίδες, προγράμματα και αδόμητα κείμενα και οι απαντήσεις μπορούν να περιέχουν κείμενο ή απεικονίσεις. Τα περισσότερα πειράματα αυτής της εφαρμογής είχαν σχέση με την Γεωγραφία. Το σύστημα, όμως, θα μπορούσε να εφαρμοστεί και σε άλλα πεδία Γνώσης (Hobbs et al., 2004).

Όπως προέκυψε από την παραπάνω βιβλιογραφική επισκόπηση, οι πρώτες εργασίες για τα αρχικά Συστήματα Απάντησης Ερωτήσεων (Lindsay, Green et al., Thompson and Craig, Kontos et al., Winograd, Moldovan, Hobbs et al.), περιορίζονταν σε συστήματα που δίνουν απαντήσεις σε ερωτήσεις 'ενώνοντας' προτάσεις ή τμήματα προτάσεων χωρίς 'μεταγνωσιακές διαδικασίες'. Ωστόσο, λόγω των νέων απαιτήσεων αναζήτησης για την παραγωγή απαντήσεων κατόπιν συλλογισμού που συνδυάζει πληροφορίες και παράγει πληροφορίες που δεν είναι ρητά δοσμένες στη βάση δεδομένων και κειμένων προέκυψε η ιδέα της δημιουργίας Συστημάτων Απάντησης Ερωτήσεων που να διαθέτουν λειτουργίες εμπνευσμένες από 'μεταγνωσιακές διαδικασίες' του ανθρώπου.

Στο επόμενο Κεφάλαιο εκτίθεται το θεωρητικό μοντέλο (model) των *‘μεταγνωσιακών’* διαδικασιών του ανθρώπου που πρότεινε ο Flavell (Metacognition and Cognitive Monitoring, 1979), και οι οποίες αποτελούν πηγή έμπνευσης και της δικιάς μας εργασίας. Ειδικότερα πηγή έμπνευσής μας αποτέλεσε η θεωρητική προσέγγιση και η υπολογιστική εφαρμογή του υλοποιηθέντος συστήματος *περί στρατηγικών* για την παραγωγή απαντήσεων αυτών των συστημάτων. Επίσης για την περαιτέρω ανασκόπηση των νεωτέρων εξελίξεων της μελέτης του ψυχολογικού φαινομένου «metacognition», ενδεικτικά επιλέξαμε τις πρόσφατες σχετικές προσεγγίσεις που παρουσιάζονται στα άρθρα των Zion et al., Vosniadou & Kyriakopoulou, Chi Min & VanLehn.

ΚΕΦΑΛΑΙΟ 3

Η Μεταγνωσιακή λειτουργία στη Ψυχολογία.

3.1 Ορισμός της μεταγνωσιακής λειτουργίας (metacognition)

Ο όρος «metacognition» (μεταγνωσιακές λειτουργίες) είναι συνδεδεμένος με τον Flavell (1976), ο οποίος τον χρησιμοποίησε για πρώτη φορά επίσημα στο άρθρο του «Metacognitive aspects of problem solving» και τον εμπνεύστηκε από μια προηγούμενη ιδέα του για την «metamemory» (Flavell, 1971; Georgiades, 2004). Σύνδεσε ο Flavell (1979), τον όρο «metacognition» με την γνώση που έχουν οι μαθητές για την δικιά τους γνώση και τον όρισε ως «τη γνώση και νοητική λειτουργία για τα γνωσιακά φαινόμενα» (knowledge and cognition about cognitive phenomena) (Flavell, 1979, σελ.906). Επίσης ισχυρίζεται (Flavell, ό.π.) ότι η επίβλεψη διαφόρων γνωσιακών δραστηριοτήτων γίνεται μέσω των ενεργοποιήσεων των διαδικασιών που αναφέρονται στη συνέχεια και οι οποίες στο σύνολό τους δεν αποτελούν αντικείμενο της έρευνάς μας παρά μόνον πηγή έμπνευσης, κυρίως σε ότι αφορά την ενεργοποίηση των κατάλληλων στρατηγικών που χρησιμοποιούνται για την επίτευξη ενός στόχου. Το σύστημα που παρουσιάζεται στην παρούσα εργασία ενεργοποιεί την αλλαγή της στρατηγικής ανάλογα με την αποτυχία ή την επιτυχία στην απάντηση της ερώτησης που αποτελεί και τον στόχο του συστήματός μας. Οι διαδικασίες που ενεργοποιούνται και αλληλεπιδρούν μεταξύ τους για την επίβλεψη των διαφόρων γνωσιακών δραστηριοτήτων κατά τον Flavell (1979), είναι οι:

- 1 Metacognitive knowledge (μεταγνώση),
 - 1α person category (όταν αναφέρεται στο άτομο),
 - 1β task category (όταν αναφέρεται στη δραστηριότητα),
 - 1γ strategy category (όταν αναφέρεται στην επιλογή στρατηγικής για την επίτευξη του στόχου),
- 2 Metacognitive experience (μεταγνωσιακή εμπειρία),
- 3 Metacognitive goals or tasks (μεταγνωσιακοί στόχοι),
- 4 Metacognitive action or strategies (μεταγνωσιακές στρατηγικές).

Κατά τον Flavell (1976), ο όρος «metacognition» (μεταγνωσιακές λειτουργίες) αναφέρεται στη γνώση κάποιου, σχετικά με τις γνωσιακές του διαδικασίες, ή τα προϊόντα που προκύπτουν καθώς και οτιδήποτε σχετίζεται με αυτά, όπως οι ιδιότητες που έχουν σχέση με την μάθηση πληροφοριών ή δεδομένων. Ισχυρίστηκε ότι σε οποιοδήποτε είδος γνωσιακής αλληλεπίδρασης ενός ανθρώπου με άλλους ανθρώπους ή

ενός ανθρώπου με ένα μη ανθρώπινο περιβάλλον, μπορεί να γίνουν διάφορες ενέργειες επεξεργασίας πληροφοριών και ότι οι «μεταγνωσιακές λειτουργίες» αφορούν κυρίως την ενεργή επίβλεψη, τη συνακόλουθη ρύθμιση και την ενορχήστρωση όλων αυτών των διαδικασιών που εξαρτώνται από τα γνωστικά αντικείμενα ή δεδομένα και εφαρμόζονται για την επίτευξη ενός συγκεκριμένου σκοπού. Έδωσε δε διάφορα παραδείγματα στα οποία φαίνεται ο σημαντικός ρόλος των «μεταγνωσιακών λειτουργιών» (μεταμνήμης, μεταμάθησης, μεταπροσοχής, μεταγλώσσας): Π.χ. αν κάποιος παρατηρήσει ότι έχει μεγαλύτερη δυσκολία να μάθει το Α θέμα από το Β, ή αν του έλθει ξαφνικά η ιδέα ότι πρέπει να ελέγξει ξανά το Γ, πριν αποφασίσει να το δεχτεί σαν γεγονός, ή αν σκέφτηκε ξαφνικά ότι πρέπει να εξετάσει με προσοχή κάθε εναλλακτική λύση πριν αποφασίσει τι θα επιλέξει. Το ίδιο συμβαίνει και αν ένα άτομο κρίνει ότι πρέπει να κρατήσει σημειώσεις για το Δ διαφορετικά θα το ξεχάσει (Flavell, 1976, σελ.232).

Τον ορισμό που έδωσε ο Flavell (ό.π.) ακολούθησαν πλήθος άλλων ορισμών, όρων και αναλύσεων για την «metacognition». Ο Georgiades (ό.π.), στην επισκόπησή του για τις τρεις δεκαετίες της «metacognition», αναφέρει ότι στην έρευνά τους οι Paris & Jacobs (1984), Cross & Paris (1988), Paris & Winograd (1990), προσδιόρισαν δύο ουσιαστικά χαρακτηριστικά γνωρίσματα στον δικό τους ορισμό της «metacognition»: την «αυτό-αξιολόγηση» (self-appraisal) και την «αυτό-διαχείριση» (self-managemeng) της «νοητικής λειτουργίας» (cognition). Η «αυτό-αξιολόγηση» περιλαμβάνει στοχασμούς για την κατανόηση, τις ικανότητες και τη συναισθηματική κατάσταση των μαθητών κατά την διάρκεια της μαθησιακής διαδικασίας, ενώ η «αυτό-διαχείριση» αναφέρεται σε «metacognitions» που βρίσκονται σε ενεργή κατάσταση, δηλαδή σε νοητικές διεργασίες που βοηθούν στην καλύτερη οργάνωση των επιμέρους τμημάτων ενός προβλήματος για την επίλυσή του (Paris & Winograd 1990:8). Επίσης στην επισκόπηση αναφέρονται και άλλοι όροι που χρησιμοποιήθηκαν επίσης στη βιβλιογραφία για να περιγράψουν την αντίληψη των προβλημάτων, τις περιπτώσεις και τους τρόπους με τους οποίους τα σκεπτόμαστε ή τα περιγράφουμε και αυτοί είναι οι: «Metalearning» (White & Gustone 1989), «deutero-learning» (Bateson 1983) και «mindfulness» (Salomon & Globerson 1987). Ο Georgiades (ό.π.), ισχυρίζεται ότι το πλήθος των ορισμών, των όρων και των αναλύσεων για το τι είναι και αφορούν οι «μεταγνωσιακές λειτουργίες» έχουν γίνει αιτία σύγχυσης ενός τμήματος της βιβλιογραφίας. Αναφέρει επίσης ότι ο μεν Weinert (1987), εύρισκε κάποια 'ασάφεια' ('vague') και 'ανακρίβεια' ('imprecise') στον

ανεπίσημο ορισμό της «metacognition», οι δε Adey & Shayer (1994), παρατήρησαν την σύγχυση μεταξύ των εκπαιδευτών, όχι μόνο πάνω στην έννοια του όρου της «metacognition» αλλά στην αναγνώρισή του.

Στην παρούσα εργασία για την αποφυγή οποιασδήποτε σύγχυσης που μπορεί να προκύψει από τις ψυχολογικές συνεκδοχές του επιθέτου «metacognitive» και οι οποίες μπορεί να συνδεθούν με την ανθρώπινη «metacognition», προτείνεται η χρήση του όρου «metagnostic» (Kontos et al., 2009) αντί του όρου «metacognitive» που χρησιμοποιήθηκε στο Kontos & Armaos (2007) για τα συστήματα απάντησης ερωτήσεων, τα οποία επιδεικνύουν ιδιότητα εμπνεόμενη από «self-awareness» (αυτό-επίγνωση). Ο λόγος έγκειται στο ότι το σύστημα που παρουσιάζεται εδώ, δεν στοχεύει καθόλου στη μοντελοποίηση της ανθρώπινης «metacognition», αλλά αντ' αυτού παράγει απαντήσεις, οι οποίες επιδεικνύουν μηχανική «metacognition». Αυτό επιδεικνύεται από το γεγονός ότι οι εξηγήσεις που παράγονται αναφέρουν ρητά το ιστορικό των συλλογισμών τους. Με τον τρόπο αυτό η απάντηση των ερωτήσεων του συστήματος εμπνέεται μόνο ως προς το σκέλος της ρητότητας που περιλαμβάνεται στον ορισμό της «Explicit self-awareness» κατά Schubert (2005), και όχι ως προς τις άλλες φιλόδοξες «ανθρωπομορφικές» (human-like) ιδιότητες οι οποίες δεν αφορούν το σύστημα.

Ο Flavell (ό.π.) σχολιάζει την έρευνα των Resnick and Glaser (1976), οι οποίοι μελέτησαν την απόδοση των παιδιών στη λύση προβλημάτων, καθώς και τα αίτια των αποτυχιών τους στη λύση. Επισημαίνει την προσέγγιση των ερευνητών στο πρόβλημα της μάθησης και της ανάπτυξης των παιδιών, δηλαδή τον τρόπο με τον οποίο τα παιδιά συγκεντρώνουν, συντονίζουν ή ενσωματώνουν τις ήδη υπάρχουσες γνώσεις και δεξιότητες σε νέες διαδικασίες. Ισχυρίστηκε ότι η μη αναμενόμενη αποτυχία των παιδιών στην έρευνα των Resnick and Glaser, έχει σχέση με τις «μεταγνωσιακές λειτουργίες» και συγκεκριμένα με έναν υποχώρο της, τη «μεταμνήμη» (metamemory) (Kreutzer et al, 1975). Ο Flavell (ό.π.), υποστήριξε ότι η αποθήκευση και η ανάκτηση των πληροφοριών είναι «μεταγνωσιακές λειτουργίες», οι οποίες μπορούν βαθμιαία να αποκτηθούν από ένα παιδί. Δηλαδή ότι το παιδί μπορεί να μάθει, να κρίνει τις καταστάσεις εκείνες από τις οποίες μπορεί να αντλήσει συγκεκριμένες πληροφορίες και να τις αποθηκεύσει ώστε να τις χρησιμοποιήσει μελλοντικά στην επίλυση κάποιου προβλήματος. Επιπλέον το παιδί μπορεί να εκπαιδευτεί στις «μεταγνωσιακές λειτουργίες» που είναι απαραίτητες για να διατηρεί μια πληροφορία ενημερωμένη και έτοιμη να ανακτηθεί όποτε τη χρειαστεί. Ο Flavell

(ό.π.), χρησιμοποιεί τους όρους «αποθήκευση» και «ανάκτηση» δίχως να τους δώσει το πρόθεμα «μνήμη». Θεωρεί ότι η «αποθήκευση» και η «ανάκτηση» μπορούν να αναφέρονται και σε εξωτερικές πηγές όπως είναι τα βιβλία, οι μαγνητοταινίες, οι υπολογιστές αλλά και η μνήμη ενός άλλου ανθρώπου. Επίσης ισχυρίζεται ότι τα παιδιά όσο μεγαλώνουν μαθαίνουν πολλά για το «πώς», «πού» και «πότε» μπορούν να αποθηκεύουν και να ανακτούν μια πληροφορία. Η διαδικασία για το «πώς» περιλαμβάνει μια ποικιλία στρατηγικών αποθήκευσης και ανάκτησης. Η διαδικασία για το «πού» σχετίζεται με τους πόρους αποθήκευσης και ανάκτησης, ενώ η διαδικασία για το «πότε» συνδέεται με τις διάφορες καταστάσεις που απαιτούν ενέργειες και συγκεκριμένες προσπάθειες ανάκτησης και εφαρμογής αυτών που έχουν αποθηκευτεί (Flavell, 1976, σελ.233).

Κατά τον Flavell (1976), η έρευνα αναφορικά με την μάθηση και την ανάπτυξη της μνήμης των παιδιών θα κατέχει πλέον δεσπόζουσα θέση μεταξύ των ενδιαφερόντων των ερευνητών. Αναφέρει δε (Flavell, 1979) το συμπέρασμα που κατέληξαν οι ερευνητές: *οι «μεταγνωσιακές λειτουργίες» διαδραματίζουν ένα σημαντικό ρόλο στην προφορική μετάδοση των πληροφοριών, στην προφορική πειθώ, στην προφορική κατανόηση, στην κατανόηση της ανάγνωσης, στη γραφή, στην απόκτηση γλώσσας (εκμάθηση της πρώτης γλώσσας), στην προσοχή, στη μνήμη, στην επίλυση προβλήματος και στους διάφορους τύπους του αυτοελέγχου και της αυτοεκπαίδευσης.* Ως εκ τούτου θεωρώντας ότι η φύση και η ανάπτυξη των «μεταγνωσιακών λειτουργιών», όπως και της γνωσιακής επίβλεψης και ρύθμισης αναδεικνύει έναν ενδιαφέροντα και πολλά υποσχόμενο χώρο έρευνας, προσπαθεί να δώσει απάντηση για το τι υπάρχει σε αυτόν τον τομέα που θα μπορούσε να μάθει ένα παιδί ή ένας έφηβος. Δηλαδή ποια γνώση και ποια συμπεριφορά παρόμοια με αυτήν ενός ενήλικα θα μπορούσε να αποτελέσει εδώ τον αναπτυξιακό στόχο, προς τον οποίον ένα παιδί θα προχωρούσε βαθμιαία; Η απάντηση του Flavell (1978) δίνεται στη συνέχεια.

3.2 Το θεωρητικό πλαίσιο του Flavell (1978)

Ο Flavell (1978) παρουσίασε ένα θεωρητικό πλαίσιο ως απάντηση στο παραπάνω ερώτημα, στο οποίο η επίβλεψη των διαφόρων γνωσιακών δραστηριοτήτων γίνεται μέσω των ενεργοποιήσεων και αλληλεπιδράσεων των διαδικασιών:

1 Την «μεταγνώση» (Metacognitive knowledge), η οποία είναι ένα τμήμα της συνολικής αποθηκευμένης γνώσης ενός παιδιού ή ενός ενήλικα και αφορά στις γνώσεις τόσο για το ποιοι παράγοντες και ποιες μεταβλητές ενεργούν και

αλληλεπιδρούν στις γνωσιακές δραστηριότητες, όσο και για τον τρόπο επηρεασμού της πορείας και της έκβασης των γνωσιακών δραστηριοτήτων. Στο σύστημα που κατασκευάστηκε, ο τρόπος με τον οποίο επιτυγχάνεται η ανάλυση του κειμένου ενός οποιουδήποτε γνωστικού πεδίου, δίνει τη δυνατότητα χειρισμού των προτάσεων του για την εξαγωγή ρητής ή άρρητης γνώσης, απαραίτητης για την βέλτιστη επιλογή στρατηγικής και την παραγωγή απάντησης σε ερώτηση που εισάγεται στο σύστημα.

Την «μεταγνώση» (metacognitive knowledge) ο Flavell (1979) τη διακρίνει περαιτέρω στην γνώση για τους παράγοντες και τις μεταβλητές ενός ατόμου (person category), στην γνώση των μεταβλητών μιας συγκεκριμένης δραστηριότητας (task category) και στην γνώση των μεταβλητών μιας στρατηγικής (strategy category):

1α η γνώση για τους παράγοντες και τις μεταβλητές ενός ατόμου (person category) αναφέρεται σε όλα όσα πιστεύει κάθε άτομο για τη φύση του και για τους άλλους. Τη διάκριση αυτή τη διαχωρίζει περαιτέρω ως προς τις ενδοατομικές, διατομικές και καθολικές πεποιθήσεις.

1β η γνώση των μεταβλητών μιας συγκεκριμένης δραστηριότητας (task category). Τη διάκριση αυτή τη διαχωρίζει περαιτέρω στην περίπτωση που αφορά διαθέσιμη πληροφορία για τη φύση της γνωσιακής δραστηριότητας και στην περίπτωση που αφορά απαιτήσεις που έχουν σχέση με τη δραστηριότητα ή τον στόχο. Παράδειγμα που αφορά την συγκεκριμένη δραστηριότητα (task category) έχουμε όταν ένα παιδί μαθαίνει ότι κάποιες γνωσιακές δραστηριότητες είναι πιο απαιτητικές και πιο δύσκολες από άλλες, ακόμη και αν του δίνονται οι ίδιες πληροφορίες (η γνώση ότι θα απαιτηθεί περισσότερος χρόνος για να διαβαστεί και να κατανοηθεί ένα επιστημονικό κείμενο, από ότι ένα μυθιστόρημα) (Livingston, 1997).

1γ η γνώση των μεταβλητών μιας στρατηγικής (strategy category) που αφορά τη γνώση που μπορεί να αποκτηθεί, ως προς τις αποτελεσματικές στρατηγικές επίτευξης των στόχων και των επιμέρους στόχων για τις διάφορες γνωσιακές δραστηριότητες. Για παράδειγμα όταν ένα παιδί θεωρεί ότι ο καλύτερος τρόπος για να μάθει και να μπορέσει να θυμηθεί ένα πλήθος πληροφοριών, είναι να δώσει περισσότερη προσοχή στα σημαντικότερα σημεία των πληροφοριών και να προσπαθήσει να τα διατυπώσει με δικές του λέξεις.

Στην παρούσα εργασία όταν το σύστημα που περιγράφεται εφαρμόζεται στο μικρόκοσμο της Γεωμετρίας, οι μεταβλητές μιας στρατηγικής αφορούν τις οντότητες και τα συνώνυμά τους που περιέχονται σε μια ερώτηση, η πράξη που τα συνδέει και η χρήση ή όχι προαπαιτούμενης γνώσης για την απάντηση της ερώτησης.

Κατά τον Flavell (1979), το μεγαλύτερο μέρος της μεταγνώσης (metacognitive knowledge) έχει σχέση με τις αλληλεπιδράσεις και τους συνδυασμούς μεταξύ δύο ή και τριών από τους διαχωρισμούς που αναφέρθηκαν και αφορούν τη γνώση των μεταβλητών ενός ατόμου, μιας δραστηριότητας, και μιας στρατηγικής). Σαν παράδειγμα συνδυασμού και των τριών αναφέρει την περίπτωση, κατά την οποία ένας πιστεύει (διαφορετικά από κάποιον άλλον) ότι θα μπορούσε να χρησιμοποιήσει τη στρατηγική A (και όχι τη B) για την δραστηριότητα X (σε αντίθεση με ότι θα έκανε για την δραστηριότητα Y). Ισχυρίζεται δε ο Flavell (ό.π.) ότι η «μεταγνώση» επιφέρει σημαντικά αποτελέσματα στις γνωσιακές δραστηριότητες τόσο των παιδιών όσο και των ενηλίκων, διότι επηρεάζει την επιλογή, την αξιολόγηση, την αναθεώρηση, και την εγκατάλειψη ορισμένων γνωσιακών δραστηριοτήτων, στόχων και στρατηγικών για την επίτευξη ενός σκοπού.

2 Την «μεταγνωσιακή εμπειρία» (Metacognitive experience), την οποία ο Flavell (1979) περιγράφει ως την εμπειρία που βιώνει κάποιος σε μια στιγμή σύγχυσης αλλά και την απορία κάποιου για το αν αντιλαμβάνεται τις προθέσεις κάποιου άλλου. Οι «μεταγνωσιακές εμπειρίες» μπορούν να εμφανιστούν πριν, μετά ή κατά τη διάρκεια μιας γνωσιακής δραστηριότητας (π.χ. η αίσθηση που έχει κάποιος ότι θα αποτύχει στην επόμενη δραστηριότητα). Πολλές «μεταγνωσιακές εμπειρίες» έχουν να κάνουν με τη θέση που κατέχει κάποιος μέσα σε μια δραστηριότητα και το είδος της προόδου που επιτυγχάνει ή θα επιτύχει (αυτό συμβαίνει όταν κάποιος θεωρεί ότι έχει απομνημονεύσει τις οδηγίες που του έχουν δοθεί ή ξεκινάει την επίλυση ενός προβλήματος γιατί νομίζει ότι είναι εύκολη η λύση του). Ο Flavell (ό.π.) επίσης υποστηρίζει ότι κάποιες «μεταγνωσιακές εμπειρίες» περιγράφονται καλύτερα ως στοιχεία της «μεταγνώσης» (metacognitive knowledge) όπως στην περίπτωση που επιλύουμε ένα δύσκολο πρόβλημα και θυμόμαστε ξαφνικά ένα άλλο παρόμοιο που επιλύσαμε. Όμως δεν συμβαίνει αυτό πάντα, διότι η αίσθηση κάποιου ότι απέχει από τον στόχο του, σαν «μεταγνωσιακή εμπειρία», δεν αποτελεί στοιχείο της «μεταγνώσης» (metacognitive knowledge), παρόλο που οι ενέργειες που γίνονται εξ αιτίας αυτής της αίσθησης καθοδηγούνται από την «μεταγνώση» (metacognitive knowledge). Έτσι ο Flavell (ό.π.), θεωρεί ότι η «μεταγνώση» (metacognitive knowledge) και οι «μεταγνωσιακές εμπειρίες» σχηματίζουν μερικώς επικαλυπτόμενα σύνολα. Επιπλέον υποστηρίζει ότι οι «μεταγνωσιακές εμπειρίες» έχουν σημαντική επίδραση στους γνωσιακούς στόχους, στη «μεταγνώση» (metacognitive knowledge) και στις γνωσιακές στρατηγικές.

3 Τους «Μεταγνωσιακούς σκοπούς ή στόχους» (Metacognitive goals or tasks) που αποτελούν τον τρίτο κύριο διαχωρισμό που πρότεινε ο Flavell (1979) και αφορά τους αντικειμενικούς σκοπούς και την επιθυμητή κατάληξη των γνωσιακών δραστηριοτήτων. Στους σκοπούς ή στόχους περιλαμβάνεται η κατανόηση, η απομνημόνευση γεγονότων, η παραγωγή κειμένων, η λύση ενός μαθηματικού προβλήματος και γενικά η βελτίωση της γνώσης ενός ατόμου σε κάποιο θέμα. Ισχυρίζεται ο Flavell (1979), ότι για την επίτευξη ενός σκοπού ή στόχου μιας δραστηριότητας είναι απαραίτητη η υπάρχουσα «μεταγνώση» (metacognitive knowledge) του ατόμου για το σκοπό ή το στόχο και η οποία θα οδηγήσει στη δυνατότητα πραγματοποίησής του.

4 Τις «**μεταγνωσιακές στρατηγικές**» (metacognitive action or strategies), οι οποίες μας ενδιαφέρουν ιδιαίτερα στην έρευνά μας, και αποτελούν τον τέταρτο από τους κύριους διαχωρισμούς που πρότεινε ο Flavell (1979). Οι «μεταγνωσιακές στρατηγικές» αφορούν τις διαδικασίες επίβλεψης προόδου των γνωσιακών δραστηριοτήτων ενός ατόμου, *για να εξασφαλισθεί η επιτυχής ολοκλήρωση ενός στόχου*. Αυτές οι διαδικασίες βοηθούν στη ρύθμιση και επίβλεψη της μάθησης και αποτελούνται από γνωσιακές δραστηριότητες σχεδιασμού, επίβλεψης και ελέγχου της κατάληξης αυτών των δραστηριοτήτων (Livingston, 1997). Στο σύστημα εφαρμόζεται μια σειρά στρατηγικών, οι οποίες ανάλογα με την γραφή των οντοτήτων και την ανάγκη χρήσης ή μη προαπαιτούμενης γνώσης ρυθμίζει την διαδικασία για την απάντηση της ερώτησης που εισάγεται στο σύστημα. Ισχυρίζεται δε ο Flavell (1979), ότι οι μεταγνωσιακές στρατηγικές ενεργοποιούνται από τις μεταγνωσιακές εμπειρίες με σκοπό την επίτευξη ενός γνωσιακού ή μεταγνωσιακού στόχου. Ένα παράδειγμα μεταγνωσιακής στρατηγικής έχουμε όταν ένας μαθητής μετά την ανάγνωση ενός κειμένου μπορεί να έχει την εντύπωση («μεταγνωσιακή εμπειρία») ότι δεν το ξέρει αρκετά καλά και το ξαναδιαβάζει (γνωσιακή στρατηγική) για να βελτιώσει τη γνώση του (γνωσιακός στόχος). Ίσως όμως αναρωτηθεί ο μαθητής («μεταγνωσιακή εμπειρία») κατά πόσο το έχει κατανοήσει (μεταγνωσιακός στόχος) και μέσω της αυτοεξέτασης (μεταγνωσιακή στρατηγική ελέγχου της κατανόησης) να αποφασίσει ότι πρέπει να το ξαναδιαβάσει (ρύθμιση). Κατά τον Flavell (1979), γίνεται *επίκληση γνωσιακών στρατηγικών για να επιτευχθεί γνωσιακή πρόοδος και επίκληση μεταγνωσιακών στρατηγικών για να την 'επιβλέψουν'* (monitor) (Flavell, 1979). Κατά τον Georgiades (ό.π.), η επίτευξη της γνωσιακής προόδου στον ισχυρισμό του Flavell, αναφέρεται και στη «διευκόλυνση» (facilitate) της μάθησης

καθώς επίσης και στην ολοκλήρωση του σκοπού. Σε πολλές περιπτώσεις όμως ανεξαρτήτως του λόγου (επίτευξη γνωσιακής προόδου ή ‘επίβλεψη’), για τον οποίο γίνεται η επίκληση μίας στρατηγικής (γνωσιακή ή μεταγνωσιακής) επιτυγχάνονται και οι δύο στόχοι. Υπάρχει περίπτωση, όμως, να γίνεται επίκληση της ίδιας στρατηγικής και για τους δύο στόχους, κάτι που συμβαίνει στο σύστημα της έρευνάς μας, διότι με την ίδια στρατηγική εξασφαλίζεται και η επιτυχής ολοκλήρωση του στόχου που είναι η απάντηση της ερώτησης αλλά και η επίβλεψη της ακολουθούμενης διαδικασίας (επιλογή κανόνων συμπερασμού).

Συνοψίζοντας, μπορούμε να πούμε ότι ο Flavell (1978) μέσω της έρευνάς του, ενθάρρυνε μεν την ανάπτυξη των «μεταγνωσιακών λειτουργιών» στους μαθητές των σχολείων υποστηρίζοντας ότι «η αύξηση της ποσότητας και της ποιότητας στις «μεταγνωσιακές λειτουργίες» και στις δεξιότητες επίβλεψης των παιδιών μέσω της συστηματικής κατάρτισης, μπορεί να είναι εφικτή και επιθυμητή» (Flavell, 1978), θεώρησε όμως ότι πρέπει να μελετηθούν σε βάθος τα αποτελέσματα της έρευνάς του, πριν χρησιμοποιηθούν σε οποιεσδήποτε εκπαιδευτικές παρεμβάσεις για να αποφευχθούν ενδεχόμενες υπερβολές ή αστοχίες.

3.3 Νεώτερες σχετικές εξελίξεις στον τομέα της Γνωσιακής Επιστήμης και Ψυχολογίας

Από τον Flavell (1976; 1978; 1979) και μετά - μέχρι και σήμερα - έχουν λάβει χώρα ποικίλες έρευνες στο χώρο της Γνωσιακής Επιστήμης και της Ψυχολογίας για την περαιτέρω διερεύνηση του ψυχολογικού φαινομένου «metacognition».

Μια από τις προσπάθειες των ψυχολόγων να δημιουργήσουν υπολογιστικά μοντέλα για να εξετάσουν τη σχέση που έχουν οι «μεταγνωσιακές λειτουργίες» ως προς τη συμπεριφορά και δράση των ανθρώπων, είναι αυτή της Lynn Reder (1996) και των συνεργατών της στην εφαρμογή του SAC (Sources of Activation Confusion). Το SAC προβλέπει με επιτυχία αν τα υποκείμενα θα χρησιμοποιήσουν την διαδικασία ανάκτησης της μνήμης ή αν θα εξετάσουν εκ νέου τη στρατηγική που θα εφαρμόσουν, για να απαντήσουν σε μια ερώτηση.

Οι ερευνητές Chi et al. (1989) και Chi, M.T.H. (1995), ισχυρίζονται ότι η μάθηση που αποδίδει περισσότερο είναι εκείνη που συνοδεύεται από την παραγωγή ερωτήσεων εκ μέρους του υποκειμένου, το οποίο και τις απαντά. Η διαδικασία αυτή καλείται *self-explanation effect* και έχει μοντελοποιηθεί υπολογιστικά από τον VanLehn και τους συνεργάτες του (VanLehn et al., 1990, 1992). Σε σχέση με τις έρευνες των Chi και

VanLehn, οι Recker & Pirolli (1995), χρησιμοποίησαν το μοντέλο μάθησης SURF και έδειξαν ότι αυτό μπορούσε να εξηγήσει τις διαφορές που παρουσιάζονται σε όσους προσπαθούν να μάθουν προγραμματισμό σε LISP με τη βοήθεια γραπτών οδηγιών. Συγκεκριμένα ότι τα άτομα που εξηγούν στον εαυτό τους εκείνα που κατανοούν κατά τη διάρκεια της διδασκαλίας ή τις στρατηγικές που χρησιμοποιούνται αποδίδουν καλύτερα από εκείνους που δεν ενεργούν με αυτόν τον τρόπο. Ισχυρίζονται δε ότι κάποια υποκείμενα εκμεταλλεύονται την μεταγνωσιακή ανατροφοδότηση, κατά τον έλεγχο της κατανόησης, για να κρίνουν το πότε μαθαίνουν (Recker & Pirolli, 1994).

Τέλος ενδεικτικά επιλέγουμε να αναφερθούμε στις παρακάτω πρόσφατες σχετικές προσεγγίσεις:

- 1 The effects of metacognitive instruction embedded within an asynchronous learning network on scientific inquiry skills (Zion et al., *Bar-Ilan University, Israel*, 2005)

Στη συγκεκριμένη μελέτη οι ερευνητές εξετάζουν τα αποτελέσματα τεσσάρων μεθόδων εκμάθησης όσον αφορά την αναζήτηση δεξιοτήτων που αποκτώνται κατόπιν εκπαίδευσης 407 μαθητών ηλικίας 15 ετών σε μαθήματα θετικών επιστημών. Οι τέσσερις μέθοδοι εκμάθησης είναι:

A. η ασύγχρονη εκμάθηση μέσα σε ένα ψηφιακό περιβάλλον αναζήτησης, κατά την οποία οι μαθητές σε μια εικονική (virtual) τάξη από διαφορετικά σχολεία δίχως γεωγραφικούς περιορισμούς, ανταλλάσσουν πληροφορίες άμεσα μέσω του δικτύου και λαμβάνουν ‘μεταγνωσιακή’ καθοδήγηση «metacognitive-guidance».

B. όταν η ασύγχρονη εκμάθηση μέσα στο ψηφιακό περιβάλλον αναζήτησης «Asynchronous Learning Network» - (ALN), γίνεται δίχως οι μαθητές να λαμβάνουν ‘μεταγνωσιακή’ καθοδήγηση.

Γ. όταν η εκμάθηση γίνεται πρόσωπο με πρόσωπο «face-to-face interaction» και περιλαμβάνει ‘μεταγνωσιακή’ καθοδήγηση.

Δ. όταν η εκμάθηση γίνεται πρόσωπο με πρόσωπο «face-to-face interaction» δίχως ‘μεταγνωσιακή’ καθοδήγηση.

Η μελέτη εξέτασε τις επιδράσεις των τεσσάρων μεθόδων εκμάθησης, όσον αφορά την αναζήτηση δεξιοτήτων που αποκτώνται κατόπιν εκπαίδευσης, σε μαθητές με ικανότητα μάθησης σε επιστημονικά θέματα γενικά και ειδικά στο πεδίο της Μικροβιολογίας. Τα αποτελέσματα έδειξαν ότι η ομάδα έρευνας της πρώτης μεθόδου μάθησης υπερέιχε των άλλων ομάδων έρευνας. Η ερευνητική ομάδα της τέταρτης

μεθόδου μάθησης είχε τη χαμηλότερη επίδοση, ενώ δεν βρέθηκε κάποια σημαντική διαφορά μεταξύ των ομάδων έρευνας των δύο άλλων μεθόδων μάθησης. Κατά τους ερευνητές η πρώτη μέθοδος εκμάθησης είχε σημαντική συνεισφορά και στην επίτευξη του σχεδιασμού των πειραμάτων αλλά και στην εξαγωγή συμπερασμάτων από τους μαθητές. Ισχυρίζονται επίσης ότι η χρήση της ‘μεταγνωσιακής’ κατάρτισης σε ένα περιβάλλον ALN αυξάνει τις επιδράσεις του ALN στις επιδόσεις των σπουδαστών στα επιστημονικά μαθήματα.

2. The Problem of Metaconceptual Awareness in Theory Revision. (Vosniadou, S. & Kyriakopoulou, N., Department of Philosophy and History of Science, National Kapodistrian University of Athens, 2006).

Στην έρευνα των Vosniadou & Kyriakopoulou (2006) περιγράφεται ένα πείραμα, στο οποίο εξετάζεται εάν τα παιδιά της πρωτοβάθμιας εκπαίδευσης έχουν «μετα-εννοιολογική» (metaconceptual) επίγνωση των διαδικασιών αναθεώρησης της θεωρίας. Στο πείραμα τα παιδιά έπρεπε να επιλέξουν μεταξύ φαινομενολογικών και επιστημονικών απεικονίσεων (depictions) διαφορετικών αστρονομικών φαινομένων και να δηλώσουν ποιες από αυτές τις απεικονίσεις ήταν πιο κοντά στο «είναι» (Reality) και ποιες στο «φαίνεσθαι» (Appearance).

Τα αποτελέσματα της έρευνας έδειξαν ότι όσο αυξάνονταν η ηλικία των παιδιών, τόσο αυξάνονταν ο αριθμός των προτιμήσεων στις επιστημονικές απεικονίσεις και ότι τα παιδιά που επέλεξαν και τις δύο αναπαραστάσεις (representations), φαινομενολογικές και επιστημονικές των αστρονομικών φαινομένων, δεν μπορούσαν να αποφασίσουν ποιες απεικονίσεις αναπαριστούσαν καλύτερα το «είναι» και ποιες το «φαίνεσθαι».

Η «μετα-εννοιολογική» επίγνωση όπως ισχυρίζονται οι Vosniadou & Kyriakopoulou (ό.π.) απαιτεί την ικανότητα ταυτόχρονης σύγκρισης δύο εναλλακτικών αναπαραστάσεων του ίδιου αστρονομικού φαινομένου, όπου η μία αναπαράσταση βασίζεται στη φαινομενολογική απεικόνιση και η άλλη βασίζεται σε επιστημονικό μοντέλο. Επίσης οι Vosniadou & Kyriakopoulou (ό.π.) υποστηρίζουν ότι η λήψη απόφασης για το ποιές απεικονίσεις είναι πιο κοντά στο «φαίνεσθαι» και ποιές είναι πιο κοντά στην πραγματικότητα, απαιτεί την ικανότητα να διατηρούνται στο νου και να συγκρίνονται διαφορετικές αναπαραστάσεις που απεικονίζουν την ίδια κατάσταση του κόσμου. Αυτό απαιτεί «μεταγνωστικές» (metacognitive) ικανότητες και πιο συγκεκριμένα την ικανότητα της κατανόησης ότι οι αναπαραστάσεις των

καταστάσεων του κόσμου είναι θεωρητικές οντότητες και υποθέσεις που μπορούν να εξετασθούν, να βρεθούν λανθασμένες και να αντικατασταθούν από άλλες.

3. The Impact of Explicit Strategy Instruction on Problem-solving Behaviors across Intelligent Tutoring Systems. (Chi. M. & VanLehn, K., 2007).

Οι ερευνητές Chi & VanLehn (2007) ισχυρίζονται ότι η ρητή διδασκαλία μιας στρατηγικής στην επίλυση προβλημάτων, βελτίωσε την απόδοση των σπουδαστών όχι μόνο στον τομέα (Πιθανότητες) στον οποίον αυτοί διδάχτηκαν αρχικά τη στρατηγική αλλά και στον επόμενο τομέα (Φυσική), διαφορετικό από τον προηγούμενο δίχως να έχουν διδαχτεί ξανά οι σπουδαστές την ίδια στρατηγική. Υποστηρίζουν ότι εφόσον οι δύο τομείς έρευνας (Πιθανότητες και Φυσική) δεν έχουν επικαλυπτόμενες περιοχές, όσον αφορά το γνωστικό τους αντικείμενο, είναι πιθανόν η ίδια στρατηγική επίλυσης προβλημάτων να μεταφέρθηκε από το ένα πεδίο στο άλλο. Ισχυρίζονται επιπλέον ότι η ρητή διδασκαλία στρατηγικής ενεργεί ως αποτελεσματική υποστηρικτική υποδομή για την απόκτηση των αρχών (principles) ενός επιστημονικού πεδίου. Εφιστά την προσοχή των σπουδαστών στις ξεχωριστές (individual) αρχές του πεδίου με συνέπεια να μαθαίνουν να σκέπτονται την επίλυση προβλημάτων ως εφαρμογή των ξεχωριστών αυτών αρχών. Αυτή η μεταγνώση από τον πρώτο τομέα μπορεί να επηρεάζει τον τρόπο με τον οποίο οι σπουδαστές εστιάζουν στη συνέχεια την προσοχή τους στις αρχές του επόμενου τομέα διότι αντιλαμβάνονται ότι θα τις χρειαστούν αργότερα και πρέπει να τις γνωρίζουν καλά.

4. Eliminating the Gap between the High and Low Students through Meta-Cognitive Strategy Instruction. (Chi. M. & VanLehn, K., 2008).

Οι ερευνητές Chi & VanLehn (2008), έδειξαν ότι η διδασκαλία στρατηγικής επίλυσης προβλημάτων μέσω του συστήματος: Intelligent Tutoring Systems (ITSs), εξαλείφει τη διαφορά απόδοσης μεταξύ των σπουδαστών, όχι μόνο στο πεδίο (Πιθανότητες) για το οποίο οι σπουδαστές διδάχτηκαν αυτή τη στρατηγική, αλλά και σε ένα δεύτερο (Φυσική) στο οποίο δεν διδάχτηκαν αυτή τη στρατηγική. Η στρατηγική την οποία προτείνουν οι Chi & VanLehn ό.π. περιλαμβάνει δύο κύρια συστατικά: το ένα επιλύει προβλήματα μέσω μιας αλυσιδωτής διαδικασίας συλλογισμών, από τον στόχο προς κάποια προηγούμενη θέση της διαδικασίας επίλυσης του προβλήματος, το οποίο στη διεθνή βιβλιογραφία καλείται 'Backward Chaining' (BC) και το άλλο εφιστά την προσοχή των σπουδαστών στα χαρακτηριστικά κάθε μιας εκ των αρχών του πεδίου.

Οι Chi & VanLehn (ό.π.), υποστηρίζουν ότι το αποτελεσματικό στοιχείο της στρατηγικής που διδάχτηκαν οι σπουδαστές δεν αφορά τη διαδικασία συλλογισμού BC, αλλά την ικανότητα της έμφασης στις αρχές του πεδίου. Για το λόγο αυτό προτείνουν οι σπουδαστές χαμηλής επίδοσης να διδάσκονται και να εστιάζουν στις εφαρμογές των ξεχωριστών αρχών του πεδίου.

Στο κεφάλαιο αυτό εκτέθηκαν ο ορισμός και το θεωρητικό πλαίσιο της *‘μεταγνωσιακής λειτουργίας’* «metacognition» καθώς επίσης και οι πρόσφατες προσεγγίσεις από τους ερευνητές του τομέα αυτού της Ψυχολογίας.

Στο επόμενο κεφάλαιο παρουσιάζεται η επίδραση του τομέα των *‘μεταγνωσιακών λειτουργιών’* της Ψυχολογίας και άλλων τομέων σκέψης στην Τεχνητή Νοημοσύνη με την αναφορά και τον σχολιασμό ορισμένων σχετικών άρθρων.

ΚΕΦΑΛΑΙΟ 4

Συστήματα Τεχνητής Νοημοσύνης εμπνευσμένα από Μεταγνωσιακές Διαδικασίες

4.1 Γενικά περί ‘μεταγνωσιακών’ διαδικασιών

Στο προηγούμενο κεφάλαιο έγινε μια σύντομη ανασκόπηση της βιβλιογραφίας σχετικής με τη ‘μεταγνωσιακή’ διαδικασία στην Ψυχολογία. Υπάρχουν όμως και σε άλλους τομείς της ανθρώπινης σκέψης αναφορές σε έννοιες σχετικές με τις ‘μεταγνωσιακές’ διαδικασίες.

Για παράδειγμα οι φιλόσοφοι ανέκαθεν έδειχναν μεγάλο ενδιαφέρον για τις μεταγνωσιακές διαδικασίες. Ο Άγ. Αυγουστίνος (354-430 μ.Χ.) στο σύγγραμμά του : *Περί της Αγίας Τριάδος (De Trinitate, 15 βιβλ., 399-416)*, αναρωτιέται: «ποιο μπορεί να είναι το νόημα της εντολής: γνώρισε τον εαυτό σου;». Το γνωμικό ‘Γνώθι σαυτόν’ που ήταν γραμμένο στον πρόναο του μεγάλου ναού των Δελφών (Παυσανίας, *Ελλάδος Περιήγησις, βιβλ.10, ‘Φωκικά’, κεφ.24.1,1-13*), σημαίνει κατά τον Cox (2005), ότι «θα έπρεπε να στοχαστεί ο νους για τον εαυτό του». Το ίδιο οι μαθηματικοί και οι φιλόσοφοι αντελήφθησαν τουλάχιστον από τον καιρό του Σωκράτη τα προβλήματα που συνδέονται με τις αυτό-αναφερόμενες προτάσεις όπως στο «παράδοξο του ψεύτη» με τον Κρητικό φιλόσοφο Επιμενίδα (Αναπολιτάνος, 1985; Perlis, 2006).

Πρόσφατα ο Hofstadter (1989) επιχειρηματολόγησε ότι η έννοια του στοχασμού και των «βρόχων» (Loops) που δημιουργούνται όταν ένα αντικείμενο στρέφει προς τον εαυτό του είναι κάτι το οποίο μελετάται εντός και εκτός του επιστημονικού χώρου. Ομοίως «βρόχοι» (Loops) παρατηρούνται στα μαθηματικά με τις αποδείξεις του Gödel, στην τέχνη με τη ζωγραφική του Escher, αλλά και στη μουσική με τις συνθέσεις του Bach (Hofstadter et al., 1989).

4.2 Τεχνητή Νοημοσύνη (TN), «μετα-συλλογισμός» και ενδοσκόπηση

Επιστήμονες της T.N. έχουν εξετάσει την δυνατότητα να δημιουργηθούν συστήματα εφοδιασμένα με ιδιότητες ανάλογες με τις «μεταγνωσιακές» (metacognitive). Οι Marvin Minsky και John McCarthy μελέτησαν τα θέματα της μηχανικής «αυτό-γνώσης» (self-knowledge) και των ενδοσκοπικών δυνατοτήτων των μηχανών ήδη από τα μέσα της δεκαετίας του ’50. Ο Minsky (1968,1985), ισχυρίστηκε ότι μια μηχανή για να μπορεί να δίνει επαρκείς απαντήσεις σε ερωτήσεις που αφορούν τον κόσμο και

τον εαυτό της, θα πρέπει να έχει ένα εκτελεστικό μοντέλο και για τον ίδιο της τον εαυτό. Η ευφυής μηχανή κατά τον Minsky πρέπει να έχει ένα υπολογιστικό μοντέλο του εξωτερικού κόσμου, του οποίου μια εκτελεστική προσομοίωση να μπορεί να απαντά σε ερωτήσεις για ενέργειες που συμβαίνουν στον πραγματικό κόσμο και ένα αναδρομικό μοντέλο του εαυτού της για να απαντά σε ερωτήσεις που αφορούν το ίδιο το μοντέλο, τις δικές του διαθέσεις και τη δική του συμπεριφορά μέσα στον κόσμο.

Ο McCarthy (1968), υποστήριξε ότι μια μηχανή για να συμπεριφέρεται σε ικανοποιητικό βαθμό με ευφυή τρόπο, πρέπει να αναπαριστά τη γνώση της με τρόπο δηλωτικό και ότι οι μηχανές μπορούν να εξετάζουν τις πεποιθήσεις τους, όταν αυτές παρουσιάζονται ρητώς. Επίσης έδειξε ενδιαφέρον (McCarthy, 1995), για τη σχέση μεταξύ των «μεταγνωσιακών λειτουργιών» μιας μηχανής και των «μεταγνωσιακών λειτουργιών» του ανθρώπου (the relation between machine and human metacognition).

Έχουν γίνει διάφορα συνέδρια με αντικείμενο την έρευνα για το σκοπό αυτό στις δεκαετίες του '80 και '90 όπως το Workshop on Meta-level Architectures and Reflection που οργανώθηκε στο Alghero της Ιταλίας τον Οκτώβριο του 1986 (Maes&Nardi, 1988) αλλά και το Second International Conference on Meta-level Architecture and Reflection που οργανώθηκε στο Saint-Malo της Γαλλίας τον Ιούλιο του 1999 (Cointe,1999).

Το ενδιαφέρον της μελέτης των υπολογιστικών μοντέλων του συλλογισμού, ανάλογου με αυτόν που επιδεικνύεται από τους ανθρώπους, καταδεικνύεται και από τις εργασίες των ερευνητών στα διάφορα συνέδρια. Πρόσφατα το 2005 έγινε το AAAI Spring Symposium on Metacognition in Computation (Anderson et al., 2005), νωρίτερα έγιναν τα: AISB 2000 symposium on How to Design a Functioning Mind (Davis, 2005), St. Thomas Common Sense Symposium: Designing Architectures for Human-Level Intelligence (Minsky et al., 2004), DARPA Workshop on Self-Aware Computer Systems (McCarthy et al., 2004). Σύμφωνα με τον Wellman (1983), η «μεταγνωσιακή λειτουργία» (metacognition) διαφέρει από την «νοητική λειτουργία» (cognition) στο ότι ο ίδιος ο εαυτός είναι το αντικείμενο αναφοράς της διαδικασίας ή της γνώσης. Στις περισσότερες ερμηνείες (Hayes-Roth, Waterman & Lenat, 1983; Kuokka, 1990) η έκφραση meta-X εξηγείται ως «X που αφορά το X» και συχνά η metaknowledge και η metamemory, συμπεριλαμβάνονται στη μελέτη της «μεταγνωσιακής λειτουργίας» διότι παίζουν σημαντικό ρόλο στον αυτο-έλεγχο και στις επιπλέον «μεταγνωσιακές λειτουργίες».

Στην επιστήμη της πληροφορικής μια κλασική υπολογιστική λειτουργία εμπνευσμένη από «μεταγνωσιακή λειτουργία» αφορά το πρόβλημα της επιλογής αλγορίθμου (Rice, 1976). Σε αυτήν την εργασία εξετάζεται η εισαγωγή των δεδομένων σε ένα πρόγραμμα ως ένα ιδιαίτερο πρόβλημα, καθώς επίσης και το σύνολο των αλγορίθμων που έχουν τη δυνατότητα να δώσουν λύσεις. Ο σκοπός είναι να επιλεγεί ο ταχύτερος αλγόριθμος, όμως οι αποφάσεις για την επιλογή του πρέπει να βασίζονται όχι μόνο στα χαρακτηριστικά του προβλήματος που τίθεται, αλλά και στη γνώση της απόδοσης του αλγορίθμου. Οι Lagoudakis et al., (2001, 2002) περιγράφουν αυτήν την εργασία με την βοήθεια των προβλημάτων της απλής ταξινόμησης (quick sort, insertion sort και merge sort). Η ταξινόμηση γίνεται με βάση κάποια χαρακτηριστικά των προγραμμάτων και η απόφαση επιλογής του ταχύτερου αλγορίθμου διατυπώνεται ως «Διαδικασία Απόφασης» (Markov Decision Process - MDP).

Ο Cox (ό.π.), ισχυρίζεται ότι στη βιβλιογραφία των «μεταγνωσιακών λειτουργιών», η διάκριση μεταξύ του «συλλογισμού» (reasoning) και του «μετασυλλογισμού» (metareasoning) δεν είναι πάντα σαφής. Και αυτό φαίνεται στη μελέτη κατανόησης μιας ιστορίας, η οποία αποτελείται από χαρακτήρες και συμβάντα που αλλάζουν την κατάσταση αυτών των χαρακτήρων ή τον χώρο στον οποίο συμβαίνει η ιστορία. Διότι για την ανάλυση μιας ιστορίας απαιτείται από ένα σύστημα ένας συλλογισμός για τα συμβάντα, τις καταστάσεις, αλλά και για το λόγο που οι χαρακτήρες επέλεξαν να ενεργήσουν έτσι όπως ενήργησαν.

Τέτοια συστήματα Επεξεργασίας της Φυσικής Γλώσσας (ΕΦΓ) ή σύμφωνα με τη διεθνή βιβλιογραφία Natural Language Processing (NLP) κατά τον Cox (ό.π.), δημιουργούν μια αναπαράσταση της ιστορίας. Επίσης εκτελούν ορισμένους υπολογισμούς, οι οποίοι μεταβάλλουν την αναπαράσταση μέχρι να επιτευχθεί ικανοποιητική ερμηνεία της ιστορίας. Και η επεξεργασία της Φυσικής Γλώσσας αφορά όπως ισχυρίζεται ο Cox (ό.π.), «νοητικές» καταστάσεις και «νοητικά» συμβάντα παρόμοια με τις καταστάσεις και τα συμβάντα του μικρόκοσμο της ιστορίας. Εάν δε αυτές οι καταστάσεις και τα συμβάντα αναπαρίστανται σε ένα τρίτο μικρόκοσμο, τότε μπορούν επίσης να υποστούν λογική επεξεργασία όπως και η ίδια η ιστορία. Έτσι το αποτέλεσμα της «ενδοσκόπησης» είναι ένας συλλογισμός που αφορά τη δραστηριότητα του συλλογισμού. Για τον λόγο αυτό είναι μια διαδικασία συλλογισμού δευτέρας τάξης με την ονομασία «μετασυλλογισμός» (metareasoning) ή γενικότερα «μεταγνωσιακή λειτουργία» (metacognition).

4.3 Εφαρμογές συστημάτων QA σε φυσική γλώσσα που επιδεικνύουν «ρητή αυτοεπίγνωση».

Στα προηγούμενα αναφέρθηκαν οι πρώτες εφαρμογές συστημάτων QA τα οποία έδιναν απαντήσεις σε συγκεκριμένες ερωτήσεις ‘ενώνοντας’ προτάσεις ή τμήματα προτάσεων δίχως μεταγνωσιακές διαδικασίες. Στη συνέχεια αναπτύχθηκαν εφαρμογές συστημάτων QA που είχαν τη δυνατότητα μαζί με την αλληλεπίδραση με το χρήστη σε φυσική γλώσσα, να επιδεικνύουν και «ρητή αυτό-επίγνωση» (explicit self-awareness) (Morbinì, 2006, σελ.34). Κατά τον Schubert (2005), «ρητή αυτο-επίγνωση» σε μια τεχνητή «αυτόνομη διαδικασία» (agent), επιδεικνύεται όταν η «αυτο-γνώση» (self-knowledge) του συστήματος κωδικοποιείται σε μορφή που να μπορεί εύκολα να εξεταστεί, να ερμηνευτεί και να μπορεί να γίνει φανερή από την «αυτόνομη διαδικασία» μέσω της χρήσης μιας συνηθισμένης γλώσσας (δηλ. να μπορεί κάποιος να παρακολουθεί ποια γνώση χρησιμοποίησε και ποιες μεθόδους για να δώσει μια απάντηση). Επίσης να μπορεί να εφαρμοστεί αυτή η γνώση και σε άλλη παρόμοια συμπερασματική διαδικασία, όπως συμβαίνει με όλη την άλλη γνώση (δηλ. να μην είναι αποκλειστικά διαθέσιμη) (Schubert, 2005).

Τέτοια συστήματα είναι τα : SHRDLU (Winograd, 1972) (το οποίο ήδη αναφέρθηκε πιο πάνω), HOMER (Vere and Bickmore, 1990), CASSIE (Shapiro, 1989) και COMA με το σύστημα απάντησης ερωτήσεων EPILOG ως υποδομή (Schubert et al., 1993). Τα δύο πρώτα SHRDLU και HOMER, δεν αναπτύσσονται περαιτέρω ενώ τα CASSIE και COMA είναι τα πιο πρόσφατα και είναι ακόμη υπό ανάπτυξη (Morbinì, ό.π., σελ.34).

Επίσης όπως αναφέρει ο Morbinì, (ό.π., σελ.34) δεν αφορά την ερευνά του και δεν σχολιάζει τα διάφορα υπολογιστικά μοντέλα που στοχεύουν στην προσομοίωση της ανθρώπινης σκέψης/γνώσης όπως είναι τα : SOAR, IDA και ACT-R. Ομοίως το σύστημα, το οποίο περιγράφεται σε αυτήν εδώ την εργασία δεν ανήκει σε αυτήν την κατηγορία των προαναφερθέντων συστημάτων που προσπαθούν να προσομοιάσουν την ανθρώπινη νόηση. Εμπνέεται όμως από την ανθρώπινη συμπεριφορά και συνεισφέρει στην τεχνολογική πρόοδο των ηλεκτρονικών υπολογιστών.

Το σύστημα SHRDLU έχει την ικανότητα :

- να χειρίζεται διαλόγους μεταξύ χρήστη και προγράμματος/υπολογιστικού συστήματος που αφορούν ένα μικρόκοσμο, για τον οποίον του έχουν δοθεί πληροφορίες,

- να πραγματοποιεί επίλυση αναφοράς, δηλαδή να ταυτοποιεί αντωνυμίες με κάποια προαναφερθείσα οντότητα,
- να ξεχωρίζει αντικείμενα ως προς την θέση τους, το χρώμα τους και το σχήμα τους,
- να δίνει ονόματα σε αντικείμενα και σε σύνολα αντικειμένων που ορίζονται με γεωμετρικούς όρους,
- να εξηγεί τις ενέργειές του, για να έχει τη δυνατότητα να απαντά σε ερωτήσεις της μορφής όπως «How did you do it?» (το 'it' αναφέρεται στον συγκεκριμένο στόχο) ή «Why did you do that?» (το that αναφέρεται στη συγκεκριμένη ενέργεια),
- να θυμάται τις ενέργειές του ώστε να απαντά σε ερωτήσεις της μορφής όπως «When did you pick up?» (Morbinì, ό.π., σελ.36-37).

Χαρακτηριστικό γνώρισμα του συστήματος κατά τον Morbinì (ό.π., σελ.36-41).είναι η σύνδεση των τμημάτων του προγράμματος που αφορούν την σύνταξη, την σημασιολογία και το συμπέρασμα. Επίσης η ικανότητά του στον μερισμό (parsing) των λέξεων του κειμένου και στην σημασιολογική τους ανάλυση. Το σύστημα δεν είναι εύκολα επεκτάσιμο και ενώ υπήρξαν προσπάθειες για την επανεμφάνισή του σε εφαρμογές, καμία δεν υπήρξε απόλυτα επιτυχής. Αργότερα σε κείμενό του ο Winograd (1990), είχε αρνητική θέση ως προς τη δυνατότητα των μηχανών να 'καταλαβαίνουν' όπως οι άνθρωποι¹. Ως προς την ικανότητα του συστήματος να επιδεικνύει «αυτό-επίγνωση», το πιο εντυπωσιακό χαρακτηριστικό του γνώρισμα είναι η δυνατότητά του να δίνει εξηγήσεις για τις ενέργειές του (Morbinì, ό.π., σελ.41).

Το σύστημα HOMER ήταν η προσομοίωση ενός υποβρυχίου ικανού :

- να «επικοινωνεί» με ανθρώπους σε φυσική γλώσσα και να εκτελεί εντολές,
- να σχεδιάζει κινήσεις,
- να σχεδιάζει και να επανασχεδιάζει τις ενέργειές του, κάτι πολύ σημαντικό γιατί ενεργούσε σε ένα δυναμικό περιβάλλον,
- να αιτιολογεί τις ενέργειές του,

¹ Winograd, (1990) : «Computers can never be creative or insightful or possess real judgement».

- να ενεργεί και να είναι σε θέση να διακρίνει αντικείμενα σε ένα εικονικό περιβάλλον και
- να εκφράζει τις δικές του εμπειρίες.

Ως χαρακτηριστικά του γνωρίσματα ο Morbini (ό.π.), αναφέρει την ικανότητά του να σχεδιάζει και να επανασχεδιάζει τις ενέργειές του, τη μνήμη επεισοδίων του αλλά και τις αντιληπτικές του διαδικασίες. Ενώ ως κύρια προβλήματα του είχε: το λεξικό, τη σημασιολογία, την πρόσβαση στη μνήμη και τον διαχωρισμό της μνήμης σχεδιασμού και της μνήμης επεισοδίων, κάτι που καθιστούσε αδύνατη την εξήγηση των ενεργειών του (Morbini, ό.π., σελ.42, 57).

Το σύστημα CASSIE (the Cognitive Agent of the SnePS System, an Intelligent Entity) σχεδιάστηκε από τον Stuart C. Shapiro (1989) και στοχεύει μακροπρόθεσμα στην ικανότητα χρήσης της φυσικής γλώσσας. Οι κύριες ενότητες του είναι

- το GATN (Generalized Augmented Transition Network) που μεταφράζει τα αγγλικά σε σημασιολογικό δίκτυο ώστε να χρησιμοποιηθεί από τον συμπερασμό και από το 'τμήμα αναπαράστασης της γνώσης' (knowledge representation module),
- το SnePS/SNIP (Semantic Network Processing System/SnePS Inference Package) που είναι μια 'μηχανή αναπαράστασης γνώσης και συμπερασμού' (knowledge representation and inference engine).

Κατά τον Morbini (2006), το CASSIE δεν έχει κάποιο ιδιαίτερο χαρακτηριστικό γνώρισμα που να ξεχωρίζει από τα άλλα συστήματα που ήδη παρουσιάστηκαν (Morbini, ό.π., σελ.57-61).

Το σύστημα COMA (COncscious MAchine) ξεκίνησε από τον Schubert το 1999 με στόχο τη δημιουργία μίας «αυτόνομη διαδικασία» (ενός agent) με δυνατότητα να έχει όλες τις απαιτούμενες πληροφορίες για τη δομή της, να έχει τη δυνατότητα πρόσβασης στο σύνολο των κανόνων με τους οποίους έχει εφοδιαστεί, και για κάθε έργο ή στόχο να έχει την δυνατότητα της λήψης απόφασης εκτέλεσής του χρησιμοποιώντας σαν υποδομή το σύστημα απάντησης ερωτήσεων EPILOG (Schubert et al., 1993). Η «αυτό-επίγνωση» (self-awareness) του agent όπως αναφέρεται από τον Schubert (2005), πρέπει να είναι εμφανής στην αλληλεπίδραση με τον χρήστη, δηλαδή θα πρέπει ένας παρατηρητής να είναι ικανός να καταλαβαίνει κάθε βήμα του συλλογισμού του agent. Η ιεραρχία που χρησιμοποιείται στο σύστημα EPILOG για τον ορισμό των μορφών και των σχέσεων των πραγμάτων που έχει πληροφορίες, αποτελεί πρόβλημα λόγω της αρνητικής επίδρασής της στις απαντήσεις

ερωτήσεων διότι δεν είναι πλήρως ενσωματωμένη στη συμπερασματική του διαδικασία (Morbinì, ό.π., σελ.62-63). Δεν είναι σε θέση να απαντήσει στη γενική ερώτηση «What do you know about flowers?» με «Flowers are plants». Σε αυτό το πρόβλημα ο Schubert (2005), προτείνει την κωδικοποίηση της γνώσης των συμπερασμάτων που προέρχονται από την εκτέλεση ορισμένων λειτουργιών του συστήματος. Άλλο πρόβλημα του συστήματος που επισημαίνει ο Morbinì (ό.π.), αφορά την οργάνωση της μνήμης του, ιδίως όταν δεν διευκρινίζονται οι μορφές των μεταβλητών στη διατύπωση των εισαγομένων οντολογικών γνώσεων. Η μορφή των μεταβλητών είναι σημαντική για την πρόσβαση στη μνήμη διότι χρησιμοποιείται στο τρόπο καταγραφής της (Morbinì, ό.π., σελ.68).

Ο Morbinì (ό.π.), αναφέρει ότι ενώ το COMA στερείται της ικανότητας να δέχεται κείμενα γραμμένα στην Αγγλική γλώσσα, τα άλλα συστήματα SHRDLU, HOMER και CASSIE, έχουν αρκετά καλούς «μεριστές» (parsers) για ένα περιορισμένο υποσύνολό της και ένα σύστημα κανόνων που να συνδέει την συντακτική δομή μιας πρότασης με τις οντότητες, τις ιδιότητές τους και τις μεταξύ τους σχέσεις, που έχουν καταγραφεί σε μορφή επεξεργάσιμη από το πρόγραμμα του υπολογιστή (Morbinì, ό.π., σελ.70).

Μια άλλη μελλοντική τάση αφορά την ανάπτυξη συστημάτων, τα οποία θα μπορούν να διεξάγουν έναν επεξηγηματικό διάλογο με τον χρήστη τους απαντώντας σε διάφορες ερωτήσεις τύπου «Why» χρησιμοποιώντας κάποια προσομοίωση συμπεριφοράς των συστημάτων. Μια ερώτηση τύπου «Why» αναζητά το λόγο για τον οποίον το σύστημα εμφανίζει μια συγκεκριμένη συμπεριφορά.

Οι έρευνες πάνω στα μοντέλα ανακάλυψης συστημάτων QA προετοιμάζουν το έδαφος για σημαντικές εξελίξεις και έτσι δικαιολογούνται οι προσπάθειες ανάπτυξης εργαλείων και πόρων με στόχο την επίλυση των προβλημάτων που υπάρχουν στα μοντέλα ανακάλυψης με βάση τα μεγάλα και σύνθετα κείμενα. Τα κείμενα αυτά μπορεί να είναι και αναφορές πειραματικών δεδομένων τα οποία θα χρησιμοποιηθούν στην υποστήριξη της ανακάλυψης και της προσαρμογής των μοντέλων στα υπολογιστικά συστήματα (Kontos, J., and Malagardi, I., 2006).

Οι δραστηριότητες της Ι.Ε. και της Απάντησης Ερωτήσεων έχουν ως στόχο να προσδιορίσουν στα κείμενα τα τμήματα των πληροφοριών, τα οποία είναι σχετικά με μια δοσμένη ερώτηση. Ωστόσο, οι δύο δραστηριότητες διαφέρουν ως προς τον τύπο των πληροφοριών που αναζητούνται. Συγκεκριμένα ενώ στην Ι.Ε. οι πληροφορίες που αναζητούνται έχουν μεγαλύτερο εύρος και το είδος των πληροφοριών είναι γνωστό

εκ των προτέρων, στο σύστημα Απάντησης Ερωτήσεων πρέπει να απαντήσει σε ένα ευρύ φάσμα απρόβλεπτων ερωτήσεων από τους χρήστες.

Στο επόμενο κεφάλαιο εισάγονται ορισμένες βασικές έννοιες από τους επιστημονικούς τομείς της Ευκλείδειας Γεωμετρίας και της αλληλεπίδρασης πρωτεϊνών. Στις έννοιες αυτές βασιστήκαμε για την δημιουργία παραδειγμάτων που χρησιμοποιήθηκαν για την αξιολόγηση του συστήματος του συστήματος 'AMYNTA' της παρούσας διατριβής.

ΚΕΦΑΛΑΙΟ 5

Ιδέες και μέθοδοι των επιμέρους Επιστημονικών κλάδων στους οποίους στηρίχτηκε η έρευνα στην εφαρμογή του συστήματος «ΑΜΥΝΤΑΣ»

5.1 Η αποδεικτική διαδικασία στη Γεωμετρία από τους αρχαίους Έλληνες Μαθηματικούς των Κλασσικών Χρόνων

5.1.1 Γενικά

Όπως αναφέρει ο Heath στην εισαγωγή του βιβλίου του: “*A History OF GREEK MATHEMATICS*” (1921;2006, vol.1, p. v): «τα θεμέλια των Μαθηματικών και ένα μεγάλο τμήμα του περιεχομένου του είναι ελληνικά. Οι Έλληνες καθόρισαν τις πρώτες αρχές (*principles*), επινόησαν τις μεθόδους ‘*ab initio*’ (εξ’ αρχής – Λατ. *Initium* = αρχή), και καθόρισαν την ορολογία». Επίσης υποστηρίζει ο Heath (ό.π.) ότι εάν θα ήθελε κάποιος να καταλάβει πλήρως τον τρόπο σκέψης των Ελλήνων καλό θα ήταν να άρχιζε με την Γεωμετρία τους «if one would understand the Greek genius fully, it would be a good plan to begin with their geometry», διότι η ικανότητά τους στα Μαθηματικά ήταν απλά μια πτυχή της ικανότητά τους στη φιλοσοφία (Heath, ό.π., p. 3). Όταν ο Πλάτωνας αναφέρει στην *Πολιτεία* (Ζ’ 534e, μετ. Ι. Γρυπάρης), ότι: «η διαλεκτική είναι βαλμένη ψηλά στην κορυφή σαν επιστέγασμα πάνω απ’ τάλλα μαθήματα», ως μαθήματα εννοεί την Αριθμητική, την επιστήμη της μέτρησης (Γεωμετρία) και την Αστρονομία (Heath, ό.π., p. 10). Συγκεκριμένα ο Πλάτωνας αναφέρει στους *Νόμους* ότι: «Για τους ελεύθερους απομένουν τρία μαθήματα ακόμα: οι λογαριασμοί και η μελέτη των αριθμών, οι μετρήσεις ευθειών, επιφανειών κι όγκων και οι αμοιβαίες σχέσεις των άστρων κατά την πορεία της τροχιάς τους» (Πλάτων, *Νόμοι*, vii, 817E). Τη Γεωμετρία, το ένα από τα μαθήματα στα οποία αναφέρεται ο Πλάτων, ο Πυθαγόρας προσπάθησε να την καταστήσει θεωρητική επιστήμη (Heath, ό.π. p. 2) και υπήρξε ο πρώτος ο οποίος προσπάθησε να τη μελετήσει φτάνοντας την έρευνα μέχρι τις πρώτες της αρχές (Heath, ό.π. p. 166).

Ο Fritz (1955), αν και ισχυρίζεται ότι «οι Βαβυλώνιοι χίλια περίπου έτη προ των απαρχών των Ελληνικών Μαθηματικών ήσαν εις θέσιν να λύσουν κατά προσέγγισιν αρκετά πολύπλοκα προβλήματα», στη συνέχεια τονίζει ότι «ουδαμού ευρέθη ένδειξις περί του ότι οι Βαβυλώνιοι ακόμη και οι Αιγύπτιοι επεχείρησαν ποτέ να παραγάγουν τας μαθηματίας προτάσεις με αυστηράν λογικήν από πρώτας Αρχάς» (Fritz, 1955, pp. 13-14, στο SZABO, 1973 σελ.254).

Στη Γεωμετρία των αρχαίων Ελλήνων θα βρούμε σχεδόν όλες τις μεθόδους απόδειξης που χρησιμοποιούνται σήμερα, όπως «τη μέθοδο της εις άτοπον απαγωγής» ‘*Reductio ad absurdum*’, (Heath, 1956, vol.1, p. 136), «τη μέθοδο της ανάλυσης και σύνθεσης» ‘*Analysis and Synthesis*’, (Heath, ό.π. p. 137), καθώς και «τη μέθοδο της πλήρους επαγωγής ή του αναδρομικού συλλογισμού» (Σταμάτης, Ευκλείδου Γεωμετρία - Θεωρία Αριθμών, σελ.5). Η τελευταία μέθοδος αναφέρεται στον Αριστοτέλη (*Αναλυτικά ύστερα Α’*, 73b 32): «το καθόλου δε υπάρχει τότε όταν επί του τυχόντος και πρώτου δεικνύεται» (Μία δηλαδή μαθηματική πρότασις έχει καθόλου ισχύ, εάν είναι δυνατόν ν’ αποδειχθή, ότι ισχύει εις την πρώτην τυχούσαν περίπτωση, εις την οποίαν αυτή αναφέρεται) (Σταμάτης, ό.π. σελ.6).

Ο Αριστοτέλης αναφέρεται στις ‘*αρχικές έννοιες*’ (*Αναλυτικά ύστερα Α’*, 72a 15), στα ‘*αξιώματα*’ (ό.π., 72a 17), στους ‘*ορισμούς*’ (ό.π., 72a 21), και στην ‘*απόδειξη*’ «*απόδειξιν δε λέγω συλλογισμόν επιστημονικόν*» (ό.π., 71b 19), όπως επίσης «*εξ’ αναγκαίων άρα συλλογισμός έστιν η απόδειξις*» (ό.π., 73a 24).

Τα *Στοιχεία* του Ευκλείδη αποτελούν κλασσικό παράδειγμα αξιωματικής θεμελίωσης και στο έργο αυτό ο Ευκλείδης κατά τον Πρόκλο (*Εις Πρώτον Ευκλείδου Στοιχείων – Πρόλογος*, 68, 8) «*συγκέντρωσε τα στοιχειώδη και συστηματοποίησε πολλά θεωρήματα του Ευδόξου και τελειοποίησε πολλά του Θεαίτητου, δίνοντας αδιάμευστες αποδείξεις σε όσα αποδεικνύονταν ακόμα ασθενώς από τους προγενέστερους*». Επίσης ο Ευκλείδης στα *Στοιχεία*, «*έχει συμπεριλάβει όλους τους τρόπους των συλλογισμών, οι οποίοι άλλοτε αντλούν τη βεβαιότητά τους από τα αίτια και άλλοτε στηρίζονται σε τεκμήρια*» (Πρόκλος, ό.π., 69, 10).

Στα δεκατρία βιβλία των *Στοιχείων* του ο Ευκλείδης περιλαμβάνει Ορισμούς, Αιτήματα, Κοινές έννοιες ή Αξιώματα και Προτάσεις. Στην πληρέστερή της μορφή μια πρόταση που συναντάμε στη γεωμετρία περιλαμβάνει σύμφωνα με τον Πρόκλο (*Εις Πρώτον Ευκλείδου Στοιχείων – Ζητημάτων Α’*, 203, 4-5), έξι μέρη:

- Την *πρόταση* ή τη *διατύπωση* σε γενικούς όρους.
- Την *έκθεση*. Δηλαδή την ανάλυση των δεδομένων και την αναπαράσταση της πρότασης με συγκεκριμένο γεωμετρικό σχήμα, το οποίο έχει γράμματα του ελληνικού αλφαβήτου.
- Τον *διορισμό* ή *ορισμό* ή *προσδιορισμό* δηλαδή την επαναδιατύπωση με σκοπό να διευκρινιστεί και να καταστεί σαφές το ζητούμενο της πρότασης.
- Την *κατασκευή*,
- Την *απόδειξη* και

- Το συμπέρασμα.

5.1.2 Τα γεωμετρικά σχήματα στη διαδικασία της απόδειξης

Η έκθεση που είναι το δεύτερο από τα έξι μέρη της πρότασης που αναφέρθηκαν προηγουμένως και αφορά στην αναπαράσταση της πρότασης με συγκεκριμένο γεωμετρικό σχήμα, δείχνει κατά τον SZABO (*Αρχαία των Ελληνικών Μαθηματικών*, 1973, σ.258), ότι η χρήση των γεωμετρικών σχημάτων ήταν ένα αναπόσπαστο στοιχείο των κειμένων των αρχαίων Ελληνικών Μαθηματικών. Φανερώνει δε ότι σε μια προγενέστερη βαθμίδα της εξέλιξης των Μαθηματικών, θα πρέπει να έπαιξε σημαντικό ρόλο το προφανές της παραστατικότητας '*die anschauliche Evidenz*' (Fritz, 1955, p.94, στο SZABO, ό.π., σελ.258).

Επίσης ο SZABO (ό.π.) ισχυρίζεται ότι οι αρχαίοι Έλληνες μαθηματικοί τη διαδικασία της απόδειξης την εξέφραζαν με το ρήμα «δείκνυμι» (δείχνω), διότι απεδείκνυαν την ορθότητα ή το λανθασμένο μιας μαθηματικής πρότασης με το να την καθιστούν αρχικά ορατή με συγκεκριμένο τρόπο «*την περί ής ο λόγος κατάστασιν των πραγμάτων κατά τρόπον συγκεκριμένον*». Και ίσως το να καθιστούν δυνατό να δείξουν το συγκεκριμένο, να ήταν ο αρχικός τρόπος με τον οποίον έκαναν τη μαθηματική απόδειξη (SZABO, ό.π., σελ.258).

Αυτό ακριβώς δείχνει το παράδειγμα της διδασκαλίας των μαθηματικών της κλασσικής εποχής από το εδάφιο του πλατωνικού διαλόγου «Μένων» (Πλάτων, Μένων, 82B–85E). Στο διάλογο αυτό ο Σωκράτης ρωτά έναν δούλο πώς είναι δυνατόν να διπλασιαστεί το εμβαδόν τετραγώνου πλευράς μήκους δύο ποδών, σε τετράγωνο σχήμα. Συγχρόνως ο Σωκράτης σχεδιάζει ενώπιόν του, για την καλύτερη κατανόηση της ερώτησης, ένα τετράγωνο πλευράς μήκους δύο ποδών το οποίο πρέπει να διπλασιασθεί. Στην αρχή ο δούλος υποδεικνύει ότι το μήκος της πλευράς πρέπει να είναι διπλάσιο (τέσσερα πόδια) πράγμα που ο Σωκράτης με σχεδιασμό του νέου τετραγώνου δείχνει ότι δεν είναι το ζητούμενο τετράγωνο. Το ίδιο συμβαίνει και όταν υποδεικνύει ότι το μήκος πρέπει να είναι τρία πόδια. Τελικά ο Σωκράτης με ένα τέταρτο σχήμα αποδεικνύει ότι το τετράγωνο που σχηματίζεται με πλευρά τη διαγώνιο του αρχικού τετραγώνου έχει ακριβώς το διπλάσιο εμβαδόν. Όπως ισχυρίζεται ο SZABO (ό.π., σελ.259), με το τελευταίο του σχήμα ο Σωκράτης, δίνει την ορθή απάντηση της σχετικής μαθηματικής πρότασης και την ορατή απόδειξη της ορθότητάς της. Σ' αυτό το εδάφιο του Πλατωνικού διαλόγου περιγράφεται και ο τρόπος με τον οποίον ελέγχονταν και αποδεικνύονταν οι μαθηματικές προτάσεις, σε

μια από τις βαθμίδες της εξέλιξης των Ελληνικών Μαθηματικών, στην περίοδο της κλασσικής εποχής. *Οτιδήποτε αποδεικνυόταν με εμπειρικό και παραστατικό τρόπο και ήταν οπτικά καταληπτό, το υποστήριζαν στη συνέχεια με μία βήμα προς βήμα λογική σκέψη* (SZABO, ό.π., σελ.258).

Ο Winterstein (2004) στη διατριβή του, αναφερόμενος στα γεωμετρικά σχήματα θεωρεί ότι *«είναι αναπόσπαστα στοιχεία των μαθηματικών κειμένων και χρησιμοποιούνται πάντοτε στις αναπαραστάσεις και τους συλλογισμούς. Μπορούν επίσης να δείξουν γιατί μια μαθηματική πρόταση είναι αληθής, και να εξηγήσουν την διαδικασία μιας απόδειξης ή να αποδείξουν άμεσα ένα θεώρημα»*.

Την αξία των σχημάτων στη διαδικασία της απόδειξης των προτάσεων των Στοιχείων του Ευκλείδη την επισημαίνει και ο Netz (2004).

Για την πρώτη πρόταση του πρώτου βιβλίου των *Στοιχείων*: *«Επί της δοθείσης πεπερασμένης ευθείας να κατασκευασθή ισόπλευρον τρίγωνον»* (ΣΤΑΜΑΤΗΣ, τομ.1, σελ.43), ο Miller (2001, p. 1) στη διατριβή του, ισχυρίζεται ότι η είναι καθαρά σχεδιαστική και ότι ο Ευκλείδης κάνει χρήση κάποιων συμβάσεων, όσον αφορά τα σημεία, τις γραμμές και τους κύκλους στο ευκλείδειο επίπεδο. Για την απόδειξη της πρότασης ο Ευκλείδης υπέθεσε ότι οι κύκλοι με κέντρα τα σημεία A, B και ακτίνα AB τέμνονται στο σημείο Γ, και κατά τον Heath (ό.π., vol.1, p. 242), το υπέθεσε δίχως την προϋπόθεση κάποιου αξιώματος: *«it is commonplace that Euclid has no right to assume, without premising some postulate, that the two circles will meet in a point C»*.

5.1.3 Από το παραστατικό στο ιδεατό – σύμμετρα και ασύμμετρα μεγέθη

Κατά τον Reidemeister (1949), η περίοδος των Αρχαίων Ελληνικών Μαθηματικών της κλασσικής εποχής, χαρακτηρίζεται από τη μετάβαση του παραστατικού εις το ιδεατό *«εις τα ελληνικά (Πυθαγόρεια) Μαθηματικά επετελέσθη η μετάβασις από την παράστασιν εις το ιδεατόν»* (Reidemeister, 1949, p. 51, στο SZABO, ό.π., σελ.261). Επίσης ο SZABO (ό.π., σελ.300), ισχυρίζεται ότι η ενσυνείδητη απομάκρυνση από την παράσταση κατά τη διαδικασία της απόδειξης, ήταν μια εξέλιξη που προετοιμάστηκε υπό την επίδραση της ελεατικής φιλοσοφίας, με την απόρριψη της πρακτικής εμπειρίας και την αμφισβήτηση οποιασδήποτε γνώσης που αποκτάται με τις αισθήσεις. Αυτή την άποψη συμμερίστηκε αργότερα και ο Πλάτων εξηγώντας ότι η Αριθμητική *«δεν ανέχεται με κανέναν τρόπο να βασίζονται οι αριθμοί της σε ορατά και αισθητά αντικείμενα»* (Πλάτων, Πολιτεία 525d), *«διότι οι αριθμοί μόνον με τη*

διάνοια μπορούν να γίνουν αντιληπτοί, και δεν είναι δυνατόν να τους χρησιμοποιήσει κανείς με κανένα άλλο τρόπο» (Πλάτων, Πολιτεία 526a).

Σύμφωνα με τον SZABO (ό.π., σελ.270) συγχρόνως με την αντιεμπειρική και αντιπαραστατική τάση, εμφανίστηκε μια νέα τεχνική απόδειξης στα Μαθηματικά, η οποία δεν επικαλείται πλέον το προδήλως ορατό και επαληθεύει ένα μαθηματικό ισχυρισμό, με το να αποδεικνύει το αδύνατο του αντιθέτου ισχυρισμού. Ισχυρίζεται ο SZABO (ό.π., σελ.270-271), ότι η μεταβολή αυτή συντελέστηκε με την ανακάλυψη της γραμμικής ασυμμετρίας, η οποία ήταν ένα σημαντικό επίτευγμα των Αρχαίων των Ελληνικών Μαθηματικών. Στο βιβλίο X των *Στοιχείων* ο Ευκλείδης στον πρώτο ορισμό αναφέρει ότι «*σύμμετρα μεγέθη λέγονται τα μετρούμενα δια του αυτού μέτρου, ασύμμετρα δε εκείνα δια τα οποία δεν υπάρχει κοινόν μέτρον*» (Σταμάτης, 1975, *Περί Ασυμμέτρων*, том.3, σελ.20). Ένα σχόλιο στο δέκατο βιβλίο των *Στοιχείων* του Ευκλείδη που αποδίδεται στον Πάππο τον Αλεξανδρέα (4^{ος} μ.Χ. αι.) αναφέρει ότι η ασυμμετρία μελετήθηκε για πρώτη φορά από τους Πυθαγορείους (Heath, ό.π., p. 154), «*ήλθον δε την αρχήν επί την της συμμετρίας ζήτησιν οι Πυθαγόρειοι πρώτοι αυτήν εξευρόντες εκ της των αριθμών κατανοήσεως*» (Χριστιανίδης, 2003, σ.75). Η δημιουργία όμως και η ανάπτυξης της θεωρίας, όπως αυτή εκτίθεται στο X βιβλίο των *Στοιχείων*, πρέπει να θεωρηθεί ότι είναι έργο των μαθηματικών της Ακαδημίας του Πλάτωνος και συγκεκριμένα του Θεαίτητου του Αθηναίου, περίπου το 417-369 π.Χ. και του Ευδόξου του Κνιδίου, περίπου το 408-355 π.Χ. (Σταμάτης, ό.π., σελ.7). Αναφορά στον τρόπο απόδειξης της ασυμμετρίας περιέχεται στα «Αναλυτικά πρότερα» του Αριστοτέλη (Αριστοτέλη, *Αναλυτικά πρότερα Α*, 23,41a,26), σύμφωνα με την οποία «*οίον ότι ασύμμετρος η διάμετρος δια το γίνεσθαι τα περιττά ίσα τοις αρτίοις συμμέτρου τεθείσης*» (είναι ασύμμετρος η διαγώνιος για τον λόγο ότι οι περιττοί αριθμοί θα γίνονταν ίσοι με τους αρτίους, αν εθεωρείτο σύμμετρος).

5.1.4 Η ασυμμετρία και ο αλγόριθμος της ανθυφαίρεσης

Στο διάλογο του Πλάτωνα, *Θεαίτητος* (147d7-148b7), ο γέροντας γεωμέτρης Θεόδωρος ο Κυρηναίος, διδάσκαλος του Πλάτωνα στα μαθηματικά, ομιλεί περί ασυμμετρίας στους νεαρούς Θεαίτητο και Σωκράτη, αποδεικνύοντας ότι οι πλευρές των τετραγώνων με εμβαδόν 3,5,6,7,8,10,11,12,13,14,15,17 και 18 τετραγωνικών ποδών δεν είναι σύμμετρες προς την «ποδιαία» (το μήκος του ενός ποδός). Στο συγκεκριμένο κείμενο του Πλάτωνα (ό.π.) δεν υπάρχει η απόδειξη της ασυμμετρίας. Προσπάθεια ανακατασκευής της απόδειξης του Θεοδώρου έχει διατυπωθεί από τον

Zeuthen (1910) βασιζόμενη στη μέθοδο του αλγορίθμου της ανθυφαίρεσης (Χριστιανίδης, ό.π., σ.123). Ο αλγόριθμος της ανθυφαίρεσης ήταν γνωστός στη Γεωμετρία των Αρχαίων Ελλήνων από τη μέθοδο της εύρεσης του μεγίστου κοινού διαιρέτη δύο αριθμών (Σταμάτης, 1975, τ.3, σελ.6). Άρα ήταν δυνατή η απόδειξη του Θεοδώρου την ιστορική εκείνη περίοδο με τη μέθοδο του αλγορίθμου της ανθυφαίρεσης.

Η μεν αριθμητική εφαρμογή της μεθόδου της ανθυφαίρεσης υπάρχει στη Πρόταση 1, Στοιχ. VII: *«Εάν δοθώσι δύο άνισοι αριθμοί, ανθυφαιρήται δε πάντοτε ο μικρότερος από του μεγαλύτερου, εάν ο λειπόμενος ουδέποτε καταμετρή τον πρό εαυτού, μέχρις ότου υπολειφθή μονάς, οι εξ αρχής αριθμοί θα είναι πρώτοι προς αλλήλους»* (Σταμάτης, 1975, τ.2, σελ.135). Η δε γεωμετρική εφαρμογή της ανθυφαίρεσης υπάρχει στην Πρόταση 2, Στοιχ. X: *«Εάν δοθώσι δύο άνισα μεγέθη και ανθυφαιρείται πάντοτε το μικρότερον από του μεγαλύτερου, το εκάστοτε δε υπόλοιπον ουδέποτε καταμετρή το προ εαυτού, τα μεγέθη θα είναι ασύμμετρα»* (Σταμάτης, 1975, τ.3, σ.23).

5.1.5 Το άπειρο της ανθυφαίρεσης και το κριτήριο ασυμμετρίας δύο μεγεθών – γραμμική ασυμμετρία

Ο SZABO (ό.π.), ισχυρίζεται ότι στην περίπτωση της εφαρμογής της αριθμητικής ανθυφαίρεσης από τον Ευκλείδη, αυτή γίνεται μόνον νοερώς και κατά τρόπο εντελώς αφηρημένο (SZABO, ό.π., σελ.278). Επίσης αναφέρει (SZABO, ό.π., σελ.279), ότι στην περίπτωση της εφαρμογής της γεωμετρικής ανθυφαίρεσης από τον Ευκλείδη σε δύο ασύμμετρα μεγέθη, η διαδικασία προχωρεί μέχρι του απείρου και αυτό αποτελεί κριτήριο ασυμμετρίας των δύο μεγεθών. Υποστηρίζει δε (SZABO, ό.π., σελ.285), ότι το άπειρο της ανθυφαίρεσης είναι ένα κριτήριο που εφαρμόζεται μόνο θεωρητικά και όχι πρακτικά στην περίπτωση ασυμμετρίας δυο μεγεθών.

Όμως η μεγάλη διαφορά μεταξύ της αριθμητικής και γεωμετρικής ανθυφαίρεσης έγκειται στο ότι στη μεν αριθμητική ανθυφαίρεση δύο πρώτων προς αλλήλους άνισων αριθμών, αυτή οδηγεί στον ελάχιστο *«οικοδομικό λίθο»* των αριθμών, την *«μονάδα»*, στη δε γεωμετρική ανθυφαίρεση δύο ασύμμετρων μεγεθών, αυτή δεν μπορεί να οδηγήσει σε κανένα τελικό αποτέλεσμα, διότι στη Γεωμετρία δεν υπάρχει ελάχιστο (SZABO, ό.π., σελ.279).

Κατά τον SZABO (ό.π.), το ότι η διαδικασία της ανθυφαίρεσης δύο ασύμμετρων μεγεθών δεν μπορεί ποτέ να πραγματοποιηθεί με πρακτικό και παραστατικό τρόπο παρά μόνο να κατανοηθεί νοερώς, είχε σαν συνέπεια να οδηγήσει τους Έλληνες

μαθηματικούς στη γνώση ενός μαθηματικού γεγονότος: της γραμμικής ασυμμετρίας (SZABO, ό.π., σελ.288). Κατά τον SZABO (ό.π.), η ύπαρξη του γεγονότος της γραμμικής ασυμμετρίας δεν μπορεί να αποδειχθεί με παραστατικό τρόπο, διότι η απόδειξή της δεν συνίσταται στη σχηματική αναπαράσταση, αλλά στο πώς θα την αντιληφθούμε (SZABO, ό.π., σελ.290).

5.1.6 Η έμμεσος αποδεικτική διαδικασία (η εις άτοπον απαγωγή)

Η απόδειξη της γραμμικής ασυμμετρίας κατά τον SZABO (ό.π.), προϋποθέτει τη δυνατότητα μιας νοεράς επεξεργασίας του μαθηματικού προβλήματος και μιας νέας αποδεικτικής διαδικασίας με καθαρώς θεωρητικά μέσα, η οποία δεν χρησιμοποιεί την Παραστατικότητα και τον Εμπειρισμό αλλά τα δεδομένα της σκέψεως (SZABO, ό.π., σελ.294-295). Τη νέα αποδεικτική διαδικασία που καλείται «έμμεσος αποδεικτική διαδικασία» (*reductio ad absurdum*), οι Έλληνες Μαθηματικοί τη χρησιμοποίησαν στους αριθμούς και στα γεωμετρικά σχήματα, τα οποία τα μελετούσαν με αφηρημένο εντελώς τρόπο και ως διανοητικά στοιχεία μόνο. Η νέα τεχνική εμφανίζεται στην απόδειξη της ασυμμετρίας που περιέχεται στα «Αναλυτικά πρότερα» (41a,25 και 50a,35) του Αριστοτέλη. Μία εφαρμογή της εμμέσου αποδεικτικής διαδικασίας εμφανίζεται και στο εδάφιο του Πλατωνικού διαλόγου *Παρμενίδης* (136a). Σε αυτόν ο Ελεάτης φιλόσοφος Παρμενίδης συστήνει στο νεαρό Σωκράτη τη διαδικασία που εφαρμόζει ο Ζήνων, κατά την οποία σε κάθε περίπτωση πρέπει να προβάλλεται όχι μόνο μία υπόθεση για την οποία να εξετάζεται ποιες συνέπειες μπορούν να προκύψουν, αλλά κατόπιν μετά από αυτήν, να εξετάζεται με τον ίδιο τρόπο και η αντίθετη υπόθεση και άποψη. Παράδειγμα αυτής της μαθηματικής αποδεικτικής διαδικασίας υπάρχει στο εδάφιο του Πλατωνικού διαλόγου *Θεαίτητος* (162e-164b). Σε αυτό ο Σωκράτης αφού πρώτα συμβουλεύει ότι δεν πρέπει κανείς να αρκείται εις το πιθανό αλλά να χρησιμοποιεί το αυστηρό των Μαθηματικών, στη συνέχεια ακολουθεί μια αυστηρή αποδεικτική διαδικασία, η οποία είναι μια διαδικασία έμμεσης απόδειξης. Ο SZABO (1973, σελ.336), ισχυρίζεται ότι η προτροπή της χρήσης «στο αυστηρό των Μαθηματικών», δηλαδή της μεθόδου της εμμέσου αποδείξεως, δείχνει ότι ο Πλάτωνας τη θεωρούσε ιδιαίτερο χαρακτηριστικό γνώρισμα της μαθηματικής σκέψης. Επίσης ο SZABO (ό.π.) επισημαίνει ότι στο βιβλίο VII των Στοιχείων του Ευκλείδη, οι 15 από τις 36 πρώτες Προτάσεις του οι οποίες θεωρούνται ότι είναι παλιές (Waerden, 1947/49), αποδεικνύονται εμμέσως (SZABO, ό.π. σελ.335).

5.1.7 Η αποδεικτική διαδικασία στην πρώτη Πρόταση των Στοιχείων του

Ευκλείδη

Σύμφωνα με τον Πρόκλο (*Εις Πρώτον Ευκλείδου Στοιχείων - Ζητημάτων Α'*, 200.21-201.2), γενικά η επιστήμη διαιρείται σε δύο μέρη: το ένα μέρος ασχολείται με τις πρωταρχικές θέσεις, και το άλλο με όσα αποδεικνύονται και κατασκευάζονται από αυτές. Για την περίπτωση δε της Γεωμετρίας το δεύτερο μέρος διαιρείται επίσης σε δύο μέρη: στην επεξεργασία των προβλημάτων και στην «των θεωρημάτων εύρεσιν» (Πρόκλος, ό.π. 201.3-5). Επίσης ο Πρόκλος (*Εις Πρώτον Ευκλείδου Στοιχείων – Πρόλογος*, 57.10-27), υποστηρίζει ότι η Γεωμετρία για τη μελέτη των μεγεθών και των σχημάτων χρησιμοποιεί τη σύνθεση και την ανάλυση και εφαρμόζει όλες τις διαλεκτικές μεθόδους: διαιρέσεις, ορισμούς, αποδείξεις και αναλύσεις. Τις διαιρέσεις για «τα εις ευρέσει των ειδών», τους ορισμούς για να περιγράψει την ουσία κάποιων πραγμάτων, τις αποδείξεις για τις μεταβάσεις από «των αρχών εις τα ζητούμενα» και τις αναλύσεις για τις επιστροφές «από των ζητουμένων επί τας αρχάς» (Πρόκλος, ό.π., 69.14-19). Στο κείμενο των *Στοιχείων*, σύμφωνα με τον Πρόκλο (ό.π., 69.10-12), ο Ευκλείδης έχει συμπεριλάβει όλους τους τρόπους συλλογισμών, είτε αυτοί αντλούν τη βεβαιότητά τους από τα αίτια «από των αιτίων λαμβάνοντας την πίστιν», είτε στηρίζονται σε τεκμήρια «από τεκμηρίων ωρμημένους» και χρησιμοποιεί όλες τις διαλεκτικές μεθόδους.

Μια εφαρμογή του συστήματός που περιγράφεται στη διατριβή σε ένα επόμενο κεφάλαιο και σχετίζεται με το επιστημονικό πεδίο της Γεωμετρίας αφορά την πρώτη Πρόταση του πρώτου βιβλίου των *Στοιχείων* του Ευκλείδη. Ο Πρόκλος (*Εις Πρώτον Ευκλείδου Στοιχείων - Ζητημάτων Α'*, 208.3) κατατάσσει τη συγκεκριμένη Πρόταση στο τμήμα της Γεωμετρίας που ασχολείται με την επεξεργασία των προβλημάτων «πρόβλημα εστι δήλον». Υποστηρίζει δε ότι τα προβλήματα: μας προτρέπουν να κατασκευάσουμε ένα σχήμα, να το τοποθετήσουμε σε μια θέση, να το εφαρμόσουμε σε κάποιο άλλο, να το περιγράψουμε, να το προσαρμόσουμε, να το φέρουμε σε επαφή με κάποιο άλλο (Πρόκλος, ό.π. 201.10-15).

Επίσης ο Πρόκλος (ό.π., σελ.21), ισχυρίζεται ότι κάθε πρόβλημα και θεώρημα αποτελείται από τα εξής μέρη: την πρόταση, την έκθεση, τον προσδιορισμό, την κατασκευή, την απόδειξη και το συμπέρασμα, με πιο αναγκαία από αυτά την πρόταση, την απόδειξη και το συμπέρασμα. Συγκεκριμένα για την απόδειξη ο Πρόκλος (ό.π.) αναφέρει ότι: «η απόδειξη εξάγει επιστημονικά το ζητούμενο από τις προτάσεις που έχουν γίνει ήδη αποδεκτές» και ότι η απόδειξη άλλοτε «αποδεικνύει το

ζητούμενο με τη μεσολάβηση των ορισμών – γιατί αυτή είναι η τέλεια μορφή της απόδειξης – ενώ άλλοτε προσπαθεί να αποδείξει κάτι μέσω τεκμηρίων» (Πρόκλος, ό.π., σελ.27).

Έτσι και το συγκεκριμένο κείμενο της απόδειξης της πρώτης πρότασης των *Στοιχείων* αποδεικνύει την πρόταση (‘το ζητούμενο’ κατά τον Πρόκλο σύμφωνα με τα ανωτέρω), με τη μεσολάβηση του ορισμού του κύκλου (Πρόκλος, ό.π., σελ.35), στοιχείο το οποίο θα χρησιμοποιήσει το σύστημα για την παραγωγή της απάντησης όπως αυτή περιγράφεται στο 1^ο παράδειγμα του αντίστοιχου κεφαλαίου.

5.2 Το φαινόμενο της διαδικασίας βρόχου ή ανατροφοδότησης των πρωτεϊνών p53 και mdm2 στο πεδίο της Βιοϊατρικής

Κατά τους ερευνητές Bar-Or et al. (2000) η πρωτεΐνη p53 που εμπλέκεται στην καρκινογένεση, διαδραματίζει ένα βασικό ρόλο στην παρεμπόδιση της ανάπτυξης του καρκίνου και στην αδρανοποίησή του σε πολλές κακοήθειες του ανθρώπινου σώματος. Επίσης ισχυρίζονται (Bar-Or et al., ό.π.), ότι όταν η p53 είναι ενεργοποιημένη, κάποιες από τις κυτταρικές επιπτώσεις της μπορεί να είναι αμετάκλητες (διακοπή του κύκλου ανάπτυξης των κυττάρων «cell-cycle arrest», ή τον θάνατο των κυττάρων καθώς συμμετέχει στις διαδικασίες αποκατάστασης του DNA, «repair process») και ως εκ τούτου αποτελεί σημαντικό στοιχείο η κάτω από αυστηρό έλεγχο διατήρηση της λειτουργίας της p53 στα κανονικά κύτταρα. Υποστηρίζουν (Bar-Or et al., ό.π.), ότι ένας βασικός φορέας για τη ρύθμιση της p53 είναι η πρωτεΐνη Mdm2 και θεωρούν ότι μπορεί η απουσία της Mdm2 να απορυθμίζει την p53 σε τέτοιο βαθμό, ώστε μια υπέρμετρη δραστηριότητά της να γίνεται αιτία θανάτου των εμβρυακών κυττάρων «embryonic death». Ωστόσο θεωρούν (Bar-Or et al., ό.π.), ότι μια έντονη δραστηριότητα της Mdm2 μπορεί να οδηγήσει σε ανασχεση την p53 και με αυτόν τον τρόπο να υποβοηθήσει τον καρκίνο χωρίς την ανάγκη να αλλάξει το ίδιο το γονίδιο της p53. Η ανωτέρω παραγωγή των «ταλαντώσεων» (oscillation) των πρωτεϊνών p53 και Mdm2 μέσω του φαινομένου της διαδικασίας του βρόχου ή άλλως της ανατροφοδότησης «feedback loop», έχει εξηγηθεί λεπτομερώς στο άρθρο των ερευνητών Bar-Or, Maya, Segel, Alon, Levine, Oren (2000) «Generation of oscillations by the p53-Mdm2 feedback loop: A theoretical and experimental study».

Όλα όσα περιγράφηκαν στα προηγούμενα κεφάλαια αποτέλεσαν πηγή έμπνευσης για την παρούσα διατριβή, στόχος της οποίας ήταν η δημιουργία ενός

ευπροσάρμοστου (flexible) Συστήματος Απάντησης Ερωτήσεων που ονομάστηκε ‘ΑΜΥΝΤΑΣ’ και το οποίο δεν περιορίζεται σε απαντήσεις ερωτήσεων ενός μόνο επιστημονικού πεδίου. Η εφαρμογή του πραγματοποιείται σε απαντήσεις ερωτήσεων που αφορούν κείμενα από εντελώς διαφορετικά πεδία, όπως αυτά της Γεωμετρίας και της Βιοϊατρικής και που προκύπτουν από συμπερασματικές διαδικασίες. Επιπλέον το σύστημα, αιτιολογεί την επιλογή της εκάστοτε απάντησης παράγοντας επεξηγήσεις και επιδεικνύει το χαρακτηριστικό της ‘αυτοεπίγνωσης’ το οποίο έχουμε ονομάσει ‘metagnostic’ ώστε να αποφεύγεται η χρήση ψυχολογικών όρων που θα μπορούσε να προκαλέσει σύγχυση.

Στο επόμενο κεφάλαιο, δίδονται αφενός μεν μια περιγραφή της δομής και της λειτουργίας του συστήματος ‘ΑΜΥΝΤΑΣ’ αφετέρου δε η αξιολόγησή του με εφαρμογή σε τρία παραδείγματα, εκ των οποίων τα δύο αφορούν τον χώρο της Ευκλείδειας Γεωμετρίας και το τρίτο αφορά τον χώρο της Βιοϊατρικής και συγκεκριμένα την αλληλεπίδραση των πρωτεϊνών p53 και mdm2.

ΚΕΦΑΛΑΙΟ 6

Γενική περιγραφή και παραδείγματα εφαρμογής του υλοποιηθέντος συστήματος

6.1 Γενική περιγραφή του υλοποιηθέντος συστήματος Απάντησης Ερωτήσεων «*AMYNTAS*» – Αυτόματο Μεταγνωστικό Υπολογιστικό Τροποποιήσιμο Απαντητικό Σύστημα

Το σύστημα αποτελείται από τα εξής υποσυστήματα:

1. Υποσύστημα επεξεργασίας της ερώτησης (σελ. 75)
2. Υποσύστημα ανάλυσης κειμένου (σελ. 79)
3. Υποσύστημα αυτόματης εξαγωγής γνώσης (σελ. 95)
4. Υποσύστημα απάντησης ερωτήσεων (σελ. 101)
5. Υποσύστημα σχεδίασης σχημάτων (σελ. 126)

Το σύστημα Απάντησης Ερωτήσεων που περιγράφεται στην παρούσα εργασία έχει τη δυνατότητα να απαντά σε ερωτήσεις από κείμενα και να δίνει εξηγήσεις για το περιεχόμενο των απαντήσεων. Η παραγωγή αυτού του είδους των εξηγήσεων στοχεύει να εφοδιάσει τον χρήστη με την τεκμηρίωση της απάντησης που δίνεται στις ερωτήσεις. Μια τέτοιας μορφής εξήγηση είναι ιδιαίτερος σημαντική στο πεδίο της Βιοϊατρικής διότι μια απάντηση δεν θα είναι εύκολα αποδεκτή από έναν χρήστη της πληροφορίας αυτού του πεδίου, εκτός εάν δίνονται τεκμηριωμένες εξηγήσεις. Η τρέχουσα βιβλιογραφία (Κεφάλαιο 4.3) που αφορά συστήματα απάντησης ερωτήσεων και συγκεκριμένα των Συστημάτων Απάντησης Ερωτήσεων (ΣΑΕ) από βιοϊατρικά κείμενα δείχνει την έλλειψη ικανότητας παραγωγής εξηγήσεων αυτής της μορφής.

Σύμφωνα και με την πολύ πρόσφατη επισκόπηση των S.J. Athenikos & H. Han (in press), το περιγραφόμενο σύστημα στην εργασία των J. Kontos, I. Malagardi & J. Peros είναι το μόνο υπάρχον μέχρι σήμερα σύστημα ΣΑΕ με συλλογισμό/συμπερασμό απευθείας (δηλαδή χωρίς προηγούμενη μετάφραση σε τυπική αναπαράσταση) από βιοϊατρικό κείμενο. Το σύστημα αυτό δεν χρησιμοποιεί ‘μεταεπίπεδο’ για την παραγωγή εξηγήσεων. Το 2007, για πρώτη φορά ανακοινώνεται εργασία των J. Kontos & I. Armaos με την εισαγωγή ‘μεταεπιπέδου’ σε ΣΑΕ απευθείας από κείμενα. Το ολοκληρωμένο αυτό σύστημα έχει πλέον τη δυνατότητα παραγωγής εξηγήσεων (Kontos, Armaos & Malagardi, 2009) και αποτελεί αντικείμενο μελέτης και συμβολή της παρούσας διατριβής.

Πληροφορίες που αφορούν την πρόοδο του συλλογισμού που εκτελείται περιέχονται στην εσωτερική καταγραφή των διαδοχικών καταστάσεων του συστήματος. Αυτή η καταγραφή βασίζεται στην στρατηγική που ακολουθείται στους γλωσσολογικούς κανόνες καθώς και στην εφαρμογή των κανόνων συμπερασμού που χρησιμοποιούνται για την απάντηση της ερώτησης.

Τα συμπεράσματα των γεγονότων συνάγονται άμεσα από το κείμενο χωρίς να χρησιμοποιούνται οποιεσδήποτε τυπικές αναπαραστάσεις αυτού, το οποίο αποτελεί βασική διαφορά από τα περιγραφέντα στο προηγούμενο κεφάλαιο συστήματα. Μια επιπλέον διαφορά είναι ότι έχει τη δυνατότητα να προσαρμόζεται σε διαφορετικά γνωστικά πεδία όπως τα πεδία της Γεωμετρίας και της Βιοϊατρικής. Μια αξιολόγηση προτάσεων που ελήφθησαν από τον διαδικτυακό τόπο Pubmed έδωσε επιτυχή αποτελέσματα. Το σύστημα έχει τη δυνατότητα να εφαρμοστεί είτε για Βιοϊατρική εκπαίδευση ή στην Απάντηση Ερωτήσεων από Βιοϊατρικά κείμενα που χρησιμοποιούνται από τους επιστήμονες της Βιοϊατρικής.

6.2 Περιγραφή παραδειγμάτων εφαρμογής του συστήματος

Η περιγραφή των παραδειγμάτων που θα ακολουθήσει αφορά το μέρος του συστήματος στο οποίο επιλέγεται η κατάλληλη στρατηγική και γίνεται η επίβλεψη της διαδικασίας για την επίτευξη του στόχου που είναι η παραγωγή της αιτιολογημένης απάντησης.

Η εκτέλεση των παραδειγμάτων που περιγράφονται παρακάτω προϋποθέτουν ότι έχουν προηγηθεί από το σύστημα μερικές διαδικασίες οι οποίες αφορούν:

- α) την αυτόματη επεξεργασία και ανάλυση σε φυσική γλώσσα του κειμένου που εισάγεται,
- β) την επιλογή της ερώτησης από τον χρήστη, με την αυτόματη ανάλυση και επεξεργασία της σε φυσική γλώσσα από το σύστημα, και τέλος
- γ) την εξαγωγή υπονοούμενης οντολογίας από τις προτάσεις του κειμένου.

➤ Η διαδικασία αυτόματης επεξεργασίας και ανάλυσης σε φυσική γλώσσα του κειμένου που εισάγεται, πραγματοποιείται μέσω του προγράμματος Ανάλυσης Κειμένου (ΠΑΡΑΡΤΗΜΑ 1, σελ. 136). Το πρόγραμμα περιγράφεται και τεκμηριώνεται αναλυτικά στο ΚΕΦΑΛΑΙΟ 7: Υποσύστημα ανάλυσης κειμένου (σελ. 79). Η περιγραφή της επεξεργασίας του κειμένου, δίχως αυτό να αναιρεί τη γενικότητα του προγράμματος, αφορά κείμενο της Γεωμετρίας από το 1^ο βιβλίο των

Στοιχείων του Ευκλείδη όπως αυτό υπάρχει στη δεύτερη έκδοση του βιβλίου του Sir Thomas Heath (1956) στην Αγγλική γλώσσα (σελ. 59). Η ίδια διαδικασία ακολουθείται και για οποιοδήποτε άλλο κείμενο όπως αυτό του παραδείγματος της Βιολογίας. Το κείμενο που πρόκειται να εισαχθεί στο σύστημα διαφοροποιείται ‘με το χέρι’. Οι προτάσεις του κειμένου παραμένουν αναλλοίωτες και οι μόνες διαφοροποιήσεις από το πρωτότυπο κείμενο είναι ότι στη μια γραμμή γίνεται η αρίθμηση των προτάσεων που θα εισαχθούν για την επεξεργασία τους από το σύστημα και στην επόμενη γραμμή γράφεται η ίδια η πρόταση του κειμένου, αντικαθιστώντας τα σημεία στίξεως με τη λέξη της ονομασίας τους (full stop, comma, semicolon) λόγω της Prolog.

Το αποτέλεσμα της επεξεργασίας που υποβλήθηκε το κείμενο της Γεωμετρίας και το οποίο θα είναι το κείμενο εισαγωγής στο σύστημα φαίνεται στον πίνακα που ακολουθεί:

114	.. now comma since the point a is the centre of the circle cdb comma ac is equal to ab fullstop . .
115	.. again comma since the point b is the centre of the circle cae comma bc is equal to ba fullstop . .
117	.. therefore each of the straight lines ca comma cb is equal to ab fullstop . .
118	.. and things which are equal to the same thing are also equal to one another semicolon . .
119	.. therefore ca is also equal to cb fullstop . .
1110	.. therefore the three straight lines ca comma ab comma bc are equal to one another fullstop . .
1111	.. therefore the triangle abc is equilateral semicolon . .

Πίνακας 1: Το κείμενο της Γεωμετρίας ως κείμενο εισαγωγής στο σύστημα

Στη συνέχεια μια διαδικασία αυτόματης επεξεργασίας και ανάλυσης αφαιρεί τις ‘αδιάφορες’ λέξεις (π.χ. also) και χωρίζει μεταξύ τους τις προτάσεις που έχουν ρήμα χωρίς να επεμβαίνει σε αυτές που δηλώνουν προαπαιτούμενη γνώση (π.χ. η περίοδος No 118 στον Πίνακα 1). Το αποτέλεσμα φαίνεται στον επόμενο πίνακα:

1141	.. the point a is the centre of the circle cdb . .
1142	.. ac is equal to ab . .
1151	.. the point b is the centre of the circle cae . .

```

1152
.. bc is equal to ba . .
1171
.. each of the straight lines ca comma cb is equal to ab . .
1181
.. and things which are equal to the same thing are also equal to one another . .
1191
.. ca is also equal to cb . .
11101
.. the three straight lines ca comma ab comma bc are equal to one another . .
11111
.. the triangle abc is equilateral . .

```

Πίνακας 2: Το κείμενο μετά την πρώτη επεξεργασία από το σύστημα

Η διαδικασία αυτόματης επεξεργασίας και ανάλυσης του κειμένου ακολούθως παράγει αρχεία με τις προτάσεις του κειμένου σε φυσική γλώσσα.

Σε ένα αρχείο περιέχονται ολόκληρες οι προτάσεις με το ρήμα τους χωρίς τα περιττά σημεία στίξης. Σε άλλο αρχείο περιέχονται οι ίδιες προτάσεις χωρισμένες όμως σε τρία τμήματα: στο αριστερό μέρος της πρότασης ως προς το ρήμα, στο ρήμα και στο δεξί μέρος της πρότασης ως προς το ρήμα. Στους πίνακες που ακολουθούν παρουσιάζονται τα περιεχόμενα των δύο παραγομένων αρχείων που περιγράφηκαν:

```

1141
the point a is the centre of the circle cdb
1142
ac is equal to ab
1151
the point b is the centre of the circle cae
1152
bc is equal to ba
1171
each of the straight lines ca comma cb is equal to ab
1181
things which are equal to the same thing are equal to one another
1191
ca is equal to cb
11101
the three straight lines ca comma ab comma bc are equal to one another
11111
the triangle abc is equilateral

```

Πίνακας 3: Το αρχείο που παράγει το σύστημα με το κείμενο χωρισμένο σε προτάσεις


```

1141
the point a
is
the centre of the circle cdb
1142
ac
is
equal to ab
1151
the point b
is
the centre of the circle cae
1152
bc
is
equal to ba
1171
each of the straight lines ca comma cb
is
equal to ab
1181
things which
are
equal to the same thing are equal to one another
1191
ca
is
equal to cb
11101
the three straight lines ca comma ab comma bc
are
equal to one another
11111
the triangle abc
is
equilateral
0

```

Πίνακας 4: Το αρχείο που παράγει το σύστημα με τις προτάσεις του κειμένου της Γεωμετρίας χωρισμένες σε τρία μέρη -το ρήμα στο μέσον-

Οι προτάσεις των αρχείων που περιγράφηκαν θα χρησιμοποιηθούν από τα υπόλοιπα τμήματα του συστήματος με σκοπό την απάντηση των ερωτήσεων που θα εισαχθούν από τον ενδιαφερόμενο χρήστη.

➤ Η διαδικασία επιλογής της ερώτησης από τον χρήστη, καθώς και η αυτόματη ανάλυση και επεξεργασία της σε φυσική γλώσσα από το σύστημα, πραγματοποιείται μέσω του προγράμματος που περιγράφεται και τεκμηριώνεται αναλυτικά στο ΚΕΦΑΛΑΙΟ 7.1: Υποσύστημα επεξεργασίας της ερώτησης (σελ. 75).

Για διευκόλυνση του χρήστη του συστήματος υπάρχει δυνατότητα επιλογής μιας ερώτησης από ένα κατάλογο ερωτήσεων μιας συγκεκριμένης περιοχής ενός

γνωστικού πεδίου π.χ. της Γεωμετρίας για την πρώτη πρόταση του πρώτου βιβλίου των Στοιχείων του Ευκλείδη ή της Βιοϊατρικής για τις πρωτεΐνες «p53» και «mdm2»:

```

q1:-question("why ac is equal to ab").
q2:-question("why ca is equal to ab").
q3:-question("why ac is equal to ba").
q4:-question("why ca is equal to ba").
q5:-question("why ab is equal to ac").
q6:-question("why ba is equal to ac").
q7:-question("why ab is equal to ca").
q8:-question("why ba is equal to ca").
q9:-question("why bc is equal to ba").
q10:-question("why cb is equal to ba").
q11:-question("why bc is equal to ab").
q12:-question("why cb is equal to ab").
q13:-question("why ba is equal to bc").
q14:-question("why ab is equal to bc").
q15:-question("why ba is equal to cb").
q16:-question("why ab is equal to cb").
q17:-question("why ca is equal to cb").
q18:-question("why ac is equal to cb").
q19:-question("why ca is equal to bc").
q20:-question("why ac is equal to bc").
q21:-question("why cb is equal to ca").
q22:-question("why bc is equal to ca").
q23:-question("why cb is equal to ac").
q24:-question("why bc is equal to ac").
q53:-question("why p53 regulates mdm2").
q54:-question("what inhibits p53").
q55:-question("is there a process loop of p53").
q56:-question("is there a process loop of mdm2").
q57:-question("what is influenced by p53").

```

Πίνακας 5: Ενδεικτικός κατάλογος των ερωτήσεων

Από το τμήμα αυτό του συστήματος παράγεται ένα αρχείο το οποίο περιέχει την επιλεγμένη ερώτηση, τις οντότητές της, το ρήμα της καθώς και τη σχέση η οποία συνδέει τις οντότητες. Το περιεχόμενο του αρχείου αυτού για την ερώτηση q1 φαίνεται στον πίνακα 6:

Question.....	why ac is equal to ab
Lentity.....	ac
Verbgrou...	is equal to
Rentity.....	ab
Verb.....	is
Relation.....	equality

Πίνακας 6: Το περιεχόμενο του αρχείου που παράγει το Υποσύστημα επεξεργασίας της ερώτησης για την ερώτηση q1

➤ Η διαδικασία του συστήματος για την παραγωγή οντολογίας υπονοούμενης από τις προτάσεις του κειμένου πραγματοποιείται μέσω του προγράμματος Αυτόματης Εξαγωγής Γνώσης (ΠΑΡΑΡΤΗΜΑ 2, σελ. 141). Το πρόγραμμα περιγράφεται και τεκμηριώνεται αναλυτικά στο ΚΕΦΑΛΑΙΟ 7.3: Υποσύστημα αυτόματης εξαγωγής της γνώσης (σελ. 95). Από το τμήμα αυτό του συστήματος παράγονται ξεχωριστά αρχεία που περιέχουν οντότητες των πεδίων που προκύπτουν από τις προτάσεις του κειμένου. Στους πίνακες που ακολουθούν περιέχονται - κατά τον συμβολισμό της Γεωμετρίας - οι κύκλοι που αναφέρονται στις προτάσεις του κειμένου, τα ευθύγραμμα τμήματα που προκύπτουν με τα συνώνυμά τους, τα άκρα των ευθυγράμμων τμημάτων καθώς και τα σημεία που βρίσκονται στις περιφέρειες των κύκλων.

```
isa_circle("circle","cdb","a")
isa_circle("circle","dbc","a")
isa_circle("circle","bcd","a")
isa_circle("circle","cae","b")
isa_circle("circle","aec","b")
isa_circle("circle","eca","b")
```

Πίνακας 7: Οι κύκλοι
(Εξαγόμενη οντολογική γνώση από τις προτάσεις του κειμένου)

```
isa_synwnyma("synwnyma","ca","ac")
isa_synwnyma("synwnyma","ac","ca")
isa_synwnyma("synwnyma","da","ad")
isa_synwnyma("synwnyma","ad","da")
isa_synwnyma("synwnyma","ba","ab")
isa_synwnyma("synwnyma","ab","ba")
```

Πίνακας 8: Τα ευθύγραμμα τμήματα και τα συνώνυμά τους

```
isa_oncicle("oncicle","cdb","c")
isa_oncicle("oncicle","cdb","d")
isa_oncicle("oncicle","cdb","b")
isa_oncicle("oncicle","dbc","c")
isa_oncicle("oncicle","dbc","d")
isa_oncicle("oncicle","dbc","b")
isa_oncicle("oncicle","bcd","c")
isa_oncicle("oncicle","bcd","d")
isa_oncicle("oncicle","bcd","b")
```

Πίνακας 9: Εξαγόμενη οντολογική γνώση από τα ονόματα των κύκλων που αναφέρονται στο κείμενο

```
isa_endsline("endsline","ca","c","a")
isa_endsline("endsline","ac","a","c")
isa_endsline("endsline","da","d","a")
isa_endsline("endsline","ad","a","d")
isa_endsline("endsline","ba","b","a")
isa_endsline("endsline","ab","a","b")
```

Πίνακας 10: Εξαγόμενη οντολογική γνώση από τα ονόματα των ευθυγράμμων τμημάτων που αναφέρονται στο κείμενο

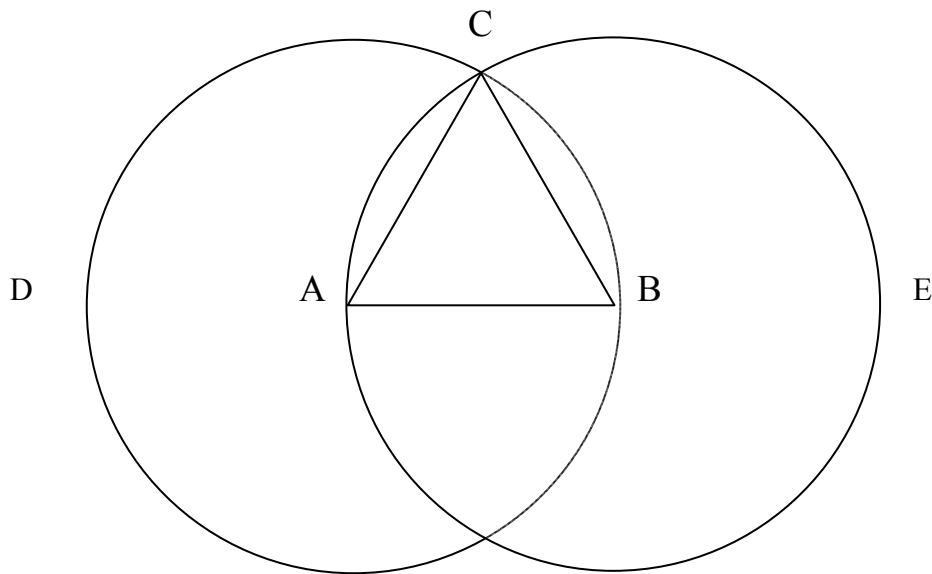
6.2.1 Παράδειγμα 1ο : Παραγωγή απάντησης όταν αυτή είναι ρητή δήλωση σε κείμενο Γεωμετρίας.

➤ Αναφορικά με την εφαρμογή στρατηγικών για την επεξεργασία της ερώτησης, για τον εντοπισμό των οντοτήτων της ερώτησης και της σχέσης των οντοτήτων που τις συνδέει, σε μορφή πρότασης στο σώμα του κειμένου, και τέλος για την παραγωγή της απάντησης όταν αυτή είναι ρητή δήλωση υπό μορφή πρότασης στο κείμενο.

Στο παράδειγμα περιγράφονται τα βήματα με τα οποία το σύστημα απαντά σε ερωτήσεις φυσικής γλώσσας. Το σώμα του κειμένου σε φυσική γλώσσα, από το οποίο θα παραχθούν οι απαντήσεις των ερωτήσεων, είναι το κείμενο της απόδειξης της πρώτης πρότασης του πρώτου βιβλίου των Στοιχείων του Ευκλείδη (σελ. 59). Το σύστημα κάνει οντολογική ανάλυση των ερωτήσεων (Κεφάλαιο 7.1) και στη συνέχεια βρίσκει στο κείμενο των Στοιχείων σε ποια πρότασή του απαντάται το αντίστοιχο τμήμα της ερώτησης με τις οντότητές του. Επειδή όλες οι προτάσεις του κειμένου είναι γραμμένες βάσει αιτιακών σχέσεων, η απάντηση που παράγεται (σελ. 60), δίνεται με τις ίδιες τις προτάσεις του κειμένου, τεκμηριωμένη με τους απαραίτητους ορισμούς, τα αξιώματα και τις έννοιες των Στοιχείων του Ευκλείδη.

Έστω η ερώτηση που εισάγεται στο σύστημα: «why ac is equal to ab ?», όπου ac και ab είναι αντίστοιχα τα ευθύγραμμα τμήματα AC και AB. Η μετατροπή των κεφαλαίων γραμμάτων σε αντίστοιχα μικρά έγινε λόγω της Prolog.

Το κείμενο από το οποίο θα αναζητήσει το σύστημα την απάντησή της είναι η απόδειξη της πρώτης πρότασης του πρώτου βιβλίου των Στοιχείων του Ευκλείδη: «On a given finite straight line to construct an equilateral triangle» και προέρχεται από τον Heath L. T. (1956). *“The Thirteen books of Euclid’s Elements”*, Dover N.Y.



Σχήμα 1: Από την απόδειξη της πρώτης πρότασης των ‘Στοιχείων’ από το βιβλίο του Heath

Το κείμενο της απόδειξης από το βιβλίο του Heath (ό.π., Vol1, σελ.241) είναι το:

«Let AB be the given finite straight line.
 Thus it is required to construct an equilateral triangle on the straight line AB.
 With centre A and distance AB let the circle BCD be described; [Post.3]
 again, with centre B and distance BA let the circle ACE be described; [Post.3]
 and from the point C, in which the circles cut one another, to the points A, B let
 the straight lines CA, CB be joined. [Post.1]
 Now, since the point A is the centre of the circle CDB, AC is equal to AB. [Def.15]
 Again, since the point B is the centre of the circle CAE, BC is equal to BA. [Def.15]
 But CA was also proved equal to AB;
 therefore each of the straight lines CA, CB is equal to AB.
 And things which are equal to the same thing are also equal to one another; [C.N.1]
 therefore CA is also equal to CB.
 Therefore the three straight lines CA, AB, BC are equal to one another.
 Therefore the triangle ABC is equilateral;
 and it has been constructed on the given finite straight line AB.
 (Being) what it was required to do».

Πίνακας 11: Το κείμενο της απόδειξης της πρώτης πρότασης των ‘Στοιχείων’ από το βιβλίο του Heath

Το σύστημα για να απαντήσει στην ερώτηση «why ac is equal to ab», θα αναζητήσει πρώτα στο σώμα του κειμένου την πρόταση που αποτελεί και τμήμα της ερώτησης: «ac is equal to ab» στο οποίο αναφέρονται οι οντότητες και η σχέση που τις συνδέει. Αυτό επιτυγχάνεται από το σύστημα μέσω διαδικασιών, που περιγράφονται λεπτομερώς στο Κεφάλαιο 7.4: «Υποσύστημα απάντησης ερωτήσεων». Το τμήμα αυτό της συγκεκριμένης ερώτησης υπάρχει στη περίοδο του κειμένου: «Now, since the point A is the centre of the circle CDB, AC is equal to AB» (Πίνακας 11).

Στη συνέχεια το σύστημα θα αναζητήσει την αιτιολόγηση της αλήθειας αυτής της πρότασης, η οποία μπορεί είτε να είναι ρητή στο κείμενο είτε άρρητη και θα χρησιμοποιηθεί στην απάντηση της ερώτησης. Στο συγκεκριμένο παράδειγμα που περιγράφεται εδώ, η αιτιολόγηση που δίνεται από το κείμενο είναι ρητή στην περίοδο που αναφέρθηκε προηγουμένως με την πρόταση: «the point A is the centre of the circle CDB» και προηγείται της προτάσεως: «AC is equal to AB» στην ίδια περίοδο. Το σύστημα κατά την παραγωγή της απάντησης εκτός από την ρητή αιτιολόγηση που υπάρχει στο κείμενο, περιγράφει και τον συλλογισμό που υπονοεί η ρητή δήλωση του κειμένου από τον οποίο συνάγεται το ζητούμενο της ερώτησης. Η περιγραφή του συλλογισμού κάνει χρήση και αναφορά της προαπαιτούμενης γνώσης του πεδίου, με τις οντότητες της ερώτησης που εισάγεται στο σύστημα. Αυτό εξασφαλίζει την ορθότητα της επιλογής κατά την αναζήτηση που πραγματοποίησε το σύστημα, και εφοδιάζει τον χρήστη του συστήματος με την τεκμηρίωση της απάντησης που παράγει αυτόματα το σύστημα. Στη συγκεκριμένη περίπτωση η προαπαιτούμενη γνώση αναφέρεται στον ορισμό του κύκλου και η απάντηση που παράγεται αυτόματα από το σύστημα είναι:

```
Your question is :
[why <ac> is equal to <ab> ?].
I analysed this question using strategy : <strategy1>
which involves : nosynonymes, left, right, noinversion,
equality, nopk
My ansuer using Euclid's proof text offers the justification :
because < the point a is the centre of the circle cdb >.

This justification is based on the following :
<ac>=<ab>
because the ends of <ac> are <a>, <c>
and the ends of <ab> are <a>, <b>
and the common end <a> is the center of the circle <cdb>.
That is, <c>, <b> are on the circumference of the same circle,
and therefore <ac>, <ab> are equal as radii on the same circle.
```

The above follows from the definition of the circle.
In this case, circle c is a plane figure contained by one line such that all the straight lines ac , ab falling upon it from one point a among those lying within the figure are equal to one another i.e. $ac=ab$ where the point a is the centre of the circle.

Πίνακας 12: Η επεξήγηση της απάντησης που παράγεται αυτόματα από το σύστημα όταν η απάντηση δηλώνεται ρητά στο κείμενο

Ο τρόπος με τον οποίον παράγονται όλα τα μέρη της απάντησης από το σύστημα περιγράφονται λεπτομερώς στο Κεφάλαιο 7.4: «Υποσύστημα απάντησης ερωτήσεων». Αν στην ερώτηση οι οντότητες δεν αναφέρονται όπως στο κείμενο: «why ac is equal to ab », αλλά με ένα συνώνυμό τους π.χ. «why ca is equal to ba » ή έχει λάβει η μια τη θέση της άλλης στη σχέση που τις συνδέει π.χ. «why ab is equal to ac » είτε και τα δύο συγχρόνως π.χ. «why ba is equal to ca », τότε το σύστημα επιλέγει από ένα προκαθορισμένο κατάλογο στρατηγικές (Πίνακας 13), διαδοχικά τη μια μετά την άλλη, έως ότου βρεθεί εκείνη για την οποία δεν θα αποτύχουν οι προτεινόμενες διαδικασίες ορθής αναζήτησης, όπως περιγράφηκε ανωτέρω, με σκοπό την αιτιολογημένη απάντηση της ερώτησης. Θα πρέπει επίσης να επισημανθεί ότι το σύστημα είναι πάντα ενήμερο της τρέχουσας στρατηγικής και των επιλογών της καθώς και των στρατηγικών που απέτυχαν.

```
next(s,strategy1).
next(strategy1,strategy2).
next(strategy2,strategy3).
next(strategy3,strategy4).
next(strategy4,strategy5).
next(strategy5,strategy6).
next(strategy6,strategy7).
next(strategy7,strategy8).
next(strategy8,strategy9).
next(strategy9,strategy10).
next(strategy10,strategy11).
next(strategy11,strategy12).
next(strategy12,strategy13).
next(strategy13,strategy14).
next(strategy14,strategy15).
next(strategy15,strategy16).
next(strategy16,strategy17).
next(strategy17,strategy18).
next(strategy18,strategy19).
next(strategy19,strategy20).
```

```
next(strategy20,strategy21).
next(strategy21,strategy22).
next(strategy22,strategy23).
next(strategy23,strategy24).
```

Πίνακας 13: Η διαδοχή της επιλογής των στρατηγικών

Στον Πίνακα 14 εμφανίζονται όλες οι στρατηγικές του συστήματος με τα ορίσματά τους:

```
strategy(nosynonymes,left,right,noinversion,equality,nopk,strategy1).
strategy(nosynonymes,left,right,meinversion,equality,nopk,strategy2).
strategy(mesynonymes,left,noright,noinversion,equality,nopk,strategy3).
strategy(mesynonymes,left,noright,meinversion,equality,nopk,strategy4).
strategy(mesynonymes,noleft,right,noinversion,equality,nopk,strategy5).
strategy(mesynonymes,noleft,right,meinversion,equality,nopk,strategy6).
strategy(mesynonymes,left,right,noinversion,equality,nopk,strategy7).
strategy(mesynonymes,left,right,meinversion,equality,nopk,strategy8).
strategy(nosynonymes,left,right,noinversion,equality,pk,strategy9).
strategy(nosynonymes,left,right,meinversion,equality,pk,strategy10).
strategy(mesynonymes,left,noright,noinversion,equality,pk,strategy11).
strategy(mesynonymes,left,noright,meinversion,equality,pk,strategy12).
strategy(mesynonymes,noleft,right,noinversion,equality,pk,strategy13).
strategy(mesynonymes,noleft,right,meinversion,equality,pk,strategy14).
strategy(mesynonymes,left,right,noinversion,equality,pk,strategy15).
strategy(mesynonymes,left,right,meinversion,equality,pk,strategy16).
strategy(nosynonymes,left,right,noinversion,causality,nopk,strategy17).
strategy(mesynonymes,left,noright,noinversion,causality,nopk,strategy18).
strategy(mesynonymes,noleft,right,noinversion,causality,nopk,strategy19).
strategy(mesynonymes,left,right,noinversion,causality,nopk,strategy20).
strategy(nosynonymes,left,right,noinversion,causality,pk,strategy21).
strategy(mesynonymes,left,noright,noinversion,causality,pk,strategy22).
strategy(mesynonymes,noleft,right,noinversion,causality,pk,strategy23).
strategy(mesynonymes,left,right,noinversion,causality,pk,strategy24).
```

Πίνακας 14: οι στρατηγικές του συστήματος με τα ορίσματά τους

Το κατηγορήμα της στρατηγικής με τα ορίσματά της έχει τη μορφή:

```
strategy(Synonym,LSide,RSide,Inversion,Relstrategy,Pk,strategy1).
```

Η επιλογή μιας στρατηγικής καθορίζει τον τρόπο με τον οποίο το πρόγραμμα θα διεκπεραιώσει τις διαδικασίες ανάλογα με τον τύπο της ερώτησης. Π.χ. άλλη διαδικασία θα ακολουθηθεί για μια σχέση ισότητας, της οποίας η αιτιολόγηση αναφέρεται ρητά στο κείμενο και άλλη όταν η αιτιολόγηση υπονοείται μέσω κάποιων πρωταρχικών αρχών του πεδίου όπως είναι οι κοινές έννοιες των Στοιχείων του Ευκλείδη.

Οι δυνατές τιμές των ορισμάτων του κατηγορήματος 'strategy' έχουν ως εξής :

Synonym= nosynonymes, mesynonymes
LSide= left, noleft
RSide= right, noright
Inversion= noinversion, meinversion
Relstrategy= equality
Pk= pk, nopk
Strategy= strategy1, strategy2,,,

Πίνακας 15: Οι τιμές των ορισμάτων του κατηγορήματος ‘strategy’

Τα ορίσματα στη θέση ‘Synonym’ καθορίζουν εάν οι οντότητες της ερώτησης θα βρεθούν στο κείμενο ως έχουν ή θα βρεθούν συνώνυμά τους. Τα ορίσματα στη θέση ‘LSide’ καθορίζουν εάν θα βρεθεί στο κείμενο συνώνυμο της οντότητας που βρίσκεται αριστερά του ρήματος της ερώτησης. Τα ορίσματα στη θέση ‘RSide’ καθορίζουν εάν θα βρεθεί στο κείμενο συνώνυμο της οντότητας που βρίσκεται δεξιά του ρήματος της ερώτησης. Τα ορίσματα στη θέση ‘Inversion’ καθορίζουν εάν θα βρεθεί στο κείμενο η σχέση με τις θέσεις των οντοτήτων σε αντιμετάθεση ως προς αυτές που έχουν στην εισαγόμενη ερώτηση του συστήματος. Το όρισμα στη θέση ‘Relstrategy’ καθορίζει τη σχέση των οντοτήτων επιλέγοντας ορισμένες στρατηγικές έναντι άλλων με συνέπεια την μείωση του χρόνου επεξεργασίας. Τα ορίσματα στη θέση ‘Pk’ καθορίζουν εάν για την ολοκλήρωση των διαδικασιών με σκοπό την απάντηση της ερώτησης είναι απαραίτητη ή μη η χρήση πρότερης γνώσης. Τέλος τα ορίσματα στη θέση ‘Strategy’ ενημερώνουν συνεχώς ποια είναι η τρέχουσα στρατηγική που εφαρμόζεται από το σύστημα για την ολοκλήρωση της διαδικασίας.

Η στρατηγική που εφαρμόζει το σύστημα για την αναζήτηση της πρότασης του κειμένου που απαντά ρητά στο ερώτημα που εισάγεται, εμπνέεται από τις απόψεις του Πρόκλου που αφορούν τη συγκεκριμένη πρόταση των Στοιχείων του Ευκλείδη και έχουν περιγραφή ήδη στο κεφάλαιο για τη Γεωμετρία (σελ. 49). Είναι δε το κείμενο της απόδειξης, μια σειρά διαδοχικών αιτιακών συμπερασμών, με όλα τα συμπεράσματά του να προκύπτουν από λογικές συνεπαγωγές προτάσεων, οι οποίες έχουν προηγηθεί στο κείμενο. Το σύστημά μας λαμβάνοντας αυτό υπόψη του, και μέσω μιας διαδικασίας αναζήτησης, μπορεί και εντοπίζει την πρόταση στην αλυσίδα των λογικών συμπερασμών, η οποία αποτελεί την εξήγηση του συμπεράσματος, όταν αυτό τίθεται ως ερώτημα. Μετά την ανεύρεση από το σύστημα, στο κείμενο της απόδειξης, το τμήμα της ερώτησης που αφορά τη σχέση με τις οντότητες (ή το

αντίστοιχό του, για τους λόγους που αναφέρθηκαν προηγουμένως), ελέγχει την αμέσως προηγούμενή του πρόταση. Η προηγούμενη πρόταση, αν είναι μέρος της αιτιακής αλυσίδας συμπερασμών, μπορεί να παραπέμπει σε διαδικασίες προαπαιτούμενης γνώσης του πεδίου με συμπέρασμα την πρόταση που ακολουθεί, μπορεί όμως και να είναι μια απλή δήλωση από την οποία με απλό συλλογισμό να προκύπτει η πρόταση που ακολουθεί.

6.2.2 Παράδειγμα 2ο : Παραγωγή απάντησης όταν αυτή δεν δηλώνεται ρητά σε κείμενο Γεωμετρίας.

➤ Αναφορικά με την εφαρμογή στρατηγικών για την επεξεργασία της ερώτησης για τον εντοπισμό των οντοτήτων της ερώτησης και της σχέσης των οντοτήτων που τις συνδέει, σε μορφή πρότασης στο σώμα του κειμένου και τέλος για την παραγωγή της απάντησης όταν αυτή δεν είναι ρητή δήλωση υπό μορφή πρότασης στο κείμενο, αλλά είναι αποτέλεσμα συλλογισμών προαπαιτούμενης γνώσης του πεδίου.

Στο παράδειγμα που θα περιγραφεί, ισχύει ότι και στο προηγούμενο 1^ο παράδειγμα, σε ότι αφορά την εφαρμογή των στρατηγικών για την επεξεργασία της ερώτησης και σε ότι τον εντοπισμό στο σώμα του κειμένου, των οντοτήτων της ερώτησης με τη σχέση που τις συνδέει, υπό μορφή πρότασης. Δεν ισχύει όμως το ίδιο και για την παραγωγή της απάντησης, επειδή αυτή δεν δηλώνεται ρητά σε πρόταση του κειμένου, αλλά υπονοείται από πρόταση που δηλώνει προαπαιτούμενη γνώση – «things which are equal to the same thing are also equal to one another» - του επιστημονικού πεδίου το οποίο για το παράδειγμα είναι ο χώρος της Ευκλείδειας Γεωμετρίας. Το σύστημα για την παραγωγή της απάντησης σε αυτήν την περίπτωση, εκτελεί όλες τις διαδικασίες συλλογισμού που υπονοούνται από την προαπαιτούμενη γνώση εμπνεόμενο από τον ισχυρισμό του Αριστοτέλη (Αναλυτικών Ύστερων, 71a,1-2) «Πάσα διδασκαλία και πάσα μάθησις διανοητική εκ προϋπαρχούσης γίνεται γνώσεως». Με τον τρόπο αυτό δίνεται η δυνατότητα στον χρήστη του συστήματος, να παρακολουθεί τα βήματα του συλλογισμού που αιτιολογεί με προτάσεις του σώματος του κειμένου, την ορθότητα της απάντησης.

Έστω η ερώτηση που εισάγεται στο σύστημα: «why bc is equal to ac ?».

Το σύστημα για να απαντήσει στην ερώτηση «why bc is equal to ac ?», θα αναζητήσει πρώτα στο σώμα του κειμένου, με διαδικασίες που περιγράφηκαν στο 1^ο

παράδειγμα (σελ. 58), την πρόταση: «bc is equal to ac» που αποτελεί και τμήμα της ερώτησης στο οποίο αναφέρονται οι οντότητες και η σχέση που τις συνδέει. Επειδή αυτή η πρόταση, ως έχει, δεν υπάρχει στο κείμενο, το σύστημα πριν δηλώσει αποτυχία απάντησης σ' αυτήν την ερώτηση, θα αναζητήσει, όπως έχει ήδη περιγραφεί (σελ. 63-64), την κατάλληλη στρατηγική, η οποία κάθε φορά αφού αντικαταστήσει κάποια ή όλες τις οντότητες της πρότασης με τα συνώνυμά τους (Πίνακας 8) και επιλέξει την αντιμετάθεσή τους ως προς την σχέση που τις συνδέει, θα επαναλαμβάνει τις διαδικασίες αναζήτησής της στο κείμενο. Για την ερώτηση: «why bc is equal to ac ?» επιτυγχάνει εκείνη η στρατηγική, η οποία θα αντικαταστήσει τις δύο οντότητες bc και ac αντίστοιχα με τα συνώνυμά τους cb και ca, και θα επιλέξει επίσης να τις αντιμεταθέσει ως προς τη σχέση τους. Τότε η πρόταση που προκύπτει είναι η: «ca is equal to cb», η οποία υπάρχει στη περίοδο του κειμένου: «And things which are equal to the same thing are also equal to one another; therefore CA is also equal to CB» (το «also» είναι αδιάφορη λέξη για το σύστημα).

Στη συνέχεια το σύστημα θα αναζητήσει την αιτιολόγηση του συμπεράσματος το οποίο περιγράφεται από την πρόταση: «ca is equal to cb», η οποία μπορεί να είναι ρητή στο κείμενο ή άρρητη και θα χρησιμοποιηθεί για την απάντηση της ερώτησης. Για το παράδειγμα που περιγράφεται εδώ, η αιτιολόγηση που δίνεται από το κείμενο για το συμπέρασμα: «CA is also equal to CB» υπονοείται από την πρόταση που προηγείται του συμπεράσματος στην ίδια περίοδο: «And things which are equal to the same thing are also equal to one another» και παραπέμπει σε προαπαιτούμενη γνώση του πεδίου: την πρώτη από τις κοινές έννοιες των Στοιχείων του Ευκλείδη.

Το σύστημα λόγω της συγκεκριμένης προαπαιτούμενης γνώσης, θα επιλέξει τη στρατηγική εκείνη, η οποία αναζητά στο κείμενο προτάσεις, διαφορετικές μεταξύ τους, που έχουν δύο οντότητες και μια σχέση να τις συνδέει, με την προϋπόθεση η μια από τις οντότητες (ή το συνώνυμό της) και η σχέση μεταξύ των δύο συγκεκριμένων οντοτήτων, να είναι ίδιες με αυτές της πρότασης που αποτελεί και τμήμα της ερώτησης. Στο παράδειγμα, η πρόταση: «CA is also equal to CB», περιέχει τις οντότητες «CA» και «CB». Το συνώνυμο της οντότητας «CA» είναι το «AC» και υπάρχει στην πρόταση: «AC is equal to AB». Ομοίως το συνώνυμο της οντότητας «CB» είναι το «BC» και υπάρχει στην πρόταση: «BC is equal to BA». Οι άλλες οντότητες «AB» και «BA», που περιέχονται στις δύο αυτές προτάσεις του κειμένου, είναι συνώνυμα, σύμφωνα δε με την προαπαιτούμενη γνώση του πεδίου (η πρώτη

από τις κοινές έννοιες), οι άλλες οντότητες των προτάσεων: «AC» και «BC» (ή τα αντίστοιχα συνώνυμά τους «CA» και «CB»), συνδέονται με την ίδια σχέση: «AC is equal to BC» (ή «CA is equal to CB»).

Στο κείμενο υπάρχει η πρόταση: «CA is equal to CB».

Το σύστημα κατά την παραγωγή της απάντησης στην ερώτηση αναφέρει, την στρατηγική που επέλεξε, τις διαδικασίες που εφάρμοσε για την ταύτιση του τμήματος της ερώτησης με τμήμα του κειμένου, και την αιτιολόγηση του συμπεράσματος που προκύπτει από τον συλλογισμό που υπονοεί η προαπαιτούμενη γνώση του πεδίου, κάνοντας χρήση των οντοτήτων της ερώτησης. Η απάντηση που παράγεται από το σύστημα είναι:

```
Your question is:
why <bc> is <equal> <to> <ac> ?
I analysed this question using strategy : <strategy16>
which involves : mesynonymes, left, right, meinversion, equality, pk
My answer using Euclid's proof text offers the justification :
because
<bc> is equal to <ab> and
<ac> is equal to <ba>
furthermore
because
<ab>, <ba> are synonymes, which means that
<bc>, <ac> are equal to each other
because of the common notion
< things which are equal to the same thing are equal to one another>.
```

Πίνακας 16: Η επεξήγηση της απάντησης που δεν παράγεται αυτόματα από το σύστημα όταν η απάντηση δεν δηλώνεται ρητά στο κείμενο -της Γεωμετρίας-

6.2.3 Παράδειγμα 3ο : Παραγωγή απάντησης όταν αυτή δεν δηλώνεται ρητά σε Βιοϊατρικά κείμενα.

Στο τμήμα αυτό περιγράφεται ένα παράδειγμα από το χώρο της Βιοϊατρικής και συγκεκριμένα την αλληλεπίδραση των πρωτεϊνών p53 και mdm2. Το παράδειγμα αυτό είναι εμφανές ότι αφορά την εφαρμογή του συστήματος σε ένα πεδίο τελείως διαφορετικό των άλλων δύο προηγούμενων παραδειγμάτων. Εντούτοις το σύστημα προσαρμόστηκε με σχετική ευκολία διότι οι μόνες απαιτήσεις για την προσαρμογή αυτή αφορούσαν την αλλαγή της οντολογίας και του λεξικού, καθώς επίσης την προσθήκη κανόνων γραμματικής στο πρόγραμμα του συστήματος.

Θα δείξουμε την επιλογή της στρατηγικής από το σύστημα που αφορά:

1. την επεξεργασία της ερώτησης,
2. τον εντοπισμό προτάσεων του κειμένου στις οποίες υπάρχουν οντότητες που συνδέονται με αιτιακή σχέση και
3. την παραγωγή της απάντησης, στην οποία περιέχονται εξηγήσεις που παράγονται είτε άμεσα είτε κατόπιν εφαρμογής παραγωγικού συμπερασμού με χρήση πληροφοριών που καταγράφει το σύστημα σε εσωτερικές βάσεις δεδομένων και αφορούν τις διαδοχικές καταστάσεις του συστήματος.

Η ρητή αναφορά του ιστορικού όλων των συλλογισμών για την εξαγωγή της πληροφορίας που παράγεται αυτόματα στο κείμενο, στην απάντηση της εισαγόμενης ερώτησης, παρέχει στον χρήστη της μια αρκετά πειστική εξήγηση για της ορθότητά της.

Στο παράδειγμα που θα περιγραφεί, ισχύει ό,τι και στα προηγούμενα παραδείγματα, σε ό,τι αφορά τις διαδικασίες αναζήτησης: των προτάσεων στο σώμα του κειμένου που είναι σε φυσική γλώσσα, και τον εντοπισμό των οντοτήτων μέσα στην κάθε πρόταση, οι οποίες περιγράφονται λεπτομερώς στο ΚΕΦΑΛΑΙΟ 7.4 Το σώμα του κειμένου είναι προτάσεις από βιοϊατρικά κείμενα στην αγγλική γλώσσα, που πάρθηκαν μέσω διαδικτύου από το Pubmed και αφορούν τις πρωτεΐνες «p53» και «Mdm2»:

The p53 protein regulates the mdm2 gene. The mdm2 oncogene can inhibit p53 mediated transactivation. The mdm2 gene enhances the tumorigenic potential of cells.

Πίνακας 17: Το κείμενο της Βιοϊατρικής ως κείμενο εισαγωγής στο σύστημα

Το σύστημα αρχικά, στο τμήμα του για την επεξεργασία της ερώτησης, όπως αναφέρθηκε και στα προηγούμενα παραδείγματα, αναλύει την ερώτηση στα συστατικά της μέρη χρησιμοποιώντας τη γραμματική της ερώτησης και τα αποθηκεύει για να τα χρησιμοποιήσει στο τμήμα παραγωγής της απάντησης. Η αυτόματη ανάλυση και επεξεργασία της ερώτησης σε φυσική γλώσσα πραγματοποιείται μέσω του προγράμματος που περιγράφεται και τεκμηριώνεται αναλυτικά στο ΚΕΦΑΛΑΙΟ 7.4: «Υποσύστημα επεξεργασίας της ερώτησης» (σελ. 75). Από το τμήμα αυτό του συστήματος παράγεται ένα αρχείο το οποίο περιέχει την επιλεγμένη ερώτηση, και τα κύρια συστατικά της μέρη: οι οντότητές, η ρηματική ομάδα, το ρήμα και το είδος της που δηλώνεται. Για μια ερώτηση όπως η: «is there a

process loop of p53?» το παραπάνω τμήμα του συστήματος θα παράγει ένα αρχείο το περιεχόμενο του οποίου φαίνεται στον πίνακα 18:

Question.....	is there a process loop of p53
Lentity.....	blank
Verbgrou...	is there
Rentity.....	p53
Verb.....	is
Relation.....	causality

Πίνακας 18: Το περιεχόμενο του αρχείου που παράγει το Υποσύστημα επεξεργασίας της ερώτησης για την ερώτηση q55

Το «blank» στο αρχείο δηλώνει ότι δεν υπάρχει στην ερώτηση οντότητα αριστερά του ρήματος «is» και ως στοιχείο αφορά την διαδικασία του συστήματος που θα χρησιμοποιήσει αυτό το αρχείο.

Στη συνέχεια όπως έχει ήδη αναφερθεί, το σύστημα προχωρά στην αυτόματη ανάλυση των προτάσεων του κειμένου, χρησιμοποιώντας μια γραμματική κειμένου, λεξικά και οντολογίες. Μετά την αυτόματη ανάλυση το σύστημα παράγει αρχεία τα οποία περιέχουν είτε μια-μια ολόκληρες τις προτάσεις με το ρήμα τους (Πίνακας 19), είτε τις ίδιες προτάσεις χωρισμένες όμως σε τρία τμήματα: το αριστερό μέρος της πρότασης ως προς το ρήμα, το ρήμα και το δεξί μέρος της πρότασης ως προς το ρήμα (Πίνακας 20).

1411
the p53 protein regulates the mdm2 gene
1421
the mdm2 oncogene can inhibit p53 mediated transactivation
1431
the mdm2 gene enhances the tumorigenic potential of cells
0

Πίνακας 19: Οι προτάσεις του κειμένου της Βιοϊατρικής μετά από επεξεργασία από το σύστημα

1411
 the p53 protein
 regulates
 the mdm2 gene
 1421
 the mdm2 oncogene can
 inhibit
 p53 mediated transactivation
 1431
 the mdm2 gene
 enhances
 the tumorigenic potential of cells
 equilateral
 0

Πίνακας 20: Το αρχείο που παράγει το σύστημα με τις προτάσεις του κειμένου της Βιοϊατρικής χωρισμένες σε τρία μέρη -το ρήμα στο μέσον-

Με την επεξεργασία αυτή των προτάσεων του κειμένου, το σύστημα με αυτόματο τρόπο εντοπίζει το ρήμα της κάθε πρότασης και τις οντότητες, αν αυτές υπάρχουν, για κάθε ένα από τα δύο τμήματα της πρότασης που προέκυψαν από τον χωρισμό τους ως προς το ρήμα (ΚΕΦΑΛΑΙΟ 7.4: Υποσύστημα επεξεργασίας της ερώτησης).

Έστω η ερώτηση που εισάγεται στο σύστημα: «is there a process loop of p53?». Το σύστημα, όπως έχει ήδη αναφερθεί, θα αναζητήσει την απάντηση της ερώτησης από βιοϊατρικά κείμενα σε φυσική γλώσσα. Συνεπώς λόγω της εστίασης του συστήματος στο πεδίο της Βιοϊατρικής, για την επίτευξη της διαδικασίας της ανάλυσης της ερώτησης και του κειμένου, θα πρέπει το λεξικό του συστήματος να περιλαμβάνει ορισμένες λέξεις και όρους που είναι οντότητες της Βιοϊατρικής.

isalexi("enhance", "verb"),
 isalexi("enhances", "verb"),
 isalexi("inhibit", "verb"),
 isalexi("inhibits", "verb"),
 isalexi("regulate", "verb"),
 isalexi("regulates", "verb"),

Πίνακας 21: Το Λεξικό ρημάτων για την εφαρμογή του πεδίου της Βιοϊατρικής

isalexi("p53", "entity"),
 isalexi("mdm2", "entity")

Πίνακας 22: Το Λεξικό οντοτήτων για την εφαρμογή του πεδίου της Βιοϊατρικής

Επίσης ο συλλογισμός της στρατηγικής που θα επιλέξει το σύστημα, πρέπει να λάβει υπόψη του τη σημασιολογία των όρων και των οντοτήτων του πεδίου που αναφέρονται είτε στην ερώτηση είτε στο λεξικό. Στη συγκεκριμένη ερώτηση αναφέρεται ο όρος «διαδικασία βρόχου» (process loop) και η οντότητα «p53» του πεδίου της Βιοϊατρικής.

Το φαινόμενο της ανατροφοδότησης του βρόχου (feedback loop) των πρωτεϊνών p53 και Mdm2 που περιγράφεται στην έρευνα των Bar-Or et al. (2000) και αναφέρθηκε στο ΚΕΦΑΛΑΙΟ 5.3 (σελ. 50), έδωσε την έμπνευση για τον συλλογισμό της στρατηγικής που ακολούθησε το σύστημα για την παραγωγή της απάντησης.

Το σύστημα σύμφωνα με όσα περιγράφηκαν πιο πάνω, εντοπίζει και αποθηκεύει από την ανάλυση και επεξεργασία της συγκεκριμένης ερώτησης: την οντότητα της πρωτεΐνης «p53», και τη μεταξύ των πρωτεϊνών σχέση της «αιτιότητας», η οποία θα εκφράζεται στις προτάσεις του σώματος του κειμένου από τα ρήματα «regulates» και «inhibit». Για τον λόγο αυτό οι προτάσεις που επιλέγει ως σώμα κειμένου το σύστημα με αυτόματο τρόπο, έχουν ως ρήμα τους το «regulates» ή το «inhibit» (Πίνακας 21).

Για την αυτόματη παραγωγή της απάντησης στο ερώτημα: «is there a process loop of p53?» το σύστημα πρέπει να διαπιστώσει αν υπάρχει ή όχι ένας αιτιακός βρόχος ανατροφοδότησης μιας αλυσιδωτής διαδικασίας σχετικών τμημάτων από τις προτάσεις και τις αναφορές του σώματος του κειμένου. Ο συλλογισμός που ακολουθεί το σύστημα για τη διαπίστωση της ύπαρξης ενός αιτιακού βρόχου ανατροφοδότησης, ξεκινάει με την αναζήτηση της οντότητας «p53», που εντόπισε το σύστημα στην ερώτηση (Πίνακας 18), σε ποια από τις προτάσεις του κειμένου υπάρχει και μάλιστα στο αριστερό της τμήμα ως προς το ρήμα. Μια διαδικασία του συστήματος επεξεργάζεται διαδοχικά μια-μια τις προτάσεις και εντοπίζει την ύπαρξη της οντότητας της πρωτεΐνης «p53» που αναφέρεται στην ερώτηση, στην πρόταση: «the p53 protein regulates the mdm2 gene» στο αριστερό της τμήμα: «the p53 protein» πριν το ρήμα «regulates»:

1411
the p53 protein *regulates* the mdm2 gene
 1421
 the mdm2 oncogene can inhibit p53 mediated transactivation
 1431
 the mdm2 gene enhances the tumorigenic potential of cells

Πίνακας 23: Οι προτάσεις που επεξεργάζεται το σύστημα για τον εντοπισμό του τμήματος αριστερά του ρήματος

Για τον εντοπισμό της οντότητας η διαδικασία του συστήματος συγκρίνει μια-μια τις λέξεις του τμήματος (chunk) της πρότασης, με τις οντότητες που υπάρχουν στο λεξικό του συστήματος (Πίνακας 22). Εάν η προς σύγκριση λέξη δεν είναι μια από τις υπάρχουσες οντότητες, η διαδικασία προχωράει στην επόμενη λέξη του τμήματος μέχρις ότου τις εξαντλήσει όλες. Τα όποια αποτελέσματα αυτής της αναζήτησης το σύστημα τα καταγράφει στην παραγόμενη απάντηση της ερώτησης.

Για το συγκεκριμένο παράδειγμα το τμήμα είναι το: «the p53 protein», και τα αποτελέσματα αυτής της διαδικασίας καταγράφονται στην απάντηση ως εξής:

in the chunk <the p53 protein> of the sentence <1411>
 is not the fist token and therefore I test the rest of the tokens.
 Repeating the search
 I found that the entity <p53> is one of the tokens of the chunk <the p53 protein>

Πίνακας 24: Το τμήμα από την παραγόμενη απάντηση της ερώτησης που αφορά την αναζήτηση της οντότητας p53

Στο επόμενο βήμα του συλλογισμού μια διαδικασία του συστήματος, αναζητά και εντοπίζει στην ίδια πρόταση το τμήμα: «the mdm2 gene» που βρίσκεται δεξιά του ρήματος.

1411 the p53 protein *regulates* **the mdm2** gene

Πίνακας 25: Η πρόταση του κειμένου που επεξεργάζεται το σύστημα για τον εντοπισμό του τμήματος δεξιά του ρήματος

Στη συνέχεια το σύστημα ελέγχει και εντοπίζει αν σε αυτό το τμήμα της πρότασης υπάρχει οντότητα και ποια από αυτές που αναφέρονται στο λεξικό του. Στο συγκεκριμένο τμήμα υπάρχει οντότητα και είναι η «mdm2» (Πίνακας 25). Σε

διαφορετική περίπτωση αν δεν υπήρχε οντότητα, το σύστημα θα απέρριπτε την πρόταση και θα ξεκινούσε πάλι από την αρχή τη διαδικασία αναζήτησης της οντότητας «p53» με την επόμενη διαδοχικά πρόταση του σώματος του κειμένου, έως ότου ολοκληρωθεί το αρχείο των προτάσεων.

Για το παράδειγμα που περιγράφεται οι διαδικασίες του συστήματος έχουν εντοπίσει από το σώμα του κειμένου την πρόταση εκείνη που στο αριστερό της τμήμα υπάρχει η οντότητα που βρίσκεται και στην ερώτηση, ενώ στο δεξιό της τμήμα μετά το ρήμα υπάρχει μια άλλη οντότητα από το σύνολο των οντοτήτων που είναι στο λεξικό του συστήματος. Σχηματικά μπορούμε να το περιγράψουμε ως: «p53» «regulates» «mdm2»:

1411 the **p53** protein **regulates** the **mdm2** gene

Πίνακας 26: Η πρόταση του κειμένου που επεξεργάστηκε το σύστημα με τις δύο οντότητες p53 και mdm2 αντίστοιχα αριστερά και δεξιά του ρήματος

Σύμφωνα με το συλλογισμό της στρατηγικής που επιλέχτηκε αυτόματα, το σύστημα για να διαπιστώσει την ύπαρξη ενός αιτιακού βρόχου ανατροφοδότησης της «p53», πρέπει επαναλαμβάνοντας τις διαδικασίες που εκτέλεσε προηγουμένως, να αναζητήσει κατ' αρχήν την οντότητα «mdm2» που εντόπισε στο δεξί τμήμα της πρότασης που επεξεργάστηκε (Πίνακας 26), σε ποια από τις προτάσεις του σώματος του κειμένου υπάρχει στο αριστερό της τμήμα ως προς το ρήμα. Η αναζήτηση εντοπίζει την οντότητα «mdm2» στην πρόταση: «the mdm2 oncogene can inhibit p53 mediated transactivation» στο αριστερό της τμήμα: «the mdm2 oncogene can» πριν το ρήμα «inhibit»:

1411
the p53 protein regulates the mdm2 gene
1421
the mdm2 oncogene can *inhibit* p53 mediated transactivation
1431
the mdm2 gene enhances the tumorigenic potential of cells

Πίνακας 27: Το επόμενο βήμα επεξεργασίας μετά από εκείνο του Πίνακα 23

Στη συνέχεια όπως και προηγουμένως το σύστημα αναζητά και εντοπίζει στην ίδια πρόταση το τμήμα που βρίσκεται δεξιά του ρήματος: «inhibit» και είναι το: «p53 mediated transactivation»:

1421 the mdm2 oncogene can *inhibit* **p53** mediated transactivation

Πίνακας 28: Η επόμενη πρόταση του κειμένου που επέλεξε να επεξεργαστεί το σύστημα για τον εντοπισμό του τμήματος δεξιά του ρήματος

Στο τμήμα αυτό της πρότασης το σύστημα ελέγχει αν υπάρχει οντότητα και ποια, από αυτές που αναφέρονται στο λεξικό του. Για το συγκεκριμένο τμήμα η αναζήτηση εντοπίζει την οντότητα «p53» (Πίνακας 28). Σε διαφορετική περίπτωση αν δεν υπήρχε οντότητα, το σύστημα θα απέρριπτε την πρόταση και θα ξεκινούσε πάλι με τη διαδικασία αναζήτησης της οντότητας «mdm2» στην επόμενη διαδοχικά πρόταση του σώματος του κειμένου, έως ότου ολοκληρωθεί το αρχείο των προτάσεων.

Με τις διαδικασίες του συστήματος που περιγράφηκαν πιο πάνω, εντοπίζεται από το σώμα του κειμένου μια δεύτερη πρόταση, που στο αριστερό της τμήμα, υπάρχει η οντότητα που είχε εντοπίσει το σύστημα στο δεξί τμήμα της πρώτης πρότασης ως προς το ρήμα, ενώ στο δεξί τμήμα της δεύτερης πρότασης μετά το ρήμα υπάρχει οντότητα από το λεξικό του συστήματος. Σχηματικά μπορούμε να το περιγράψουμε ως: «mdm2» «inhibit» «p53»:

1421 the **mdm2** oncogene can *inhibit* **p53** mediated transactivation

Πίνακας 29: Η πρόταση του κειμένου που επεξεργάστηκε το σύστημα με τις δύο οντότητες mdm2 και p53 αντίστοιχα αριστερά και δεξιά του ρήματος

Συνεπώς το σύστημα ελέγχει και διαπιστώνει από πρόταση του κειμένου (No 1411, Πίνακας 19) ότι η οντότητα «p53» επηρεάζει («regulates») την οντότητα «mdm2» και στη συνέχεια από άλλη πρόταση του κειμένου (No 1421, Πίνακας 19) ότι η οντότητα «mdm2» επηρεάζει («inhibit») την οντότητα «p53». Το επόμενο βήμα του συλλογισμού αφορά τη σύγκριση των δύο οντοτήτων που εντόπισε το σύστημα, τη μία στο αριστερό τμήμα ως προς το ρήμα της πρώτης πρότασης (Πίνακας 26) και την

άλλη στο δεξί τμήμα ως προς το ρήμα της τελευταίας των προτάσεων που επιλέχτηκαν από το σώμα του κειμένου (Πίνακας 29). Η σύγκριση δείχνει ότι πρόκειται για την ίδια οντότητα: «p53», πράγμα το οποίο σημαίνει ότι υπάρχει βρόχος ανατροφοδότησής της.

Το σύστημα αφού διαπίστωσε με την στρατηγική που επέλεξε, ότι στις προτάσεις του σώματος του κειμένου υπάρχει ένας αιτιακός βρόχος ανατροφοδότησης, παράγει στη συνέχεια την απάντηση της ερώτησης από βιοϊατρικά κείμενα και παρέχει εξηγήσεις που επιδεικνύουν αυτοεπίγνωση (Πίνακας 30). Αυτό το επιτυγχάνει μέσω της αναπαράστασης και επεξεργασίας της κατάστασής του, καθώς και της αυτόματης ανάλυσης της ερώτησης και του κειμένου.

Ο τρόπος με τον οποίον παράγονται όλα τα μέρη της απάντησης από το σύστημα περιγράφονται λεπτομερώς στο κεφάλαιο 7.4 «Υποσύστημα απάντησης ερωτήσεων».

Η εξήγηση του παραδείγματος όπως αυτή παράγεται κατά λέξη από το σύστημα έχει ως εξής:

```
Your question is : <is there a process loop of p53>
To be answered using the text
-----
I succeeded answering this question using strategy :<strategy17>
With its parameters taking the values :
<nosynonymes>, <left>, <right>, <noinversion>, <causality>, <nopk>
which means that I am testing for causality using no synonyms,
no inversion and no prerequisite knowledge
My answer using the input text is 'yes there is loop of p53'
and I offer the following explanation and meta-explanation of my answering process :
I am searching in the input text for the entity <p53> as mentioned in the question
in the chunk <the p53 protein> of the sentence <1411>
is not the first token and therefore I test the rest of the tokens.
Repeating the search
I found that the entity <p53> is one of the tokens of the chunk <the p53 protein>
which is the chunk to the left of the verb of the sentence <1411>.
I found that to the right of the verb in the phrase <1411> is the
chunk <the mdm2 gene > and about the entity
is not the first token and therefore I test the rest of the tokens.
Repeating the search
the entity <mdm2> is one of the tokens of the chunk <the mdm2 gene >
which is the chunk to the left of the verb of the sentence <1411>.
I found the entity <p53> in the sentence
<1411>: <the p53 protein regulates the mdm2 gene >
I search for the entity <mdm2> as follows:
I am searching in the input text for the entity <mdm2> as mentioned in the question
in the chunk <the p53 protein> of the sentence <1411>
is not the first token and therefore I test the rest of the tokens.
Repeating the search
I found that the entity <mdm2> is not a token of the chunk <the p53 protein>
of the sentence and for this reason I will test the following sentence
in the chunk <the mdm2 oncogene can> of the sentence <1421>
is not the first token and therefore I test the rest of the tokens.
```

Repeating the search
I found that the entity <mdm2> is one of the tokens of the chunk <the mdm2 oncogene can>
which is the chunk to the left of the verb of the sentence <1421>.
I found that to the right of the verb in the phrase <1421> is the chunk <p53 mediated transactivation > and about the entity <p53> is the first token of the chunk
the entity <p53> is one of the tokens of the chunk <p53 mediated transactivation >
which is the chunk to the left of the verb of the sentence <1421>.
In my effort for answering the question <is there a process loop of p53>
I found that from the following sentences
<1411>: <the p53 protein regulates the mdm2 gene >
<1421>: <the mdm2 oncogene can inhibit p53 mediated transactivation >
From which it follows that there is a loop for <p53>.

Πίνακας 30: Η επεξήγηση της απάντησης που παράγεται αυτόματα από το σύστημα όταν η απάντηση δεν δηλώνεται ρητά στο κείμενο -της Βιοϊατρικής-

Στο επόμενο κεφάλαιο δίνεται αναλυτική περιγραφή όλων των προγραμμάτων και των αρχείων με τα οποία υλοποιούνται τα πέντε υποσυστήματα.

ΚΕΦΑΛΑΙΟ 7

Αναλυτική περιγραφή των υποσυστημάτων

7.1 Υποσύστημα επεξεργασίας της ερώτησης

Η ερώτηση που εισάγεται στο σύστημα αναλύεται και δίνει τιμές σε ορίσματα εσωτερικών βάσεων δεδομένων όπως: `question()` για ολόκληρη την πρόταση της ερώτησης, `leftentityofquestion()` για την αριστερή ως προς το ρήμα οντότητα της πρότασης, `verbgroupofquestion()` για την ρηματική έκφραση της πρότασης, `rightentityofquestion()` για την δεξιά ως προς το ρήμα οντότητα της πρότασης, `verbofquestion()` για το ρήμα της πρότασης, και `relationofquestion()` για την σχέση που αναφέρεται και ισχύει μεταξύ των οντοτήτων της πρότασης.

Οι εσωτερικές βάσεις με τα ορίσματα της ερώτησης καταγράφονται σε ένα αρχείο «`q_temp1`» το οποίο επεξεργάζεται το πρόγραμμα του συστήματος για την παραγωγή της απάντησης.

Για διευκόλυνση του χρήστη του συστήματος υπάρχει δυνατότητα επιλογής μιας ερώτησης από ένα κατάλογο ερωτήσεων μιας συγκεκριμένης περιοχής ενός γνωστικού πεδίου όπως για την πρώτη πρόταση του πρώτου βιβλίου των Στοιχείων του Ευκλείδη ή για τις πρωτείνες «`p53`» και «`mdm2`»:

```
q1:-question("why ac is equal to ab").
q2:-question("why ca is equal to ab").
q3:-question("why ac is equal to ba").
q4:-question("why ca is equal to ba").
q5:-question("why ab is equal to ac").
q6:-question("why ba is equal to ac").
q7:-question("why ab is equal to ca").
q8:-question("why ba is equal to ca").
q9:-question("why bc is equal to ba").
q10:-question("why cb is equal to ba").
q11:-question("why bc is equal to ab").
q12:-question("why cb is equal to ab").
q13:-question("why ba is equal to bc").
q14:-question("why ab is equal to bc").
q15:-question("why ba is equal to cb").
q16:-question("why ab is equal to cb").
q17:-question("why ca is equal to cb").
q18:-question("why ac is equal to cb").
q19:-question("why ca is equal to bc").
q20:-question("why ac is equal to bc").
q21:-question("why cb is equal to ca").
q22:-question("why bc is equal to ca").
q23:-question("why cb is equal to ac").
q24:-question("why bc is equal to ac").
```

```

q53:-question("why p53 regulates mdm2").
q54:-question("what inhibits p53").
q55:-question("is there a process loop of p53").
q56:-question("is there a process loop of mdm2").
q57:-question("what is influenced by p53").

```

Το πρόγραμμα καλεί τον κανόνα «question(_)

(για την περίπτωση που η ερώτηση έχει τη μορφή «why ac is equal to ab»
ή τη μορφή «why p53 regulates mdm2»):

```

question(Question):-
f(Question,Head,Rest1),f(Rest1,LEntity,Rest2),
(γίνεται κλήση του κανόνα vg(_,_,_) ο οποίος θα δώσει τιμές στα ορίσματα των
εσωτερικών βάσεων)

```

```

vg(Rest2,Verbgroup,REntity),
write("H ERWTHSH EINAI : <","Head,"> <","LEntity,">
<","Verbgroup,"> <","REntity,"> ?"),nl,
retractall(the_question(_)),
assert(the_question(Question)),
retractall(leftentityofquestion(_)),
assert(leftentityofquestion(LEntity)),
retractall(verbgroupofquestion(_)),
assert(verbgroupofquestion(Verbgroup)),
retractall(rightentityofquestion(_)),
assert(rightentityofquestion(REntity)),
(γίνεται κλήση του κανόνα «extract_question» για την εγγραφή των
ορισμάτων των εσωτερικών βάσεων σε ένα εξωτερικό αρχείο «q_temp1»)
extract_question.

```

(για την περίπτωση που η ερώτηση έχει τη μορφή «what inhibits p53»):

```

question(Question):-
f(Question,Head,Rest1),
(γίνεται κλήση του κανόνα vg(_,_,_) ο οποίος θα δώσει τιμές στα ορίσματα των
εσωτερικών βάσεων)

```

```

vg(Rest1,Verbgroup,REntity),
write("H ERWTHSH EINAI : <","Head,"> <","Verbgroup,">
<","REntity,"> ?"),nl,
retractall(the_question(_)),
assert(the_question(Question)),
retractall(leftentityofquestion(_)),
assert(leftentityofquestion(blank)),
retractall(verbgroupofquestion(_)),
assert(verbgroupofquestion(Verbgroup)),
retractall(rightentityofquestion(_)),

```

```

    assert(rightentityofquestion(REntity)),
(γίνεται κλήση του κανόνα «extract_question» για την εγγραφή των
ορισμάτων των εσωτερικών βάσεων σε ένα εξωτερικό αρχείο «q_temp1»)
    extract_question.

```

(για την περίπτωση που η ερώτηση έχει τη μορφή «is there a process loop of p53»):

```

question(Question):-
(γίνεται κλήση του κανόνα vg(_,_,_,_) ο οποίος θα δώσει τιμές στα ορίσματα
των εσωτερικών βάσεων)

```

```

    vg(Question,Verbgroupp,Process,REntity),
    prepositionofquestion(P),
    write("H ERWTHSH EINAI : <\",Verbgroupp,\"> <\",Process,\">
<\",P,\"> <\",REntity,\"> ?"),nl,
    retractall(the_question(_)),
    assert(the_question(Question)),
    retractall(leftentityofquestion(_)),
    assert(leftentityofquestion(blank)),
    retractall(verbgrouppofquestion(_)),
    assert(verbgrouppofquestion(Verbgroupp)),
    retractall(rightentityofquestion(_)),
    assert(rightentityofquestion(REntity)),
(γίνεται κλήση του κανόνα «extract_question» για την εγγραφή των
ορισμάτων των εσωτερικών βάσεων σε ένα εξωτερικό αρχείο «q_temp1»)
    extract_question.

```

Στη συνέχεια παρουσιάζονται αναλυτικά οι δύο κανόνες που κλήθηκαν από τον κανόνα «question(_):»:

(επιλογή για την περίπτωση που η πρώτη λέξη του προς ανάλυση τμήματος δεν είναι το «equals»)

```

vg(R1,V,RE):-
    f(R1,V,R2),f(R2,RE,""),
    retractall(verbofquestion(_)),
    assert(verbofquestion(V)),
    V<>"equals",
    retractall(relationofquestion(_)),
    assert(relationofquestion("causality")).

```

(επιλογή για την περίπτωση που η ρηματική φράση είναι «is equal to»)

```

vg(R1,VG,RE):-
    f(R1,V,R2),f(R2,R,R3),f(R3,P,R4),f(R4,RE,""),
    R=equal,
    c(V,"",Vb),c(Vb,R,VR),c(VR,"",VRb),c(VRb,P,VG),

```



```

retractall(verbquestion(_)),
  assert(verbquestion(V)),
retractall(relationquestion(_)),
  assert(relationquestion("equality")),
retractall(prepositionquestion(_)),
  assert(prepositionquestion(P)).

```

(επιλογή για την περίπτωση που η ρηματική φράση είναι «is influenced by»)

```

vg(R1, VG, RE) :-
  f(R1, V1, R2), f(R2, V2, R3), f(R3, P, R4), f(R4, RE, ""),
  V2<>"equals",
  c(V1, " ", Vb), c(Vb, V2, VR), c(VR, " ", VRb), c(VRb, P, VG),
  retractall(verbgroupquestion(_)),
  assert(verbgroupquestion(VG)),
  retractall(verbquestion(_)),
  assert(verbquestion(causalverb)),
  retractall(relationquestion(_)),
  assert(relationquestion("causality")),
  retractall(prepositionquestion(_)),
  assert(prepositionquestion(P)).

```

(επιλογή για την περίπτωση που η ρηματική φράση είναι «is there» ακολουθεί το

«a process loop» και συνεχίζει με μια πρόθεση και μια οντότητα)

```

vg(Q, V12, W123, RE) :-
  f(Q, V1, R1), f(R1, V2, R2),
  f(R2, W1, R3), f(R3, W2, R4), f(R4, W3, R5),
  f(R5, P, R6), f(R6, RE, ""),
  c(V1, " ", V1b), c(V1b, V2, V12),
  c(W1, " ", W1b), c(W1b, W2, W12), c(W12, "
", W12b), c(W12b, W3, W123),
  retractall(verbquestion(_)),
  assert(verbquestion(V1)),
  W123="a process loop",
  retractall(relationquestion(_)),
  assert(relationquestion("causality")),
  retractall(prepositionquestion(_)),
  assert(prepositionquestion(P)).

```

(κλήση του κανόνα «extract_question» για την καταχώρηση στο εξωτερικό

αρχείο «q_temp1» τις τιμές των ορισμάτων των εσωτερικών βάσεων)

```

extract_question:-
  the_question(Question), write("the question is =
", Question), nl,
  leftentityofquestion(LEntity), write("left entity of
question = ", LEntity), nl,
  verbgroupofquestion(Verbgroup), write("verbgroup of
question = ", Verbgroup), nl,

```

```

rightentityofquestion(REntity),write("right entity of
question = ",REntity),nl,
verbofquestion(Verb),write("verb of question =
",Verb),nl,
relationofquestion(Relation),write("relation of question
= ",Relation),nl,
openwrite(q_tmpl,"q_tmpl.pro"),writedevice(q_tmpl),
write(Question),nl,
write(LEntity),nl,
write(Verbgroup),nl,
write(REntity),nl,
write(Verb),nl,
write(Relation),nl,
closefile(q_tmpl).

```

7.2 Υποσύστημα ανάλυσης κειμένου

Το κείμενο το οποίο αποτελεί το σώμα της επεξεργασίας του προγράμματος αφορά τις αποδείξεις των πρώτων προτάσεων του 1^{ου} βιβλίου των Στοιχείων του Ευκλείδη, όπως αυτές υπάρχουν στη δεύτερη έκδοση του βιβλίου του Sir Thomas Heath (1956) στην Αγγλική γλώσσα. Από το πλήρες κείμενο των προτάσεων των Στοιχείων λαμβάνονται μόνο τα τμήματα εκείνα που αποτελούν την διαδικασία της απόδειξης των. Το κείμενο παραμένει αναλλοίωτο και οι μόνες διαφοροποιήσεις από το πρωτότυπο είναι ότι στη μια γραμμή γίνεται η αρίθμηση των περιόδων (από τελεία σε τελεία) και στην επόμενη γράφεται η περίοδος του κειμένου, αντικαθιστώντας τα σημεία στίξεως με τη λέξη της ονομασίας τους (full stop, comma, semicolon). Στην αρίθμηση, τα πρώτα ψηφία σχηματίζουν τον αριθμό του βιβλίου των Στοιχείων που βρίσκεται το πλήρες κείμενο της πρότασης του Ευκλείδη. Τα αμέσως επόμενα ψηφία σχηματίζουν τον αριθμό της πρότασης των Στοιχείων στην οποία ανήκει η περίοδος της απόδειξης και τα τελευταία ψηφία σχηματίζουν τον αριθμό της περιόδου στην απόδειξη της πρότασης. Για λόγους που έχουν να κάνουν με τον έλεγχο των ορίων του κειμένου από το πρόγραμμα, έχει γραφεί στην αρχή ο αριθμός 1, στο τέλος ο αριθμός 0, με τις αντίστοιχες περιόδους να έχουν σχόλια που αφορούν την προέλευση του κειμένου. Ένα τμήμα του κειμένου από την αρχή και το τέλος είναι το ακόλουθο:

```

1
. . euclid the thirteen books of the elements book i fullstop . .
114
. . now comma since the point a is the centre of the circle cdv comma
ac is equal to ab fullstop . .
115
. . again comma since the point b is the centre of the circle cae
comma bc is equal to ba fullstop . .
.....

```

```

136
. . so that ae is also equal c fullstop . .
137
. . therefore comma the two straight lines ab comma c comma from ab
the greater ae has been cut off equal to c the less fullstop . .
0
. . by sir thomas little heath fullstop . .

```

Κατά τη διαδικασία ανάλυσης του κειμένου, δημιουργούνται εσωτερικές βάσεις με στοιχεία απαραίτητα στο πρόγραμμα για την εξαγωγή κατάλληλης πληροφορίας (Information Extraction) χρήσιμης στην απάντηση ερωτήσεων (Question Answering). Το πρόγραμμα το οποίο θα επεξεργαστεί το κείμενο των προτάσεων από τα Στοιχεία του Ευκλείδη ξεκινάει με την εκτέλεση του κανόνα `sentences`:

```

1/a  sentences:-clearfilesentences,
        openfilesentences,
        openfileeuclidperiods,
        tokenisation_periods(N),
        getp(N),
        loopsentences.
1/b  sentences:-loopsentences.
1/c  sentences:-closefile(periods),closefile(sentence).

```

Στον κανόνα `sentences` εκτελείται πρώτα η διαδικασία της διαγραφής όλων των εγγραφών του αρχείου `"sentence.pro"` που ορίζεται με τον κανόνα :

```

1/a.1 clearfilesentences:-openwrite(sentence,"sentence.pro"),
        closefile(sentence).

```

(Η εντολή `openwrite` ανοίγει ένα αρχείο για να γράψει πάνω από τις παλιές εγγραφές)

Στη συνέχεια ανοίγει το αρχείο `"sentence.pro"`:

```

1/a.2 openfilesentences:-openappend(sentence,"sentence.pro").

```

(Η εντολή `openappend` ανοίγει το αρχείο για να γράψει νέες εγγραφές χωρίς να σβήσει τις προηγούμενες).

Και τελικώς ανοίγει το αρχείο `"euclid.pro"` στο οποίο βρίσκεται το κείμενο και το οποίο για την διαδικασία εκτέλεσης του προγράμματος έχει το εσωτερικό-συμβολικό όνομα `periods`:

```

1/a.3 openfileeuclidperiods:-openread(periods,"euclid.pro").

```

(Η εντολή `openread` ανοίγει το αρχείο μόνο για ανάγνωση).

Ο κανόνας `1/a sentences` στη συνέχεια εκτελεί την εντολή `tokenisation_periods(N)`, η οποία αναλύει μια περίοδο της απόδειξης από την πρόταση των Στοιχείων

```
1/a.4 tokenisation_periods(N) :-not (eof (periods) ) ,
      tokenisation ,
      inputperiod (N,S) ,
      process (N,S) .
```

για την οποία εφόσον δεν υπάρχει «end of file» του αρχείου `periods` (το συμβολικό όνομα του “`euclid.pro`”) θα δώσει πρώτα εντολή εκτέλεσης του κανόνα `tokenisation` :

```
1/a.4.2 tokenisation:-retractall(w(_,_,_)),retractall(lastw(_)) .
```

ώστε να σβήσει όλες τις εγγραφές των εσωτερικών βάσεων δεδομένων `w(_,_,_)` και `lastw(_)` .

Και μετά θα διαβάσει μια περίοδο του κειμένου δίνοντας εντολή εκτέλεσης του κανόνα `inputperiod(N,S)` :

```
1/a.4.3 inputperiod(N,S) :-readdevice (periods) ,
      readint (N) ,
      readln (S) ,
      readdevice (keyboard) .
```

ο οποίος αφού δώσει προσωρινή εκτροπή «ρευμάτων» του προγράμματος, διαβάζει τον αριθμό της περιόδου : `readint (N)` , και κατόπιν διαβάζει σαν ‘string’ την ίδια την περίοδο : `readln(S)` .

Στη συνέχεια θα εκτελέσει τον κανόνα `process(N,S)` για να διαβάσει λέξη-λέξη την περίοδο :

```
1/a.4.4 process(SN,S) :-pr1(SN,S,1) .
```

όπου `SN` ο αριθμός της περιόδου, και `s` η περίοδος.

Στη συνέχεια εκτελείται ο κανόνας `pr1(SN,S,1)` , όπου με τον αριθμό 1 αριθμείται η πρώτη λέξη της περιόδου.

Συγκεκριμένα η πρώτη επιλογή του κανόνα αφορά την περίπτωση κατά την οποία η περίοδος του κειμένου αφορά τη πρώτη κοινή έννοια του 1^{ου} βιβλίου των Στοιχείων:

```
1/a.4.4.1/a pr1(SN,S,N) :-S=". . and things which are equal
      to the same thing are also equal
      to one another semicolon . ." ,
      pr(SN,S,N) ,! .
```

```
1/a.4.4.1/b pr1(SN,S,N) :-f(S,W,"") ,
      sto(SN,N,W) .
```

```
1/a.4.4.1/c pr1(SN,S,N) :-f(S,W,T) ,
      sto(SN,N,W) ,
      M=N+1 ,
      retractall(lastw(_)) ,assert(lastw(N)) ,
      pr1(T,M,SN) .
```

Στην πρώτη επιλογή του κανόνα, μετά την επιβεβαίωση ότι αφορά ειδικά αυτή την περίοδο του κειμένου, εκτελείται ο κανόνας $pr(SN, S, N)$ ο οποίος θα κάνει την ανάλυση της περιόδου και θα της αφαιρέσει τη λέξη «and» :

```
1/a.4.4.1/a.2/a
    pr(SN, S, N) :- f(S, W, " ") ,
                    sto(SN, N, W) .
1/a.4.4.1/a.2/b
    pr(SN, S, N) :- f(S, W, T) ,
                    sto(SN, N, W) ,
                    M=N+1,
                    retractall(lastw(_)) , assert(lastw(N)) ,
                    pr0(T, M, SN) .
```

Η ανάλυση της περιόδου με τον κανόνα ' $pr(SN, S, N)$ ' γίνεται με το κατηγορήμα `fronttoken`, το οποίο έχει δηλωθεί για το πρόγραμμα με το γράμμα `f` και έχει σαν πρώτο όρισμα το αλφαριθμητικό `string` της περιόδου, σαν δεύτερο όρισμα την πρώτη λέξη του αλφαριθμητικού και σαν τρίτο όρισμα το υπόλοιπο αλφαριθμητικό.

```
1/a.4.4.1/a.2/a.1
    f(S, W, T) :- fronttoken(S, W, T) .
```

Η πρώτη επιλογή του κανόνα γίνεται στην περίπτωση που το αλφαριθμητικό αποτελείται από μια λέξη μόνο, οπότε δεν υπολείπεται καμία άλλη. Τότε καταγράφει αυτή τη λέξη σε μια εσωτερική βάση δεδομένων $w(SN, WN, W)$ με τον κανόνα `sto(SN, N, W)` :

```
1/a.4.4.1/a.2/a.2
    sto(SN, WN, W) :- assert(w(SN, WN, W)) .
```

Η εσωτερική βάση δεδομένων $w(SN, WN, W)$ σε κάθε λέξη `w` δίνει δύο δείκτες αρίθμησης. Ο ένας `SN` είναι ο αριθμός της περιόδου όπως είναι στο κείμενο και ο άλλος `WN` είναι η αύξουσα σειρά της θέσης που κατέχει η λέξη μέσα στην περίοδο από αριστερά προς τα δεξιά.

Εάν όμως το αλφαριθμητικό έχει περισσότερες από μια λέξεις (το τρίτο όρισμα θα έχει κάποιο στοιχείο) τότε θα εκτελεστεί η δεύτερη επιλογή του κανόνα $pr(SN, S, N)$. Σε αυτήν την επιλογή, μετά την καταγραφή της πρώτης λέξης του αλφαριθμητικού στη βάση δεδομένων, αυξάνει τον μετρητή αρίθμησης για την επόμενη λέξη και καταχωρεί σε μια βάση δεδομένων την αρίθμηση της τελευταίας λέξης που κατέγραψε : `assert(lastw(N))` . Τέλος εκτελείται ο κανόνας $pr0(T, M, SN)$ για την ανάλυση του υπόλοιπου αλφαριθμητικού με τον ίδιο τρόπο.

```
1/a.4.4.1/a.2/b.5/a
    pr0(S, N, SN) :- f(S, W, T) , N=3, W="and" ,
                    pr0(T, N, SN) .
```

```

1/a.4.4.1/a.2/b.5/b
  pr0 (S,N,SN) :- f (S,W,T) ,
                sto (SN,N,W) , M=N+1 ,
                retractall (lastw(_)) , assert (lastw(N)) ,
                pr0 (T,M,SN) .

```

```

1/a.4.4.1/a.2/b.5/c
  pr0 ("",_,_) .

```

Η πρώτη επιλογή του κανόνα (1/a.4.4.1/a.2/b.5/a pr0(S,N,SN)) εκτελείται όταν κατά την ανάλυση συμβαίνει να είναι η λέξη «and» το τρίτο αλφαριθμητικό στοιχείο. Τα δύο προηγούμενα στοιχεία είναι οι δύο τελείες που καθορίζουν το αριστερό όριο της περιόδου. Η επιλογή δεν καταγράφει τη λέξη και ξαναεκτελείται ο κανόνας pr0(S,N,SN) για τα επόμενα στοιχεία. Εάν το τρίτο αλφαριθμητικό στοιχείο της περιόδου δεν είναι η λέξη «and» τότε το πρόγραμμα εκτελεί τη δεύτερη επιλογή του κανόνα που βρίσκει και καταγράφει τη νέα λέξη, αυξάνει τον μετρητή αρίθμησης και ξαναεκτελεί τον κανόνα. Όταν δεν θα υπάρχει άλλο αλφαριθμητικό στοιχείο θα εκτελεστεί η τρίτη επιλογή οπότε και θα επιστρέψει στο σημείο από το οποίο κλήθηκε ο κανόνας.

Μετά την επιστροφή του κανόνα στο σημείο κλήσης του, το πρόγραμμα συνεχίζει τις επόμενες επιλογές του κανόνα pr1(SN,S,N) που αφορούν στην ανάλυση των περιόδων γενικά εκτός συγκεκριμένων περιπτώσεων, όπως αυτή που εξετάστηκε προηγουμένως με την κοινή έννοια. Η δεύτερη επιλογή

```

1/a.4.4.1/b pr1 (SN,S,N) :- f (S,W,"") ,
                        sto (SN,N,W) .

```

εκτελείται αν η ανάλυση έχει φτάσει στην τελευταία λέξη της περιόδου την οποία και καταγράφει. Διαφορετικά ξεκινά την ανάλυση των λέξεων της περιόδου με τη τρίτη επιλογή

```

1/a.4.4.1/c pr1 (SN,S,N) :- f (S,W,T) ,
                        sto (SN,N,W) ,
                        M=N+1 ,
                        retractall (lastw(_)) , assert (lastw(N)) ,
                        pr11 (T,M,SN) .

```

η οποία μετά την ανάλυση της πρώτης λέξης συνεχίζει για τις υπόλοιπες με τη διαδικασία ενός αναδρομικού (recursive) κανόνα :

```

1/a.4.4.1/c.5/a
  pr11 (S,N,SN) :- f (S,W,T) ,
                  not (indiferent (W)) ,
                  sto (SN,N,W) , M=N+1 ,
                  retractall (lastw(_)) , assert (lastw(N)) ,
                  pr11 (T,M,SN) .

```

```
1/a.4.4.1/c.5/b
    pr11 (S,N,SN) :- f (S,W,T) ,
                    indifferent (W) ,
                    pr11 (T,N,SN) .
```

```
1/a.4.4.1/c.5/c
    pr11 ("",_,_) .
```

Η πρώτη επιλογή αφορά στην ανάλυση όλων των λέξεων που είναι διαφορετικές ενός συνόλου παραλειπόμενων λέξεων όπως η «also». Στην περίπτωση που το πρόγραμμα συναντά μια από τις παραλειπόμενες λέξεις εκτελείται η δεύτερη επιλογή στην οποία ο αναδρομικός κανόνας συνεχίζει με το υπόλοιπο τμήμα της περιόδου δίχως να λαμβάνει υπόψη του στην ανάλυση την παραλειπόμενη λέξη. Η τρίτη επιλογή αφορά την περίπτωση κατά την οποία δεν υπάρχει άλλο στοιχείο της περιόδου για ανάλυση.

Με αυτήν την διαδικασία ολοκληρώνεται η καταγραφή στην εσωτερική βάση δεδομένων $w(SN, WN, W)$ όλων των λέξεων του κειμένου με δείκτες αρίθμησης που αφορούν στον αριθμό της περιόδου και της θέσης της λέξης μέσα στην περίοδο από αριστερά προς τα δεξιά.

Στη συνέχεια το πρόγραμμα επιστρέφει στον κανόνα `sentence` για να δώσει την εντολή στη διαδικασία `getp(N)` η οποία θα ξεκινήσει τον διαχωρισμό της περιόδου σε τμήματα που χωρίζονται με σημείο στίξεως και περιέχουν ένα ρήμα από αυτά που αναφέρει το πρόγραμμα.

```
1/a.5/a    getp(N) :- retractall (word(_)) , assert (word(0)) ,
            p (N, vL, X) ,
            p (N, vR, Y) .
```

```
1/a.5/b    getp(_) .
```

Ο κανόνας αυτός αφού δώσει πρώτη τιμή (0) στην εσωτερική βάση δεδομένων `word()` στη συνέχεια θα δώσει την εντολή να εκτελεστεί μια από τις διαφορετικές επιλογές του ίδιου κανόνα `p(N, vL, X)`, ο οποίος θα αναλύσει το αριστερό τμήμα «X» της περιόδου «N» ως προς το σημείο στίξεως (η σταθερά `vL` υποδηλώνει ότι η ανάλυση γίνεται για το αριστερό τμήμα ως προς το σημείο στίξεως).

Η πρώτη επιλογή αφορά στην περίπτωση κατά την οποία δεν υπάρχει μέσα στη περίοδο ‘comma’ και το μόνο σημείο στίξεως είναι το ‘full stop’ στο τέλος της περιόδου.

(Με παρόμοια διαδικασία αναλύεται επίσης το άνω και το κάτω όριο του κειμένου με αριθμούς περιόδων : $S=1$ και $S=0$)

```

1/a.5/a.2/a p(S,vL,NP) :-w(S,N,V) ,
    S=1,
    word(XN) ,N>XN,
    V="fullstop",
    writedevic(sentence) ,
    write(S) ,nl,
    N1=N-1,ws(1,N1,NP) ,
    writedevic(sentence) ,
    write(". .") ,write(NP) ,write(" . .") ,nl,
    retractall(numv(_)) ,assert(numv(N)) .

```

```

1/a.5/a.2/b p(S,vL,NP) :-w(S,N,V) ,
    S=0,
    word(XN) ,N>XN,
    V="fullstop",
    writedevic(sentence) ,
    write(S) ,nl,
    N1=N-1,ws(1,N1,NP) ,
    writedevic(sentence) ,
    write(". .") ,write(NP) ,write(" . .") ,nl,
    retractall(numv(_)) ,assert(numv(N)) .)

```

```

1/a.5/a.2/c p(S,vL,NP) :-w(S,N,V) ,
    S<>1,S<>0,
    word(XN) ,N>XN,not(w(S,_,"comma")) ,
    V="fullstop",
    writedevic(sentence) ,
    S1=(10*S)+1,write(S1) ,nl,
    N1=N-1,
    ws(1,N1,NP) ,
    writedevic(sentence) ,
    write(". .") ,write(NP) ,write(" . .") ,nl,
    retractall(numv(_)) ,assert(numv(N)) .

```

Η πρώτη εντολή $w(S, N, V)$ - ανασύρει υπό ορισμένες προϋποθέσεις - όλες τις 'λέξεις' που υπάρχουν στην περίοδο αριστερά της λέξης $V="fullstop"$, από την εσωτερική βάση δεδομένων, μαζί με τον αύξοντα αριθμό τους N που έχουν μέσα στην περίοδο, και τον αριθμό S της περιόδου. Θα πρέπει όμως οι 'λέξεις' να έχουν αύξοντα αριθμό N μεγαλύτερο της τιμής $XN=0$ όπως αυτή έχει ήδη δηλωθεί στο : `assert(word(0))` (1/α.5/α), και να μην υπάρχει μεταξύ των 'λέξεων' της περιόδου S η λέξη "comma" (`not(w(S,_, "comma"))`).

Αν $S=1$, αναλύεται η περίοδος του άνω ορίου (1/a.5/a.2/a), αν $S=0$ αναλύεται η περίοδος του κάτω ορίου (1/a.5/a.2/b) και αν S είναι διάφορο του 0 και του 1 αναλύονται όλες οι υπόλοιπες περίοδοι (1/a.5/a.2/c).

Με την εντολή `writedevic()` εκτρέπει την διαδικασία εξόδου στο αρχείο των προτάσεων : `sentence`, τις οποίες παράγει το πρόγραμμα μετά την 'ανάγνωση' των περιόδων. Στο αρχείο αυτό στη μια γραμμή γράφεται ο αριθμός που χαρακτηρίζει την

κάθε πρόταση μέσω της διαδικασίας $S1=(10*S)+1, write(S1)$ και στην αμέσως επόμενη γραμμή η πρόταση NP που παράγεται από την εντολή $ws(1, N1, NP)$ με 'λέξεις' από την αρχή της περιόδου με τη 'λέξη' που έχει αύξοντα αριθμό 1 μέχρι και αυτήν που έχει αύξοντα αριθμό $N1=N-1$, που είναι η προηγούμενη της $V="fullstop"$ ως εξής :

1/a.5/a.2/c.8/a

```
ws(1, N, X) :- w(_, N, W), not(stop(W)),
               assert(lexi(N, W)),
               M=N-1,
               ws(1, M, Y), !,
               c(Y, " ", YY), c(YY, W, X).
```

1/a.5/a.2/c.8/b

```
ws(1, N, X) :- N>1, X="".
```

Στην πρώτη επιλογή αυτού του κανόνα, η 'λέξη' w που έχει αύξοντα αριθμό N (είναι η προηγούμενη της $V="fullstop"$) και δεν είναι μια από τις stop-words ($not(stop(W))$), δηλώνεται στην εσωτερική βάση : $assert(lexi(N, W))$. Στη συνέχεια αφού μειωθεί κατά ένα ο αύξων αριθμός, εκτελείται πάλι ο αναδρομικός κανόνας για την προ-προηγούμενη 'λέξη', μέχρι να βρεθεί μια από τις stop-words, οπότε εκτελείται η δεύτερη επιλογή : $ws(1, N, X) :- N>1, X=""$.

Τότε στη δεύτερη επιλογή που είναι και terminating conditions, εφόσον ο αύξων αριθμός είναι μεγαλύτερος του 1, ορίζεται στη θέση της 'λέξης' που είναι τώρα μια stopword. Συνεπώς στην εσωτερική βάση $lexi(N, W)$ υπάρχουν όλες οι 'λέξεις' αριστερά της $V="fullstop"$, εκτός της stop-word που βρίσκεται στην αρχή της περιόδου. Στη συνέχεια γίνεται επιστροφή στο σημείο που καλείται ο αναδρομικός κανόνας και επειδή δεν χρειάζεται να βρει άλλες λύσεις, το ! εμποδίζει την ανάστροφη πορεία του προγράμματος.

Κατόπιν η πρώτη επιλογή δίνει εντολή να γίνει δύο φορές η χρήση του κατηγορήματος $concat(X, Y, Z)$, στο οποίο το τρίτο όρισμα Z αντικαθίσταται από τα δύο πρώτα X, Y σε συνεχόμενη γραφή δηλαδή XY. Στο πρόγραμμα το κατηγορήμα αυτό συμβολίζεται με $c(X, Y, Z) :- concat(X, Y, Z)$, οπότε τα κατηγορήματα που εκτελούνται είναι τα :

```
c(Y, " ", YY), c(YY, W, X).
```

Το πρώτο κατηγορήμα δίνει σαν αποτέλεσμα τη 'λέξη' Y που έχει τον μικρότερο αύξοντα αριθμό, με μια κενή θέση στα δεξιά της. Το δεύτερο κατηγορήμα βάζει στα δεξιά του προηγούμενου συνδυασμού τη 'λέξη' με τον αμέσως μεγαλύτερο αύξοντα αριθμό ('λέξη' - κενή θέση - 'λέξη'). Αυτό επαναλαμβάνεται μέχρι να

χρησιμοποιηθούν στο συνδυασμό όλες οι 'λέξεις' που ανασύρθηκαν με αυτήν τη διαδικασία και με τελικό προϊόν την πρόταση x. Στη συνέχεια εκτελείται η εντολή του κανόνα

```
write(". ."),write(NP),write(" . ."),nl,
```

με την οποία γράφει την πρόταση στο αρχείο sentence στην αμέσως επόμενη γραμμή από αυτήν που έγραψε προηγουμένως τον αριθμό της. Ο κανόνας ολοκληρώνεται με την εγγραφή του αύξοντα αριθμού N της V="fullstop" στην εσωτερική βάση numv() μέσω της εντολής

```
retractall(numv(_)),assert(numv(N)).
```

Η δεύτερη επιλογή αφορά στην περίπτωση κατά την οποία δεν υπάρχει μέσα στη περίοδο 'comma' και το μόνο σημείο στίξεως είναι το 'semicolon' στο τέλος της περιόδου.

```
1/a.5/a.2/d p(S,vL,NP):w(S,N,V),
                    word(XN),N>XN,not(w(S,_, "comma")),
                    V="semicolon",
                    writedevic(sentence),
                    S1=(10*S)+1,write(S1),nl,
                    N1=N-1,ws(1,N1,NP),
                    writedevic(sentence),
                    write(". ."),write(NP),write(" . ."),nl,
                    retractall(numv(_)),assert(numv(N)).
```

Η διαδικασία και σε αυτή την επιλογή εκτελείται όπως και στην 1/a.5/a.2/c με μόνη τη διαφορά ότι εδώ η περίοδος τελειώνει σε V="semicolon".

Η τρίτη επιλογή αφορά την περίπτωση κατά την οποία μέσα στη περίοδο υπάρχει 'comma', και επιπλέον ρήμα που προηγείται (αριστερά) από το 'comma':

```
1/a.5/a.2/e p(S,vL,NP):-w(S,N,V),
                    word(XN),N>XN,
                    V="comma",!,
                    retractall(numv(_)),assert(numv(XN)),
                    N1=N-1,
                    retractall(lexi(_,_)),
                    ws(1,N1,NP),
                    retractall(word(_)),assert(word(N)),
                    verbexistence(S,vL,NP,N1).
```

Στην αρχή η επιλογή εντοπίζει από την εσωτερική βάση w(S,N,V) την εγγραφή V="comma", η οποία έχει αύξοντα αριθμό N>XN όπου XN=0 (assert(word(0)) από 1/α.5/α). Δηλαδή βρίσκει το πρώτο "comma" της περιόδου και τον αύξοντα αριθμό του. Κατόπιν κρατάει σε μια εσωτερική βάση τον αύξοντα αριθμό της 'λέξης' μετά από την οποία εντόπισε το "comma":

```
retractall(numv(_)),assert(numv(XN))
```

Στη συνέχεια σβήνει τις εγγραφές της εσωτερικής βάσης `lexi (_, _)` και εκτελεί, όπως έχει ήδη περιγραφεί στους κανόνες `1/a.5/a.2/c.8/a` και `1/a.5/a.2/c.8/b` την διαδικασία `ws(1, N1, NP)` αναλύοντας τις 'λέξεις' της περιόδου από την πρώτη μέχρι και την προηγούμενη `N1` της `V="comma"`, και δίνει όλο αυτό το τμήμα της περιόδου ως τιμή στη μεταβλητή `NP`.

Μετά δίνει νέα τιμή στην εσωτερική βάση `word()` :

```
retractall(word(_), assert(word(N)),
```

έτσι ώστε, αν χρειαστεί, να ανασύρει την εγγραφή με το επόμενο "comma" να ξεκινήσει την έρευνα από την εγγραφή με το προηγούμενο "comma" που έχει αυτόν τον αύξοντα αριθμό. Και στο τέλος δίνει εντολή να εκτελεστεί ο κανόνας `verbexistence(S, vL, NP, N1)` με τον οποίο ελέγχει την ύπαρξη ή μη ρήματος μεταξύ των 'λέξεων' αριστερά από το "comma".

```
1/a.5/a.2/e.9/a
```

```
verbexistence(S, vL, _, N) :- not(lexi(_, is)), not(lexi(_, are)),
    NV=N+1,
    retractall(word(_), assert(word(NV))),
    p(S, vL, _) , !.
```

```
1/a.5/a.2/e.9/b
```

```
verbexistence(S, _, T, N) :- writedevic(sentence),
    S1=(10*S)+1, write(S1), nl,
    write(". ."), write(T), write(" . ."), nl,
    writedevic(screen), write("sentence:", S1), nl,
    M=N+1,
    w(S, M, V),
    write(M, "=", V), nl, write(T).
```

Η πρώτη επιλογή αυτού του κανόνα εκτελείται εάν δεν υπάρχει στις εγγραφές της εσωτερικής βάσης `lexi(,)` (αυτές έγιναν στο `1/a.5/a.2/c.8/a`), ένα από τα ρήματα `is, are` τα οποία και θα αποκαλώ ρήματα-‘λέξεις’.

```
not(lexi(_, is)), not(lexi(_, are))
```

Στην περίπτωση αυτή το πρόγραμμα θα προχωρήσει να βρει την επόμενη 'λέξη' `V="comma"` και θα ελέγξει πάλι αν υπάρχει ένα από τα ρήματα-‘λέξεις’ `is, are`. Για να το κάνει αυτό δίνει τον αύξοντα αριθμό της προηγούμενης 'λέξης' `V="comma"` με το `retractall(word(_), assert(word(NV))` και με αυτόν τον αριθμό επιστρέφει να εκτελέσει ξανά τον κανόνα : `1/a.5/a.2/e p(S, vL, _)`.

Η δεύτερη επιλογή του κανόνα αφορά στην περίπτωση κατά την οποία υπάρχει κάποιο από τα ρήματα-‘λέξεις’ `is, are`. Τότε ο κανόνας θα γράψει σε μια γραμμή του αρχείου `sentence` έναν αριθμό που θα χαρακτηρίζει το τμήμα της περιόδου και στην επόμενη γραμμή το κείμενό της :

```
write(". ."), write(T), write(" . .").
```

Η πρώτη κενή θέση στο τρίτο `write(" . .")` δόθηκε διότι κατά την εκτέλεση των διαδικασιών `c(Y, " ", YY)`, `c(YY, W, X)` ('λέξη'-κενή θέση-'λέξη') δεν υπάρχει στο τέλος της πρότασης κενή θέση. Οι εντολές $M=N+1$ και `w(S, M, V)`, δίνουν τη δυνατότητα να παρουσιαστεί στην οθόνη (`writedevic(screen)`), ο αύξων αριθμός με την αντίστοιχη 'λέξη' `V="comma"`, μέσω της εντολής :

```
write(M, "=", V)
```

Επίσης αναγράφεται και η πρόταση μέχρι το `V="comma"` που περιέχει κάποιο από τα ρήματα-'λέξεις' με την εκτέλεση της εντολής :

```
write(T).
```

Μετά την ολοκλήρωση των διαδικασιών της δεύτερης επιλογής `verbexistence`, το πρόγραμμα επιστρέφει στον κανόνα, από τον οποίο κλήθηκε `1/a.5/a.2/e p(S, vL, NP)`, και αυτός με τη σειρά του στον `1/a.5/a.2/e getp(N)`.

Επειδή η πρώτη επιλογή οδηγεί στην επανάληψη της εκτέλεσης του κανόνα `1/a.5/a.2/e p(S, vL, NP)`, η έξοδος από τον βρόχο γίνεται με συνθήκη τερματισμού. Στην περίπτωση κατά την οποία, μετά το τελευταίο `V="comma"` η περίοδος έχει για τελευταία 'λέξη' την «full stop», το πρόγραμμα εκτελεί την επιλογή :

```
1/a.5/a.2/f p(S, vL, NP) :- w(S, N, V),
                             word(XN), N > XN,
                             V = "fullstop",
                             w(S, XN, "comma"),
                             not(lexi(_, is)), not(lexi(_, are)),
                             writedevic(sentence),
                             S1 = (10 * S) + 1, write(S1), nl,
                             N1 = N - 1, ws(1, N1, NP),
                             writedevic(sentence),
                             write(" . ."), write(NP), write(" . ."), nl,
                             retractall(numv(_)), assert(numv(N)).
```

Αυτή ικανοποιείται αν υπάρχει στην εσωτερική βάση `w(S, N, V)` η 'λέξη' `V="fullstop"` με αύξοντα αριθμό `N`, ο οποίος πρέπει να είναι μεγαλύτερος του αύξοντα αριθμού `XN` (από `1/a.5/a.2/e.9/α.3`), και αν υπάρχει η εγγραφή στην ίδια εσωτερική βάση της 'λέξης' `"comma"`, με αύξοντα αριθμό `XN` (απαίτηση να μην υπάρχει κάποιο από τα ρήματα-'λέξεις' `is`, `are` έχει ήδη ελεγχθεί από την `1/a.5/a.2/e.9/α`).

Στη συνέχεια γράφει στο αρχείο τον αριθμό που θα χαρακτηρίζει την πρόταση και στη συνέχεια αφού αναλύσει όλες τις λέξεις αρχίζοντας από την προηγούμενη της «full stop» μέχρι την `stop-word`, πηγαίνοντας προς τα πίσω, γράφει την πρόταση `NP`.

Στην περίπτωση κατά την οποία μετά το τελευταίο $v="comma"$ η περίοδος έχει για τελευταία ‘λέξη’ το ‘semicolon’ το πρόγραμμα εκτελεί την επιλογή :

```
1/a.5/a.2/g p(S,vL,NP) :-w(S,N,V),
                        word(XN),N>XN,
                        V="semicolon",
                        w(S,XN,"comma"),
                        not(lexi(_,is)),not(lexi(_,are)),
                        writedevic(sentence),
                        S1=(10*S)+1,write(S1),nl,
                        N1=N-1,ws(1,N1,NP),
                        writedevic(sentence),
                        write(". ."),write(NP),write(". ."),nl,
                        retractall(numv(_)),assert(numv(N)).
```

Η οποία είναι παρόμοια με την 1/a.5/a.2/f επιλογή.

Στη συνέχεια και αφού μέχρι στιγμής έχει καταγραφεί το τμήμα της περιόδου μέχρι η ‘λέξη’ $v="comma"$, να έχει στα αριστερά της ένα από τα ρήματα-‘λέξεις’ *is, are*, το πρόγραμμα επιστρέφει στον κανόνα 1/a.5/a.2/e `getp(N)`

για να αναλύσει και να συμπληρώσει την υπόλοιπη περίοδο εκτελώντας τον κανόνα `p(N,vR,Y)` :

```
1/a.5/a.3 p(S,vR,NP) :-w(S,N,V),
                        numv(XN),N>XN,
                        V="comma",!,
                        writedevic(sentence),
                        S2=(10*S)+2,write(S2),nl,
                        lastw(N2),
                        N1=N+1,
                        wo(N1,N2,NP),
                        writedevic(sentence),
                        write(". ."),write(NP),write(". ."),nl.
```

Ο κανόνας καλεί την εγγραφή $v="comma"$ από τη βάση δεδομένων που έχει αύξοντα αριθμό N μεγαλύτερο του XN , το οποίο, όταν η περίοδος έχει "comma" με ρήμα-‘λέξη’ προ αυτού, έχει τιμή τον αύξοντα αριθμό από τον οποίο το πρόγραμμα ξεκίνησε τον έλεγχο για να βρει την εγγραφή $v="comma"$.

Στη συνέχεια ο κανόνας γράφει σε μια γραμμή του αρχείου `sentence` τον αριθμό που χαρακτηρίζει το υπόλοιπο της περιόδου μετά το "comma". Κατόπιν καλεί από την εσωτερική βάση `lastw()` τον αύξοντα αριθμό $N2$ της τελευταίας λέξης της περιόδου που ανέλυσε (1/a.4.4.1/c.5/a) και αφού αυξήσει κατά ένα ($N1=N+1$) τον αύξοντα αριθμό (N) της εγγραφής $v="comma"$, εκτελεί τον κανόνα `wo(N1,N2,NP)` :

```
1/a.5/a.3.8/a wo(N1,N2,X) :-N2>N1+1,
                        w(_,N1,W),not(stop(W)),
                        M=N1+1,
                        wo(M,N2,Y),!,
                        c(W,"",Wb),c(Wb,Y,X).
```

```
1/a.5/a.3.8/b      wo (N1, N2, X) : -N=N1+1, w (N, N2, X) .
```

```
1/a.5/a.3.8/c      wo (N1, N2, X) : -N2>N1, X=" " .
```

Στην πρώτη επιλογή καλεί από την βάση δεδομένων `w(_, N1, w)` τη 'λέξη' `w`, μετά το "comma" με αύξοντα αριθμό τον `N1=N+1`, υπό την προϋπόθεση να μην είναι μια από τις stopwords και ο αύξων αριθμός της αυξημένος κατά μια μονάδα να είναι μεγαλύτερος από αυτόν της τελευταίας 'λέξης' (με αυτόν τον περιορισμό ο αναδρομικός κανόνας εκτελείται μέχρι η ανάλυση φθάσει στην stopword 'λέξη'). Στη συνέχεια εκτελείται ο αναδρομικός κανόνας, καλώντας κάθε φορά από την βάση την επόμενη 'λέξη', ο αύξων αριθμός της οποίας θα είναι μεγαλύτερος κατά μια μονάδα από αυτόν της προηγούμενης (`M=N1+1`). Όταν ολοκληρώσει αυτή τη διαδικασία εκτελεί τα διαδοχικά `concat(X, Y, Z)` για να έχει τις 'λέξεις' σε μια επαναλαμβανόμενη διαδοχή ('λέξη' - κενή θέση - 'λέξη'). Κατόπιν ακολουθούν οι συνθήκες τερματισμού αυτού του κανόνα οι οποίες θα ενεργοποιηθούν, όταν δεν θα ισχύουν οι περιορισμοί της πρώτης επιλογής.

Το πρόγραμμα στη συνέχεια επιστρέφει στον κανόνα `1/a.5/a.3 p(S, vR, NP)` για να γράψει στο αρχείο `sentence` το τμήμα αυτό της περιόδου μετά το "comma".

Με την ολοκλήρωση του κανόνα, το πρόγραμμα επιστρέφει στον `1/a.5/a getp(N)` και μετά στην προηγούμενη κλήση του κανόνα `1/a sentences` ο οποίος καλεί την εκτέλεση του `loopsentences` :

```
1/a.6 loopsentences:-retractall(word(_)),
                    tokenisation_periods(N),
                    getp(N),
                    loopsentences.
```

Εδώ, αφού πρώτα σβήσει τις εγγραφές της εσωτερικής βάσης `word()`, επαναλαμβάνει τους κανόνες `1/a.4 tokenisation_periods()` και `1/a.5 getp()` για την επόμενη περίοδο του κειμένου. Εδώ το πρόγραμμα επαναλαμβάνει τον κανόνα `loopsentences` μέχρι το τέλος του αρχείου των περιόδων (ελέγχεται από τον κανόνα `1/a.4 tokenisation_periods()`) οπότε και επιστρέφει στον `1/a sentences` για να κληθούν στη συνέχεια οι συνθήκες τερματισμού του :

```
1/b sentences:-loopsentences.
```

```
1/c sentences:-closefile(periods),closefile(sentence),nl.
```

οι οποίες κλείνουν τα αρχεία `periods` και `sentence`.

Με την ολοκλήρωση πλέον του κανόνα 1 sentences το πρόγραμμα επιστρέφει στο τμήμα goal για να συνεχίσει το πρόγραμμα μέσω της εκτέλεσης του κανόνα 2 phrases:

```
2/a phrases:-clearfileverbless,openfileverbless,clearfilephrase,
openfilephrase,openreadfilesentences,
tokenisation_sentence(N),
get_phrases(N),
loop_phrases.
```

```
2/b phrases:-loop_phrases.
```

```
2/c phrases:-closefile(sentence),closefile(verbless),
closefile(phrase).
```

Στην επιλογή 2/a phrases, μετά την διευθέτηση των απαραίτητων αρχείων, δίνεται η εντολή της εκτέλεσης του κανόνα tokenisation_sentence():

```
2/a.6 tokenisation_sentence(N):-not.eof(sentence),
tokenisation,
inputsentence(N,S),nl,
process(N,S).
```

ο οποίος έχει παρόμοια διαδικασία με εκείνη του κανόνα 1/a.4 tokenisation_periods(). Στη συνέχεια το πρόγραμμα προχωρά στην εκτέλεση του κανόνα get_phrases():

```
2/a.4/a get_phrases(N):-retractall(word(_),assert(word(0)),
protasi(N,vL,X),
retractall(word(_),assert(word(0)),
protasi(N,vR,Y).
```

```
2/a.4/b get_phrases(_).
```

Στο σημείο αυτό να παρατηρήσουμε ότι το πρόγραμμα ενώ στην εκτέλεση του κανόνα 1 sentences χώριζε τις περιόδους σε προτάσεις, με σημείο αναφοράς τη 'λέξη' "comma" που είχε όμως ένα ρήμα-'λέξη' προ αυτού, στην εκτέλεση του κανόνα 2 phrases θα κατασκευάσει τις προτάσεις έχοντας πλέον σαν σημείο αναφοράς το ρήμα-'λέξη'. Δηλαδή θα φτιάξει ένα αρχείο verbless μέσα στο οποίο θα γράψει τις προτάσεις χωρισμένες σε δύο τμήματα. Το πρώτο τμήμα θα είναι το μέρος της πρότασης αριστερά του ρήματος-'λέξη' και το δεύτερο τμήμα θα είναι το δεξί μέρος της πρότασης μετά το ρήμα-'λέξη'. Ταυτόχρονα όμως θα φτιάξει ένα αρχείο phrase που θα περιέχει το ρήμα-'λέξη' με τα δύο τμήματα αριστερά και δεξιά σε μια ενιαία πρόταση.

Ο κανόνας 2/a.4/a get_phrases(N) καλεί την εκτέλεση της διαδικασίας protasi(N,vL,X). Οι δύο πρώτες επιλογές αυτής της διαδικασίας: 2/a.7/a.2/a,

2/a.7/a.2/b αφορούν σχόλια για το κείμενο και δίνουν στις εσωτερικές βάσεις *verbless* και *phrase*, τα όρια μεταξύ των οποίων βρίσκονται τα τμήματα του κειμένου στα οποία θα γίνει η επεξεργασία. Και για τις δύο εσωτερικές βάσεις η αρχική εγγραφή γίνεται με τον αριθμό 1, ενώ η τελική γίνεται με τον 0.

Για το σχόλιο της αρχής ενεργοποιείται η επιλογή

```
2/a.7/a.2/a protasi (S,vL,NP) :-S=1,
    w(S,N2,_),lastw(N2),wo(1,N2,NP),
    writedevic(verbless),write(S),
    writedevic(phrase),write(S),
    writedevic(screen),write(S),!.
```

ενώ για το σχόλιο του τέλους ενεργοποιείται η επιλογή

```
2/a.7/a.2/b protasi (S,vL,NP) :-S=0,
    w(S,N2,_),lastw(N2),wo(1,N2,NP),
    writedevic(verbless),write(S),
    writedevic(phrase),write(S),
    writedevic(screen),write(S),!.
```

Για το υπόλοιπο κείμενο το οποίο θα είναι αντικείμενο περαιτέρω επεξεργασίας για την απόκτηση πληροφοριών, η επιλογή του κανόνα 2/a.7/a.2/c *protasi (S,vL,NP)* αναπτύσσει το αριστερό τμήμα της πρότασης, που βρίσκεται πριν το ρήμα-‘λέξη’.

```
2/a.7/a.2/c protasi (S,vL,NP) :-w(S,N,V),
    word(XN),N>XN,
    isalexi(V,verb),
    N1=N-1,ws(1,N1,NP),
    writedevic(verbless),
    write(S),nl,write(NP),nl,
    writedevic(phrase),
    write(S),nl,write(NP),write(" ",V," "),
    retractall(word(_)),assert(word(N)).
```

Οι διαδικασίες είναι παρόμοιες με εκείνες του κανόνα 1/a.5/a.2/ *p (S,vL,NP)*

Αφού εντοπιστεί η εγγραφή του ρήματος-‘λέξη’ από την εσωτερική βάση *w(S,N,V)* και γίνει η επεξεργασία κατά τα γνωστά των ‘λέξεων’ που προηγούνται αυτού μέσω της διαδικασίας *ws(1,N1,NP)*, τότε γράφεται σε μια γραμμή του αρχείου *verbless* ο αριθμός που θα χαρακτηρίζει αυτό το τμήμα της πρότασης και αμέσως στην επόμενη γραμμή το κείμενο που παρήχθη μέσω των εντολών *write(S),nl,write(NP)* χωρίς το ρήμα. Στη συνέχεια ο κανόνας γράφει στο αρχείο *phrase* στη μια γραμμή τον αριθμό που θα χαρακτηρίζει τώρα την ενιαία πρόταση και στην άλλη γραμμή το πρώτο τμήμα της πρότασης ακολουθούμενο από το σχήμα : διάστημα - ρήμα-‘λέξη’ – διάστημα, μέσω των εντολών *write(S),nl,write(NP),write(" ",V," ")*.

Μετά την ολοκλήρωση αυτού του κανόνα το πρόγραμμα επιστρέφει τη διαδικασία στον κανόνα 2/a.4/a `get_phrases(N)`, ο οποίος στη συνέχεια θα καλέσει την εκτέλεση της διαδικασίας `protasi(S,vR,NP)` η οποία αναπτύσσει το δεξιό τμήμα της πρότασης μετά το ρήμα-‘λέξη’.

```
2/a.7/a.4 protasi(S,vR,NP):-w(S,N,V),
                    word(XN),N>XN,
                    isalexi(V,verb),
                    lastw(N2),
                    N1=N+1,wo(N1,N2,NP),
                    writedevic(verbless),
                    write(V),nl,write(NP),nl,
                    writedevic(phrase),write(NP),nl.
```

Και εδώ μετά τον εντοπισμό του ρήματος-‘λέξη’, γίνεται η επεξεργασία των ‘λέξεων’ της υπόλοιπης πρότασης, με αύξοντα αριθμό $N1$ τον αμέσως μεγαλύτερο από αυτόν του ρήματος-‘λέξη’ ($N1=N+1$), μέχρι αυτόν της τελευταίας ‘λέξης’ που δίνει η εσωτερική βάση `lastw()`. Η διαδικασία εκτελείται κατά τα γνωστά μέσω του κανόνα `wo(N1,N2,NP)`. Κατόπιν γράφεται στο αρχείο `verbless`, σε συνέχεια της προηγούμενης διαδικασίας στο ίδιο αρχείο, στη μια γραμμή το ρήμα-‘λέξη’ και στην επόμενη το υπόλοιπο τμήμα της πρότασης. Μετά στο αρχείο `phrase`, πάλι σε συνέχεια της προηγούμενης διαδικασίας για το ίδιο αρχείο, γράφεται συνεχόμενα στην ίδια γραμμή το υπόλοιπο τμήμα της πρότασης, διότι το αρχικό μαζί με το ρήμα-‘λέξη’ είχε γραφεί προηγουμένως.

Μετά την ικανοποίηση των απαιτήσεων του κανόνα η διαδικασία του προγράμματος επιστρέφει στο σημείο που εκλήθη (2/a.7/a `get_phrases(N)`) και στη συνέχεια στο σημείο που εκλήθη η `get_phrases(N)`, δηλαδή στην επιλογή 2/a `phrases`, η οποία θα δώσει εντολή στον κανόνα `loop_phrases`:

```
2/a.8 loop_phrases:-retractall(word(_)),
                    tokenisation_sentence(N),
                    get_phrases(N),
                    loop_phrases.
```

Με τις εντολές `tokenisation_sentence(N)` και `get_phrases(N)` ενεργεί όπως προηγουμένως για τις υπόλοιπες περιόδους του κειμένου σε βρόχο (`loop_phrases`).

Η έξοδος από τον βρόχο και την επιστροφή στον κανόνα 2/a `phrases` ενεργοποιεί τις υπόλοιπες επιλογές του κανόνα. Με την επιλογή 2/c `phrases`:

```
2/c phrases:-closefile(sentence),
              closefile(verbless),
              closefile(phrase).
```

κλείνουν τα αρχεία και ολοκληρώνεται η διαδικασία του προγράμματος.

7.3 Υποσύστημα αυτόματης εξαγωγής γνώσης

Δεδομένων των προτάσεων 114 και 115, όπως αυτές έχουν επισημωθεί για να επεξεργαστούν από το πρόγραμμα, παράγονται τα εξωτερικά αρχεία: `circle`, `oncircle`, `synwnyma` και `endslines`. Η οντολογία αυτών των αρχείων χρησιμεύει για την επεξεργασία των ερωτήσεων και για την παραγωγή των απαντήσεων. Η παραγωγή των αρχείων γίνεται μέσω του τμήματος του αλγορίθμου που περιγράφεται στη συνέχεια :

Με την ολοκλήρωση πλέον του κανόνα 1 `sentences` το πρόγραμμα επιστρέφει στο τμήμα `goal` για να συνεχίσει το πρόγραμμα μέσω της εκτέλεσης του κανόνα 2 `extraction_knowledge`:

```
2/a extraction_knowledge:-openreadfilesentences,not(eof(sentence)),
    readdevice(sentence),
    readint(_),readln(S),
    extract_from(S),
    loop_extraction_knowledge,
    readdevice(keyboard).
```

```
2/b extraction_knowledge:-loop_extraction_knowledge.
```

```
2/c extraction_knowledge:-closefile(sentence).
```

Στην επιλογή 2/a `extraction_knowledge` δίνεται η εντολή της εκτέλεσης του κανόνα `openreadfilesentences` :

```
2/a.1 openreadfilesentences:-openread(sentence,"sentence.pro").
```

Με τον κανόνα αυτό διαβάζεται το αρχείο `sentence.pro` το οποίο για την διαδικασία εκτέλεσης του προγράμματος έχει το εσωτερικό-συμβολικό όνομα `sentence` (Η εντολή `openread` ανοίγει το αρχείο μόνο για ανάγνωση).

Στη συνέχεια με το κατηγορημα ανακατεύθυνσης 2/α.2 `readdevice`

(`sentence`) δίνεται εντολή στο πρόγραμμα μέσω του κατηγορήματος 2/α.3 `readln(S)` να διαβάσει από το αρχείο `sentence` την πρόταση (`s`). Την πρόταση (`s`) χειρίζεται ο κανόνας `extract_from(S)` :

```
2/a.4/a extract_from(S):-f(S,".",T1),
    f(T1,".",T2),
    np1(T2,Hypernym1,Entity1,T3),
    f(T3,V,T4),
    np2(T4,Hypernym2,Concept,Entity2,T5),
    f(T5,".",T6),f(T6,".",""),
    V=is,Hypernym1=point,Hypernym2=centre,
```

`dos_file_extraction1 (Concept, Entity2, Entity1) .`

`2/a.4/b extract_from() .`

Τα δύο πρώτα κατηγορήματα $f(S, ".", T1)$, $f(T1, ".", T2)$ που αντιστοιχούν σε κατηγορήματα `fronttoken` διαβάζουν τις δύο τελείες που υπάρχουν στην αρχή κάθε πρότασης του κειμένου. Μετά ο κανόνας δίνει εντολή στην εκτέλεση του `np1 (T2, Hypernym1, Entity1, T3) :`

`2/a.4/a.3 np1 (T1, H1, H2, T4) :- f (T1, the, T2) , f (T2, H1, T3) , f (T3, H2, T4) .`

Με αυτόν τον πρώτο κανόνα (noun phrase) το πρόγραμμα αναγνωρίζει το υπερώνυμο (`Hypernym1`) και την οντότητα της πρότασης (`Entity1`). Στη συνέχεια με το `fronttoken : f (T3, v, T4)`, αναγνωρίζει την επόμενη λέξη (`v`) της πρότασης (`s`).

Η επόμενη εντολή του κανόνα αφορά το υπόλοιπο μέρος της πρότασης `s`, και την επεξεργασία της μέσω του επόμενου κανόνα (noun phrase) `np2 (T4, Hypernym2, Concept, Entity2, T5) :`

`2/a.4/a.5 np2 (T1, H1, H2, E1, T5) :- f (T1, the, T2) , f (T2, H1, T3) , f (T3, of, T4) ,
np1 (T4, H2, E1, T5) .`

Με αυτόν τον κανόνα το πρόγραμμα αναγνωρίζει ένα δεύτερο υπερώνυμο (`Hypernym2`), μια έννοια (`Concept`) και μια δεύτερη οντότητα (`Entity2`) αφού καλέσει εκ νέου τον κανόνα `2/a.4/a.3 np1 ()`. Στη συνέχεια αφού διαβαστούν από το πρόγραμμα οι δύο τελείες στο τέλος της πρότασης, το πρόγραμμα βεβαιώνει ότι η πρόταση `s`, την οποία αναλύει, πρέπει να περιέχει σαν τέταρτη λέξη `v` το ρήμα `is` (`v=is`), σαν πρώτο υπερώνυμο τη λέξη (`point`) και σαν δεύτερο υπερώνυμο τη λέξη (`centre`). Αυτό αποτελεί μια ασφαλιστική δικλείδα για το ποιες προτάσεις θα επεξεργαστεί το πρόγραμμα για την εξαγωγή γνώσης. Εφόσον γίνει η επιβεβαίωση και ικανοποιηθεί ο κανόνας δίνεται η εντολή στην εκτέλεση του κανόνα που ακολουθεί `dos_file_extraction1 (Concept, Entity2, Entity1) :`

`2/a.4/a.8 dos_file_extraction1 (CIRCLE, Name, Centre) :-
frontchar (Name, H1, T) , frontchar (T, H2, TT) ,
str_char (C, H1) , str_char (D, H2) ,
circle_construction (CIRCLE, C, D, TT, Centre) ,
oncircle_construction (C, D, TT) ,
synonyma_construction (C, D, TT, Centre) ,
endsline_construction (C, D, TT, Centre) .`

Ο κανόνας αυτός λαμβάνει ως μεταβλητές την έννοια (`Concept`) και τις δύο οντότητες (`Entity2, Entity1`) που προέκυψαν από την ανάλυση της πρότασης (`s`).

Κατά την εκτέλεση αυτού του κανόνα οι μεταβλητές ονομάζονται `Circle`, `Name` και `Centre` αντίστοιχα.

Με το ενσωματωμένο κατηγορημα `frontchar(, , ,)` αποκτούνται ένας-ένας οι χαρακτήρες-γράμματα της λέξης με την οποία καλούμε την δεύτερη οντότητα (`Entity2 - Name`):

```
2/a.4/a.8.1 frontchar(Name,H1,T),frontchar(T,H2,TT)
```

Με το επόμενο ενσωματωμένο κατηγορημα μετατροπής `str_char(,)`:

```
2/a.4/a.8.2 str_char(C,H1),str_char(D,H2)
```

μετατρέπει τους χαρακτήρες `H1` και `H2` αντίστοιχα σε `string : C` και `D`. Κατόπιν δίνεται η εντολή εκτέλεσης του κανόνα `circle_construction(Circle,C,D,TT,Centre)`:

```
2/a.4/a.8.3 circle_construction(Circle,C,D,B,A):-
    concat(C,D,CD),concat(CD,B,CDB),
    assert(isa_circle(Circle,CDB,A),circles),

    concat(D,B,DB),concat(DB,C,DBC),
    assert(isa_circle(Circle,DBC,A),circles),

    concat(B,C,BC),concat(BC,D,BCD),
    assert(isa_circle(Circle,BCD,A),circles),

    save("circles.dba",circles).
```

Με τον κανόνα αυτό χρησιμοποιείται το ενσωματωμένο κατηγορημα χειρισμού αλφαριθμητικών `concat`, το οποίο τις δύο διαφορετικές μεταβλητές τύπου `string (C)` και `(D)` τις κάνει μία μεταβλητή πάλι τύπου `string (CD)`. Ομοίως τις `(CD)` και `(B)` σε `(CDB)`. Κατόπιν ο κανόνας με το ενσωματωμένο κατηγορημα `assert` δηλώνει σε μια εσωτερική βάση δεδομένων τριών ορισμάτων την οποία καλεί `isa_circle`, το όνομα της έννοιας στην πρόταση `(S) : Circle` και τα ονόματα των δύο οντοτήτων μέσα στην πρόταση `(S) : CDB` και `A`. Η εσωτερική βάση `isa_circle` δηλώνεται σε μια ειδική βάση δεδομένων που έχει το όνομα `circles`.

Στη συνέχεια επαναλαμβάνεται δύο φορές η ίδια διαδικασία με το δεύτερο όρισμα να έχει διαφορετική σειρά διαδοχής των αλφαριθμητικών `(DBC)` και `(BCD)` αντίστοιχα. Με τον τρόπο αυτό δηλώνονται στη βάση δεδομένων όλοι οι δυνατοί τρόποι με τους οποίους μπορεί να γίνει ο συμβολισμός της έννοιας `Circle` που αναφέρεται στην πρόταση `(S)`, μαζί με το σημείο `(A)`.

Η επόμενη εντολή του κανόνα `save` δημιουργεί ένα dos αρχείο `circles.dba` στο οποίο υπάρχουν οι εγγραφές της βάσης δεδομένων. Μετά την ολοκλήρωση του

κανόνα 2/a.4/a.8.3 `circle_construction`, το πρόγραμμα επιστρέφει στον κανόνα από τον οποίο κλήθηκε 2/a.4/a.8 `dos_file_extraction` ο οποίος δίνει εντολή εκτέλεσης του κανόνα `oncircle_construction(C,D,TT)` :

```
2/a.4/a.8.4 oncircle_construction(C,D,B) :-
    concat(C,D,CD), concat(CD,B,CDB),
    assert(isa_oncircle(oncircle,CDB,C), oncircles),
    assert(isa_oncircle(oncircle,CDB,D), oncircles),
    assert(isa_oncircle(oncircle,CDB,B), oncircles),

    concat(D,B,DB), concat(DB,C,DBC),
    assert(isa_oncircle(oncircle,DBC,C), oncircles),
    assert(isa_oncircle(oncircle,DBC,D), oncircles),
    assert(isa_oncircle(oncircle,DBC,B), oncircles),

    concat(B,C,BC), concat(BC,D,BCD),
    assert(isa_oncircle(oncircle,BCD,C), oncircles),
    assert(isa_oncircle(oncircle,BCD,D), oncircles),
    assert(isa_oncircle(oncircle,BCD,B), oncircles),

    save("oncircle.db", oncircles).
```

Με τον κανόνα αυτό χρησιμοποιείται το ενσωματωμένο κατηγορημα χειρισμού αλφαριθμητικών `concat` το οποίο τις δύο διαφορετικές μεταβλητές τύπου `string` (`C`) και (`D`) τις κάνει μία μεταβλητή πάλι τύπου `string` (`CD`). Ομοίως τις (`CD`) και (`B`) σε (`CDB`). Κατόπιν ο κανόνας με το ενσωματωμένο κατηγορημα `assert` δηλώνει σε μια εσωτερική βάση δεδομένων τριών ορισμάτων την οποία καλεί `isa_oncircle`, σαν πρώτο όρισμα έχει το όνομα `oncircle`, σαν δεύτερο και τρίτο όρισμα τα ονόματα των δύο οντοτήτων : `CDB` και `C`. Η εσωτερική βάση `isa_oncircle` δηλώνεται σε μια ειδική βάση δεδομένων που έχει το όνομα `oncircles`. Στην αυτή βάση δεδομένων δηλώνονται στη συνέχεια δύο νέες εγγραφές οι οποίες διαφέρουν από την πρώτη μόνο ως προς το τρίτο όρισμα, το οποίο είναι (`D`) και (`B`) αντιστοίχα. Με την διαδικασία αυτή εξάγεται γνώση με την οποία δηλώνεται ότι τα `C,D` και `B` ευρίσκονται επάνω στην περιφέρεια του κύκλου `CDB`.

Η διαδικασία επαναλαμβάνεται με το ενσωματωμένο κατηγορημα χειρισμού αλφαριθμητικών `concat` να κάνει τις δύο διαφορετικές μεταβλητές τύπου `string` (`D`) και (`B`) σε μια μεταβλητή τύπου `string` (`DB`) και τις (`DB`) και (`C`) σε (`DBC`). Στη συνέχεια δηλώνονται, όπως και προηγουμένως, τρεις νέες εγγραφές στη βάση δεδομένων με τα δεύτερα ορίσματά τους να έχουν την τιμή (`DBC`) και τα τρίτα ορίσματά τους να είναι τα `C,D` και `B` αντιστοίχα.

Και πάλι επαναλαμβάνεται η ίδια διαδικασία κατά την οποία οι δύο διαφορετικές μεταβλητές τύπου `string` (`B`) και (`C`) γίνονται μια μεταβλητή τύπου `string` (`BC`) και οι

(BC) και (D) σε (BCD). Ομοίως ακολουθούν οι δηλώσεις των τριών εγγραφών στην εσωτερική βάση με τα δεύτερα ορίσματά τους να έχουν την τιμή (BCD) και τα τρίτα ορίσματά τους να είναι τα C,D και B αντίστοιχα. Με αυτές τις δηλώσεις το πρόγραμμα γνωρίζει ότι τα σημεία C,D και B βρίσκονται πάνω στη περιφέρεια ενός κύκλου όταν αυτός συμβολίζεται με τρία σημεία σε οποιαδήποτε σειρά.

Η επόμενη εντολή του κανόνα `save` δημιουργεί ένα dos αρχείο `oncircle.db` στο οποίο υπάρχουν οι εγγραφές της βάσης δεδομένων και ολοκληρώνει τον κανόνα `2/a.4/a.8.4 oncircle_construction`. Το πρόγραμμα επιστρέφει στον κανόνα από τον οποίο κλήθηκε `2/a.4/a.8 dos_file_extraction` και συνεχίζει με τον κανόνα `synwnyma_construction(C,D,TT,CEntre)` :

```
2/a.4/a.8.5 synwnyma_construction(C,D,B,A) :-
    concat(C,A,CA), concat(D,A,DA), concat(B,A,BA),
    concat(A,C,AC), concat(A,D,AD), concat(A,B,AB),
    assert(isa_synwnyma(synwnyma,CA,AC), synwnyms),
    assert(isa_synwnyma(synwnyma,AC,CA), synwnyms),
    assert(isa_synwnyma(synwnyma,DA,AD), synwnyms),
    assert(isa_synwnyma(synwnyma,AD,DA), synwnyms),
    assert(isa_synwnyma(synwnyma,BA,AB), synwnyms),
    assert(isa_synwnyma(synwnyma,AB,BA), synwnyms),
    save("synwnyms.db", synwnyms).
```

Στον κανόνα αυτό χρησιμοποιείται το ενσωματωμένο κατηγορήμα χειρισμού αλφαριθμητικών `concat`, το οποίο τις δύο διαφορετικές μεταβλητές τύπου `string` (C) και (A) τις κάνει μία μεταβλητή πάλι τύπου `string` (CA). Ομοίως τις (D) και (A) σε (DA), τις (B) και (A) σε (BA), τις (A) και (C) σε (AC), τις (A) και (D) σε (AD) και τέλος τις (A) και (B) σε (AB). Κατόπιν ο κανόνας με το ενσωματωμένο κατηγορήμα `assert` δηλώνει σε μια εσωτερική βάση δεδομένων τριών ορισμάτων την οποία καλεί `isa_synwnyma`, σαν πρώτο όρισμα το όνομα `synwnyma` και σαν δεύτερο και τρίτο όρισμα, τα ονόματα των δύο οντοτήτων που προέκυψαν από το τρίτο όρισμα του κατηγορήματος `concat`, τα δύο αυτά ορίσματα διαφέρουν μόνο ως προς τη σειρά των γραμμάτων με τα οποία συμβολίζονται. Η εσωτερική βάση `isa_synwnyma` δηλώνεται σε μια ειδική βάση δεδομένων που έχει το όνομα `synwnyms`. Με τον τρόπο αυτό δηλώνονται τα συνώνυμα όλων των ευθυγράμμων τμημάτων που προκύπτουν από τις μεταβλητές των σημείων που είναι τα δύο πρώτα ορίσματα του κατηγορήματος `concat`. Τα ευθύγραμμα τμήματα της εσωτερικής βάσης που προέρχονται από την ανάλυση της συγκεκριμένης πρότασης είναι ακτίνες του κύκλου (CDB, A).

Η εντολή `save` δημιουργεί ένα dos αρχείο `synonyms.dba` στο οποίο υπάρχουν οι εγγραφές της εσωτερικής βάσης δεδομένων. Μετά την ολοκλήρωση του κανόνα `2/a.4/a.8.5 synonyma_construction`, το πρόγραμμα επιστρέφει στον κανόνα από τον οποίο κλήθηκε `2/a.4/a.8 dos_file_extraction` δίνεται εντολή εκτέλεσης του κανόνα `endsline_construction(C,D,TT,CEntre)` :

```
2/a.4/a.8.6 endsline_construction(C,D,B,A) :-
    concat(C,A,CA), concat(D,A,DA), concat(B,A,BA),
    concat(A,C,AC), concat(A,D,AD), concat(A,B,AB),
    assert(isa_endsline(endsline,CA,C,A),endslines),
    assert(isa_endsline(endsline,AC,A,C),endslines),
    assert(isa_endsline(endsline,DA,D,A),endslines),
    assert(isa_endsline(endsline,AD,A,D),endslines),
    assert(isa_endsline(endsline,BA,B,A),endslines),
    assert(isa_endsline(endsline,AB,A,B),endslines),
    save("endsline.dba",endslines).
```

Το ενσωματωμένο κατηγορημα χειρισμού αλφαριθμητικών `concat` χειρίζεται τις ίδιες μεταβλητές με τον ίδιο τρόπο όπως στον κανόνα `2/a.4/a.8.5 synonyma_construction(C,D,B,A)`. Στη συνέχεια ο κανόνας με το ενσωματωμένο κατηγορημα `assert` δηλώνει σε μια εσωτερική βάση δεδομένων εγγραφές `isa_endsline` με τέσσερα ορίσματα. Σαν πρώτο όρισμα έχει το όνομα `endsline`, σαν δεύτερο ένα από τα ονόματα των οντοτήτων που προέκυψαν από το τρίτο όρισμα του κατηγορήματος `concat`, και σαν τρίτο και τέταρτο όρισμα, το πρώτο και δεύτερο αντίστοιχα από το ίδιο κατηγορημα `concat`. Η εσωτερική βάση `isa_endsline` δηλώνεται σε μια ειδική βάση δεδομένων που έχει το όνομα `endslines`. Με τον τρόπο αυτό δηλώνονται τα άκρα όλων των ευθυγράμμων τμημάτων και των συνωνύμων τους.

Και πάλι η εντολή `save` δημιουργεί ένα dos αρχείο `endsline.dba` στο οποίο υπάρχουν οι εγγραφές της εσωτερικής βάσης δεδομένων. Μετά την ολοκλήρωση του κανόνα `2/a.4/a.8.6 endsline_construction`, το πρόγραμμα επιστρέφει στον κανόνα, από τον οποίο κλήθηκε `2/a.4/a.8 dos_file_extraction`. Με την ικανοποίηση αυτού του κανόνα το πρόγραμμα επιστρέφει στον κανόνα `2/a.4/a.extract_from(S)`, που τον κάλεσε. Η ικανοποίηση και αυτού του κανόνα επιστρέφει το πρόγραμμα στον κανόνα `2/a extraction_knowledge` ο οποίος δίνει εντολή στην εκτέλεση του κανόνα `loop_extraction_knowledge` :

```
2/a.5 loop_extraction_knowledge:-
    readint(_),readln(S),
    extract_from(S),
    loop_extraction_knowledge.
```

Με τον κανόνα αυτό το πρόγραμμα διαβάζει την επόμενη πρόταση του κειμένου, όπως αυτό έχει αποθηκευτεί στη βάση δεδομένων και ως μεταβλητή πλέον του κανόνα `extract_from(S)` αναλύεται, όπως περιγράφηκε προηγουμένως. Μετά την ανάλυση της πρότασης το πρόγραμμα επιστρέφει πάλι εδώ και συνεχίζει με την επανάληψη του κανόνα `loop_extraction_knowledge` έως ότου εξαντληθούν όλες οι προτάσεις του κειμένου.

Με την ικανοποίηση και αυτού του κανόνα το πρόγραμμα επιστρέφει στον κανόνα `2/a extraction_knowledge` ο οποίος ανακατευθύνει το πρόγραμμα στο `keyboard : readdevice(keyboard)`. Η ικανοποίηση του κανόνα `extraction_knowledge` επιστρέφει το πρόγραμμα στον επόμενο στόχο (goal) του προγράμματος ο οποίος είναι `3 phrases_construction`.

7.4 Υποσύστημα απάντησης ερωτήσεων

Το τμήμα αυτό του προγράμματος απαντά σε ερωτήσεις φυσικής γλώσσας, οι οποίες αφορούν στο κείμενο της απόδειξης των προτάσεων των Στοιχείων του Ευκλείδη. Το πρόγραμμα κάνει μια οντολογική ανάλυση των ερωτήσεων, βρίσκει στο κείμενο των Στοιχείων σε ποιο σημείο απαντώνται οι οντότητες της ερώτησης και, επειδή όλες οι προτάσεις του κειμένου είναι γραμμένες κατά αιτιώδη τρόπο, απαντά με κείμενο από την απόδειξη της πρότασης των Στοιχείων. Για παράδειγμα έστω η πρώτη από τις προτάσεις που αποδεικνύονται στο πρώτο βιβλίο των Στοιχείων του Ευκλείδη : «On a given finite straight line to construct an equilateral triangle». Με την ερώτηση «why ca is equal to bc» που αφορά τη σχέση δύο ευθυγράμμων τμημάτων που υπάρχει μέσα στο κείμενο της απόδειξης, ζητάμε από το πρόγραμμα να μας δώσει την απάντησή της μέσα από το ίδιο το κείμενο. Η απάντηση συνδέεται με το τμήμα της ερώτησης που ακολουθεί το «why» με τρόπο αιτιώδη, ο οποίος άλλοτε είναι άμεσα αναγνωρίσιμος και άλλοτε υπονοούμενος μέσω της προαπαιτούμενης γνώσης των ορισμών, των αξιωμάτων και των κοινών εννοιών της Ευκλείδειας Γεωμετρίας.

Στο πρόγραμμα υπάρχει μια λίστα ερωτήσεων οι οποίες μπορεί να τεθούν από τον χρήστη. Οι ερωτήσεις αφορούν στο κείμενο των αποδείξεων των προτάσεων του Ευκλείδη. Κάθε μία από τις ερωτήσεις σαν όρισμα του κατηγορήματος (predicate) question, αποτελεί και ένα διαφορετικό γεγονός (fact). Το πρόγραμμα στην

προσπάθειά του να δώσει την απάντηση, επιλέγει διαδοχικά τις στρατηγικές, μια από τις οποίες θα επιτύχει την αναζήτηση της απάντησης για τη συγκεκριμένη ερώτηση :

```
strategy (Synonym, LSide, RSide, Inversion, Relstrategy, Pk, strategy1)
```

Η επιλογή μιας στρατηγικής καθορίζει τον τρόπο με τον οποίο το πρόγραμμα θα διεκπεραιώσει τη διαδικασία ανάλογα με τον τύπο της ερώτησης. Άλλη διαδικασία θα ακολουθηθεί για μια σχέση ισότητας, της οποίας η αιτιολόγηση αναφέρεται ρητά στο κείμενο και άλλη, όταν υπονοείται μέσω κάποιας κοινής έννοιας. Οι δυνατές τιμές των ορισμάτων του κατηγορήματος `strategy` έχουν ως εξής :

```
Synonym= nosynonymes, mesynonymes
LSide= left, noleft
RSide= right, noright
Inversion= noinversion, meinversion
Relstrategy= equality
Pk= pk, nopk
Strategy= strategy1, strategy2, , , ,
```

Η ερώτηση που επιλέχτηκε να απαντηθεί από το πρόγραμμα υποβάλλεται με την μορφή : `q1:-question("why ac is equal to ab")` και για την εκτέλεση της επιλογής του κανόνα `1.1/a question(Question)` επιλέγεται η στρατηγική `strategy1`. Η συγκεκριμένη στρατηγική χρησιμοποιεί τα ευθύγραμμα τμήματα της ερώτησης, ως έχουν, και με τη σειρά που εμφανίζονται στη σχέση που τα συνδέει. Άλλες στρατηγικές χρησιμοποιούν συνώνυμα των ευθυγράμμων τμημάτων με ή χωρίς εναλλαγή των θέσεών τους ως προς τη σχέση που τα συνδέει. Επίσης για την απάντηση της ερώτησης στη συγκεκριμένη στρατηγική δεν γίνεται χρήση πρότερης γνώσης.

Η επόμενη εντολή του κανόνα `assert()` γράφει τη στρατηγική που χρησιμοποιείται, σε μια εσωτερική βάση δεδομένων, ώστε το πρόγραμμα να έχει τη δυνατότητα να γνωρίζει συνεχώς την ακολουθούμενη στρατηγική και τα ορίσματά της τα οποία χαρακτηρίζουν τις επιλογές του προγράμματος για την ολοκλήρωση της διαδικασίας.

```
1.1/a.3      assert (oldstrategy (strategy1)
```

Ο κανόνας : 1 `q1` δίνει εντολή στην εκτέλεση του κανόνα `question()` με όρισμα το `string "why ac is equal to ab":`

```
1.1/a question(Question):-retractall(cn1(_)),
                           extract_question(Question),
```

```

strategy(Synonym, LSide, RSide, Inversion,
         Relstrategy, Pk, strategy1),
retractall(oldstrategy(_)),
assert(oldstrategy(strategy1)),
nl, write("TRYING STRATEGY : strategy1"),
nl, write(" with ", Synonym, " ", LSide, "
         ", RSide, " ", Inversion, " ", Relstrategy, "
         ", Pk),
process_answer(strategy1).

```

```

1.1/b question(Question):-extract_question(Question),
oldstrategy(OldS),
next(OldS, Strategy),
questionloop(Strategy).

```

```

1.1/c question(Question):-nl, write("impossible to answer
", Question), nl.

```

Με την πρώτη επιλογή του κανόνα δίνεται η εντολή εκτέλεσης στο 1.1/a.1

```
extract_question(Question):
```

```

1.1/a.1      extract_question(Question):-f(Question, why, Rest1),
                                                f(Rest1, LEntity, Rest2),
                                                f(Rest2, is, Rest3),
                                                f(Rest3, Relation, Rest4),
                                                f(Rest4, Prep, Rest5),
                                                f(Rest5, REntity, ""),
        relation(Relation), preposition(Prep),
retractall(leftentityofquestion(_)),
assert(leftentityofquestion(LEntity)),
retractall(relationofquestion(_)),
assert(relationofquestion(Relation)),
retractall(rightentityofquestion(_)),
assert(rightentityofquestion(REntity)).

```

Με αυτόν τον κανόνα γίνεται ανάλυση της ερώτησης μέσω του κατηγορήματος fronttoken, το οποίο έχει δηλωθεί για το πρόγραμμα με το γράμμα f και έχει σαν πρώτο όρισμα το αλφαριθμητικό string της ερώτησης, σαν δεύτερο όρισμα την πρώτη λέξη του αλφαριθμητικού και σαν τρίτο όρισμα το υπόλοιπο αλφαριθμητικό. Με τον τρόπο αυτό και με τη διαδοχική εκτέλεση του κατηγορήματος fronttoken ορίζουμε τις οντότητες (LEntity, REntity) της ερώτησης και τη σχέση που τις συνδέει (Relation). Στον ίδιο κανόνα τις γράφουμε σε εσωτερικές βάσεις δεδομένων μέσω των εντολών assert().

Μετά την εγγραφή στις εσωτερικές βάσεις το πρόγραμμα επιστρέφει στο σημείο που κλήθηκε η εντολή αυτού του κανόνα (1.1/a question(Question)) και γίνεται η επιλογή της στρατηγικής, όπως περιγράφηκε πιο πάνω. Ακολούθως ο επόμενος κανόνας ξεκινάει την προσπάθεια εφαρμογής της επιλεγμένης στρατηγικής για την επιτυχή απάντηση της ερώτησης :

1.1/a.4/a `process_answer(strategy1)`

Εάν κατά τη διαδικασία της επεξεργασίας του προγράμματος δεν ικανοποιηθεί ένας κανόνας, τότε αποτυγχάνει η διαδικασία που ακολουθείται και περιγράφεται από τα ορίσματα της στρατηγικής και το πρόγραμμα δοκιμάζει μια άλλη, την αμέσως επόμενη, η οποία περιγράφεται από το επόμενο κατηγορημα από την ομάδα των στρατηγικών. Στην περίπτωση αυτή για να επιτευχθεί η αλλαγή της διαδικασίας μέσω της επόμενης στρατηγικής, εκτελείται η δεύτερη επιλογή του κανόνα `question(Question)`:

```
1.1/b question(Question) :-extract_question(Question),
                                oldstrategy(OldS),
                                next(OldS,Strategy),
                                questionloop(Strategy).
```

Η επιλογή αυτού του κανόνα αφού επαναλάβει την ανάλυση της ερώτησης μέσω του κατηγορηματος `fronttoken`, και ανασύρει την εγγραφή της εσωτερικής βάσης `oldstrategy()` που έχει το όνομα της προηγούμενης στρατηγικής που απέτυχε, δίνει την εντολή εκτέλεσης ενός εκ των `next(OldS,Strategy)` κατηγορημάτων. Σε αυτό το κατηγορημα το πρώτο όρισμα δηλώνει την παλιά αποτυχημένη στρατηγική και το δεύτερο την επόμενη που θα ακολουθηθεί σύμφωνα με τη σειρά που έχει δηλωθεί σε αυτή την ομάδα των κατηγορημάτων. Στη συνέχεια εκτελείται ο αναδρομικός κανόνας `questionloop(Strategy)` ο οποίος έχει σαν όρισμα τη νέα στρατηγική :

```
1.1/b.2/a questionloop(Strategy) :-strategy(Synonym,LSide,RSide,
                                           Inversion,Relstrategy,Pk,Strategy),
                                           retractall(oldstrategy(_)),
                                           assert(oldstrategy(Strategy)),
                                           process_answer(Strategy).
```

```
1.1/b.2/b questionloop(_):-oldstrategy(OLDS),
                    next(OLDS,NewStrategy),
                    questionloop(NewStrategy).
```

Η πρώτη επιλογή αυτού του κανόνα εκτελεί εκείνο από τα κατηγορηματα του συνόλου των στρατηγικών, το οποίο έχει την ίδια τιμή στη μεταβλητή `Strategy`. Με τον τρόπο αυτό ορίζονται και οι τιμές των υπολοίπων ορισμάτων της συγκεκριμένης στρατηγικής οι οποίες θα καθορίσουν και την περαιτέρω διαδικασία για την απάντηση της ερώτησης. Στη συνέχεια γράφει στην εσωτερική βάση το όνομα της τρέχουσας στρατηγικής και δίνει εντολή εκτέλεσης μιας εκ των υπολοίπων επιλογών του κανόνα :

1.1/a.4 process_answer(Strategy) .

Εάν και πάλι κατά τη διαδικασία της επεξεργασίας του προγράμματος δεν ικανοποιηθεί τουλάχιστον ένας κανόνας, τότε αποτυγχάνει εκ νέου η διαδικασία που περιγράφεται από τα ορίσματα της στρατηγικής και το πρόγραμμα δοκιμάζει την αμέσως επόμενη που περιγράφεται από το επόμενο κατηγορημα της ομάδας των στρατηγικών. Η αλλαγή της στρατηγικής λόγω της μη ικανοποίησης της πρώτης επιλογής του κανόνα question loop(Strategy) επιτυγχάνεται με την δεύτερη επιλογή του :

```
1.1/b.2/b question loop ( _ ) :- old strategy ( OLDS ) ,  
                                next ( OLDS , NewStrategy ) ,  
                                questionloop ( NewStrategy ) .
```

Με την επιλογή αυτή του κανόνα γίνεται ανάκληση της τρέχουσας στρατηγικής (oldstrategy(OLDS)) και με το δεύτερο όρισμα NewStrategy του κατηγορηματος next(OLDS,NewStrategy) ευρίσκεται η επόμενη στρατηγική, η οποία ενεργοποιείται μέσω του αναδρομικού κανόνα questionloop(NewStrategy), ο οποίος επαναλαμβάνει τα βήματα του 1.1/b.2/a questionloop(Strategy) μέχρι την εντολή εκτέλεσης μιας εκ των επιλογών του κανόνα process_answer(Strategy) .

Για την ερώτηση q1 θα γίνει χρήση των ορισμάτων της strategy1.

```
1.1/a.4/a process_answer(strategy1) :- strategy(Synonym,_,_,  
                                                Inversion,_,Pk,strategy1) ,  
                                                Synonym="nosynonymes",  
                                                Inversion="noinversion",  
                                                Pk="nopk",  
                                                compose_question(Q,Synonym,Inversion) ,  
                                                same_relation,  
                                                previous_phrase_in_text(Q,PrevQ) ,  
                                                pk(1,cn1,K1),PrevQ<>K1,  
                                                pk(3,cn3,K3),PrevQ<>K3,  
                                                answer(Q,PrevQ) .
```

Ο κανόνας αυτός αφού αναφερθεί στη στρατηγική που θα ακολουθήσει, ώστε να έχει στη διάθεσή του τις τιμές των ορισμάτων της, τα επιβεβαιώνει και δίνει εντολή στην εκτέλεση του κανόνα compose_question() όπου η πρώτη του επιλογή είναι :

```
1.1/a.4/a.1/a compose_question(CQ,S,I) :-  
                                                leftentityofquestion(LEntity) ,  
                                                relationofquestion(Relation) ,  
                                                rightentityofquestion(REntity) ,  
                                                c(" ",LEntity,BLE) , c(BLE," ",T1) , c("is"," ",Isb) ,  
                                                c(Isb,Relation,IsbR) , c(IsbR," ",IsbRb) , c(IsbRb,"to",T2) ,  
                                                c(" ",REntity,BRE) , c(BRE," ",T3) , c(T1,T2,T12) ,  
                                                c(T12,T3,CQ) ,
```

```

retractall(leftentityofquestiontransferred(_)),
assert(leftentityofquestiontransferred(LEntity)),
retractall(rightentityofquestiontransferred(_)),
assert(rightentityofquestiontransferred(REntity)).

```

Υπάρχουν οκτώ επιλογές αυτού του κανόνα ώστε να καλύπτει όλους τους δυνατούς τρόπους μετασχηματισμών (γραφή και σειρά εμφάνισης των ευθυγράμμων τμημάτων) της ερώτησης. Οι τρεις πρώτες εντολές του κανόνα αφορούν στην ανάκληση των οντοτήτων της ερώτησης από εσωτερικές βάσεις. Στη συνέχεια συνθέτει την ερώτηση μέσω της διαδοχικής χρήσης του κατηγορήματος `concat(X, Y, Z)`, στο οποίο το τρίτο όρισμα `Z` αντικαθίσταται από τα δύο πρώτα `X, Y` σε συνεχόμενη γραφή δηλαδή `XY`. Στο πρόγραμμα το κατηγορήμα αυτό συμβολίζεται με `c(X, Y, Z) :- concat(X, Y, Z)`. Με τη μεταβλητή `CQ` στην τελευταία χρήση του κατηγορήματος λαμβάνουμε μια μορφή του τμήματος της ερώτησης το οποίο θα προσπαθήσουμε να ταυτίσουμε στο κείμενο της απόδειξης της πρότασης του Ευκλείδη. Στη συνέχεια ο κανόνας γράφει σε εσωτερικές βάσεις τις θέσεις των οντοτήτων σύμφωνα με την ανασύνθεση που έκανε στην ερώτηση. Μετά την ολοκλήρωση της διαδικασίας αυτού του κανόνα, το πρόγραμμα επιστρέφει στο σημείο που είχε κληθεί για να συνεχίσει την εκτέλεσή του. Αν κατά τη διάρκεια της εκτέλεσης δεν ικανοποιηθεί τουλάχιστον ένας κανόνας λόγω της θέσης ή της γραφής των οντοτήτων, η διαδικασία επιστρέφει για να επιλέξει τη σύνθεση ενός άλλου μετασχηματισμού (τρόπο γραφής και σειρά εμφάνισης των ευθυγράμμων τμημάτων) της ερώτησης μέσω διαφορετικής επιλογής κανόνα `compose_question()`.

Το πρόγραμμα μετά την ικανοποίηση του κανόνα `compose_question()` επιστρέφει στον κανόνα `process_answer(Strategy)` από όπου εκλήθη για να συνεχιστεί η διαδικασία με τον κανόνα `same_relation` :

1.1/a.4/a.2 `same_relation`:-

```

leftentityofquestiontransferred(LEntity),
rightentityofquestiontransferred(REntity),
number_leftpart_verb(LEntity,N),
find_rightpart_verbless(N,Rightpartnextverb),
f(Rightpartnextverb,equal,T1),
f(T1,to,T2),
f(T2,Rightentity," "),
Rightentity=REntity.

```

Με την εκτέλεση αυτού του κανόνα το πρόγραμμα λαμβάνει από μια εσωτερική βάση το αριστερό ως προς το ρήμα μέρος της ερώτησης, όπως αυτή μετασχηματίστηκε (`leftentityofquestiontransferred(LEntity)`), αναζητά και ταυτίζει το

αντίστοιχο που υπάρχει στην απόδειξη του κειμένου και βρίσκει μέσω του κανόνα `number_leftpart_verb(LEntity,N)`, τον αριθμό με τον οποίο χαρακτηρίζεται από την αρίθμηση του αρχείου των προτάσεων του κειμένου, οι οποίες έχουν παραχθεί με το πρόγραμμα :

```
1.1/a.4/a.2.1/a    number_leftpart_verb(LEntity,N) :-
                    openread(verbless,"verbless.pro"),
                    readdevice(verbless),
                    c(" ",LEntity,BL),
                    match_leftpart_verbless(BL),
                    numberleftpart(N),
                    closefile(verbless),
                    readdevice(keyboard).
```

Στην αρχή αφού ο κανόνας ανοίξει το αρχείο "verbless.pro" και ανακατευθύνει το πρόγραμμα προς αυτό, προσθέτει με το κατηγορήμα : `concat`, μια κενή θέση αριστερά της οντότητας, επειδή έτσι είναι γραμμένη στο αρχείο, ώστε να μπορέσει να κάνει την ταυτοποίηση. Στη συνέχεια δίνει εντολή να εκτελεστεί ο αναδρομικός κανόνας `match_leftpart_verbless(BL)` :

```
1.1/a.4/a.2.1/a.1/a    match_leftpart_verbless(BL) :-
                        readint(N),
                        retractall(numberleftpart(_)),
                        assert(numberleftpart(N)),
                        N<>0,
                        readln(Leftentity),
                        retractall(leftpartverbless(_)),
                        assert(leftpartverbless(Leftentity)),
                        readln(_),readln(_),
                        leftpartverbless(Leftentity),
                        BL<>Leftentity,
                        match_leftpart_verbless(BL).
```

Με την εντολή `readint(N)` διαβάζει την πρώτη γραμμή του αρχείου που έχει τον αριθμό που αντιστοιχεί στην πρόταση του κειμένου τον οποίο και γράφει σε εσωτερική βάση με την εντολή : `assert(numberleftpart(N))`. Εφόσον ο αριθμός είναι διάφορος του μηδενός, (το μηδέν γράφεται στο τέλος του αρχείου για τη δημιουργία συνθήκης τερματισμού) με την εντολή `readln(Leftentity)` διαβάζει τη δεύτερη γραμμή του αρχείου που έχει το τμήμα της πρότασης που βρίσκεται αριστερά του ρήματος και το γράφει σε εσωτερική βάση με την εντολή: `assert(leftpartverbless(Leftentity))`. Με την εκτέλεση της εντολής `readln(_)` δύο φορές αποφεύγεται το διάβασμα των δύο επόμενων γραμμών του αρχείου που έχουν η μεν πρώτη το ρήμα, η δε δεύτερη το τμήμα της πρότασης του κειμένου που βρίσκεται δεξιά του ρήματος, με σκοπό την επόμενη φορά, αν

ξαναδιαβάσει, να βρει τον αριθμό που αντιστοιχεί στην επόμενη πρόταση. Στη συνέχεια, κάθε φορά που η τιμή της μεταβλητής του κανόνα (το αριστερό ως προς το ρήμα μέρος της ερώτησης), είναι διαφορετική από το περιεχόμενο του κειμένου που διαβάστηκε, όταν έγινε η χρήση της εντολής `realm(Left entity)`, τότε επαναλαμβάνεται η εκτέλεση του κανόνα, αν όμως αυτά τα δύο είναι ίσα εκτελείται η επόμενη επιλογή του κανόνα :

```
1.1/a.4/a.2.1/a.1/b match_leftpart_verbless(_):-numberleftpart(N),
                                     N<>0,
                                     closefile(verbless),
                                     readdevice(keyboard).
```

Με την επιλογή αυτή του κανόνα και αφού ήδη έχει προηγηθεί η εύρεση στο κείμενο της απόδειξης, του αριστερού ως προς το ρήμα μέρους της ερώτησης, το πρόγραμμα κλείνει το αρχείο και ικανοποιεί τον κανόνα. Η επόμενη επιλογή του κανόνα εκτελείται όταν το πρόγραμμα διαβάζει την εγγραφή του αρχείου `verbless` που χαρακτηρίζεται με τον αριθμό 0.

```
1.1/a.4/a.2.1/a.1/c match_leftpart_verbless(_):-
                                     numberleftpart(N),N=0,
                                     closefile(verbless),
                                     readdevice(keyboard),fail.
```

Και σε αυτήν την περίπτωση το πρόγραμμα κλείνει το αρχείο και ο κανόνας επιστρέφει στο σημείο, από το οποίο είχε κληθεί : `1.1/a.4/a.2.1 number_leftpart_verb(LEntity,N)` για να εκτελεστεί η επόμενη εντολή του `numberleftpart(N)`, η οποία επανακαλεί από την εσωτερική βάση τον αριθμό `N` και δίνει την τιμή του στο όρισμα του κανόνα. Στη συνέχεια το πρόγραμμα επιστρέφει στον κανόνα `same_relation` για να συνεχίσει με την εκτέλεση του `find_rightpart_verbless(N, Rightpartnextverb)` :

```
1.1/a.4/a.2.2 find_rightpart_verbless(N,Rightpartnextverb):-
                                     openread(verbless,"verbless.pro"),
                                     readdevice(verbless),
                                     rightpart_in_verbless(N),
                                     rightpartverbless(Rightpartnextverb),
                                     closefile(verbless),
                                     readdevice(keyboard).
```

Με την εκτέλεση αυτού του κανόνα ανοίγει το αρχείο `verbless.pro` και ανακατευθύνεται το πρόγραμμα προς αυτό. Κατόπιν δίνεται εντολή εκτέλεσης του αναδρομικού κανόνα `rightpart_in_verbless(N)` :

```
1.1/a.4/a.2.2.1/a rightpart_in_verbless(N):-
```

```

        not (eof (verbless)) ,
        readint (Num) ,
        readln ( _ ) , readln ( _ ) ,
        readln (Rightpart) ,
        retractall (rightpartverbless ( _ ) ) ,
        assert (rightpartverbless (Rightpart)) ,
        N<>Num, ! ,
rightpart_in_verbless (N) .

```

1.1/a.4/a.2.2.1/b rightpart_in_verbless (_) .

Στον κανόνα αυτό, εφόσον το πρόγραμμα δεν βρίσκεται στο τέλος του αρχείου που άνοιξε, διαβάζει στην πρώτη του γραμμή τον αριθμό, δεν διαβάζει τη δεύτερη γραμμή του αρχείου, που έχει το τμήμα της πρότασης του κειμένου που βρίσκεται αριστερά του ρήματος, ούτε την επόμενη που έχει το ρήμα της πρότασης. Διαβάζει την αμέσως επόμενη γραμμή που έχει το τμήμα της πρότασης του κειμένου που βρίσκεται δεξιά του ρήματος (`readln(Rightpart)`) και το γράφει σε μια εσωτερική βάση : `assert(rightpartverbless (Rightpart))`. Μετά ελέγχει τον αριθμό που διάβασε με τον αριθμό του ορίσματος στον κανόνα. Όσο αυτοί οι δύο αριθμοί είναι διαφορετικοί ο κανόνας θα επαναλαμβάνεται. Όταν ο αριθμός που διάβασε είναι ίδιος με τον αριθμό του ορίσματος, το πρόγραμμα σαν συνθήκη τερματισμού του κανόνα εκτελεί την δεύτερη επιλογή του : 1.1/a.4/a.2.2.1/b με την οποία ικανοποιείται και επιστρέφει στον `find_rightpart_verbless()`, ο οποίος κλείνει το αρχείο για να επιστρέψει το πρόγραμμα στον κανόνα `same_relation` για να συνεχίσει με την εκτέλεση των διαδοχικών `fronttokens`, ώστε να απομονώσει την δεξιά ως προς το ρήμα οντότητα `Rightentity`. Κατόπιν ικανοποιεί τον κανόνα αν εξισώνεται η τιμή της δεξιάς ως προς το ρήμα οντότητας της ερώτησης, με την δεξιά ως προς το ρήμα οντότητας της πρότασης του κειμένου. Αυτό σημαίνει ότι βρέθηκε στο κείμενο η ίδια πρόταση με αυτήν της ερώτησης.

Το πρόγραμμα μετά την ικανοποίηση του κανόνα `same_relation` επιστρέφει στον κανόνα `process_answer(Strategy)`, από όπου εκλήθη για να συνεχιστεί η διαδικασία με εντολή στον κανόνα `previous_phrase_in_text(Q,PrevQ)` :

```

1.1/a.4/a.3 previous_phrase_in_text (Q,PrevQ) :-
        find_number_previous_phrase (Q,NprevQ) ,
        findpreviousphrase (PrevQ,NprevQ) .

```

Ο κανόνας καλεί την εκτέλεση του `find_number_previous_phrase(Q,NprevQ)` :

```

1.1/a.4/a.3.1 find_number_previous_phrase (Q,NprevQ) :-
        openread (phrase, "phrase.pro") ,
        readdevice (phrase) ,

```



```

number_previous_phrase(Q),
numberpreviousphrase(NprevQ),
closefile(phrase),
readdevice(keyboard).

```

Με την εντολή εκτέλεσης αυτού του κανόνα ανοίγει το αρχείο "phrase.pro", ανακατευθύνεται το πρόγραμμα προς το αρχείο για να το διαβάσει και στη συνέχεια δίνεται εντολή να εκτελεστεί ο κανόνας `number_previous_phrase(Q)` :

```

1.1/a.4/a.3.1.1/a number_previous_phrase(Q) :-
    not (eof(phrase)),
    readint(NprevQ),
    readln(Sentence),
    Q<>Sentence,
    retractall(numberpreviousphrase(_)),
    assert(numberpreviousphrase(NprevQ)),
    number_previous_phrase(Q).

```

```

1.1/a.4/a.3.1.1/b number_previous_phrase(_).

```

Με τον αναδρομικό αυτό κανόνα και εφόσον η διαδικασία δεν έχει φτάσει στο τέλος του αρχείου, το πρόγραμμα διαβάζει στην πρώτη γραμμή τον αριθμό που χαρακτηρίζει την πρόταση του κειμένου (`readint(NprevQ)`) και ακολούθως στη δεύτερη γραμμή την ίδια την πρόταση με το ρήμα και τα δύο εκατέρωθεν αυτού μέρη της (`readln(Sentence)`). Όσο το όρισμα του κανόνα (το οποίο αποτελεί μέρος της ερώτησης), διαφέρει από εκείνο που διάβασε το πρόγραμμα και το οποίο έγραψε σε μια εσωτερική βάση, ο κανόνας θα επαναλαμβάνεται. Όταν αυτά τα δύο ταυτιστούν `Q=Sentence`, καλείται να εκτελεστεί η δεύτερη επιλογή του κανόνα η οποία και ικανοποιείται με αποτέλεσμα το πρόγραμμα να επιστρέφει στο σημείο, από το οποίο καλέστηκε (`find_number_previous_phrase()`), για να κλείσει το αρχείο και να επιστρέψει στο `previous_phrase_in_text()`. Η επόμενη εντολή αφορά στην εκτέλεση του κανόνα `findpreviousphrase(PrevQ,NprevQ)` :

```

1.1/a.4/a.3.2 findpreviousphrase(PrevQ,NprevQ) :-
    openread(phrase,"phrase.pro"),
    readdevice(phrase),
    previous_phrase(NprevQ),
    previousphrase(PrevQ),
    closefile(phrase),
    readdevice(keyboard).

```

Η εκτέλεση του κανόνα αυτού θα μας δώσει την απάντηση στην ερώτηση που υποβλήθηκε, όταν αυτή δεν υπονοείται από κάποια προαπαιτούμενη γνώση αλλά αναφέρεται ρητά στο κείμενο της απόδειξης των προτάσεων των Στοιχείων του

Ευκλείδη ως αποτέλεσμα διαδοχικών αιτιακών συμπερασμών. Όλα τα συμπεράσματα στα κείμενα των αποδείξεων αποτελούν προϊόν συνεπαγωγής (σειρά λογικών βημάτων) που επιτρέπουν στο πρόγραμμα με την κατάλληλη διαδικασία αναζήτησης να εντοπίζει τον κρίκο ο οποίος αποτελεί την εξήγηση του συμπεράσματος (το οποίο ερωτάται).

Ο κανόνας `findpreviousphrase (PrevQ, NprevQ)` ανοίγει το αρχείο `phrase.pro` και ανακατευθύνει το πρόγραμμα στο αρχείο για να διαβάσει το περιεχόμενό του. Το αρχείο είναι προϊόν επεξεργασίας των κειμένων των αποδείξεων των προτάσεων των Στοιχείων του Ευκλείδη και περιέχει τμήματα προτάσεων των κειμένων με ρήμα. Είναι δε οργανωμένο έτσι ώστε στη μία γραμμή να γράφεται ο αριθμός ο οποίος χαρακτηρίζει ένα μέρος από το κείμενο της πρότασης το οποίο γράφεται στην αμέσως επόμενη γραμμή. Στη συνέχεια δίνεται εντολή εκτέλεσης του αναδρομικού κανόνα `previous_phrase ()` :

```
1.1/a.4/a.3.2.1/a previous_phrase (Nphrase) :-  
    not (eof (phrase)) ,  
    readint (N) ,  
    readln (Sentence) ,  
    retractall (previousphrase (_)) ,  
    assert (previousphrase (Sentence)) ,  
    Nphrase<>N, ! ,  
    previous_phrase (Nphrase) .
```

```
1.1/a.4/a.3.2.1/b previous_phrase (_) .
```

Με τον αναδρομικό κανόνα `previous_phrase (_)` και εφόσον η διαδικασία δεν έχει φτάσει στο τέλος του αρχείου, το πρόγραμμα διαβάζει στην πρώτη γραμμή τον αριθμό που χαρακτηρίζει την πρόταση του κειμένου (`readint (N)`) και αμέσως μετά στη δεύτερη γραμμή την ίδια την ρηματική πρόταση την οποία και γράφει στη συνέχεια σε μια εσωτερική βάση : `previousphrase ()` . Ελέγχει εάν ο αριθμός του ορίσματος του κανόνα είναι διαφορετικός από αυτόν που διάβασε το πρόγραμμα στο αρχείο `phrase : Nphrase<>N`. Εάν είναι διαφορετικός, επαναλαμβάνει την διαδικασία, διαφορετικά το πρόγραμμα εκτελεί την επόμενη επιλογή του κανόνα `previous_phrase (_)` η οποία και ικανοποιείται. Αυτό σημαίνει ότι βρέθηκε στο αρχείο ο αριθμός που αντιστοιχεί στην προηγούμενη πρόταση αυτής με την οποία συντάχτηκε η ερώτηση και ότι την έχει γράψει στην εσωτερική βάση : `previousphrase (Sentence)` . Η ικανοποίηση του κανόνα οδηγεί το πρόγραμμα εκεί, από όπου αυτός εκλήθη : `findpreviousphrase (PrevQ, NprevQ)` , για να ακολουθήσει το κλείσιμο του αρχείου, η ικανοποίηση και αυτού του κανόνα, και η

επιστροφή του προγράμματος στον κανόνα `previous_phrase_in_text(Q,PrevQ)`.
 Με την ικανοποίηση και αυτού του κανόνα το πρόγραμμα επιστρέφει στον `process_answer()` για να καλέσει και να ελέγξει ότι η προηγούμενη πρόταση του κειμένου που θα δώσει σαν απάντηση στην ερώτηση δεν αποτελεί προαπαιτούμενη γνώση και άρα δεν απαιτείται περαιτέρω επεξεργασία :

```
1.1/a.4/a.4 pk(1,cn1,K1),PrevQ<>K1,
             pk(3,cn3,K3),PrevQ<>K3,
```

Τα κατηγορήματα `pk(,_,_)`, έχουν σαν τρίτο όρισμα τα κείμενα των ορισμάτων, των αξιωμάτων και των κοινών εννοιών του πρώτου βιβλίου των Στοιχείων του Ευκλείδη όπως :

```
pk(1,cn1," things which are equal to the same thing are equal to one
another ").
```

Η εκτέλεση της εντολής `PrevQ<>K1` ελέγχει ότι η προηγούμενη πρόταση του κειμένου `PrevQ` δεν είναι κάποιο από τα κείμενα `K1`, `K3` της προαπαιτούμενης γνώσης.

Μετά από αυτόν τον έλεγχο το πρόγραμμα προχωρά στην εκτέλεση της επόμενης εντολής : `answer(Q,PrevQ)`

```
1.1/a.4/a.5 answer(Q,PrevQ):-
    openappend(outfile,"c:\\prolog\\qanswers.txt"),
    writedevic(outfile),
    nl,write("the answer of question :why ",Q,"? is :"),
    nl,write("because <<",PrevQ,">>"),nl,
    writedevic(screen),
    closefile(outfile).
```

Η εντολή αυτή ανοίγει το εξωτερικό αρχείο `qanswers.txt` και γράφει την απάντηση για την ερώτηση `Q`, ώστε να είναι δυνατή η περαιτέρω επεξεργασία της από τον χρήστη. Στη συνέχεια το πρόγραμμα κλείνει το αρχείο, επιστρέφει στον κανόνα `process_answer()`, με τη σειρά του αυτός επιστρέφει στον `1.1/a/question(Question)` ο οποίος ολοκληρώνει τη διαδικασία επιστρέφοντας στον κανόνα : `1 q1`.

Εάν όμως η προηγούμενη πρόταση του κειμένου `PrevQ` αποτελεί προαπαιτούμενη γνώση και άρα απαιτείται περαιτέρω επεξεργασία, τότε το πρόγραμμα εκτελεί την επιλογή :

```
1.1/a.4/h process_answer(Strategy):-strategy(Synonym,_,_,
    Inversion,_,Pk,Strategy),
    Synonym="nosynonymes",
    Inversion="noinversion",
```

```

Pk="pk",
compose_question(Q, Synonym, Inversion),
same_relation,
previous_phrase_in_text(Q, PrevQ),
implicit_knowledge(PrevQ).

```

Η διαδικασία της επιλογής αυτής διαφέρει ως προς την προηγούμενη :

1. όταν εκτελείται ο κανόνας `same_relation` και χρειάζεται το συνώνυμο της αριστερής ως προς το ρήμα οντότητας της ερώτησης, τότε εκτελείται η δεύτερη επιλογή του κανόνα `number_leftpart_verb(L,N)` :

```

1.1/a.4/a.2.1/b number_leftpart_verb(L,N) :-cn1(on),
openread(verbless, "verbless.pro"),
readdevice(verbless),
synwnyma(L, SL),
c(" ", SL, BL),
match_leftpart_verbless(BL),
numberleftpart(N),
closefile(verbless),
readdevice(keyboard).

```

Με την πρώτη εντολή του κανόνα, το string `on` γράφεται στην εσωτερική βάση `cn1()`, κάτι που θα χρειαστεί ως προϋπόθεση για την εκτέλεση του κανόνα `implicit_knowledge(PrevQ)`. Με τις επόμενες εντολές καλείται το συνώνυμο της οντότητας και ακολουθείται η διαδικασία, όπως αυτή περιγράφηκε στην πρώτη επιλογή του κανόνα.

2. στην ύπαρξη του κανόνα `implicit_knowledge(PrevQ)` :

```

1.1/a.4/h.4 implicit_knowledge(PrevQ) :-pk(1, cn1, K1), PrevQ=K1,
assert(cn1(on)),
common_notion1.

```

ο οποίος ελέγχει ότι αν η προαπαιτούμενη γνώση αφορά στην πρώτη κοινή έννοια του πρώτου βιβλίου των Στοιχείων του Ευκλείδη και αν η εσωτερική βάση `cn1` έχει το περιεχόμενο `on`. Αν αυτά συμβαίνουν, δίνει εντολή να εκτελεστεί ο κανόνας `common_notion1`:

```

1.1/a.4/h.4.1 common_notion1:-leftpartofQrelation,
rightpartofQrelation,
numberLpartofLEntityinVerb(N),
numberRpartofREntityinVerb(MREntity),
N<>MREntity,
answer_cn1.

```

Με τον κανόνα `common_notion1` δίνεται εντολή στον `leftpartofQrelation` :

```

1.1/a.4/h.4.1.1 leftpartofQrelation:-
leftentityofquestiontransferred(LEntity),

```

```

        number_leftpart_verb(LEntity,NLEntity),
        findleftpartoftext(LEntity,NLEntity,N),
        find_rightpart_verbless(N,RightpartnextverbofLEntity),
        f(RightpartnextverbofLEntity,equal,T1),
        f(T1,to,T2),
        f(T2,RightentityofLEntity," "),
        retractall(rightpartofLEntity(_)),
        assert(rightpartofLEntity(RightentityofLEntity)),
        retractall(numberLpartofLEntityinVerb(_)),
        assert(numberLpartofLEntityinVerb(N)).

```

Ο κανόνας `leftpartofQrelation` κάνει ανάκληση της αριστερής οντότητας της ερώτησης `LEntity` όπως έχει καταγραφεί από τον κανόνα `1.1/a.4/a.1 compose_question()` και βρίσκει από το αρχείο `verbless.pro` με τον κανόνα `1.1/a.4/2.2 number_leftpart_verb()` τον αριθμό που αντιστοιχεί σε μια ταυτόσημη αριστερή οντότητα ίδιας σχέσης στο κείμενο, η οποία δεν ταυτίζεται με τη σχέση της ερώτησης, η οποία και αυτή υπάρχει στο κείμενο. Θα πρέπει να ανήκει σε μια άλλη σχέση που υπάρχει στο κείμενο και αυτό εξασφαλίζεται από τον κανόνα `findleftpartoftext(LEntity,NLEntity,N)`:

```

1.1/a.4/h.4.1.1.1/a    findleftpartoftext(LEntity,_,N):-
                        number_leftpart_verb(LEntity,N).

1.1/a.4/h.4.1.1.1/b    findleftpartoftext(LEntity,_,N):-
                        synwnyma(LEntity,SLEntity),
                        number_leftpart_verb(SLEntity,N).

```

Η πρώτη επιλογή του κανόνα αφορά την περίπτωση κατά την οποία η αριστερή οντότητα της ερώτησης θα βρεί την ταυτόσημή της στο κείμενο, ενώ η δεύτερη επιλογή αφορά την περίπτωση που στο κείμενο υπάρχει το συνώνυμο της οντότητας.

Με την επόμενη κλήση του κανόνα `find_rightpart_verbless()`, το πρόγραμμα βρίσκει το δεξί τμήμα της σχέσης του κειμένου και με διαδοχικά `fronttokens` απομονώνει την οντότητα του κειμένου που συνδέεται με την αριστερή οντότητα της ερώτησης. Στη συνέχεια ο κανόνας γράφει σε εσωτερικές βάσεις και την αριστερή οντότητα που βρέθηκε στο αρχείο του κειμένου και τον αριθμό που της αντιστοιχεί στο αρχείο.

Το πρόγραμμα επιστρέφει στον κανόνα `common_notion1` για την εκτέλεση της επόμενης εντολής `rightpartofQrelation`:

```

1.1/a.4/h.4.1.2    rightpartofQrelation:-
                    rightentityofquestiontransferred(REntity),
                    number_leftpart_verb(REntity,MREntity),
                    find_rightpart_verbless(MREntity,RightpartnextverbofREntity),
                    f(RightpartnextverbofREntity,equal,Z1),

```

```

f(Z1,to,Z2),
f(Z2,RightentityofREntity," "),
retractall(rightpartofREntity(_)),
assert(rightpartofREntity(RightentityofREntity)),
retractall(numberRpartofREntityinVerb(_)),
assert(numberRpartofREntityinVerb(MREntity)).

```

Με τον κανόνα αυτό γίνεται ανάκληση της δεξιάς οντότητας της ερώτησης REntity, όπως έχει καταγραφεί από τον κανόνα 1.1/a.4/a.1 compose_question() και βρίσκει από το αρχείο verbless.pro με τον κανόνα 1.1/a.4/2.2 number_leftpart_verb() τον αριθμό που αντιστοιχεί σε μια ταυτόσημη αριστερή οντότητα ίδιας σχέσης στο κείμενο. Με την επόμενη κλήση του κανόνα find_rightpart_verbless(), το πρόγραμμα βρίσκει το δεξί τμήμα της σχέσης του κειμένου και με διαδοχικά fronttokens απομονώνει την οντότητα του κειμένου που συνδέεται με την δεξιά οντότητα της ερώτησης. Στη συνέχεια ο κανόνας γράφει σε εσωτερικές βάσεις και την δεξιά οντότητα που βρέθηκε στο αρχείο του κειμένου και τον αριθμό που της αντιστοιχεί στο αρχείο.

Με την ικανοποίηση και αυτού του κανόνα το πρόγραμμα επιστρέφει στον κανόνα common_notion1, ο οποίος αφού διαπιστώσει ότι οι αριθμοί που αντιστοιχούν στις δύο δεξιές οντότητες είναι διαφορετικοί και άρα προέρχονται από σχέσεις που βρίσκονται σε διαφορετικά μέρη του κειμένου, δίνει εντολή στην εκτέλεση του κανόνα answer_cn1 :

```

1.1/a.4/h.4.1.3/a answer_cn1:- leftentityofquestion(LEntity),
                                rightentityofquestion(REntity),
                                rightpartofLEntity(RightentityofLEntity),
                                rightpartofREntity(RightentityofREntity),
                                synwnyma(RightentityofLEntity,RightentityofREntity),
                                openappend(outfile,"c:\\prolog\\qanswers.txt"),
                                writedevic(outfile),
nl,write("The answer of : why <",&LEntity,"> is equal to <",&REntity,">
is :"),
nl,write(" because <",&LEntity,"> is equal to
                                <",&RightentityofLEntity,">"),
nl,write(" <",&REntity,"> is equal to <",&RightentityofREntity,">"),
nl,write(" and <",&RightentityofLEntity,">, <",&RightentityofREntity,">
                                are synonymes that means"),
nl,write(" that <",&LEntity,">, <",&REntity,"> are equal to each
other"),
writedevic(screen),
closefile(outfile).

```

Η απάντηση που δίνει αυτή η επιλογή του κανόνα answer_cn1 κάνει χρήση της προαπαιτούμενης γνώσης και συγκεκριμένα της πρώτης κοινής έννοιας. Καλεί το περιεχόμενο των εσωτερικών βάσεων που αφορούν στις οντότητες της ερώτησης και

του κειμένου, και δείχνει ότι οι δύο οντότητες της ερώτησης είναι ίσες με οντότητες του κειμένου, οι οποίες όμως είναι η μια συνώνυμη της άλλης. Αυτό σημαίνει σύμφωνα με την κοινή έννοια ότι οι δύο οντότητες της ερώτησης είναι μεταξύ τους ίσες. Αν όμως οι δύο οντότητες της ερώτησης είναι ίσες με μια κοινή τρίτη οντότητα του κειμένου, τότε την απάντηση την δίνει η επόμενη επιλογή του κανόνα :

```
1.1/a.4/h.4.1.3/b answer_cn1:-
    leftentityofquestion(LEntity),
    rightentityofquestion(REntity),
    rightpartofLEntity(RightentityofLEntity),
    rightpartofREntity(RightentityofREntity),
    RightentityofLEntity=RightentityofREntity,
    openappend(outfile,"c:\\prolog\\qanswers.txt"),
    writedevic(outfile),
nl,write("answer of : why <\",LEntity,\"> is equal to <\",REntity,\"> is
: "),
nl,write(" because <\",LEntity,\"> is equal to
                                <\",RightentityofLEntity,\">"),
nl,write(" <\",REntity,\"> is equal to <\",RightentityofREntity,\">"),
nl,write(" and <\",RightentityofLEntity,\">, <\",RightentityofREntity,\">
                                are the same finite straight line, that means"),
nl,write(" that <\",LEntity,\">, <\",REntity,\"> are equal to each
other"),
writedevic(screen),
closefile(outfile).
```

Την απάντηση το πρόγραμμα τη γράφει στο εξωτερικό αρχείο qanswers.txt ως εξής:

```
Your question is :
[why <ac> is equal to <ab> ?].
I analysed this question using strategy : <strategy1>
which involves : nosynonymes, left, right, noinversion, equality,
nopk
My ansuer using Euclid's proof text offers the justification :
because < the point a is the centre of the circle cdb >.
```

```
This justification is based on the following :
<ac>=<ab>
because the ends of <ac> are <a>, <c>
and the ends of <ab> are <a>, <b>
and the common end <a> is the center of the circle <cdb>.
That is, <c>, <b> are on the circumference of the same circle,
and therefore <ac>, <ab> are equal as radii on the same circle.
```

```
The above follows from the definition of the circle.
In this case, circle <cdb>
is a plane figure contained by one line such that all the straight
lines <ac>, <ab>
falling upon it from one point <a>
among those lying within the figure are equal to one another i.e.
<ac>=<ab>
where the point <a> is the centre of the circle.
```

Για την απάντηση ερωτήσεων που αφορούν βρόχους προτάσεων κειμένων όπως π.χ. η ερώτηση της Βιοϊατρικής «is there a process loop of p53?», το πρόγραμμα μετά την επεξεργασία της ερώτησης καλεί τον κανόνα `process_answer(strategy_)` επιλέγοντας τη στρατηγική 17 λόγω του ότι απέτυχαν οι υπόλοιπες.

```
process_answer(strategy17) :-  
    oldstrategy(Strategy),  
    strategy(Synonym, LSide, RSide, Inversion, Relstrategy, Pk, strategy17),
```

από την επεξεργασία της ερώτησης δίνεται η δεξιά του ρήματος οντότητα «p53» στη θέση «REntityQ»:

```
rightentityofquestion(REntityQ),
```

και η ερώτηση στη θέση «Question»:

```
question(Question),
```

στη συνέχεια αρχίζει η παραγωγή της απάντησης στην ερώτηση και συγκεκριμένα δίνονται τα πρώτα στοιχεία που αφορούν την ακολουθούμενη στρατηγική και τις παραμέτρους της που οδήγησαν στην απάντησή της :

```
openwrite(tempst17, "c:\\prolog\\tempst17.pro"),  
writedevicetempst17),  
write("Your question is : <\", Question, \">"),  
nl, write("To be answered using the text (...)" ),  
nl, write("-----"),  
nl, write("I succeeded answering this question using  
strategy :<\", Strategy, \">"),  
nl, write("With its parameters taking the values :"),  
nl, write("<\", Synonym, \">, <\", LSide, \">, <\", RSide, \">,  
<\", Inversion, \">, <\", Relstrategy, \">, <\", Pk, \">"),  
nl, write("which means that I am testing for causality  
using no synonyms, "),  
nl, write("no inversion and no prerequisite  
knowledge"),  
nl, write("My answer using the input text is 'yes there  
is loop of p53'"),  
nl, write("and I offer the following explanation and  
meta-explanation of my answering process :"),  
closefile(tempst17),  
writedevicetempst17),
```

Το πρόγραμμα συνεχίζει μέσω του κανόνα `process_answer(_)` την κλήση του κανόνα: `findtheLEntityleftinVerbless(_)` ο οποίος βρίσκει τον αριθμό «NumberEntityV» που αντιστοιχεί στην αριθμημένη πρόταση του αρχείου

Verbless η οποία έχει αριστερά του ρήματος (regulates) την οντότητα της ερώτησης «REntityQ »:

```
findtheLEntityleftinVerbless (REntityQ, NumberEntityV, LEntityV)
```

και στη συνέχεια την κλήση του κανόνα

```
findtheREntityrightintheSameVerbless (_,_) ο οποίος βρίσκει την οντότητα «REntityV» που υπάρχει δεξιά του ρήματος (regulates) στην ίδια αριθμημένη πρόταση που αντιστοιχεί στον αριθμό «NumberEntityV»:
```

```
findtheREntityrightintheSameVerbless (NumberEntityV, _).
```

Με την κλήση των δύο αυτών κανόνων επιτυγχάνεται η εύρεση εκείνης της πρότασης, η οποία έχει στο μέν αριστερό της τμήμα ως προς το ρήμα την οντότητα που αναζητάται από την ερώτηση στο δε δεξί της τμήμα μια λέξη η οποία είναι οντότητα «CausedEntity» σύμφωνα με το λεξικό.

Ο αριθμός των δύο τμημάτων της πρότασης στο αρχείο Verbless δίνει ολόκληρη την πρόταση που υπάρχει στο αρχείο phrase με την κλήση του κανόνα:

```
findthephrase (NumberEntityV),
```

ο οποίος έχει σαν όρισμα τον συγκεκριμένο αριθμό και η πρόταση καταγράφεται σε μια εσωτερική βάση:

```
phrase (Phrase),
```

στή συνέχεια ενημερώνεται το αρχείο <tempst17> με τα αποτελέσματα που

δίνουν οι δύο κανόνες:

```
openappend(tempst17, "c:\\prolog\\tempst17.pro"),
writedevice(tempst17),
nl,write("I found the entity <\",REntityQ,\"> in the
phrase"),
nl,write("<\",NumberEntityV,\">: <\",Phrase,\">"),
closefile(tempst17),
writedevice(screen),
```

Κατόπιν το πρόγραμμα θα αναζητήσει από το αρχείο Verbless μια πρόταση η οποία θα έχει στο αριστερό της τμήμα ως προς το ρήμα (inhibit) την οντότητα CausedEntity που βρήκε μόλις πριν στο δεξί τμήμα της προηγούμενης πρότασης και το οποίο καταγράφεται στο αρχείο της απάντησης:

```
openappend(tempst17, "c:\\prolog\\tempst17.pro"),
writedevice(tempst17),
nl,write("I search for the entity <\",CausedEntity,\">
as follows:"),
closefile(tempst17),
writedevice(screen),
```

Η αναζήτηση της νέας πρότασης γίνεται με την κλήση πάλι του πρώτου κανόνα :

```
findtheLEntityleftinVerbless(CausedEntity,NumberCEntity,_,_),!,
```

Στη νέα πρόταση που θα πρέπει να έχει ως ρήμα το `inhibit` αναζητάται το δεξί της τμήμα ως προς το ρήμα και η οντότητα που υπάρχει σε αυτό με την κλήση του δεύτερου κανόνα:

```
findtheREntityrightintheSameVerbless(NumberCEntity,REofCEntityV),!,
```

Μετά την εύρεση της οντότητας (`REofCEntityV`) που υπάρχει στο δεξί τμήμα ως προς το ρήμα (`inhibit`) της νέας πρότασης, το πρόγραμμα ελέγχει αν αυτή η οντότητα είναι ίδια με την οντότητα της πρώτης πρότασης στο αριστερό της τμήμα ως προς το ρήμα (`regulates`) (δηλαδή ίδια με την οντότητα της ερώτησης):

```
LEntityV=REofCEntityV,
```

Ενημερώνεται το αρχείο παραγωγής της απάντησης για την κατάσταση του συστήματος με την πρώτη πρόταση που βρήκε στο αρχείο `Verbless`:

```
openappend(tempst17,"c:\\prolog\\tempst17.pro"),%txt"),
  writedevice(tempst17),
  nl,write("In my effort for answering the question
<\",Question,\">"),
  nl,write("I found that from the following phrases"),
  nl,write("<\",NumberEntityV,\">: <\",Phrase,\">"),
  closefile(tempst17),
  writedevice(screen),
```

και όπως προηγουμένως αναζητά όλη τη φράση από το αρχείο `phrase`:

```
findthephrase(NumberCEntity),
την οποία καταγράφει στην εσωτερική βάση:
phrase(P),
```

και ενημερώνει ξανά το αρχείο της απάντησης:

```
openappend(tempst17,"c:\\prolog\\tempst17.pro"),%txt"),
  writedevice(tempst17),
  nl,write("<\",NumberCEntity,\">: <\",P,\">"),
  nl,write("From which it follows that there is a loop
for <\",LEntityV,\">"),
  closefile(tempst17),
  writedevice(screen).
```

Οι δύο κανόνες με τους οποίους το πρόγραμμα αναζήτησε τις οντότητες και διαπίστωσε τον βρόχο και άρα την αλυσίδα των οντοτήτων που αλληλοεπιρεάζονται περιγράφονται αναλυτικά στη συνέχεια.

Ο πρώτος:

```
findtheLEntityleftinVerbless(E,N,LEV):-
```

(για την ενημέρωση του αρχείου της απάντησης)

```
openappend(tempst17,"c:\\prolog\\tempst17.pro"),%txt"),
  writedevicetempst17),
  nl,write("I am searching in the input text for the
entity <\",E,\"> as mentioned in the question"),
  closefile(tempst17),
  writedevicetempst17),
retractall(numberleftpartverbless(_)),
retractall(leftpartverbless(_)),
```

(για το άνοιγμα του αρχείου)

```
openread(verbless,"verbless.pro"),
```

(για την αναζήτηση της οντότητας σε ένα αριστερό τμήμα μιας πρότασης του αρχείου καλείται ο κανόνας `match_inLpartVerbless_Entity(_)`, ο οποίος θα αναλυθεί στη συνέχεια)

```
match_inLpartVerbless_Entity(E),
```

(για το κλείσιμο του αρχείου)

```
closefile(verbless),
readdevice(keyboard),readln(_),
```

(για την ενημέρωση των εσωτερικών βάσεων)

```
numberleftpartverbless(N),
theEntityinthePartofVerbless(LEV).
```

Και ο δεύτερος:

```
findtheREntityrightintheSameVerbless(NEV,REV):-
```

(από τον οποίο γίνεται κλήση κανόνα για την αναζήτηση του δεξιού τμήματος της πρότασης ως προς το ρήμα, από το αρχείο `Verbless` στο οποίο βρίσκεται και το αριστερό της τμήμα)

```
find_rightpart_verbless(NEV,Rightpartnextverb),
```

(για την ενημέρωση του αρχείου της απάντησης των αποτελεσμάτων της αναζήτησης)

```
openappend(tempst17,"c:\\prolog\\tempst17.pro"),
  writedevicetempst17),
  nl,write("I found that to the right of the verb in the
phrase <\",NEV,\"> is the"),
  nl,write("chunk <\",Rightpartnextverb,\"> and about the
entity"),
  closefile(tempst17),
  writedevicetempst17),
```

(γίνεται κλήση κανόνα για την αναζήτηση οντότητας σε ένα τμήμα της πρότασης)

```
find_Entity(Rightpartnextverb),
```

```
(για την ενημέρωση εσωτερικής βάσης)
theEntityinthePartofVerbless (REV) ,
(για την ενημέρωση του αρχείου της απάντησης)
openappend(tempst17, "c:\prolog\tempst17.pro"), %txt" ,
writedevice(tempst17) ,
nl,write("the entity <\",REV,\"> is one of the tokens of
the chunk <\",Rightpartnextverb,\">\"),
nl,write("which lies at the left of the verb of the
phrase <\",NEV,\">.\"),
closefile(tempst17) ,
writedevice(screen)!.

```

Ο κανόνας που καλείται για την εύρεση ολόκληρης της φράσης στο αρχείο phrase
findthephrase (NEV) :-

```
retractall(numbercurrentphrase(_)) ,
retractall(phrase(_)) ,
openread(phrase, "phrase.pro") ,
(γίνεται κλήση κανόνα για την αναζήτηση της φράσης μέσω του αριθμού NEV)
match_phrase_number (NEV) ,
closefile(phrase) ,
readdevice(keyboard) .

```

Ο κανόνας που καλείται για την αναζήτηση της φράσης, κάνοντας χρήση του
αριθμού που της αντιστοιχεί και στο αρχείο Verbless

```
match_phrase_number (Num) :-
readdevice(phrase) ,
readint(N) ,
N<>0 ,
asserta(numbercurrentphrase(N)) ,
readln(Bphrase) ,
f(Bphrase, H, T) , c(H, T, Phrase) ,
asserta(phrase(Phrase)) ,
Num<>N ,
match_phrase_number (Num) .

```

(σε περίπτωση που ο αριθμός των τμημάτων των προτάσεων του αρχείου Verbless
δεν είναι ίδιος με τον αριθμό που διάβασε από το αρχείο phrase και δεν είναι
μηδέν)

```
match_phrase_number (_):-
numbercurrentphrase(N) , N<>0 , phrase(Phrase) ,
closefile(phrase) , readdevice(keyboard) , !.

```

(σε περίπτωση που ο αριθμός των τμημάτων των προτάσεων του αρχείου Verbless
δεν είναι ίδιος με τον αριθμό που διάβασε από το αρχείο phrase και είναι μηδέν)

```
match_phrase_number (_):-
numbercurrentphrase(N) , N=0 ,
closefile(phrase) , readdevice(keyboard) , fail.

```

Ο κανόνας που καλείται για την αναζήτηση σε ποιο από τα αριστερά τμήματα των προτάσεων του Verbless υπάρχει η οντότητα Entity του λεξικού.

```
match_inLpartVerbless_Entity(Entity) :-
  readdevice(verbless),
  readint(N),
  nl,write(N),
  asserta(numberleftpartverbless(N)),
  N<>0,
  readln(BLpartVerbless),
  f(BLpartVerbless,H,T),c(H,T,LpartVerbless),
  asserta(leftpartverbless(LpartVerbless)),
  openappend(tempst17,"c:\\prolog\\tempst17.pro"),%txt"),
  writedevicetempst17),
  nl,write("to the chunk <","LpartVerbless,"> of the
phrase <","N,">"),
  closefile(tempst17),
  writedevicetempst17),
  readln(_),readln(_),
  retractall(theEntityinthePartofVerbless(_)),
  find_Entity(LpartVerbless),
  theEntityinthePartofVerbless(LEntityV),
LEntityV<>Entity,
```

(αν η οντότητα που βρήκε είναι ίδια με αυτήν για την οποία γίνεται η αναζήτηση, το πρόγραμμα θα κάνει κλήση στην επόμενη επιλογή του ίδιου κανόνα, διαφορετικά γίνεται ενημέρωση του αρχείου της απάντησης)

```
openappend(tempst17,"c:\\prolog\\tempst17.pro"),%txt"),
  writedevicetempst17),
  nl,write("I found that the entity <","Entity,"> is not
a token of the chunk <","LpartVerbless,">"),
  nl,write("of the phrase and for this reason I will
test the following phrase"),
  closefile(tempst17),
  writedevicetempst17),
  match_inLpartVerbless_Entity(Entity).
```

(επόμενη επιλογή του ίδιου κανόνα σε περίπτωση εύρεσης της ζητούμενης οντότητας)

```
match_inLpartVerbless_Entity(Entity) :-
  leftpartverbless(LpartVerbless),
  numberleftpartverbless(N),N<>0,
  openappend(tempst17,"c:\\prolog\\tempst17.pro"),%txt"),
  writedevicetempst17),
  nl,write("I found that the entity <","Entity,"> is one
of the tokens of the chunk <","LpartVerbless,">"),
  nl,write("which lies at the left of the verb of the
phrase <","N,">."),
  closefile(tempst17),
```

```
writedevic (screen) ,
closefile (verbless) , readdevice (keyboard) , !.
```

(επόμενη επιλογή του ίδιου κανόνα, όπου δεν υπάρχει η ζητούμενη οντότητα)

```
match_inLpartVerbless_Entity (Entity) :-
numberleftpartverbless (N) , N=0 ,
openappend (tempst17 , "c:\\prolog\\tempst17.pro" , %txt" ) ,
writedevic (tempst17) ,
nl , write ("the entity < , Entity , "> there isn't to any
left chunk of phrases" ) ,
closefile (tempst17) ,
writedevic (screen) ,
closefile (verbless) , readdevice (keyboard) , fail .
```

Ο κανόνας που καλείται για την αναζήτηση μιας οντότητας σε μια πρόταση:

```
find_Entity (L) :-
f (L , W , "" ) ,
(σε περίπτωση που οντότητα είναι η μοναδική λέξη)
isalexi (W , "entity" ) ,
asserta (theEntityinthePartofVerbless (W) ) ,
(για την ενημέρωση του αρχείου της απάντησης)
openappend (tempst17 , "c:\\prolog\\tempst17.pro" , %txt" ) ,
writedevic (tempst17) ,
nl , write ("is the only word-entity < , W , "> to that
chunk of the phrase" ) ,
closefile (tempst17) ,
writedevic (screen) , !.
```

επόμενη επιλογή του ίδιου κανόνα

```
find_Entity (L) :-
f (L , W , R1) ,
(σε περίπτωση που η οντότητα είναι η πρώτη από μια σειρά λέξεων)
isalexi (W , "entity" ) , R1<>"" ,
asserta (theEntityinthePartofVerbless (W) ) ,
(για την ενημέρωση του αρχείου της απάντησης)
openappend (tempst17 , "c:\\prolog\\tempst17.pro" , %txt" ) ,
writedevic (tempst17) ,
nl , write ("< , W , "> is the first token of the chunk" ) ,
closefile (tempst17) ,
writedevic (screen) , !.
```

επόμενη επιλογή του ίδιου κανόνα

```
find_Entity (L) :-
f (L , W , R1) ,
(σε περίπτωση που η πρώτη από μια σειρά λέξεων δεν είναι οντότητα)
not (isalexi (W , "entity" ) ) ,
(για την ενημέρωση του αρχείου της απάντησης)
openappend (tempst17 , "c:\\prolog\\tempst17.pro" , %txt" ) ,
writedevic (tempst17) ,
```

```

    nl,write("is not the fist token and therefore I test
the rest of the tokens."),
    closefile(tempst17),
    writedevic(screen),
(το πρόγραμμα ξεκινάει ένα βρόχο αναζητώντας την οντότητα στις επόμενες λέξεις)
    find_Entity1(R1).

```

για την έναρξη του βρόχου γίνεται κλήση του κανόνα:

```

find_Entity1(S):-
    f(S,W1,""),
(σε περίπτωση που δεν είναι οντότητα η μοναδική επόμενη λέξη)
    not(isalex1(W1,"entity")),
(για την ενημέρωση του αρχείου της απάντησης)
openappend(tempst17,"c:\\prolog\\tempst17.pro"),%txt"),
    writedevic(tempst17),
    nl,write("My search showed that there isn't entity to
that chunk."),
    closefile(tempst17),
    writedevic(screen).

```

επόμενη επιλογή του ίδιου κανόνα

```

find_Entity1(S):-
    f(S,W1,_),
(σε περίπτωση που είναι οντότητα η επόμενη λέξη)
    isalex1(W1,"entity"),
    asserta(theEntityinthePartofVerbless(W1)),
(για την ενημέρωση του αρχείου της απάντησης)
openappend(tempst17,"c:\\prolog\\tempst17.pro"),%txt"),
    writedevic(tempst17),
    nl,write("I found that"),
    closefile(tempst17),
    writedevic(screen),!.

```

επόμενη επιλογή του ίδιου κανόνα

```

find_Entity1(S):-
    f(S,W1,T),
(σε περίπτωση που δεν είναι οντότητα η επόμενη λέξη)
    not(isalex1(W1,"entity")),
(για την ενημέρωση του αρχείου της απάντησης)
openappend(tempst17,"c:\\prolog\\tempst17.pro"),%txt"),
    writedevic(tempst17),
    nl,write("Since the next word of this chunk is not an
entity I test the following token"),
    closefile(tempst17),
    writedevic(screen),
(για την επιλογή ξανά του ίδιου κανόνα στις υπόλοιπες λέξεις που ακολουθούν)
    find_Entity1(T).

```

Την απάντηση το πρόγραμμα τη γράφει στο εξωτερικό αρχείο tempst17.pro ως εξής:

Your question is : <What is influenced by p53 ?>
To be answered using the sentences:

<1411>: <the p53 protein regulates the mdm2 gene >
<1421>: <the mdm2 oncogene can inhibit p53 mediated transactivation >

I succeeded answering this question using strategy :<strategy17>
With its parameters taking the values :
<nosynonymes>, <left>, <right>, <noinversion>, <causality>, <nopk>
which means that I am testing for causality using
no synonyms, no inversion and no prerequisite knowledge.

I searched the input for the entity <p53> mentioned in the question
In the chunk <the p53 protein> of the sentence <1411>
p53 is not the first token and therefore I test the rest of the
tokens.

I found that the entity <p53> is one of the tokens of the chunk
<the p53 protein>
which is the chunk to the left of the verb of the sentence <1411>.
I found that the chunk to the right of the verb in the sentence
<1411>

is the chunk <the mdm2 gene >
and since its first token is not an entity I tested the rest of the
tokens.

I found that the entity <mdm2> is one of the tokens of the chunk
<the mdm2 gene >
which lies at the right of the verb of the sentence <1411>.

I found that the entity <p53> influences the entity <mdm2>
in the sentence <1411>: <the p53 protein regulates the mdm2 gene >

I searched then for occurrence of the entity <mdm2>
in a chunk to the left of the verb of a sentence as follows:

I searched the chunk to the left of the verb <the p53 protein>
of the sentence <1411> and found that mdm2 is not the first token
and therefore I tested the rest of the tokens.

I found that the entity <mdm2> is not a token of the chunk <the p53
protein>
of the sentence <1411> and for this reason I will test the following
sentence.

I found that <the mdm2 oncogene can> is the left chunk of the
sentence <1421>.
mdm2 is not the first token and therefore I test the rest of the
tokens.

I found that the entity <mdm2> is one of the tokens of the chunk
<the mdm2 oncogene can> which is to the left of the verb of the
sentence <1421>.

I found that the chunk to the right of the verb in the phrase <1421>
is the chunk <p53 mediated transactivation >
and that <p53> is the first token of this chunk.

Therefore the entity <p53> is one of the tokens of the chunk
<p53 mediated transactivation > which is to the right of the verb of
<1421>.

Therefore answering the question <What is influenced by p53 ?>

I found that from the following sentences:

<1411>: <the p53 protein regulates the mdm2 gene >

<1421>: <the mdm2 oncogene can inhibit p53 mediated transactivation >

it follows that p53 is influenced by p53
and therefore there is a loop for <p53>.

7.5 Υποσύστημα σχεδίασης σχημάτων

Το πρόγραμμα ο οποίος έχει τη δυνατότητα να παράγει το σχήμα της πρώτης πρότασης των Στοιχείων του Ευκλείδη ξεκινάει με την εκτέλεση του κανόνα draw:

```
1 draw:-rgo1(0,13),!,  
      l_go(13,13),!,  
      process_print_screen(0,13),  
      tomi_toxwn(0,13),nl.
```

Ο κανόνας draw μέσω της εντολής rgo1(0,13), 'σχεδιάζει' σε μια νοητική ψηφιακή επιφάνεια ένα τεταρτημόριο (άνω δεξιά) κύκλου, ο οποίος έχει κέντρο το ένα άκρο ενός οριζόντιου ευθυγράμμου τμήματος και ακτίνα ίση με το μήκος του τμήματος, το οποίο χωρίς να επηρεάζει τη γενικότητα είναι 13 διαδοχικές ψηφιακές μονάδες επιφανείας.

```
1.1/a rgo1(X,Y):-X>13/2,  
          rgo2(X,Y).
```

```
1.1/b rgo1(X,Y):-r1(NX,NY,X,Y),  
          assert(r_sintetagmenes(X,Y),dexitoxo),  
          rgo1(NX,NY).
```

Το αρχικό σημείο από το οποίο θα ξεκινήσει ο κανόνας rgo1(X,Y) το σχήμα του τεταρτημόριου είναι η ψηφιακή μονάδα επιφανείας που βρίσκεται στη θέση (0,13). Υπάρχουν δύο επιλογές του κανόνα και η απόφαση για το ποια επιλογή θα εκτελείται κάθε φορά, εξαρτάται από την τιμή που θα λαμβάνει ο X μέσω των διαδικασιών του αλγορίθμου. Εφόσον οι αρχικές τιμές X και Y του προγράμματος είναι αντίστοιχα 0 και 13 (άρα $X > 13/2$) το πρόγραμμα θα εκτελέσει την επιλογή 1.1/b rgo1(X,Y).

Η επιλογή αυτή δίνει εντολή στην εκτέλεση του κανόνα r1(NX,NY,X,Y):

```
1.1/b.1/a r1(NX,NY,X,Y):-NX=X+1,NY=Y,NX*NX+NY*NY<=185.
```

```
1.1/b.1/b r1(NX,NY,X,Y):-NX=X+1,NY=Y-1.
```

Η πρώτη επιλογή αυτού του κανόνα ικανοποιείται όταν, αφού αυξηθεί κατά μία μονάδα η τιμή του X, τότε το τετράγωνο της νέας τιμής του X, συν το τετράγωνο της τιμής του Y να είναι μικρότερο ή ίσο της τιμής του 185. Όταν αυτό δεν ισχύει και το

άθροισμα είναι μεγαλύτερο του 185, τότε δεν ικανοποιείται η πρώτη επιλογή του κανόνα και το πρόγραμμα ενεργοποιεί τη δεύτερη επιλογή η οποία αυξάνει κατά μια μονάδα την τιμή του X ($NX=X+1$) και μειώνει κατά μια μονάδα την τιμή του Y ($NY=Y-1$). Στη συνέχεια το πρόγραμμα επιστρέφει στον κανόνα 1.1/b rgo1(X,Y) ο οποίος αφού γράψει στην εσωτερική βάση δεδομένων r_sintetagmenes(,) τις προηγούμενες τιμές των X και Y , εκτελεί σε βρόγχο τον κανόνα rgo1(X,Y) με τις νέες τιμές των X και Y που είναι αντίστοιχα NX και NY . Όταν η τιμή του X γίνει μεγαλύτερη του 13/2 το πρόγραμμα συνεχίζει με την εκτέλεση του κανόνα rgo2(X,Y) :

```
1.1/a.2/a   rgo2(X,Y) :-X=13,Y=0,
              assert(r_sintetagmenes(X,Y),dexitoxo) .

1.1/a.2/b   rgo2(X,Y) :-r2(NX,NY,X,Y) ,
              assert(r_sintetagmenes(X,Y),dexitoxo) ,
              rgo2(NX,NY) .
```

Όσο οι τιμές των X και Y δεν είναι αντίστοιχα 13 και 0, δηλαδή όσο η ψηφιακή μονάδα επιφάνειας δεν έχει φθάσει στο δεξί άκρο του ευθυγράμμου τμήματος, θα εκτελείται η δεύτερη επιλογή η οποία καλεί την εκτέλεση του κανόνα r2(NX,NY,X,Y) :

```
1.1/a.2/b.1/a   r2(NX,NY,X,Y) :-NX=X+1,NY=Y-1,NX*NX+NY*NY<=185 .

1.1/a.2/b.1/b   r2(NX,NY,X,Y) :-NX=X,NY=Y-1 .
```

Η πρώτη επιλογή αυτού του κανόνα ικανοποιείται όταν, αφού αυξηθεί κατά μία μονάδα η τιμή του X , και μειωθεί κατά μία μονάδα η τιμή του Y , τότε το τετράγωνο της νέας τιμής του X , συν το τετράγωνο της τιμής του Y να είναι μικρότερο ή ίσο της τιμής του 185. Όταν αυτό δεν ισχύει και το άθροισμα είναι μεγαλύτερο του 185, τότε δεν ικανοποιείται η πρώτη επιλογή του κανόνα και το πρόγραμμα ενεργοποιεί τη δεύτερη επιλογή η οποία αφήνει ως έχει την τιμή του X ($NX=X$) και μειώνει κατά μια μονάδα την τιμή του Y ($NY=Y-1$). Στη συνέχεια το πρόγραμμα επιστρέφει στον κανόνα 1.1/a.2/b rgo2(X,Y) ο οποίος αφού γράψει στην εσωτερική βάση δεδομένων r_sintetagmenes(,) τις προηγούμενες τιμές των X και Y , εκτελεί σε βρόγχο τον κανόνα rgo2(X,Y) με τις νέες τιμές των X και Y που είναι αντίστοιχα NX και NY .

Όταν οι τιμές γίνουν $X=13$ και $Y=0$, δηλαδή όταν η ψηφιακή μονάδα επιφάνειας βρεθεί στο δεξί άκρο του ευθυγράμμου τμήματος, τότε το πρόγραμμα εκτελεί την

επιλογή 1.1/a.2/a `rgo2(X,Y)` η οποία θα γράψει στην εσωτερική βάση δεδομένων `r_sintetagmenes(X,Y)` τις τιμές των X και Y που αντιστοιχούν στο σημείο αυτό. Μετά την ικανοποίηση του κανόνα αυτού, το πρόγραμμα επιστρέφει στο σημείο από το οποίο κλήθηκε : 1.1/a `rgo1(X,Y)`. Επειδή δεν υπάρχει άλλη ενέργεια να εκτελεστεί, το πρόγραμμα επιστρέφει στον αρχικό κανόνα : 1 `draw` για την εκτέλεση της επόμενης εντολής `l_go(13,13)` :

```
1.2    l_go(LZ,LY) :-Z=13-LZ,Y=LY,
        lgo1(Z,Y).
```

Με την εντολή αυτή θα επαναληφθεί η διαδικασία κατά την οποία το πρόγραμμα θα ‘σχεδιάσει’ ένα δεύτερο τεταρτημόριο (άνω αριστερά) κύκλου, ακτίνας ίσης πάλι με 13 διαδοχικές ψηφιακές μονάδες επιφανείας και με κέντρο το δεξί άκρο του ευθυγράμμου τμήματος. Η διαδικασία παραγωγής του δεύτερου τεταρτημρίου παραμένει κατά κάποιο τρόπο ίδια με αυτή του πρώτου. Ο ρυθμός μεταβολής των X και Y δεν αλλάζει και οι μόνες διαφορές είναι ότι η μεν τιμή του X θα μειώνεται αντί να αυξάνεται, το δε αρχικό σημείο έναρξης της διαδικασίας θα είναι το (13,13). Το πρόγραμμα για να παρακάμψει τις διαφορές ορίζει νέες μεταβλητές τις $Z=13-LZ$ και $Y=LY$ έτσι ώστε ο κανόνας `lgo1(Z,Y)` να ξεκινάει με τις τιμές $Z=0$ και $Y=13$.

```
1.2.2/a    lgo1(Z,Y) :-Z>13/2,
            lgo2(Z,Y).
```

```
1.2.2/b    lgo1(Z,Y) :-l1(NZ,NY,Z,Y,X),
            assert(l_sintetagmenes(X,Y),aristerotoxo),
            lgo1(NZ,NY).
```

Η απόφαση για το ποια επιλογή του κανόνα θα εκτελείται κάθε φορά, εξαρτάται από την τιμή που θα λαμβάνει ο Z μέσω των διαδικασιών του αλγορίθμου. Εφόσον οι αρχικές τιμές Z και Y του προγράμματος είναι αντίστοιχα 0 και 13 (άρα $Z>13/2$) το πρόγραμμα θα εκτελέσει την επιλογή 1.2.2/b `lgo1(Z,Y)`.

Η επιλογή αυτή δίνει εντολή στην εκτέλεση του κανόνα `l1(NZ,NY,Z,Y,X)` :

```
1.2.2/b.1/a l1(NZ,NY,Z,Y,X) :-NZ=Z+1,NY=Y,NZ*NZ+NY*NY<=185,X=13-Z.
```

```
1.2.2/b.1/b l1(NZ,NY,Z,Y,X) :-NZ=Z+1,NY=Y-1,X=13-Z.
```

Η πρώτη επιλογή του κανόνα ικανοποιείται όταν, αφού αυξηθεί κατά μία μονάδα η τιμή του Z ($NZ=Z+1$), τότε το τετράγωνο της νέας τιμής του Z , συν το τετράγωνο της τιμής του Y να είναι μικρότερο ή ίσο της τιμής του 185. Όταν αυτό δεν ισχύει και το άθροισμα είναι μεγαλύτερο του 185, τότε δεν ικανοποιείται η πρώτη επιλογή του

κανόνα και το πρόγραμμα ενεργοποιεί τη δεύτερη επιλογή η οποία και πάλι αυξάνει κατά μία μονάδα την τιμή του Z αλλά μειώνει κατά μια μονάδα την τιμή του Y ($NY=Y-1$). Και στις δύο επιλογές του κανόνα το πρόγραμμα δίνει την τιμή $x=13-Z$, έτσι ώστε μαζί με την τιμή Y , να προσδιορίζεται η πραγματική θέση της ψηφιακής μονάδας. Στη συνέχεια το πρόγραμμα επιστρέφει στον κανόνα 1.2.2/b lgo1(z, Y) ο οποίος αφού γράψει στην εσωτερική βάση δεδομένων l_sintetagmenes(X, Y) την τιμή του X και την προηγούμενη τιμή του Y , εκτελεί σε βρόγχο τον κανόνα lgo1(z, Y) με τις νέες τιμές των Z και Y που είναι αντίστοιχα NZ και NY . Όταν η τιμή του Z γίνει μεγαλύτερη του 13/2 το πρόγραμμα συνεχίζει με την εκτέλεση του κανόνα lgo2(X, Y) :

1.2.2/a.2/a lgo2(Z, Y) :-Z=13, Y=0, X=13-Z,
assert(l_sintetagmenes(X, Y), aristerotoxo) .

1.2.2/a.2/b lgo2(Z, Y) :-l2(NZ, NY, Z, Y, X) ,
assert(l_sintetagmenes(X, Y), aristerotoxo) ,
lgo2(NZ, NY) .

Όσο οι τιμές των Z , Y και X δεν είναι αντίστοιχα 13, 0 και 0, δηλαδή όσο η ψηφιακή μονάδα επιφάνειας δεν έχει φθάσει στο αριστερό άκρο του ευθυγράμμου τμήματος, θα εκτελείται η δεύτερη επιλογή η οποία καλεί την εκτέλεση του κανόνα l2(NZ, NY, Z, Y, X) :

1.2.2/a.2/b.1/a l2(NZ, NY, Z, Y, X) :-NZ=Z+1, NY=Y-1,
NZ*NZ+NY*NY<=185, X=13-Z.

1.2.2/a.2/b.1/b l2(NZ, NY, Z, Y, X) :-NZ=Z, NY=Y-1, X=13-Z.

Η πρώτη επιλογή αυτού του κανόνα ικανοποιείται όταν, αφού αυξηθεί κατά μία μονάδα η τιμή του Z ($NZ=Z+1$), και μειωθεί κατά μία μονάδα η τιμή του Y ($NY=Y-1$), τότε το τετράγωνο της νέας τιμής του Z , συν το τετράγωνο της τιμής του Y να είναι μικρότερο ή ίσο της τιμής του 185. Όταν αυτό δεν ισχύει και το άθροισμα είναι μεγαλύτερο του 185, τότε δεν ικανοποιείται η πρώτη επιλογή του κανόνα και το πρόγραμμα ενεργοποιεί τη δεύτερη επιλογή η οποία αφήνει ως έχει την τιμή του Z ($NZ=Z$) και μειώνει κατά μια μονάδα την τιμή του Y ($NY=Y-1$). Και στις δύο επιλογές του κανόνα το πρόγραμμα δίνει την τιμή $x=13-Z$, έτσι ώστε μαζί με την τιμή Y , να προσδιορίζεται η πραγματική θέση της ψηφιακής μονάδας. Στη συνέχεια το πρόγραμμα επιστρέφει στον κανόνα 1.2.2/a.2/b lgo2(Z, Y), ο οποίος αφού γράψει στην εσωτερική βάση δεδομένων r_sintetagmenes(X, Y) τις προηγούμενες

τιμές των X και Y, εκτελεί σε βρόγχο τον κανόνα `lgo2(Z, Y)` με τις νέες τιμές των Z και Y που είναι αντίστοιχα `NZ` και `NY`.

Όταν οι τιμές γίνουν `Z=13`, `Y=0` και `X=0`, δηλαδή όταν η ψηφιακή μονάδα επιφανείας βρεθεί στο αριστερό άκρο του ευθυγράμμου τμήματος, τότε το πρόγραμμα εκτελεί την επιλογή `1.2.2/a.2/a lgo2(Z, Y)` η οποία θα γράψει στην εσωτερική βάση δεδομένων `r_sintetagmenes(X, Y)` τις τιμές των X και Y που αντιστοιχούν στο σημείο αυτό. Μετά την ικανοποίηση του κανόνα αυτού, το πρόγραμμα επιστρέφει στο σημείο από το οποίο κλήθηκε : `1.2.2/a lgo1(Z, Y)`. Επειδή δεν υπάρχει άλλη ενέργεια να εκτελεστεί, το πρόγραμμα επιστρέφει στον `1.2 l_go(LZ, LY)` και ακολούθως στον αρχικό κανόνα : `1 draw` για την εκτέλεση της επόμενης εντολής `process_print_screen(0, 13)` :

```
1.3/a process_print_screen(X, Y) :-X=13, Y=0,
                                look_up_sintetagmenes(X, Y),
                                writedevicе(screen), pwrite(X, Y) .

1.3/b process_print_screen(X, Y) :-look_up_sintetagmenes(X, Y),
                                writedevicе(screen), pwrite(X, Y),
                                k3(NX, NY, X, Y),
                                process_print_screen(NX, NY) .
```

Με την εκτέλεση αυτού του κανόνα θα εμφανιστούν στην οθόνη του υπολογιστή το σχεδιάγραμμα των τεταρτημόριων που παράγονται από το πρόγραμμα. Οι αρχικές τιμές της διαδικασίας είναι `X=0` και `Y=13` οι οποίες αντιστοιχούν στην άνω αριστερή ψηφιακή μονάδα του νοητού ψηφιακού επιπέδου. Με τις τιμές αυτές ενεργοποιείται η δεύτερη επιλογή του κανόνα, η οποία δίνει την εντολή `look_up_sintetagmenes(X, Y)` :

```
1.3/b.1/a look_up_sintetagmenes(X, Y) :-r_sintetagmenes(X, Y);
                                                l_sintetagmenes(X, Y), !.

1.3/b.1/b look_up_sintetagmenes(X, Y) :-fail; true.
```

Η εκτέλεση της πρώτης επιλογής αυτού του κανόνα ικανοποιείται εάν υπάρχει στις βάσεις δεδομένων `r_sintetagmenes(X, Y)` ή `l_sintetagmenes(X, Y)` εγγραφή με τιμές `X=0` και `Y=13`. Εάν υπάρχει, επιστρέφει στο σημείο από όπου κλήθηκε, διαφορετικά η αποτυχία της πρώτης επιλογής οδηγεί το πρόγραμμα στη δεύτερη επιλογή η οποία ικανοποιείται και το πρόγραμμα επιστρέφει στον κανόνα `1.3/b process_print_screen(X, Y)`. Η επόμενη εντολή του κανόνα `writedevicе(screen)`, καλεί το πρόγραμμα να εμφανίσει στην οθόνη του υπολογιστή, το σημείο που αντιστοιχεί στη νοητή ψηφιακή γεωμετρική μονάδα,

σύμφωνα με τους υπολογισμούς που έχουν προηγηθεί. Η εμφάνιση του σημείου γίνεται σύμφωνα με τις τιμές των X , Y , οι οποίες επιλέγουν κάθε φορά από τον κανόνα `pwrite(X, Y)`, μια από τις εντολές `write(" ")` και εμφανίζουν το σημείο σε τέτοια θέση ώστε το σύνολό τους να σχηματίζει τεταρτημόριο.

Μετά την εμφάνιση του σημείου στην οθόνη εκτελείται ο επόμενος κανόνας `k3(NX, NY, X, Y)`:

1.3/b.3/a `k3(NX, NY, X, Y) :-NX=X+1, NY=Y, NX<=13.`

1.3/b.3/b `k3(NX, NY, X, Y) :-NX=X-13, NY=Y-1.`

Η πρώτη επιλογή του κανόνα ικανοποιείται, όταν αυξάνοντας την τιμή του X κατά μία μονάδα ($NX=X+1$) αυτή εξακολουθεί να παραμένει μικρότερη ή ίση της τιμής 13 ($NX<=13$) και η τιμή του Y να παραμένει ίδια ($NY=Y$). Εάν δεν ικανοποιείται η πρώτη επιλογή το πρόγραμμα συνεχίζει με τη δεύτερη επιλογή, η οποία ελαττώνει την τιμή του Y κατά μία μονάδα ($NY=Y-1$) και ορίζει νέα τιμή στο X ($NX=X-13$). Με τον κανόνα αυτό έχουμε τη δυνατότητα να διασχίζουμε όλη τη νοητή ψηφιακή επιφάνεια από αριστερά προς τα δεξιά και από πάνω προς τα κάτω. Το πρόγραμμα με την ικανοποίηση της επιλογής επιστρέφει στον κανόνα 1.3/b `process_print_screen(X, Y)` σε μια διαδικασία βρόγχου με την οποία διαπιστώνει κάθε φορά την ύπαρξη ή μη σημείου στη συγκεκριμένη θέση.

Όταν οι τιμές γίνουν $X=13$ και $Y=0$, δηλαδή όταν η ψηφιακή μονάδα επιφανείας βρεθεί στην κάτω δεξιά γωνία του νοητικού ψηφιακού επιπέδου, τότε το πρόγραμμα εκτελεί την επιλογή 1.3a `process_print_screen(X, Y)`. Η επιλογή αυτή μέσω του κανόνα `look_up_sintetagmenes(X, Y)`, ελέγχει αν υπάρχει εγγραφή με αυτές τις τιμές για να εμφανίσει τη ψηφιακή μονάδα στην οθόνη. Με αυτήν την ενέργεια ολοκληρώνεται η διαδικασία του κανόνα `process_print_screen(0, 13)` και το πρόγραμμα επιστρέφει στον κανόνα `draw` για να εκτελέσει την επόμενη διαδικασία `tomi_toxwn(0, 13)`.

1.4/a `tomi_toxwn(X, Y) :-X=13, Y=0,
pixels_tomis(X, Y).`

1.4/b `tomi_toxwn(X, Y) :-pixels_tomis(X, Y),
k4(NX, NY, X, Y),
tomi_toxwn(NX, NY).`

Με την διαδικασία αυτή το πρόγραμμα θα βρει αν υπάρχει διασταύρωση των δύο τόξων. Αυτό θα το διαπιστώσει αν υπάρχουν στο νοητό ψηφιακό επίπεδο 4 ψηφία σε διάταξη τέτοια που να σχηματίζουν ένα νοητό ψηφιακό τετράγωνο με τα διαγώνια

ψηφία να ανήκουν σε μια από τις δύο διαφορετικές εσωτερικές βάσεις $r_sintetagmenes(X, Y)$ ή $l_sintetagmenes(MX, Y)$.

Όσο οι τιμές των X και Y δεν είναι αντίστοιχα 13 και 0, δηλαδή, όσο το πρόγραμμα δεν έχει φθάσει στη ψηφιακή μονάδα επιφάνειας στο δεξί άκρο του ευθυγράμμου τμήματος, θα εκτελείται η δεύτερη επιλογή η οποία καλεί την εκτέλεση του κανόνα $pixels_tomis(X, Y)$:

```
1.4/b.1/a pixels_tomis(X, Y) :-k5(MX, MY, X, Y),
                                r_sintetagmenes(X, Y), l_sintetagmenes(MX, Y),
                                l_sintetagmenes(X, MY), r_sintetagmenes(MX, MY),
                                assert(pixels_tomis_toxwn(X, Y), tomi_toxwn),
                                assert(pixels_tomis_toxwn(MX, Y), tomi_toxwn),
                                assert(pixels_tomis_toxwn(X, MY), tomi_toxwn),
                                assert(pixels_tomis_toxwn(MX, MY), tomi_toxwn).
```

```
1.4/b.1/b pixels_tomis(X, Y) :-fail;true.
```

Η πρώτη επιλογή αυτού του κανόνα ικανοποιείται αν για τις τιμές X, Y που καθορίζουν μια θέση στο νοητό ψηφιακό επίπεδο, υπάρχουν οι τιμές MX, MY του διαγώνιου ψηφίου στην ίδια εσωτερική βάση που αντιστοιχεί στο δεξί νοητό τόξο και επιπλέον οι τιμές MX, Y και X, MY αντίστοιχα για το αριστερό και το κάτω ψηφίο (διαγώνια μεταξύ τους), να έχουν εγγραφεί στην άλλη εσωτερική βάση που αντιστοιχεί στο αριστερό νοητό τόξο. Οι τιμές MX ($MX=X+1$) και MY ($MY=Y-1$) διαμορφώνονται από την εντολή $k5(MX, MY, X, Y)$.

Στη συνέχεια ο κανόνας καταγράφει τις τέσσερις αυτές ψηφιακές γεωμετρικές μονάδες σε μια εσωτερική βάση δεδομένων. Εάν δεν ικανοποιείται η πρώτη επιλογή του κανόνα, τότε εκτελείται η δεύτερη επιλογή η οποία καθώς ισχύει σε κάθε περίπτωση, επιστρέφει το πρόγραμμα στον κανόνα $1.4/b\ tomi_toxwn(X, Y)$ για να συνεχίσει στην επόμενη θέση της νοητής ψηφιακής επιφάνειας, η οποία καθορίζεται από τον κανόνα $k4(NX, NY, X, Y)$:

```
1.4/b.2/a k4(NX, NY, X, Y) :-NX=X+1, NY=Y, NX<=13.
```

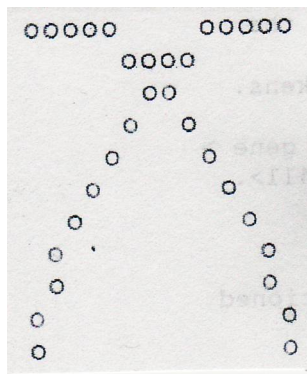
```
1.4/b.2/b k4(NX, NY, X, Y) :-NX=X-13, NY=Y-1.
```

Η εκτέλεση αυτού του κανόνα, δίνει τη δυνατότητα στο πρόγραμμα να σαρώσει όλη τη νοητή ψηφιακή επιφάνεια. Με την πρώτη επιλογή το πρόγραμμα εξαντλεί όλες τις θέσεις από αριστερά προς τα δεξιά για κάθε γραμμή Y και με τη δεύτερη μετακινείται διαδοχικά σε όλες τις γραμμές από πάνω προς τα κάτω. Μετά τον καθορισμό της νέας θέσης της ψηφιακής μονάδας και την επιστροφή του προγράμματος στο σημείο που εκλήθη ($1.4/b\ tomi_toxwn(X, Y)$) εκτελείται πάλι ο κανόνας σε βρόχο μέχρι να

γίνουν οι τιμές των X και Y αντίστοιχα 13 και 0. Όταν γίνει αυτό, οπότε το πρόγραμμα θα έχει φθάσει στη ψηφιακή μονάδα επιφάνειας που βρίσκεται στο δεξί άκρο του ευθυγράμμου τμήματος, ενεργοποιείται η πρώτη επιλογή `tomi_toxwn(X,Y)` :

```
1.4/a tomi_toxwn(X,Y) :-X=13,Y=0,  
pixels_tomis(X,Y).
```

η οποία και δίνει την εντολή να εκτελεστεί κατά τα γνωστά ο κανόνας `pixels_tomis(X,Y)`.



Σχήμα 2: Τα δύο ημικύκλια όπως τυπώνονται από το πρόγραμμα σχεδίασης

Στο σχήμα δίνονται τα δύο ημικύκλια όπως τυπώνονται από το πρόγραμμα σχεδίασης.

ΚΕΦΑΛΑΙΟ 8

Συμπεράσματα

Η παρούσα διατριβή κινήθηκε στο πλαίσιο της Γνωσιακής Επιστήμης με έμφαση στην Τεχνητή Νοημοσύνη και ειδικότερα την υλοποίηση Μεταγνωστικών Συστημάτων Απάντησης Ερωτήσεων.

Στόχος και συμβολή μας ήταν η δημιουργία Συστημάτων Απάντησης Ερωτήσεων (ΣΑΕ) τα οποία όσο το δυνατόν περισσότερο διευκολύνουν το χρήστη.

Το θεωρητικό πλαίσιο στο οποίο στηρίχτηκε η διατριβή ήταν ένα σώμα από άρθρα, υπολογιστικά μοντέλα και αποδείξεις θεωρημάτων της Γνωσιακής Επιστήμης, της Τεχνητής Νοημοσύνης, των Μαθηματικών και της Βιοϊατρικής (Flavell, 1974; M. Cox, 2005; Vosniadou & Kyriakopoulou, 2006; Veenman, Van Hout-Wolters & Afflerbach, 2006; Alevin, McLaren, Roll & Koedinger, 2006).

Έχοντας ως βασικό υλικό ποικίλα κείμενα σε ηλεκτρονική μορφή (π.χ. αποδείξεις θεωρημάτων του Ευκλείδη, βιοϊατρικά) κατασκευάστηκε ένα πρόγραμμα με μια νέα αρχιτεκτονική Συστήματος Απάντησης Ερωτήσεων (ΣΑΕ) που διαθέτει λειτουργίες εμπνευσμένες από τις «μεταγνωσιακές λειτουργίες» (metacognition).

Η κύρια συμβολή της διατριβής ήταν η κατασκευή και λειτουργία ενός ΣΑΕ του οποίου η αρχιτεκτονική στηρίζεται στη μεταγνωστική διαχείριση στρατηγικών για την αντιμετώπιση των εισαγομένων ερωτήσεων. Επιπλέον διαθέτει ισχυρή προσαρμοστικότητα λόγω της απευθείας ανάλυσης του κειμένου δίχως προηγούμενη μετάφραση σε τυπική γλώσσα με εφαρμογή σε πολύ διαφορετικά πεδία γνώσης όπως της Γεωμετρίας και της Βιοϊατρικής, δεδομένου ότι είναι σε θέση να απαντά ερωτήσεις με την εξαγωγή γνώσης απευθείας από κείμενα δίχως προηγούμενη μετατροπή τους σε τυπική αναπαράσταση. Ο μηχανισμός εξαγωγής γνώσης του συστήματος χρησιμοποιείται και για την αυτόματη δημιουργία «οντολογικών» γνώσεων. Ένας άλλος μηχανισμός που υλοποιήθηκε ως μέρος του συστήματος είναι ο μηχανισμός παραγωγής επεξηγήσεων των απαντήσεων. Συγκεκριμένα οι επεξηγήσεις αυτές που παράγονται αυτόματα αποδεικνύουν την μεταγνωστική λειτουργία του συστήματος. Δηλαδή με την παρακολούθηση και καταγραφή της απάντησης μιας ερώτησης δημιουργείται αυτομάτως κείμενο που υποστηρίζει την ορθότητα της απάντησης. Να σημειωθεί ότι στην περίπτωση που χρησιμοποιούνται ως υλικό κείμενα του Ευκλείδη, το σύστημα ενισχύει σχηματικά-διαγραμματικά την επεξήγηση που δίνεται στην απάντηση της ερώτησης που εισάγεται σε αυτό. Σε

γενικές γραμμές, το σύστημα επιτελεί «μεταγνωστική» (*metagnostic*) απάντηση ερωτήσεων από κείμενα σε οποιουδήποτε διαφορετικούς τομείς.

Το σύστημα αυτό θα μπορούσε να εφαρμοστεί στο χώρο της εκπαίδευσης ενισχύοντας αποτελεσματικά την εκμάθηση τόσο της Γεωμετρίας όσο και άλλων γνωστικών αντικειμένων. Επίσης να έχει εφαρμογή στην παιδαγωγική διαδικασία των μεθόδων της δημιουργικής μάθησης (ανακαλυπτική διαδικασία).

Επιπλέον μπορεί να εφαρμοστεί στη βιοϊατρική εκπαίδευση για την απάντηση ερωτήσεων από βιοϊατρικά κείμενα για τη χρήση των ερευνητών ή και του βιοϊατρικού προσωπικού.

Μια άλλη εξέλιξη του συστήματος θα ήταν η προσθήκη ενός υποσυστήματος τεχνητής όρασης το οποίο θα συνδύαζε την οπτική ανάλυση των γεωμετρικών σχημάτων με την αποδεικτική διαδικασία.

ΠΑΡΑΡΤΗΜΑ

1. Πρόγραμμα Ανάλυσης Κειμένου

```
1/a.3 openfileeuclidperiods:-openread(periods,"euclid.pro").
1/a.2 openfilesentences:-openappend(sentence,"sentence.pro").
1/a.1 clearfilesentences:-openwrite(sentence,"sentence.pro"),
    closefile(sentence).
2/a.1 clearfileverbless:-openwrite(verbless,"verbless.pro"),
    closefile(verbless).
2/a.2 openfileverbless:-openappend(verbless,"verbless.pro").
2/a.3 clearfilephrase:-openwrite(phrase,"phrase.pro"),
    closefile(phrase).
2/a.4 openfilephrase:-openappend(phrase,"phrase.pro").
2/a.5 openreadfilesentences:-openread(sentence,"sentence.pro").
1/a.4 tokenisation_periods(N):-not(eof(periods)),
    tokenisation,
    inputperiod(N,S),nl,
    process(N,S).
2/a.6 tokenisation_sentence(N):-not(eof(sentence)),
    tokenisation,
    inputsentence(N,S),nl,
    process(N,S).
1/a.4.2 tokenisation:-retractall(w(_,_,_)),retractall(lastw(_)).
1/a.4.3 inputperiod(N,S):-readdevice(periods),
    readint(N),nl,
    readln(S),nl,
    readdevice(keyboard).
2/a.6.3 inputsentence(N,S):-readdevice(sentence),
    readint(N),nl,
    readln(S),nl,
    readdevice(keyboard).
1/a.4.4 process(SN,S):-pr1(SN,S,1).
1/a.4.4.1/a.2/a pr(SN,S,N):-f(S,W,""),
    sto(SN,N,W).
1/a.4.4.1/a.2/b pr(SN,S,N):-f(S,W,T),
    sto(SN,N,W),
    M=N+1,
    retractall(lastw(_)),assert(lastw(N)),
    pr0(T,M,SN).
1/a.4.4.1/a.2/b.5/a pr0(S,N,SN):-f(S,W,T),N=3,W="and",
    pr0(T,N,SN).
1/a.4.4.1/a.2/b.5/b pr0(S,N,SN):-f(S,W,T),
```

```

                                sto(SN,N,W),M=N+1,
                                retractall(lastw(_)),assert(lastw(N)),
                                pr0(T,M,SN).

1/a.4.4.1/a.2/b.5/c      pr0("",_,_).

1/a.4.4.1/a pr1(SN,S,N):-S=". . and things which are equal
                                to the same thing are also equal
                                to one another semicolon . .",
                                pr(SN,S,N),!.

1/a.4.4.1/b pr1(SN,S,N):-f(S,W,""),
                                sto(SN,N,W).

1/a.4.4.1/c pr1(SN,S,N):-f(S,W,T),
                                sto(SN,N,W),
                                M=N+1,
                                retractall(lastw(_)),assert(lastw(N)),
                                pr11(T,M,SN).

1/a.4.4.1/c.5/a      pr11(S,N,SN):-f(S,W,T),
                                not(indiferent(W)),
                                sto(SN,N,W),M=N+1,
                                retractall(lastw(_)),assert(lastw(N)),
                                pr11(T,M,SN).

1/a.4.4.1/c.5/b      pr11(S,N,SN):-f(S,W,T),
                                indiferent(W),
                                pr11(T,N,SN).

1/a.4.4.1/c.5/c      pr11("",_,_).

1/a.4.4.1/a.2/a.1 f(S,W,T):-fronttoken(S,W,T).

1/a.4.4.1/a.2/a.2 sto(SN,WN,W):-assert(w(SN,WN,W)).

1/a.5/a.2/a p(S,vL,NP):-w(S,N,V),
                                S=1,
                                word(XN),N>XN,
                                V="fullstop",
                                writedevic(sentence),
                                write(S),nl,
                                N1=N-1,ws(1,N1,NP),
                                writedevic(sentence),
                                write(". ."),write(NP),write(" . ."),nl,
                                retractall(numv(_)),assert(numv(N)).

1/a.5/a.2/b p(S,vL,NP):-w(S,N,V),
                                S=0,
                                word(XN),N>XN,
                                V="fullstop",
                                writedevic(sentence),
                                write(S),nl,
                                N1=N-1,ws(1,N1,NP),
                                writedevic(sentence),
                                write(". ."),write(NP),write(" . ."),nl,
                                retractall(numv(_)),assert(numv(N)).

1/a.5/a.2/c p(S,vL,NP):-w(S,N,V),

```

```

S<>1,S<>0,
word(XN),N>XN,not(w(S,_, "comma")),
V="fullstop",
writedevice(sentence),
S1=(10*S)+1,write(S1),nl,
N1=N-1,
ws(1,N1,NP),
writedevice(sentence),
write(". ."),write(NP),write(" . ."),nl,
retractall(numv(_)),assert(numv(N)).

```

```

1/a.5/a.2/d p(S,vL,NP):-w(S,N,V),
word(XN),N>XN,not(w(S,_, "comma")),
V="semicolon",
writedevice(sentence),
S1=(10*S)+1,write(S1),nl,
N1=N-1,ws(1,N1,NP),
write(". ."),write(NP),write(" . ."),nl,
retractall(numv(_)),assert(numv(N)).

```

```

1/a.5/a.2/e p(S,vL,NP):-w(S,N,V),
word(XN),N>XN,
V="comma",!,
retractall(numv(_)),assert(numv(XN)),
N1=N-1,
retractall(lexi(_, _)),
ws(1,N1,NP),
retractall(word(_)),assert(word(N)),
verbexistance(S,vL,NP,N1).

```

```

1/a.5/a.2/f p(S,vL,NP):-w(S,N,V),
word(XN),N>XN,
V="fullstop",
w(S,XN,"comma"),
not(lexi(_,is)),not(lexi(_,are)),
writedevice(sentence),
S1=(10*S)+1,write(S1),nl,
N1=N-1,ws(1,N1,NP),
writedevice(sentence),
write(". ."),write(NP),write(" . ."),nl,
retractall(numv(_)),assert(numv(N)).

```

```

1/a.5/a.2/g p(S,vL,NP):-w(S,N,V),
word(XN),N>XN,
V="semicolon",
w(S,XN,"comma"),
not(lexi(_,is)),not(lexi(_,are)),
writedevice(sentence),
S1=(10*S)+1,write(S1),nl,
N1=N-1,ws(1,N1,NP),
write(". ."),write(NP),write(" . ."),nl,
retractall(numv(_)),assert(numv(N)).

```

```

1/a.5/a.3 p(S,vR,NP):-w(S,N,V),
numv(XN),N>XN,
V="comma",!,
writedevice(sentence),
S2=(10*S)+2,write(S2),nl,

```

```

        lastw(N2),
        N1=N+1,
        wo(N1,N2,NP),
        write(" . . "),write(NP),write(" . . "),nl,
        nl,write(N,"=",V),nl,write(" ",NP),nl.

2/a.7/a.2/a protasi(S,vL,NP):-S=1,
        w(S,N2,_),lastw(N2),wo(1,N2,NP),
        writedevic(verbless),write(S),
        writedevic(phrase),write(S),
        writedevic(screen),write(S),!.

2/a.7/a.2/b protasi(S,vL,NP):-S=0,
        w(S,N2,_),lastw(N2),wo(1,N2,NP),
        writedevic(verbless),write(S),
        writedevic(phrase),write(S),
        writedevic(screen),write(S),!.

2/a.7/a.2/c protasi(S,vL,NP):-w(S,N,V),
        word(XN),N>XN,
        isalexi(V,verb),
        N1=N-1,ws(1,N1,NP),
        writedevic(verbless),
        write(S),nl,write(NP),nl,
        writedevic(phrase),
        write(S),nl,write(NP),write(" ",V," "),
        retractall(word(_)),assert(word(N)).

2/a.7/a.4 protasi(S,vR,NP):-w(S,N,V),
        word(XN),N>XN,
        isalexi(V,verb),
        lastw(N2),
        N1=N+1,wo(N1,N2,NP),
        writedevic(verbless),
        write(V),nl,write(NP),nl,
        writedevic(phrase),write(NP),nl,

1/a.5/a getp(N):-retractall(word(_)),assert(word(0)),
        p(N,vL,X),
        p(N,vR,Y).
1/a.5/a getp(_).

2/a.7/a get_phrases(N):-retractall(word(_)),assert(word(0)),
        protasi(N,vL,X),
        retractall(word(_)),assert(word(0)),
        protasi(N,vR,Y).
2/a.7/b get_phrases(_).

1/a sentences:-clearfilesentences,
        openfilesentences,
        openfileeuclidperiods,
        tokenisation_periods(N),
        getp(N),
        loopsentences.
1/b sentences:-loopsentences.
1/c sentences:-closefile(periods),
        closefile(sentence).

2/a phrases:-clearfileverbless,
        openfileverbless,

```

```

clearfilephrase,
openfilephrase,
openreadfilesentences,
tokenisation_sentence(N),
get_phrases(N),
loop_phrases.
2/b phrases:-loop_phrases.
2/c phrases:-closefile(sentence),
closefile(verbless),
closefile(phrase).

1/a.6 loopsentences:-retractall(word(_)),
tokenisation_periods(N),
getp(N),
loopsentences.

2/a.8 loop_phrases:-retractall(word(_)),
tokenisation_sentence(N),
get_phrases(N),
loop_phrases.

1/a.5/a.2/c.8/a ws(1,N,X):-w(_,N,W),not(stop(W)),
assert(lexi(N,W)),
M=N-1,
ws(1,M,Y),!,
c(Y," ",YY),c(YY,W,X).

1/a.5/a.2/c.8/b ws(1,N,X):-N>1,X="".

1/a.5/a.3.8/a wo(N1,N2,X):-N2>N1+1,
w(_,N1,W),not(stop(W)),
M=N1+1,
wo(M,N2,Y),!,
c(W," ",Wb),c(Wb,Y,X).

1/a.5/a.3.8/a.4/a wo(N1,N2,X):-N=N1+1,w(N,N2,X).
1/a.5/a.3.8/a.4/b wo(N1,N2,X):-N2>N1,X="".

1/a.5/a.2/e.9/a
verbexistance(S,vL,_,N):-not(lexi(_,is)),not(lexi(_,are)),
NV=N+1,
retractall(word(_)),assert(word(NV)),
p(S,vL,_,!).

1/a.5/a.2/e.9/a
verbexistance(S,_,T,N):-writedevic(sentence)
S1=(10*S)+1,write(S1),nl,
write(". ."),write(T),write(" . ."),nl,
writedevic(screen),write("sentence:",S1),nl,
M=N+1,
w(S,M,V),
write(M,"=",V),nl,write(T).

goal

1 sentences,
2 phrases.

```

2. Πρόγραμμα Εξαγωγής Γνώσης

```
openreadfilesentences:-openread(sentence,"sentence.pro").

extraction_knowledge:-
  openreadfilesentences,not(eof(sentence)),
  readdevice(sentence),
  readint(_),readln(S),
  extract_from(S),
  loop_extraction_knowledge,
  readdevice(keyboard).
extraction_knowledge:-loop_extraction_knowledge.
extraction_knowledge:-closefile(sentence).

extract_from(S):-
  f(S,".",T1),f(T1,".",T2),
  np1(T2,Hypernym1,Entity1,T3),f(T3,V,T4),%Entity1=a
  np2(T4,Hypernym2,Concept,Entity2,T5),%Hypernym2=centre,
  Concept=circle,Entity2=cdb
  f(T5,".",T6),f(T6,".",""),%Concept=circle,Entity2=cdb
  V=is,Hypernym1=point,Hypernym2=centre,
  dos_file_extraction1(Concept,Entity2,Entity1).
extract_from(_).

np1(T1,H1,H2,T4):-f(T1,the,T2),f(T2,H1,T3),
  f(T3,H2,T4).
np2(T1,H1,H2,E1,T5):-f(T1,the,T2),f(T2,H1,T3),f(T3,of,T4),
  np1(T4,H2,E1,T5).

dos_file_extraction1(Circle,Name,CEntre):-
  frontchar(Name,H1,T),frontchar(T,H2,TT),
  str_char(C,H1),str_char(D,H2),
  circle_construction(Circle,C,D,TT,CEntre),
  oncircle_construction(C,D,TT),
  synnyma_construction(C,D,TT,CEntre),
  endsline_construction(C,D,TT,CEntre).

f(String,HeadWord,Tail):-fronttoken(String,HeadWord,Tail).

circle_construction(Circle,C,D,B,A):-
  concat(C,D,CD),concat(CD,B,CDB),
  assert(isa_circle(Circle,CDB,A),circles),

  concat(D,B,DB),concat(DB,C,DBC),
  assert(isa_circle(Circle,DBC,A),circles),

  concat(B,C,BC),concat(BC,D,BCD),
  assert(isa_circle(Circle,BCD,A),circles),

  save("circles.dba",circles).

oncircle_construction(C,D,B):-
  concat(C,D,CD),concat(CD,B,CDB),
  assert(isa_oncircle(oncircle,CDB,C),oncircles),
  assert(isa_oncircle(oncircle,CDB,D),oncircles),
  assert(isa_oncircle(oncircle,CDB,B),oncircles),

  concat(D,B,DB),concat(DB,C,DBC),
  assert(isa_oncircle(oncircle,DBC,C),oncircles),
  assert(isa_oncircle(oncircle,DBC,D),oncircles),
```



```

assert (isa_oncicle(oncicle,DBC,B) , oncircles) ,

concat (B,C,BC) , concat (BC,D,BCD) ,
assert (isa_oncicle(oncicle,BCD,C) , oncircles) ,
assert (isa_oncicle(oncicle,BCD,D) , oncircles) ,
assert (isa_oncicle(oncicle,BCD,B) , oncircles) ,

save ("oncicle.dba" , oncircles) .

synwnyma_construction(C,D,B,A) :-
concat (C,A,CA) , concat (D,A,DA) , concat (B,A,BA) ,
concat (A,C,AC) , concat (A,D,AD) , concat (A,B,AB) ,
assert (isa_synwnyma (synwnyma,CA,AC) , synwnyms) ,
assert (isa_synwnyma (synwnyma,AC,CA) , synwnyms) ,
assert (isa_synwnyma (synwnyma,DA,AD) , synwnyms) ,
assert (isa_synwnyma (synwnyma,AD,DA) , synwnyms) ,
assert (isa_synwnyma (synwnyma,BA,AB) , synwnyms) ,
assert (isa_synwnyma (synwnyma,AB,BA) , synwnyms) ,
save ("synwnyms.dba" , synwnyms) .

synwnyma_construction(C,D,B,A) :-
concat (C,A,CA) , concat (D,A,DA) , concat (B,A,BA) ,
concat (A,C,AC) , concat (A,D,AD) , concat (A,B,AB) ,
assert (isa_synwnyma (synwnyma,CA,AC) , synwnyms) ,
assert (isa_synwnyma (synwnyma,AC,CA) , synwnyms) ,
assert (isa_synwnyma (synwnyma,DA,AD) , synwnyms) ,
assert (isa_synwnyma (synwnyma,AD,DA) , synwnyms) ,
assert (isa_synwnyma (synwnyma,BA,AB) , synwnyms) ,
assert (isa_synwnyma (synwnyma,AB,BA) , synwnyms) ,
save ("synwnyms.dba" , synwnyms) .

endsline_construction(C,D,B,A) :-
concat (C,A,CA) , concat (D,A,DA) , concat (B,A,BA) ,
concat (A,C,AC) , concat (A,D,AD) , concat (A,B,AB) ,
assert (isa_endsline(endsline,CA,C,A) , endslines) ,
assert (isa_endsline(endsline,AC,A,C) , endslines) ,
assert (isa_endsline(endsline,DA,D,A) , endslines) ,
assert (isa_endsline(endsline,AD,A,D) , endslines) ,
assert (isa_endsline(endsline,BA,B,A) , endslines) ,
assert (isa_endsline(endsline,AB,A,B) , endslines) ,
save ("endsline.dba" , endslines) .

loop_extraction_knowledge:-
readint(_) , readln(S) ,
extract_from(S) ,
loop_extraction_knowledge.

```

3. Πρόγραμμα Απάντησης Ερωτήσεων

```

pk(15,def15," a circle is a plane figure contained by one line such
that all the straight lines falling upon it from one point among
those lying within the figure are equal to one another ").
pk(15,def15a," circle").
pk(15,def15b,"is a plane figure contained by one line such that all
the straight lines").
pk(15,def15c,"falling upon it from one point").
pk(15,def15d,"among those lying within the figure are equal to one
another").

pk(16,def16," and the point is called the centre of the circle ").

```

```

pk(16,def16a,"where the point").
pk(16,def16b,"is the centre of the circle").

pk(29,cn1," things which are equal to the same thing are equal to one
another ").

pk(30,cn2," if equals be added to equals the wholes are equal ").

pk(31,cn3," if equals be subtracted from equals the remainders are
equal ").

pk(32,cn4," things which coincide with one another are equal to one
another ").

pk(33,cn5," the whole is greater than the part ").

question(Question):-
    recordtheanswers,
    extract_question(Question),
    strategy(Synonym,LSide,RSide,Inversion,Relstrategy,Pk,strategy1),
    retractall(oldstrategy(_)),
    assert(oldstrategy(strategy1)),
    process_answer(strategy1).

question(Question):-
    extract_question(Question),
    oldstrategy(OldS),
    next(OldS,Strategy),
    questionloop(Strategy).
question(Question):-nl,write("impossible to answer ",Question),nl.

recordtheanswers:-
    nl,write("reset file qanswers.txt ? y(es) or n(o)"),
    nl,readln(A),
    preference(A).

preference(A):-A=y,
    openwrite(outfile,"c:\\prolog\\qanswers.txt"),
    closefile(outfile).
preference(A):-
    A=n.
preference(A):-
    A<>"y",A<>"n",
    recordtheanswers.

questionloop(Strategy):-
    strategy(Synonym,LSide,RSide,Inversion,Relstrategy,Pk,Strategy),
    oldstrategy(OldS),
    nl,write("TRYING STRATEGY :",Strategy),
    nl,write(" with ",Synonym," ",LSide," ",RSide," ",Inversion,"
",Relstrategy," ",Pk),
    nl,write("BECAUSE ",OldS," FAILED"),
    readln(_),
    retractall(oldstrategy(_)),
    assert(oldstrategy(Strategy)),
    process_answer(Strategy).

questionloop(_):-
    oldstrategy(OLDS),
    next(OLDS,NewStrategy),
    questionloop(NewStrategy).

```

```

extract_question(Question):-
  f(Question,why,Rest1),
  f(Rest1,LEntity,Rest2),
  f(Rest2,is,Rest3),f(Rest3,Relation,Rest4),f(Rest4,Prep,Rest5),
  f(Rest5,REntity,""),
  nl,write("The question is :<","LEntity,"> is <","Relation,">
<","Prep,"> <","REntity,"> ?"),
  relation(Relation),preposition(Prep),
  retractall(leftentityofquestion(_)),
  assert(leftentityofquestion(LEntity)),
  retractall(relationofquestion(_)),
  assert(relationofquestion(Relation)),
  retractall(prepositionofquestion(_)),
  assert(prepositionofquestion(Prep)),
  retractall(rightentityofquestion(_)),
  assert(rightentityofquestion(REntity)).

f(X,Y,Z):-fronttoken(X,Y,Z).
c(X,Y,Z):-concat(X,Y,Z).

process_answer(strategy1):-
  strategy(Synonym,_,_,Inversion,_,Pk,strategy1),
  Synonym="nosynonymes",Inversion="noinversion",Pk="nopk",
  compose_question(Q,Synonym,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  pk(29,cn1,K1),PrevQ<>K1,
  pk(31,cn3,K3),PrevQ<>K3,
  expl_answer(PrevQ).

process_answer(strategy2):-
  strategy(Synonym,_,_,Inversion,_,Pk,strategy2),
  Synonym="nosynonymes",Inversion="meinversion",Pk="nopk",
  compose_question(Q,Synonym,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  pk(29,cn1,K1),PrevQ<>K1,
  pk(31,cn3,K3),PrevQ<>K3,
  expl_answer(PrevQ).

process_answer(strategy3):-
  strategy(Synonym,LSide,_,Inversion,_,Pk,strategy3),
  Synonym="mesynonymes",LSide="left",Inversion="noinversion",Pk="nopk",
  compose_question(Q,Synonym,LSide,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  pk(29,cn1,K1),PrevQ<>K1,
  pk(31,cn3,K3),PrevQ<>K3,
  expl_answer(PrevQ).

process_answer(strategy4):-
  strategy(Synonym,LSide,_,Inversion,_,Pk,strategy4),
  ynonym="mesynonymes",LSide="left",Inversion="meinversion",Pk="nopk",
  compose_question(Q,Synonym,LSide,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  pk(29,cn1,K1),PrevQ<>K1,
  pk(31,cn3,K3),PrevQ<>K3,
  expl_answer(PrevQ).

```

```

process_answer(strategy5):-
  strategy(Synonym,_,RSide,Inversion,_,Pk,strategy5),
  Synonym="mesynonymes",RSide="right",Inversion="noinversion",Pk="nopk"
,
  compose_question(Q,Synonym,RSide,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  pk(29,cn1,K1),PrevQ<>K1,
  pk(31,cn3,K3),PrevQ<>K3,
  expl_answer(PrevQ).

process_answer(strategy6):-
  strategy(Synonym,_,RSide,Inversion,_,Pk,strategy6),

ynonym="mesynonymes",RSide="right",Inversion="meinversion",Pk="nopk",
  compose_question(Q,Synonym,RSide,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  pk(29,cn1,K1),PrevQ<>K1,
  pk(31,cn3,K3),PrevQ<>K3,
  expl_answer(PrevQ).

process_answer(strategy7):-
  strategy(Synonym,LSide,RSide,Inversion,_,Pk,strategy7),

ynonym="mesynonymes",LSide="left",RSide="right",Inversion="noinversion",Pk="nopk",
  compose_question(Q,Synonym,LSide,RSide,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  pk(29,cn1,K1),PrevQ<>K1,
  pk(31,cn3,K3),PrevQ<>K3,
  expl_answer(PrevQ).

process_answer(strategy8):-
  strategy(Synonym,LSide,RSide,Inversion,_,Pk,strategy8),

ynonym="mesynonymes",LSide="left",RSide="right",Inversion="meinversion",Pk="nopk",
  compose_question(Q,Synonym,LSide,RSide,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  pk(29,cn1,K1),PrevQ<>K1,
  pk(31,cn3,K3),PrevQ<>K3,
  expl_answer(PrevQ).

process_answer(strategy9):-
  strategy(Synonym,_,_,Inversion,_,Pk,strategy9),
  Synonym="nosynonymes",Inversion="noinversion",Pk="pk",
  compose_question(Q,Synonym,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  prerequisiteknowledge(PrevQ).

process_answer(strategy10):-
  strategy(Synonym,_,_,Inversion,_,Pk,strategy10),
  Synonym="nosynonymes",Inversion="meinversion",Pk="pk",
  compose_question(Q,Synonym,Inversion),
  same_relation,
  previous_phrase_in_text(Q,PrevQ),
  prerequisiteknowledge(PrevQ).

```

```

process_answer(strategy11):-
strategy(Synonym,LSide,_,Inversion,_,Pk,strategy11),
Synonym="mesynonymes",LSide="left",Inversion="noinversion",Pk="pk",
compose_question(Q,Synonym,LSide,Inversion),
same_relation,
previous_phrase_in_text(Q,PrevQ),
prerequisiteknowledge(PrevQ).

process_answer(strategy12):-
strategy(Synonym,LSide,_,Inversion,_,Pk,strategy12),
Synonym="mesynonymes",LSide="left",Inversion="meinversion",Pk="pk",
compose_question(Q,Synonym,LSide,Inversion),
same_relation,
previous_phrase_in_text(Q,PrevQ),
prerequisiteknowledge(PrevQ).

process_answer(strategy13):-
strategy(Synonym,_,RSide,Inversion,_,Pk,strategy13),
Synonym="mesynonymes",RSide="right",Inversion="noinversion",Pk="pk",
compose_question(Q,Synonym,RSide,Inversion),
same_relation,
previous_phrase_in_text(Q,PrevQ),
prerequisiteknowledge(PrevQ).

process_answer(strategy14):-
strategy(Synonym,_,RSide,Inversion,_,Pk,strategy14),
Synonym="mesynonymes",RSide="right",Inversion="meinversion",Pk="pk",
compose_question(Q,Synonym,RSide,Inversion),
same_relation,
previous_phrase_in_text(Q,PrevQ),
prerequisiteknowledge(PrevQ).

process_answer(strategy15):-
strategy(Synonym,LSide,RSide,Inversion,_,Pk,strategy15),
Synonym="mesynonymes",LSide="left",RSide="right",Inversion="noinversi
on",Pk="pk",
compose_question(Q,Synonym,LSide,RSide,Inversion),
same_relation,
previous_phrase_in_text(Q,PrevQ),
prerequisiteknowledge(PrevQ).

process_answer(strategy16):-
strategy(Synonym,LSide,RSide,Inversion,_,Pk,strategy16),
Synonym="mesynonymes",LSide="left",RSide="right",Inversion="meinversi
on",Pk="pk",
compose_question(Q,Synonym,LSide,RSide,Inversion),
same_relation,
previous_phrase_in_text(Q,PrevQ),
prerequisiteknowledge(PrevQ).

prerequisiteknowledge(PrevQ):-
pk(29,cn1,K1),
PrevQ=K1,
implicit_knowledge(PrevQ,cn1).

prerequisiteknowledge(PrevQ):-pk(31,cn3,K3),
PrevQ=K3,
implicit_knowledge(PrevQ,cn3).

```

```

equalinesegmentsradii (LEntity,REntity) :-
endslinesegment (LEntity,End1,End2),endslinesegment (REntity,End3,End4)
,
  End1=End3,
  oncircle (Circle,End2),oncircle (Circle,End4),
  circles (Circle,End1),
  pk (15,def15a,DEF15a),pk (15,def15b,DEF15b),pk (15,def15c,DEF15c),
  pk (15,def15d,DEF15d),
  pk (16,def16a,DEF16a),pk (16,def16b,DEF16b),
  nl,write("<"LEntity,">="REntity,">"),
  nl,write("because the ends of <"LEntity,"> are <"End1,">,
<"End2,">"),
  nl,write("and the ends of <"REntity,"> are <"End3,">,
<"End4,">"),
  nl,write("and the common end <"End1,"> is the center of the circle
<"Circle,">."),
  nl,write("That is, <"End2,">, <"End4,"> are on the circumference
of the same circle,"),
  nl,write("and therefore <"LEntity,">, <"REntity,"> are equal as
radii on the same circle."),nl,
  nl,write("The above follows from the definition of the circle."),
  nl,write("In this case,"DEF15a," <"Circle,">"),
  nl,write(DEF15b," <"LEntity,">, <"REntity,">"),
  nl,write(DEF15c," <"End1,">"),
  nl,write(DEF15d," i.e. <"LEntity,">="REntity,">"),
  nl,write(DEF16a," <"End1,"> ",DEF16b,"."),nl,nl,
  closefile (outfile),
  writedevic (screen),
  nl,write("<"LEntity,">="REntity,">"),
  nl,write("because the ends of <"LEntity,"> are <"End1,">,
<"End2,">"),
  nl,write("and the ends of <"REntity,"> are <"End3,">,
<"End4,">"),
  nl,write("and the common end <"End1,"> is the center of the circle
<"Circle,">."),
  nl,write("That is, <"End2,">, <"End4,"> are on the circumference
of the same circle,"),
  nl,write("and therefore <"LEntity,">, <"REntity,"> are equal as
radii on the same circle."),nl,
  nl,write("The above follows from the definition of the circle."),
  nl,write("In this case,"DEF15a," <"Circle,">"),
  nl,write(DEF15b," <"LEntity,">, <"REntity,">"),
  nl,write(DEF15c," <"End1,">"),
  nl,write(DEF15d," i.e. <"LEntity,">="REntity,">"),
  nl,write(DEF16a," <"End1,"> ",DEF16b,"."),nl,nl.

equalinesegmentsradii (LEntity,REntity) :-
endslinesegment (LEntity,End1,End2),endslinesegment (REntity,End3,End4)
,
  End2=End4,
  oncircle (Circle,End1),oncircle (Circle,End3),
  circles (Circle,End2),
  pk (15,def15a,DEF15a),pk (15,def15b,DEF15b),pk (15,def15c,DEF15c),
  pk (15,def15d,DEF15d),
  pk (16,def16a,DEF16a),pk (16,def16b,DEF16b),
  nl,write("<"LEntity,">="REntity,">"),
  nl,write("because the ends of <"LEntity,"> are <"End1,">,
<"End2,">"),
  nl,write("and the ends of <"REntity,"> are <"End3,">,
<"End4,">"),

```

```

    nl,write("and the common end <\",End2,\"> is the center of the circle
<\",Circle,\">\"),
    nl,write("That is, <\",End1,\">, <\",End3,\"> are on the circumference
of the same circle,\"),
    nl,write("and therefore <\",LEntity,\">, <\",REntity,\"> are equal as
radii on the same circle.\"),nl,
    nl,write("The above follows from the definition of the circle.\"),
    nl,write("In this case,\"DEF15a,\" <\",Circle,\">\"),
    nl,write(DEF15b,\" <\",LEntity,\">, <\",REntity,\">\"),
    nl,write(DEF15c,\" <\",End2,\">\"),
    nl,write(DEF15d,\" i.e. <\",LEntity,\">=<\",REntity,\">\"),
    nl,write(DEF16a,\" <\",End2,\"> \",DEF16b,\".\"),nl,nl,
    closefile(outfile),
    writedevic(screen),
    nl,write("<\",LEntity,\">=<\",REntity,\">\"),
    nl,write("because the ends of <\",LEntity,\"> are <\",End1,\">,
<\",End2,\">\"),
    nl,write("and the ends of <\",REntity,\"> are <\",End3,\">,
<\",End4,\">\"),
    nl,write("and the common end <\",End2,\"> is the center of the circle
<\",Circle,\">\"),
    nl,write("That is, <\",End1,\">, <\",End3,\"> are on the circumference
of the same circle,\"),
    nl,write("and therefore <\",LEntity,\">, <\",REntity,\"> are equal as
radii on the same circle.\"),nl,
    nl,write("The above follows from the definition of the circle.\"),
    nl,write("In this case,\"DEF15a,\" <\",Circle,\">\"),
    nl,write(DEF15b,\" <\",LEntity,\">, <\",REntity,\">\"),
    nl,write(DEF15c,\" <\",End2,\">\"),
    nl,write(DEF15d,\" i.e. <\",LEntity,\">=<\",REntity,\">\"),
    nl,write(DEF16a,\" <\",End2,\"> \",DEF16b,\".\"),nl,nl.

```

```

equalinesegmentsradii (LEntity,REntity):-
endslinesegment (LEntity,End1,End2),endslinesegment (REntity,End3,End4)

```

```

,
    End1=End4,
    oncircle(Circle,End2),oncircle(Circle,End3),
    circles(Circle,End1),
    pk(15,def15a,DEF15a),pk(15,def15b,DEF15b),pk(15,def15c,DEF15c),
    pk(15,def15d,DEF15d),
    pk(16,def16a,DEF16a),pk(16,def16b,DEF16b),
    nl,write("<\",LEntity,\">=<\",REntity,\">\"),
    nl,write("because the ends of <\",LEntity,\"> are <\",End1,\">,
<\",End2,\">\"),
    nl,write("and the ends of <\",REntity,\"> are <\",End3,\">,
<\",End4,\">\"),
    nl,write("and the common end <\",End1,\"> is the center of the circle
<\",Circle,\">\"),
    nl,write("That is, <\",End2,\">, <\",End3,\"> are on the circumference
of the same circle,\"),
    nl,write("and therefore <\",LEntity,\">, <\",REntity,\"> are equal as
radii on the same circle\"),nl,
    nl,write("The above follows from the definition of the circle.\"),
    nl,write("In this case,\"DEF15a,\" <\",Circle,\">\"),
    nl,write(DEF15b,\" <\",LEntity,\">, <\",REntity,\">\"),
    nl,write(DEF15c,\" <\",End1,\">\"),
    nl,write(DEF15d,\" i.e. <\",LEntity,\">=<\",REntity,\">\"),
    nl,write(DEF16a,\" <\",End1,\"> \",DEF16b,\".\"),nl,nl,
    closefile(outfile),
    writedevic(screen),
    nl,write("<\",LEntity,\">=<\",REntity,\">\"),

```

```

    nl,write("because the ends of <","LEntity,"> are <","End1,">,
<","End2,">"),
    nl,write("and the ends of <","REntity,"> are <","End3,">,
<","End4,">"),
    nl,write("and the common end <","End1,"> is the center of the circle
<","Circle,">"),
    nl,write("That is, <","End2,">, <","End3,"> are on the circumference
of the same circle,"),
    nl,write("and therefore <","LEntity,">, <","REntity,"> are equal as
radii on the same circle"),nl,
    nl,write("The above follows from the definition of the circle."),
    nl,write("In this case,","DEF15a," <","Circle,">"),
    nl,write(DEF15b," <","LEntity,">, <","REntity,">"),
    nl,write(DEF15c," <","End1,">"),
    nl,write(DEF15d," i.e. <","LEntity,">=<","REntity,">"),
    nl,write(DEF16a," <","End1,"> ",DEF16b,"."),nl,nl.

```

```

equalinesegmentsradii (LEntity,REntity):-
endslinesegment (LEntity,End1,End2),endslinesegment (REntity,End3,End4)

```

```

,
    End2=End3,
    oncircle(Circle,End1),oncircle(Circle,End4),
    circles(Circle,End2),
    pk(15,def15a,DEF15a),pk(15,def15b,DEF15b),pk(15,def15c,DEF15c),
    pk(15,def15d,DEF15d),
    pk(16,def16a,DEF16a),pk(16,def16b,DEF16b),
    nl,write("<","LEntity,">=<","REntity,">"),
    nl,write("because the ends of <","LEntity,"> are <","End1,">,
<","End2,">"),
    nl,write("and the ends of <","REntity,"> are <","End3,">,
<","End4,">"),
    nl,write("and the common end <","End2,"> is the center of the circle
<","Circle,">"),
    nl,write("That is, <","End1,">, <","End4,"> are on the circumference
of the same circle,"),
    nl,write("and therefore <","LEntity,">, <","REntity,"> are equal as
radii on the same circle"),nl,
    nl,write("The above follows from the definition of the circle."),
    nl,write("In this case,","DEF15a," <","Circle,">"),
    nl,write(DEF15b," <","LEntity,">, <","REntity,">"),
    nl,write(DEF15c," <","End2,">"),
    nl,write(DEF15d," i.e. <","LEntity,">=<","REntity,">"),
    nl,write(DEF16a," <","End2,"> ",DEF16b,"."),nl,nl,
closefile(outfile),
writedevic(screen),
    nl,write("<","LEntity,">=<","REntity,">"),
    nl,write("because the ends of <","LEntity,"> are <","End1,">,
<","End2,">"),
    nl,write("and the ends of <","REntity,"> are <","End3,">,
<","End4,">"),
    nl,write("and the common end <","End2,"> is the center of the circle
<","Circle,">"),
    nl,write("That is, <","End1,">, <","End4,"> are on the circumference
of the same circle,"),
    nl,write("and therefore <","LEntity,">, <","REntity,"> are equal as
radii on the same circle"),nl,
    nl,write("The above follows from the definition of the circle."),
    nl,write("In this case,","DEF15a," <","Circle,">"),
    nl,write(DEF15b," <","LEntity,">, <","REntity,">"),
    nl,write(DEF15c," <","End2,">"),
    nl,write(DEF15d," i.e. <","LEntity,">=<","REntity,">"),

```



```

nl,write(DEF16a," <","End2,"> ",DEF16b,"."),nl,nl.

expl_answer(PrevQ):-
leftentityofquestion(LEntity),
relationofquestion(Relation),
prepositionofquestion(Prep),
rightentityofquestion(REntity),
oldstrategy(Strategy),
strategy(Synonym,LSide,RSide,Inversion,Relstrategy,Pk,Strategy),
openappend(outfile,"c:\\prolog\\qanswers.txt"),
writedevic(outfile),
nl,write("Your question is :"),
nl,write("[why <","LEntity,"> is ","Relation," ",Prep," <","REntity,">
?]."),
nl,write("I analysed this question using strategy :
<","Strategy,">"),
nl,write("which involves : ",Synonym," ",LSide," ",RSide,"
",Inversion," ",Relstrategy," ",Pk),
nl,write("My ansuer using Euclid's proof text offers the
justification :"),
nl,write("because <","PrevQ,">."),nl,
nl,write("This justification is based on the following :"),
equalinesegmentsradii(LEntity,REntity).

impl_answer_cn1:-
leftentityofquestion(LEntity),
relationofquestion(Relation),
prepositionofquestion(Prep),
rightentityofquestion(REntity),
rightpartofLEntity(RightentityofLEntity),
rightpartofREntity(RightentityofREntity),
synwnyma(RightentityofLEntity,RightentityofREntity),
pk(29,cn1,CN1),
openappend(outfile,"c:\\prolog\\qanswers.txt"),
writedevic(outfile),
nl,write("The answer of question :"),
nl,write("why <","LEntity,"> is <","Relation,"> <","Prep,">
<","REntity,"> ?"),
nl,write(" is,"),
nl,write("because"),
nl,write("<","LEntity,"> is equal to <","RightentityofLEntity,">
and"),
nl,write("<","REntity,"> is equal to <","RightentityofREntity,">"),
nl,write("furthermore <","RightentityofLEntity,">,
<","RightentityofREntity,"> are synonymes, that means"),
nl,write("that <","LEntity,">, <","REntity,"> are equal to each
other"),
nl,write("because of the common notion"),
nl,write("<","CN1,">"),nl,
writedevic(screen),
closefile(outfile).

impl_answer_cn1:-
leftentityofquestion(LEntity),
relationofquestion(Relation),
prepositionofquestion(Prep),
rightentityofquestion(REntity),
rightpartofLEntity(RightentityofLEntity),
rightpartofREntity(RightentityofREntity),
RightentityofLEntity=RightentityofREntity,
pk(29,cn1,CN1),

```

```

openappend(outfile,"c:\\prolog\\qanswers.txt"),
writedevic(outfile),
nl,write("The answer of question :"),
nl,write("why <","LEntity,"> is <","Relation,"> <","Prep,">
<","REntity,"> ?"),
nl,write(" is,"),
nl,write("because"),
nl,write("<","LEntity,"> is equal to <","RightentityofLEntity,">
and"),
nl,write("<","REntity,"> is equal to <","RightentityofREntity,"> that
means"),
nl,write("that <","LEntity,">, <","REntity,"> are equal to each
other"),
nl,write("because of the common notion"),
nl,write("<","CN1,">"),nl,
writedevic(screen),
closefile(outfile).

```

```

compose_question(CQ,S,I):-
S="nosynonymes",I="noinversion",
leftentityofquestion(LEntity),
relationofquestion(Relation),
rightentityofquestion(REntity),

% DIADIKASIA CONCAT
c(" ",LEntity,BLE),c(BLE," ",T1),
c("is"," ",Isb),c(Isb,Relation,IsbR),c(IsbR,"
",IsbRb),c(IsbRb,"to",T2),
c(" ",REntity,BRE),c(BRE," ",T3),
c(T1,T2,T12),c(T12,T3,CQ),

retractall(leftentityofquestiontransfered(_)),
assert(leftentityofquestiontransfered(LEntity)),

retractall(rightentityofquestiontransfered(_)),
assert(rightentityofquestiontransfered(REntity)).

```

```

compose_question(CQ,S,I):-
S="nosynonymes",I="meinversion",
leftentityofquestion(LEntity),
relationofquestion(Relation),
rightentityofquestion(REntity),

% DIADIKASIA CONCAT
c(" ",LEntity,BLE),c(BLE," ",T1),
c("is"," ",Isb),c(Isb,Relation,IsbR),c(IsbR,"
",IsbRb),c(IsbRb,"to",T2),
c(" ",REntity,BRE),c(BRE," ",T3),
c(T3,T2,T32),c(T32,T1,CQ),

retractall(leftentityofquestiontransfered(_)),
assert(leftentityofquestiontransfered(REntity)),

retractall(rightentityofquestiontransfered(_)),
assert(rightentityofquestiontransfered(LEntity)).

```

```

compose_question(CQ,S,L,I):-
S="mesynonymes",L="left",I="noinversion",
leftentityofquestion(LEntity),
relationofquestion(Relation),
rightentityofquestion(REntity),

```

```

synwnyma (LEntity, SLEntity),

    c(" ", SLEntity, BLE), c(BLE, " ", T1),
    c("is", " ", Isb), c(Isb, Relation, IsbR), c(IsbR, "
", IsbRb), c(IsbRb, "to", T2),
    c(" ", REntity, BRE), c(BRE, " ", T3),
    c(T1, T2, T12), c(T12, T3, CQ),

    retractall(leftentityofquestiontransferred(_)),
    assert(leftentityofquestiontransferred(SLEntity)),
    retractall(rightentityofquestiontransferred(_)),
    assert(rightentityofquestiontransferred(REntity)).

compose_question(CQ, S, L, I) :-
    S="mesynonymes", L="left", I="meinversion",
    leftentityofquestion(LEntity),
    relationofquestion(Relation),
    rightentityofquestion(REntity),
    synwnyma(LEntity, SLEntity),

    c(" ", SLEntity, BLE), c(BLE, " ", T1),
    c("is", " ", Isb), c(Isb, Relation, IsbR), c(IsbR, "
", IsbRb), c(IsbRb, "to", T2),
    c(" ", REntity, BRE), c(BRE, " ", T3),
    c(T3, T2, T32), c(T32, T1, CQ),

    retractall(leftentityofquestiontransferred(_)),
    assert(leftentityofquestiontransferred(REntity)),
    retractall(rightentityofquestiontransferred(_)),
    assert(rightentityofquestiontransferred(SLEntity)).

compose_question(CQ, S, R, I) :-
S="mesynonymes", R="right", I="noinversion",
    leftentityofquestion(LEntity),
    relationofquestion(Relation),
    rightentityofquestion(REntity),
    synwnyma(REntity, SREntity),

    c(" ", LEntity, BLE), c(BLE, " ", T1),
    c("is", " ", Isb), c(Isb, Relation, IsbR), c(IsbR, "
", IsbRb), c(IsbRb, "to", T2),
    c(" ", SREntity, BRE), c(BRE, " ", T3),
    c(T1, T2, T12), c(T12, T3, CQ),

    retractall(leftentityofquestiontransferred(_)),
    assert(leftentityofquestiontransferred(LEntity)),
    retractall(rightentityofquestiontransferred(_)),
    assert(rightentityofquestiontransferred(SREntity)).

compose_question(CQ, S, R, I) :-
S="mesynonymes", R="right", I="meinversion",
    leftentityofquestion(LEntity),
    relationofquestion(Relation),
    rightentityofquestion(REntity),
    synwnyma(REntity, SREntity),

    c(" ", LEntity, BLE), c(BLE, " ", T1),
    c("is", " ", Isb), c(Isb, Relation, IsbR), c(IsbR, "
", IsbRb), c(IsbRb, "to", T2),

```

```

c(" ", SREntity, BRE), c(BRE, " ", T3),
c(T3, T2, T32), c(T32, T1, CQ),

    retractall(leftentityofquestiontransferred(_)),
    assert(leftentityofquestiontransferred(SREntity)),
    retractall(rightentityofquestiontransferred(_)),
    assert(rightentityofquestiontransferred(LREntity)).

compose_question(CQ, S, L, R, I) :-
S="mesynonymes", L="left", R="right", I="noinversion",
    leftentityofquestion(LREntity),
    relationofquestion(Relation),
    rightentityofquestion(REntity),
    synwnyma(LREntity, SLEntity),
    synwnyma(REntity, SREntity),

    c(" ", SLEntity, BLE), c(BLE, " ", T1),
    c("is", " ", Isb), c(Isb, Relation, IsbR), c(IsbR, "
", IsbRb), c(IsbRb, "to", T2),
    c(" ", SREntity, BRE), c(BRE, " ", T3),
    c(T1, T2, T12), c(T12, T3, CQ),

    retractall(leftentityofquestiontransferred(_)),
    assert(leftentityofquestiontransferred(SLEntity)),
    retractall(rightentityofquestiontransferred(_)),
    assert(rightentityofquestiontransferred(SREntity)).

compose_question(CQ, S, L, R, I) :-
S="mesynonymes", L="left", R="right", I="meinversion",
    leftentityofquestion(LREntity),
    relationofquestion(Relation),
    rightentityofquestion(REntity),
    synwnyma(LREntity, SLEntity),
    synwnyma(LREntity, SLEntity),
    synwnyma(REntity, SREntity),

    c(" ", SLEntity, BLE), c(BLE, " ", T1),
    c("is", " ", Isb), c(Isb, Relation, IsbR), c(IsbR, "
", IsbRb), c(IsbRb, "to", T2),
    c(" ", SREntity, BRE), c(BRE, " ", T3),
    c(T3, T2, T32), c(T32, T1, CQ),

    retractall(leftentityofquestiontransferred(_)),
    assert(leftentityofquestiontransferred(SREntity)),
    retractall(rightentityofquestiontransferred(_)),
    assert(rightentityofquestiontransferred(SLEntity)).

same_relation:-
    leftentityofquestiontransferred(LREntity),
    rightentityofquestiontransferred(REntity),
    number_leftpart_verb(LREntity, N),
    find_rightpart_verbless(N, Rightpartnextverb),
    f(Rightpartnextverb, equal, T1), f(T1, to, T2),
    f(T2, Rightentity, " "),
    Rightentity=REntity.

number_leftpart_verb(LREntity, N) :- openread(verbless, "verbless.pro"),
readdevice(verbless),
    c(" ", LREntity, BL),

```

```

match_leftpart_verbless(BL),
numberleftpart(N),nl,
closefile(verbless),
readdevice(keyboard).

numberofEntityComposeCN1indiferentrelationofQ(LEntityCompose,NewNLEnt
ityCompose):-
  openread(verbless,"verbless.pro"),
  readdevice(verbless),
  numberleftpart(OldNLEntityCompose),nl,
  retractall(temporarydbl(_)),
  assert(temporarydbl(OldNLEntityCompose)),
  temporarydbl(OldNLEntityCompose),
  c(" ",LEntityCompose,BL),
  match_leftpart_verbless(BL),
  numberleftpart(NewNLEntityCompose),nl,
  OldNLEntityCompose<>NewNLEntityCompose,
  closefile(verbless),
  readdevice(keyboard).

match_leftpart_verbless(BL):-
  readint(N),retractall(numberleftpart(_)),
  assert(numberleftpart(N)),
  N<>0,
  readln(Leftentity),
  retractall(leftpartverbless(_)),
  assert(leftpartverbless(Leftentity)),
  readln(_),readln(_),
  leftpartverbless(Leftentity),
  BL<>Leftentity,
  match_leftpart_verbless(BL).

match_leftpart_verbless(_):-
  numberleftpart(N),N<>0,
  closefile(verbless),readdevice(keyboard).

match_leftpart_verbless(_):-
  numberleftpart(N),N=0,
  closefile(verbless),
  readdevice(keyboard),fail.

find_rightpart_verbless(N,Rightpartnextverb):-
  openread(verbless,"verbless.pro"),
  readdevice(verbless),
  rightpart_in_verbless(N),
  rightpartverbless(Rightpartnextverb),
  closefile(verbless),
  readdevice(keyboard).

rightpart_in_verbless(N):-
  not(eof(verbless)),
  readint(Num),
  readln(_),readln(_),
  readln(Rightpart),
  retractall(rightpartverbless(_)),
  assert(rightpartverbless(Rightpart)),
  N<>Num,!,
  rightpart_in_verbless(N).

rightpart_in_verbless(_).

```

```

previous_phrase_in_text(Q,PrevQ):-
  find_number_previous_phrase(Q,NprevQ),
  findpreviousphrase(PrevQ,NprevQ).

find_number_previous_phrase(Q,NprevQ):-
  openread(phrase,"phrase.pro"),
  readdevice(phrase),
  number_previous_phrase(Q),
  numberpreviousphrase(NprevQ),
  closefile(phrase),
  readdevice(keyboard).

number_previous_phrase(Q):-
  not(eof(phrase)),
  readint(NprevQ),
  readln(Sentence),
  Q<>Sentence,
  retractall(numberpreviousphrase(_)),
  assert(numberpreviousphrase(NprevQ)),
  number_previous_phrase(Q).

number_previous_phrase(_).

findpreviousphrase(PrevQ,NprevQ):-
  openread(phrase,"phrase.pro"),
  readdevice(phrase),
  previous_phrase(NprevQ),
  previousphrase(PrevQ),
  closefile(phrase),
  readdevice(keyboard).

previous_phrase(Nphrase):-
  not(eof(phrase)),
  readint(N),
  readln(Sentence),
  retractall(previousphrase(_)),
  assert(previousphrase(Sentence)),
  Nphrase<>N,!,
  previous_phrase(Nphrase).

previous_phrase(_).

implicit_knowledge(PrevQ,cn1):-
  pk(29,cn1,K1),
  PrevQ=K1,
  common_notion1,
  impl_answer_cn1.

common_notion1:-leftpartofQrelationCN1,
  rightpartofQrelationCN1,
  numberLpartofLEntityinVerb(N),
  numberRpartofREntityinVerb(MREntity),
  N<>MREntity,
  rightpartofLEntity(RightentityofLEntity),
  rightpartofREntity(RightentityofREntity),
  synnynma(RightentityofLEntity,RightentityofREntity).

common_notion1:-
  leftpartofQrelationCN1,
  rightpartofQrelationCN1,
  numberLpartofLEntityinVerb(N),

```

```

numberRpartofREntityinVerb (MREntity) ,
N<>MREntity,
rightpartofLEntity (RightentityofLEntity) ,
rightpartofREntity (RightentityofREntity) ,
RightentityofLEntity=RightentityofREntity.

leftpartofQrelationCN1:-
leftentityofquestiontransferred (LEntityCompose) ,
numberofEntityComposeCN1indiferentrelationofQ (LEntityCompose, NLEntity
Compose) ,
find_rightpart_verbless (NLEntitycompose, RightpartnextverbofLEntity) ,
f (RightpartnextverbofLEntity, equal, T1) ,
f (T1, to, T2) ,
f (T2, RightentityofLEntity, " ") ,
retractall (rightpartofLEntity (_)) ,
assert (rightpartofLEntity (RightentityofLEntity)) ,
retractall (numberLpartofLEntityinVerb (_)) ,
assert (numberLpartofLEntityinVerb (NLEntitycompose)) .

leftpartofQrelationCN1:-
leftentityofquestiontransferred (LEntityCompose) ,
synnyma (LEntityCompose, SLEntity) ,
numberofEntityComposeCN1indiferentrelationofQ (SLEntity, NSLEntity) ,
find_rightpart_verbless (NSLEntity, RightpartnextverbofLEntity) ,
f (RightpartnextverbofLEntity, equal, T1) ,
f (T1, to, T2) ,
f (T2, RightentityofLEntity, " ") ,
retractall (rightpartofLEntity (_)) ,
assert (rightpartofLEntity (RightentityofLEntity)) ,
retractall (numberLpartofLEntityinVerb (_)) ,
assert (numberLpartofLEntityinVerb (NSLEntity)) .

rightpartofQrelationCN1:-
nl, write ("5rightpart/") , readln (_) ,
rightentityofquestiontransferred (REntityCompose) ,
number_leftpart_verb (REntityCompose, MREntityComposeinText) ,

find_rightpart_verbless (MREntityComposeinText, RightpartnextverbofREnt
ity) ,
f (RightpartnextverbofREntity, equal, Z1) ,
f (Z1, to, Z2) ,
f (Z2, RightentityofREntity, " ") ,
retractall (rightpartofREntity (_)) ,
assert (rightpartofREntity (RightentityofREntity)) ,
retractall (numberRpartofREntityinVerb (_)) ,
assert (numberRpartofREntityinVerb (MREntityComposeinText)) .

rightpartofQrelationCN1:-
rightentityofquestiontransferred (REntityCompose) ,
synnyma (REntityCompose, SREntity) ,
number_leftpart_verb (SREntity, MREntityComposeinText) ,

find_rightpart_verbless (MREntityComposeinText, RightpartnextverbofREnt
ity) ,
f (RightpartnextverbofREntity, equal, Z1) ,
f (Z1, to, Z2) ,
f (Z2, RightentityofREntity, " ") ,

retractall (rightpartofREntity (_)) ,
assert (rightpartofREntity (RightentityofREntity)) ,
retractall (numberRpartofREntityinVerb (_)) ,

```

```

assert (numberRpartofREntityinVerb (MREntityComposeinText) ) .

strategy (nosynonymes, left, right, noinversion, equality, nopk, strategy1) .
strategy (nosynonymes, left, right, meinversion, equality, nopk, strategy2) .
strategy (mesynonymes, left, noright, noinversion, equality, nopk, strategy3
) .
strategy (mesynonymes, left, noright, meinversion, equality, nopk, strategy4
) .
strategy (mesynonymes, noleft, right, noinversion, equality, nopk, strategy5
) .
strategy (mesynonymes, noleft, right, meinversion, equality, nopk, strategy6
) .
strategy (mesynonymes, left, right, noinversion, equality, nopk, strategy7) .
strategy (mesynonymes, left, right, meinversion, equality, nopk, strategy8) .
strategy (nosynonymes, left, right, noinversion, equality, pk, strategy9) .
strategy (nosynonymes, left, right, meinversion, equality, pk, strategy10) .
strategy (mesynonymes, left, noright, noinversion, equality, pk, strategy11)
.
strategy (mesynonymes, left, noright, meinversion, equality, pk, strategy12)
.
strategy (mesynonymes, noleft, right, noinversion, equality, pk, strategy13)
.
strategy (mesynonymes, noleft, right, meinversion, equality, pk, strategy14)
.
strategy (mesynonymes, left, right, noinversion, equality, pk, strategy15) .
strategy (mesynonymes, left, right, meinversion, equality, pk, strategy16) .

next (s, strategy1) .
next (strategy1, strategy2) .
next (strategy2, strategy3) .
next (strategy3, strategy4) .
next (strategy4, strategy5) .
next (strategy5, strategy6) .
next (strategy6, strategy7) .
next (strategy7, strategy8) .
next (strategy8, strategy9) .
next (strategy9, strategy10) .
next (strategy10, strategy11) .
next (strategy11, strategy12) .
next (strategy12, strategy13) .
next (strategy13, strategy14) .
next (strategy14, strategy15) .
next (strategy15, strategy16) .

```

4. Πρόγραμμα της σχεδίασης του σχήματος της πρώτης πρότασης των Στοιχείων

```

clauses

pwrite (X, Y) :- X=0, Y=13, write ("o" );
                X=1, Y=13, write ("o" );
                X=2, Y=13, write ("o" );
                X=3, Y=13, write ("o" );
                X=4, Y=13, write ("o" );
                X=5, Y=12, nl, write ("      o" );
                X=6, Y=12, write ("o" );
                X=7, Y=11, write ("o" );
                X=8, Y=10, write ("  o" );
                X=9, Y=9,  write ("    o" );

```



```

X=10,Y=8, write("      o");
X=11,Y=7, write("      o");
X=12,Y=6, write("      o");
X=12,Y=5, write("      o");
X=13,Y=4, write("      o");
X=13,Y=3, write("      o");
X=13,Y=2, write("      o");
X=13,Y=1, write("      o");
X=13,Y=0, write("*****o");

      X=13,Y=13, write("o");
      X=12,Y=13, write("o");
      X=11,Y=13, write("o");
X=10,Y=13, write("o");
X=9,Y=13, write(" o");
X=8,Y=12, write("o");
X=7,Y=12, write("o");
X=6,Y=11,nl,write("      o");
X=5,Y=10,nl,write("      o");
X=4,Y=9,nl, write("      o");
X=3,Y=8,nl, write("      o");
X=2,Y=7,nl, write("      o");
X=1,Y=6,nl, write(" o");
X=1,Y=5,nl, write(" o");
X=0,Y=4,nl, write("o");
X=0,Y=3,nl, write("o");
X=0,Y=2,nl, write("o");
X=0,Y=1,nl, write("o");
X=0,Y=0,nl, write("o").

pwrite(_,_) .
%maine loop
rgo1(X,Y):-X>13/2,
rgo2(X,Y) .

rgo1(X,Y):-r1(NX,NY,X,Y),
          assert(r_sintetagmenes(X,Y),dexitoxo),
          rgo1(NX,NY) .

r1(NX,NY,X,Y):-NX=X+1,NY=Y,NX*NX+NY*NY<=185 .
r1(NX,NY,X,Y):-NX=X+1,NY=Y-1 .

rgo2(X,Y):-X=13,Y=0,
          assert(r_sintetagmenes(X,Y),dexitoxo) .

rgo2(X,Y):-r2(NX,NY,X,Y),
          assert(r_sintetagmenes(X,Y),dexitoxo),
          rgo2(NX,NY) .

r2(NX,NY,X,Y):-NX=X+1,NY=Y-1,NX*NX+NY*NY<=185 .
r2(NX,NY,X,Y):-NX=X,NY=Y-1 .

l_go(LZ,LY):-Z=13-LZ,Y=LY,lgo1(Z,Y) .

lgo1(Z,Y):-Z>13/2,%dioti apo (7,11):(8,11),8*8+11*11=185,
          lgo2(Z,Y) .

lgo1(Z,Y):-l1(NZ,NY,Z,Y,X),
          assert(l_sintetagmenes(X,Y),aristerotoxo),
          lgo1(NZ,NY) .

l1(NZ,NY,Z,Y,X):-NZ=Z+1,NY=Y,NZ*NZ+NY*NY<=185,X=13-Z .

```

```

l1 (NZ,NY,Z,Y,X) :-NZ=Z+1,NY=Y-1,X=13-Z.

lgo2(Z,Y) :-Z=13,Y=0,X=13-Z, % limit horizontal
           assert(l_sintetagmenes(X,Y),aristerotoxo).

lgo2(Z,Y) :-l2(NZ,NY,Z,Y,X),
           assert(l_sintetagmenes(X,Y),aristerotoxo),
           lgo2(NZ,NY).
l2(NZ,NY,Z,Y,X) :-NZ=Z+1,NY=Y-1,NZ*NZ+NY*NY<=185,X=13-Z.
l2(NZ,NY,Z,Y,X) :-NZ=Z,NY=Y-1,X=13-Z.
process_print_screen(X,Y) :-X=13,Y=0,
                           look_up_sintetagmenes(X,Y),
                           writedevic(screen),pwrite(X,Y).

process_print_screen(X,Y) :-look_up_sintetagmenes(X,Y),
                           writedevic(screen),pwrite(X,Y),
                           k3(NX,NY,X,Y),
                           process_print_screen(NX,NY).

look_up_sintetagmenes(X,Y) :-
r_sintetagmenes(X,Y);l_sintetagmenes(X,Y),!.

look_up_sintetagmenes(X,Y) :-fail>true.
k3(NX,NY,X,Y) :-NX=X+1,NY=Y,NX<=13.
k3(NX,NY,X,Y) :-NX=X-13,NY=Y-1.
tomi_toxwn(X,Y) :-X=13,Y=0,
                pixels_tomis(X,Y).

tomi_toxwn(X,Y) :-pixels_tomis(X,Y),
                k4(NX,NY,X,Y),
                tomi_toxwn(NX,NY).

k4(NX,NY,X,Y) :-NX=X+1,NY=Y,NX<=13.
k4(NX,NY,X,Y) :-NX=X-13,NY=Y-1.

pixels_tomis(X,Y) :-k5(MX,MY,X,Y),
                   r_sintetagmenes(X,Y), l_sintetagmenes(MX,Y),
                   l_sintetagmenes(X,MY), r_sintetagmenes(MX,MY),

assert(pixels_tomis_toxwn(X,Y),tomi_toxwn),assert(pixels_tomis_toxwn(
MX,Y),tomi_toxwn),

assert(pixels_tomis_toxwn(X,MY),tomi_toxwn),assert(pixels_tomis_toxwn(
MX,MY),tomi_toxwn).

pixels_tomis(X,Y) :-fail>true.

k5(MX,MY,X,Y) :-MX=X+1, MY=Y-1.

save_sintetag_ltoxo_txt:-save("propos1a.txt",aristerotoxo).
save_sintetag_ltoxo_dba:-save("propos1a.dba",aristerotoxo).

save_sintetag_rtoxo_txt:-save("propos1b.txt",dexitoxo).
save_sintetag_rtoxo_dba:-save("propos1b.dba",dexitoxo).

save_sintetag_tomis_toxwn_txt:-save("propos1c.txt",tomi_toxwn).
save_sintetag_tomis_toxwn_dba:-save("propos1c.dba",tomi_toxwn).

goal
rgo1(0,13),!,
l_go(13,13),!,

```

```
process_print_screen(0,13),  
tomi_toxwn(0,13),  
save_sintetag_ltoxo_txt,  
save_sintetag_ltoxo_dba,  
save_sintetag_rtoxo_txt,  
save_sintetag_rtoxo_dba,  
save_sintetag_tomis_toxwn_txt,  
save_sintetag_tomis_toxwn_dba.
```

Κατάλογος Πινάκων και Σχημάτων

ΠΙΝΑΚΕΣ	σελ.
Πίνακας 1: Το κείμενο της Γεωμετρίας ως κείμενο εισαγωγής στο σύστημα	55
Πίνακας 2: Το κείμενο μετά την πρώτη επεξεργασία από το σύστημα	55 – 56
Πίνακας 3: Το αρχείο που παράγει το σύστημα με το κείμενο χωρισμένο σε προτάσεις	56
Πίνακας 4: Το αρχείο που παράγει το σύστημα με τις προτάσεις του κειμένου της Γεωμετρίας χωρισμένες σε τρία μέρη -το ρήμα στο μέσον-	57
Πίνακας 5: Ενδεικτικός κατάλογος των ερωτήσεων	58
Πίνακας 6: Το περιεχόμενο του αρχείου που παράγει το Υποσύστημα επεξεργασίας της ερώτησης για την ερώτηση q1	58
Πίνακας 7: Οι κύκλοι (Εξαγόμενη οντολογική γνώση από τις προτάσεις του κειμένου)	59
Πίνακας 8: Τα ευθύγραμμα τμήματα και τα συνώνυμά τους	59
Πίνακας 9: Εξαγόμενη οντολογική γνώση από τα ονόματα των κύκλων που αναφέρονται στο κείμενο	59
Πίνακας 10: Εξαγόμενη οντολογική γνώση από τα ονόματα των ευθυγράμμων τμημάτων που αναφέρονται στο κείμενο	59
Πίνακας 11: Το κείμενο της απόδειξης της πρώτης πρότασης των ‘Στοιχείων’ από το βιβλίο του Heath	61
Πίνακας 12: Η επεξήγηση της απάντησης που παράγεται αυτόματα από το σύστημα όταν η απάντηση δηλώνεται ρητά στο κείμενο	63
Πίνακας 13: Η διαδοχή της επιλογής των στρατηγικών	64
Πίνακας 14: Οι στρατηγικές του συστήματος με τα ορίσματά τους	64
Πίνακας 15: Οι τιμές των ορισμάτων του κατηγορήματος ‘strategy’	65
Πίνακας 16: Η επεξήγηση της απάντησης που δεν παράγεται αυτόματα από το σύστημα όταν η απάντηση δεν δηλώνεται ρητά στο κείμενο -της Γεωμετρίας-	68
Πίνακας 17: Το κείμενο της Βιοϊατρικής ως κείμενο εισαγωγής στο σύστημα	69
Πίνακας 18: Το περιεχόμενο του αρχείου που παράγει το Υποσύστημα επεξεργασίας της ερώτησης για την ερώτηση q55	70

ΠΙΝΑΚΕΣ

σελ.

Πίνακας 19: Οι προτάσεις του κειμένου της Βιοϊατρικής μετά από επεξεργασία από το σύστημα	70
Πίνακας 20: Το αρχείο που παράγει το σύστημα με τις προτάσεις του κειμένου της Βιοϊατρικής χωρισμένες σε τρία μέρη -το ρήμα στο μέσον-	71
Πίνακας 21: Το Λεξικό ρημάτων για την εφαρμογή του πεδίου της Βιοϊατρικής	71
Πίνακας 22: Το Λεξικό οντοτήτων για την εφαρμογή του πεδίου της Βιοϊατρικής	71
Πίνακας 23: Οι προτάσεις που επεξεργάζεται το σύστημα για τον εντοπισμό του τμήματος αριστερά του ρήματος	73
Πίνακας 24: Το τμήμα από την παραγόμενη απάντηση της ερώτησης που αφορά την αναζήτηση της οντότητας p53	73
Πίνακας 25: Η πρόταση του κειμένου που επεξεργάζεται το σύστημα για τον εντοπισμό του τμήματος δεξιά του ρήματος	73
Πίνακας 26: Η πρόταση του κειμένου που επεξεργάστηκε το σύστημα με τις δύο οντότητες p53 και mdm2 αντίστοιχα αριστερά και δεξιά του ρήματος	74
Πίνακας 27: Το επόμενο βήμα επεξεργασίας μετά από εκείνο του Πίνακα 23	74
Πίνακας 28: Η επόμενη πρόταση του κειμένου που επέλεξε να επεξεργαστεί το σύστημα για τον εντοπισμό του τμήματος δεξιά του ρήματος	75
Πίνακας 29: Η πρόταση του κειμένου που επεξεργάστηκε το σύστημα με τις δύο οντότητες mdm2 και p53 αντίστοιχα αριστερά και δεξιά του ρήματος	75
Πίνακας 30: Η επεξήγηση της απάντησης που παράγεται αυτόματα από το σύστημα όταν η απάντηση δεν δηλώνεται ρητά στο κείμενο -της Βιοϊατρικής-	77

ΣΧΗΜΑΤΑ

Σχήμα 1: Από την απόδειξη της πρώτης πρότασης των 'Στοιχείων' από το βιβλίο του Heath	61
Σχήμα 2: Τα δύο ημικύκλια όπως τυπώνονται από το πρόγραμμα σχεδίασης	136

Βιβλιογραφία

- A. Augustine. (354-430 A.D). De Trinitate, in: J. Burnaby (Ed.), Augustine: Later Works, in: Library of Christian Classics, Bk. 10, Sec. 7, vol. 8, SCM Press, 1955. Transl. by J. Burnaby, original work published around 1600, in M. T. Cox, (2005). Metacognition in computation: A selected research review. *Artificial Intelligence* 169 pp.104-141.
- Adey, P. and Shayer, M. (1994). *Really Raising Standards* (London: Routledge).
- Aleven, V., McLaren, B., Roll, I., Koedinger, K. (2006). Toward Meta-cognitive Tutoring: A Model of Help-Seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education* 16, pp.101-130.
- Anderson, M. L. (2005). How to study the mind: An introduction to embodied cognition, in: F. Santoianni, C. Sabatano (Eds.), Brain Development in Learning Environments: Embodied and Perceptual Advancements, Cambridge Univ. Press, New York, in press, in M. T. Cox, (2005). Metacognition in computation: A selected research review. *Artificial Intelligence* 169 pp.104-141.
- Appelt, D. and D. Israel (1999). Introduction to information extraction technology, in Fei – Yu Xu. Bootstrapping Relation Extraction from Semantic Seeds. A Dissertation submitted to the Philosophy Faculty of Saarland University for the Degree of Ph.D., 2008.
- Athenikos, S. J. (2009). Biomedical question answering: A survey. Computer Methods and Programs in Biomedicine. doi:10.1016/j.cmpb.2009.10.003. (in press)
- Barr, A. (1977). Meta-knowledge and memory, Technical Report, HPP-77-37, Stanford University, Department of Computer Science, Stanford, CA, in M. T. Cox, (2005). Metacognition in computation: A selected research review. *Artificial Intelligence* 169 pp.104-141.
- Bar-Or, R. L., Maya, R., Segel, L. A., Alon, U., Levine, A. J., Oren, M., (2000). Generation of oscillations by the p53-Mdm2 feedback loop: A theoretical and experimental study. Departments of Molecular Cell Biology and Applied Mathematics and Computer Science. The Weizmann Institute of Science, Rehovot, Israel, and The Rockefeller University, New York. Edited by George F. Vande Woude, Van Andel Research Institute, Grand Rapids, MI.
- Bateson, G. (1983). *Steps to an Ecology of the Mind* (London: Paladin).

- Chi, M. T. H. (1995). Revising the mental model as one learns, Plenary address to the Seventeenth Annual Conference of the Cognitive Science Society, Pittsburgh, July 23, in M. T. Cox, (2005). Metacognition in computation: A selected research review. *Artificial Intelligence* 169 pp.104-141.
- Chi, M., VanLehn, K. (2007). The impact of explicit strategy instruction on problem-solving behaviors across intelligent tutoring systems, in D. McNamara & G. Trafton (Eds.). *Proceedings of the 29th Annual Conference of the Cognitive Science Society*. (pp. 167-172). Mahwah, NJ: Erlbaum.
- Chi, M., VanLehn, K. (2008). Eliminating the gap between the high and low students through meta-cognitive strategy instructions, in B. P. Woolf, E. Aimeur, R. Nkambou & S. Lajoie (Eds.). *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*. pp.603-613. Amsterdam: IOS Press.
- Cointe P. (Ed.) (1999). Meta-Level Architectures and Reflection: Second International Conference, Reflection '99, Springer, Berlin.
- Cooper, W. S. (1964). Fact Retrieval and Deductive Question–Answering Information Retrieval Systems. *Journal of the ACM*, Vol. 11, No. 2, pp. 117-137.
- Cowie J., and Lehnert, W., (1996). Information Extraction. Communications of the ACM. Vol. 39, No. 1, pp. 80-91, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in Encyclopedia of Human Computer Interaction. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Cox, M. T. (1996). Introspective multistrategy learning: Constructing a learning strategy under reasoning failure, Technical Report, GIT-CC-96-06. Ph.D. Dissertation, College of Computing, Georgia Institute of Technology, Atlanta , hcs.bbn.com/cox/thesis/, in M. T. Cox, (2005). Metacognition in computation: A selected research review. *Artificial Intelligence* 169 pp.104-141.
- Cox, M. T. (2005). Metacognition in computation: A selected research review. *Artificial Intelligence* 169 pp.104-141.
- Cross, D. R. and Paris, S. C. (1988). Developmental and instrumental analysis of children's metacognition and reading comprehension. *Journal of Educational Psychology*, 80, pp.131–142.
- Davis, D. N. (2005). *Visions of Mind: Architectures for Cognition and Affect*, N. D. Darryl (Ed.) Idea Group Inc., U.S.A and U.K.

- Demner-Fushman, D., (2006). Complex Question Answering based on a semantic domain model of clinical medicine. Thesis, University of Maryland, College Park.
- Diekema, A., R. (2003). What do You Mean? Finding Answers to Complex Questions. *New Directions on Question Answering. Papers from 2003 AAAI Spring Symposium*. The AAAI Press. USA, pp. 87-93, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in Encyclopedia of Human Computer Interaction. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Doan-Nguyen, H. and Kosseim, L. (2004). Improving the Precision of a Closed-Domain Question Answering System with Semantic Information. *Proceedings of Recherche d' Information Assistee Ordinateur (RIAO-2004)*. pp. 850-859. Avignon, France. April, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in Encyclopedia of Human Computer Interaction. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Doddington, G., A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel (2004). The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation. *Proceedings of LREC 2004* , 837-840, in Fei – Yu Xu. Bootstrapping Relation Extraction from Semantic Seeds. A Dissertation submitted to the Philosophy Faculty of Saarland University for the Degree of Ph.D., 2008.
- Edelman, M.G., (1996). *AIΘEPΑΣ ΘΕΪΚΟΣ, ΛΑΜΠΙΕΡΗ ΦΩΤΙΑ*, ΚΑΤΟΠΤΡΟ, σελ. 44-47.
- Fei–Yu Xu. (2008). Bootstrapping Relation Extraction from Semantic Seeds. A Dissertation submitted to the Philosophy Faculty of Saarland University for the Degree of Ph.D..
- Feigenbaum, E., A. and Feldman, J. (1963). *Computers and Thought*. McGraw Hill. New York.
- Flanell, J. H. (1971). First discussant's comments. What is memory development the development of? *Human Development*, 14, pp. 272-278, in P. Georgiades. From the general to the situated: three decades of metacognition. *International Journal of Science Education*, (2004) February, No 3, pp. 365-383.
- Flavell, J. H. (1976). Metacognitive aspects of problem solving, in L. B. Resnick (Ed.), *The nature of intelligence* pp. 231-236. Hillsdale, NJ: Erlbaum.

- Flavell, J. H. (1978). Cognitive monitoring. Paper presented at the Conference on Children's Oral Communication Skills, University of Wisconsin, October.
- Flavell, J. H. (1978). Metacognition, in E. Langer (Chair), Current perspectives on awareness and cognitive processes. Symposium presented at the meeting of the American Psychological Association, Toronto, August.
- Flavell, J. H. (1979). Metacognition and Cognitive Monitoring, A New Area of Cognitive-Developmental Inquiry. *American Psychologist*, vol.34, No. 10, pp. 906-911.
- Flavell, J. H. (1979). Monitoring social-cognitive enterprises: Something else that may develop in the area of social cognition. Paper prepared for the Social Science Research Council Committee on Social and Affective Development During Childhood, January.
- Fritz, K. V. (1955). Die αρχαί in der griechischen Mathematik, «Archiv für Begriffsgeschichte» (Bonn) 1 pp. 13-103, in A. SZABO (1973). *Απαρχαί των Ελληνικών Μαθηματικών*, Τ.Ε.Ε. Αθήνα.
- Gardner, H. (1987). *The Mind's New Science. A History of the Cognitive Revolution*. New York: Basic Books.
- Georghiades, P. (2004). From the general to the situated: three decades of metacognition. *International Journal of Science Education*, February, No 3, pp. 365-383.
- Green, B. F., Wolf, A. K., Chomsky, C., Laughery, K. (1961). BASEBALL: An Automatic Question Answerer. *Proceedings of the Western Joint Computer Conference 19*, pp. 219-224, in E. A. Feigenbaum and J. Feldman, (Eds.), *Computers and Thought*. MIT Press (1995).
- Green, L. E. S., Berkeley, E.C., Gotlieb, C. (1959). Conversation with a computer. *Computers and Automation* 8, 10, pp. 9-11, in R.F. Simmons, Answering English Questions by Computer: A Survey. *Communications of the ACM*, 8(1) pp. 53-70.
- Grishman R., (1997). Information Extraction: Techniques and Challenges, in Paziienza, M. T. *Information Extraction. LNAI Tutorial*. Springer, pp. 10-27, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)

- Grishman, R., Sundheim, B. (1996). Message Understanding Conference – 6: A brief history, in *Proceedings of the 16th International Conference on Computational Linguistics*, Copengagen, in Fei – Yu Xu. Bootstrapping Relation Extraction from Semantic Seeds. A Dissertation submitted to the Philosophy Faculty of Saarland University for the Degree of Ph.D., 2008.
- Harabagiu, S., M., Maiorano, S., J. and Pasca, M., A. (2003). Open-domain textual question answering techniques. *Natural Language Engineering*, Vol. 9 (3), pp. 231-267, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Hayes-Roth, F., Waterman, D. A., Lenat D. B. (Eds.). (1983). *Building Expert Systems*, Addison-Wesley, London.
- Heath, T. L. (1956). *The Thirteen Books of Euclid's Elements*, New York, second edition unabridged.
- Heath, T. L. (1921;2006). *A History OF GREEK MATHEMATICS*, vol. 1 & 2, Elibron Classics Replica Edition, unabridged facsimile of the edition published in 1921 by the Clarendon Press, Oxford.
- Hobbs, J., Dungan, J., Waldinger, R., Appelt, D. E., Fry, J., Israel, D., Jarvis, P., Martin, D., Riehemann, S., Stickel, M. E., Tyson, M. (2004). Deductive Question Answering from Multiple Resources, in *New Directions in Question Answering*, AAAI.
- Hofstadter, D. R., Gödel, Escher (1989). *Bach: An Eternal Golden Braid*, Vintage Books, New York, 1989.
- Keyser, C. J. (2005). *Mathematical Philosophy, a study of fate and freedom; lectures for educated laymen*, E. P. Dutton & Company publishers. By Michigan historical Reprint Series.
- Kontos J. (1992) ARISTA: Knowledge Engineering with Scientific Texts. *Information and Software Technology*, Vol. 34, No 9, pp. 611-616, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)

- Kontos J. and Papakonstantinou G., (1970). A question-answering system using program generation, in *Proceedings of the ACM International Computing Symposium*, Bonn, Germany, pp. 737-750.
- Kontos J. and Kossidas A., (1971). On the Question-Answering System DELFI and its Application. *Proceedings of the Symposium on Artificial Intelligence*, Rome, Italy, pp. 31-36.
- Kontos J. and Malagardi I. (1999). Information Extraction and Knowledge Acquisition from Texts using Bilingual Question-Answering. *Journal of Intelligent and Robotic Systems*, Vol 26, No. 2, pp. 103-122, October, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Kontos J. and Malagardi I. (2001) A Search Algorithm for Knowledge Acquisition from Texts. *HERCMA 2001, 5th Hellenic European Research on Computer Mathematics & its Applications Conference*, Athens, Greece, pp. 226-230.
- Kontos J., Elmaoglou A. and Malagardi I. (2002). ARISTA Causal Knowledge Discovery from Texts. *Proceedings of the 5th International Conference on Discovery Science DS 2002*, Luebeck, Germany, pp. 348-355.
- Kontos, J., Malagardi, I., Peros, J. (2003). “The AROMA System for Intelligent Text Mining” *HERMIS International Journal of Computer mathematics and its Applications*. Vol. 4. pp.163-173. LEA, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Kontos, J., and Malagardi, I., (2006). Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK. (<http://www.idea-group.com/encyclopedia>)
- Kontos, J., Armaos, J. (2007). Metacognitive Question Answering from Euclid’s Elements Text. *Proceedings of HERCMA07*, Athens, Hellas.
- Kontos, J., Armaos, J., Malagardi, I. (2009). Metagnostic Question Answering from Biomedical Texts. *HCI*, San Diego, pp. 332-336.
- Kosseim, L., Plamondon, L. and Guillemette, L., J. (2003). Answer Formulation for Question-Answering. *Proceedings of the Sixteenth Conference of the Canadian*

- Society for Computational Studies of Intelligence. (AI'2003)*. Lecture Notes in Artificial Intelligence no 2671, pp. 24-34. Springer Verlag. June 2003. Halifax Canada, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in Encyclopedia of Human Computer Interaction. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Kreutzer, M. A., Leonard, C., & Flavell, J. H. (1975). An interview study of children's knowledge about memory. *Monographs of the Society for Research in Child Development*, 40 (4, Serial No. 112).
- Kuokka, D. R. (1990). The deliberative integration of planning, execution, and learning, Technical Report, CMUCS-90-135, Ph.D. Dissertation, Computer Science Department, Carnegie Mellon University, Pittsburgh.
- Lagoudakis, M. G., Littman, M. L., Parr, R. (2001). Selecting the right algorithm, in: C. Gomes, T. Walsh (Eds.), *Proceedings of the 2001 AAAI Fall Symposium Series: Using Uncertainty within Computation*, AAAI Press, Menlo Park, CA.
- Lagoudakis, M. G., Parr, R., Littman, M. L., (2002). Least-squares methods in reinforcement learning for control, in *Proceedings of the 2nd Hellenic Conference on Artificial Intelligence*, in *Lecture Notes on Artificial Intelligence*, vol. 2308, Springer, Berlin, pp. 249–260.
- Lindsay, R. K. (1963). Inferential memory as the basis of machines which understand natural language, in E. A. Feigenbaum and J. Feldman, (Eds.), *Computers and Thought*. MIT Press (1995).
- Livingston A. Jennifer. (1997). www.gse.buffalo.edu/fas/shuell/cep564/Metacog.ght.
- Lyons, W. (1986). *The Disappearance of Introspection*, Bradford Books/MIT Press, Cambridge, MA.
- Maes, P., Nardi D. (Eds.). (1988). *Meta-Level Architectures and Reflection*, North-Holland, Amsterdam.
- Maybury, M., T. (2003). Toward a Question answering Roadmap. *New Directions on Question Answering. Papers from 2003 AAAI Spring Symposium*. The AAAI Press. USA, pp. 8-11.
- McCarthy J. (chair), Chaudri V. (co-chair). (2004). DARPA Workshop on Self Aware Computer Systems, SRI Headquarters, Arlington, VA, April 27–28.
- McCarthy, J. (1959). Programs with common sense. Computer Science Department, Stanford University. <http://www-formal.stanford.edu/jmc/>

- McCarthy, J. (1995). Making robots conscious of their mental states. Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms, AAAI Press, Menlo Park, CA, pp. 89–96. Available as Technical Report SS-95-08.
- MILLER, N. G. (2001). A Diagrammatic Formal System for Euclidean Geometry, Thesis, Cornell University.
- Minsky, M. L. (1965). Matter, mind, and models, Published in Proceedings of International Federation of Information Processing Congress 1965, vol.1, pp.45-49.
- Minsky, M. L. (1968). Matter, mind, and models, in M.L. Minsky (Ed.), *Semantic Information Processing*, MIT Press, Cambridge, MA, pp. 425–432.
- Minsky, M. L. (1985). *The Society of Mind*, Simon and Schuster, New York.
- Minsky, M. L., Singh, P., Sloman, A. (2004). The St. Thomas common sense symposium: Designing architectures for human-level intelligence, *AI Magazine* vol.25 issue 2, pp. 113–124.
- Morbini, F. (2006). A Self-Aware Agent. Thesis Proposal for Ph.D., Department of Computer Science, University of Rochester, New York.
- Morbini, F., Schubert, L. (2005). ‘Conscious Agents’. Computer Science Department, University of Rochester, September 16. {morbini,schubert}@cs.rochester.edu
- Morbini, F., Schubert, L. (2007). Towards realistic autocognitive inference. In *Logical Formalizations of Commonsense Reasoning*, p.p. 114-118.
- Morbini, F., Schubert, L. (2009). Evaluation of EPILOG: a Reasoner for Episodic Logic. Commonsense 09, Toronto Canada.
- Netz, R. (2004). The Limits of Text in Greek Mathematics, in K. Chemla (ed.), *History of Science, History of Text*. Boston Studies in the Philosophy of Science vol. 238. Dordrecht, Springer, pp. 161-176.
- Nyberg, E. et al. (2002). The JAVELIN Question-Answering System at TREC 2002. *NIST Special Publication:11th Text Retrieval Conference (TREC), 2002*, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Paris, S. G. and Jacobs, J. E. (1984). The benefits of informed instruction for children’s reading awareness and comprehension skills. *Child Development*, 55, pp. 2083–2093.

- Paris, S. G. and Winograd, P. (1990). How metacognition can promote academic learning and instruction, in B. F. Jones and L. Idol (Eds) *Dimensions of Thinking and Cognitive Instruction* (Hillsdale, NJ: Lawrence Erlbaum), pp. 15–51.
- Perlis, D. (2006). *Theory and application of self-reference: Logic and beyond*. CSLI, Stanford.
- Phillips, A.V. (1960). A question answering routine. Memo. 16, Artificial Intelligence Project - RLE and MIT Computation Center, Memo 16 – Question-Answering Routine 1, 2, Cambridge, Massachusetts, May.
- Plamondon, L. and Kosseim, L. (2002). QUANTUM: A Function-Based Question Answering System. *Proceedings of The Fifteenth Canadian Conference on Artificial Intelligence (AI'2002). Lecture Notes in Artificial Intelligence no. 2338*, pp.281-292. Springer Verlag. Berlin . May, Calgary, Canada, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in Encyclopedia of Human Computer Interaction. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Ramakrishnan, G. et al. (2004). Is Question Answering an Acquired Skill? (2004). *WWW2004*. May, New York, USA, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery. In Encyclopedia of Human Computer Interaction. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Recker, M., Pirolli, P. (1995). Modelling individual differences in student's learning, *J. Learning Sci.* 4 (1), pp. 1–38.
- Reder, L. M., Schunn, C. D. (1996). Metacognition does not imply awareness: Strategy choice is governed by implicit learning and memory, in: L. Reder (Ed.), *Implicit Memory and Metacognition*, LEA, Mahwah, NJ, pp. 45–77.
- Reidemeister, K. (1949). *Das exakte Denken der Griechen*, Hamburg, in A. SZABO (1973). *Απαρχαί των Ελληνικών Μαθηματικών*, Τ.Ε.Ε. Αθήνα.
- Resnick, B. L. and Glaser, R. (1976). Problem Solving and Intelligence, in L. B. Resnick (Ed.), *The nature of intelligence* (pp. 205-230). Hillsdale, NJ: Erlbaum.
- Rice, J. R. (1976). The algorithm selection problem. *Advances in Computers* 15, pp. 65–118.
- Rinaldi, F. (2008). Knowledge Mining over Scientific Literature and Technical Documentation.
www.cl.uzh.ch/fileadmin/user-upload/docs/studium/dissertationen/rinaldi-diss.pdf

- Roussinof, D., Robles-Flores, J., A. (2004). Web Question Answering: Technology and Business Applications. *Proceedings of the Tenth Americas Conference on Information Systems*. New York. August, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Russell, B., Whitehead, N. A. (1910, 2d, ed. 1927). *Principia Mathematica*. Cambridge Un. Press, London. Στο Δ. Α. ΑΝΑΠΟΛΙΤΑΝΟΣ, (1985). *ΕΙΣΑΓΩΓΗ ΣΤΗ ΦΙΛΟΣΟΦΙΑ ΤΩΝ ΜΑΘΗΜΑΤΙΚΩΝ*, ΝΕΦΕΛΗ, Αθήνα.
- Sager, N., Friedman, C., Margaret, S. (1987). *Medical Language Processing: Computer Management of Narrative Data*. Addison-Wesley, in Fei – Yu Xu. Bootstrapping Relation Extraction from Semantic Seeds. A Dissertation submitted to the Philosophy Faculty of Saarland University for the Degree of Ph.D., 2008.
- Salomon, G. and Globerson, T. (1987). Skill may not be enough: the role of mindfulness in learning and transfer. *International Journal of Educational Research*, 11(6), pp. 623–637.
- Schubert, L. (2005). Some knowledge representation for Self-Awareness, in *Metacognition in Computation*, pp. 106-113.
- Schubert, L., Schaeffer, S., Hwang Chung Hee, de Haan, Johannes. (1993). *EPILOG: The Computational System for Episodic Logic. USER GUIDE*, August.
- Shapiro, S. C. (1989). The CASSIE projects: An approach to natural language competence, in João P. Martins and Ernesto M. Morgado, editors, *EPIA*, volume 390 of *Lecture Notes in Computer Science*, pp. 362-380. Springer, in F. Morbini. A Self-Aware Agent. Thesis Proposal for Ph.D., Department of Computer Science, University of Rochester, New York (2006).
- Shapiro, S. C., Rapaport, W. J., Kandefer, M., Johnson, F. L., Goldfain, A. (2007). Metacognition in SnePS. *AI Magazine* 28,1 (Spring 2007), pp.17-31.
- Simmons, F.R. (1965). Answering English Questions by Computer: A Survey. *Communications of the ACM*, 8(1) pp. 53-70.
- Simmons, F.R. (1970). Natural language question-answering systems: 1969. *Computational Linguistics*, Vol. 13(1), January, pp. 15-30.
- Szabo A. (1973). *Απαρχαί των Ελληνικών Μαθηματικών*, Τ.Ε.Ε. Αθήνα.
- Tang, J., Hong, M., Zhang, D., Liang, B., Li, J. (2007). Information Extraction: Methodologies and Applications, in *Emerging Technologies of Text Mining* :

- Techniques and Applications*, Hercules A. Prado and Edilson Ferneda (Ed.), Idea Group Inc., Hershey, USA, pp.1-33.
- Tannery, P. (1876). *Memoires Scientifiques*, pub. J-L. Heiberg & H.-G. Zenthen, vol. II Paris, pp. 48-63, in A. SZABO (1973). *Απαρχαί των Ελληνικών Μαθηματικών*, T.E.E. Αθήνα.
- Uszkoreit, H. (2002). New chances for deep linguistic processing, in *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), August 24-September 1*. Morgan Kauffmann Press, in Fei – Yu Xu. Bootstrapping Relation Extraction from Semantic Seeds. A Dissertation submitted to the Philosophy Faculty of Saarland University for the Degree of Ph.D., 2008.
- van der Waerden, B. L. (1947/49). Die Arithmetik der Pythagoreer, *Math. Ann.* 120, pp. 127-153. In A. SZABO (1973). *Απαρχαί των Ελληνικών Μαθηματικών*, T.E.E. Αθήνα, σ.334.
- VanLehn K., Jones, R. M., Chi, M. T. H. (1992). A model of the self-explanation effect, *J. Learning Sci.* 2 (1), pp. 1–60.
- VanLehn, K., Ball, W., Kowalski, B. (1990). Explanation-based learning of correctness: Towards a model of the self-explanation effect, in: *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, LEA, Hillsdale, NJ.
- Veenman, M. V. J., Van Hout-Wolters, B. H. A. M., Afflerbach, P. (2006). Metacognition and learning: conceptual and methodological considerations. *Metacognition* 1: 3-14.
- Vere, S., Bickmore, T. (1990). A basic agent. *Computational Intelligence*, 6(1), pp. 41-60, in Fabrizio Morbini : A Self-Aware Agent. *Thesis Proposal*, University of Rochester, NY. In F. Morbini. A Self-Aware Agent. Thesis Proposal for Ph.D., Department of Computer Science, University of Rochester, New York (2006).
- Voorhees E., M. (2001). The Trec Question answering Track. *Natural Language Engineering*, Vol. 7, Issue 4.pp.361-378, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>).
- Vosniadou, S., Kyriakopoulou, N. (2006). The Problem of Metaconceptual Awareness in Theory Revision, in R. Sun & N. Miyake (Eds.) *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, Vancouver, Canada. pp. 2329-2334.

- Waerden, B. L. v. d. (1947/49). Die Arithmetik der Pythagoreer, *Math. Ann.* 120, pp. 127-153, in Szabo A. (1973). *Απαρχαί των Ελληνικών Μαθηματικών*, Τ.Ε.Ε. Αθήνα. Szabo A. (1973). *Απαρχαί των Ελληνικών Μαθηματικών*, Τ.Ε.Ε. Αθήνα.
- Waldinger, R. et al. (2003). Deductive Question Answering from Multiple Resources. *New Directions in Question Answering, AAAI*, in J. Kontos, and I. Malagardi. Question Answering from Procedural Semantics to Model Discovery, in *Encyclopedia of Human Computer Interaction*. C. Ghaoui, (Ed.), Liverpool John Moores University, UK (2006). (<http://www.idea-group.com/encyclopedia>)
- Weinert, F. E. (1987). Metacognition and motivation as determinants of effective learning and understanding, in F. E. Weinert and R. H. Kluwe (Eds.) *Metacognition, Motivation, and Understanding* (Hillsdale, NJ: Lawrence Erlbaum), pp. 1–16.
- Wellman, H. M. (1983). Metamemory revisited, in: M.T.H. Chi (Ed.), *Contributions to Human Development*, vol. 9, Trends in memory development research, S. Karger, AG, Basel, Switzerland.
- White, R. T. and Gunstone, R. F. (1989). Metalearning and conceptual change. *International Journal of Science Education*, 11, Special Issue, pp. 577–586.
- Wilson, T. D., Schooler, J. W. (1991). Thinking too much: Introspection can reduce the quality of preferences and decisions, *J. Personality Social Psychol.* 60 (2), pp. 181–192.
- Winograd, T. (1972). Understanding Natural Language. *Cognitive Psychology*, 3(1), pp. 1-191, January 1972, in F. Morbini. A Self-Aware Agent. Thesis Proposal for Ph.D., Department of Computer Science, University of Rochester, New York (2006).
- Winograd, T. (1990). Thinking machines: Can there be? Are we? In Derek Partridge and Yorick Wilkis, (Eds.), *The Foundations of Artificial Intelligence: A Sourcebook*, pp. 167-189. Cambridge University Press, in F. Morbini. A Self-Aware Agent. Thesis Proposal for Ph.D., Department of Computer Science, University of Rochester, New York (2006).
- Winterstein, D. (2004). Using Diagrammatic Reasoning for Theorem Proving in a Continuous Domain. Thesis in Informatics, The University of Edinburgh).
- Zeuthen, H. G. (1910). Sur la constitution des livres arithmétiques des Eléments d'Euclide et leur rapport à la question de l'irrationalité, *Oversigt over det kongelige Danske Videnskabernes Selskabs Forhandlinger*, vol. 5, pp. 395-435. Στο Γ.

- ΧΡΙΣΤΙΑΝΙΔΗΣ, (2003). *Θέματα από την ιστορία των Μαθηματικών*, Πανεπιστημιακές Εκδόσεις Κρήτης.
- Zion, M., Michalsky, T., Mevarech, Z. R. (2005). The effects of metacognitive instruction embedded within an asynchronous learning network on scientific inquiry skills. Bar-Ilan University, Israel. *International Journal of Science Education*, vol.27, No.8, pp. 957-983.
- Αναπολιτάνος, Διονύσιος Α. (1985). *ΕΙΣΑΓΩΓΗ ΣΤΗ ΦΙΛΟΣΟΦΙΑ ΤΩΝ ΜΑΘΗΜΑΤΙΚΩΝ*, ΝΕΦΕΛΗ (Εκδ.), Αθήνα.
- Αριστοτέλης. *Αναλυτικά πρότερα Α, Β*, μετ. Η. Νικολούδης, Εκδόσεις Κάκτος 1994, Αθήνα.
- Αριστοτέλης. *Αναλυτικά ύστερα Α, Β*, μετ. Η. Νικολούδης, Εκδόσεις Κάκτος 1994, Αθήνα.
- Βοσνιάδου, Σ., Κόντος, Ι., Καφετζόπουλος, Ε., Κριμπάς, Κ., Νικολινάκος, Δ. (2004). *ΓΝΩΣΙΑΚΗ ΕΠΙΣΤΗΜΗ: Η ΝΕΑ ΕΠΙΣΤΗΜΗ ΤΟΥ ΝΟΥ*, GUTENBERG.
- Κόντος, Ι. (1996). *Τεχνητή Νοημοσύνη και Λογομηχανική (Επεξεργασία Λόγου)*, Εκδόσεις Ε. ΜΠΕΝΟΥ, Αθήνα.
- Παυσανίας, Ελλάδος Περιήγησης, βιβλ. 10, «Φωκικά», κεφ.24.1,1-13. Εισαγωγή-Μετάφραση-Σχόλια: Ν. Παπαχατζής, Εκδόσεις Ι. ΖΑΧΑΡΟΠΟΥΛΟΣ, Αθήνα.
- Πλάτων. *Θεαίτητος*, (Εισαγωγή-Μετάφραση-Σχόλια: Β. Τατάκης), Εκδόσεις Ι. ΖΑΧΑΡΟΠΟΥΛΟΣ, Αθήνα.
- Πλάτων. *λάχης-μένων*, (Εισαγωγή-Μετάφραση-Σχόλια: Β. Τατάκης), *παρμενίδης*, (Εισαγωγή-Μετάφραση-Σχόλια: Ηλ. Λάγιος), Εκδόσεις Ι. ΖΑΧΑΡΟΠΟΥΛΟΣ, Αθήνα.
- Πλάτων. *ΠΟΛΙΤΕΙΑ*, μετ. Ι. Γρυπάρης, Εκδόσεις Ι. ΖΑΧΑΡΟΠΟΥΛΟΣ, Αθήνα.
- Πλάτων, *Νόμοι Ζ*, Εισαγωγή-Μετάφραση-Σχόλια: Φιλολογική Ομάδα Κάκτου, Εκδόσεις Ι. ΖΑΧΑΡΟΠΟΥΛΟΣ, Αθήνα.
- Πρόκλος, *Έις Πρώτον Ευκλείδου Στοιχείων (Ζητημάτων Α')*, Εισαγωγή-Μετάφραση-Σχόλια: Φιλολογική Ομάδα Κάκτου, Εκδόσεις ΚΑΚΤΟΣ, Αθήνα 2008.
- Πρόκλος, *Έις Πρώτον Ευκλείδου Στοιχείων (Πρόλογος)*, Εισαγωγή-Μετάφραση-Σχόλια: Φιλολογική Ομάδα Κάκτου, Εκδόσεις ΚΑΚΤΟΣ, Αθήνα 2008.
- Σταμάτης, Ε. Σ. (1975). *ΕΥΚΛΕΙΔΟΥ ΓΕΩΜΕΤΡΙΑ-ΣΤΟΙΧΕΙΑ*. τομ. 1, Ο.Ε.Δ.Β. ΑΘΗΝΑ.

Σταμάτης, Ε. Σ. (1975). *ΕΥΚΛΕΙΔΟΥ ΓΕΩΜΕΤΡΙΑ-ΘΕΩΡΙΑ ΑΡΙΘΜΩΝ*. том. 2,
Ο.Ε.Δ.Β. ΑΘΗΝΑ.

Σταμάτης, Ε. Σ. (1975). *ΕΥΚΛΕΙΔΟΥ ΓΕΩΜΕΤΡΙΑ-ΠΕΡΙ ΑΣΥΜΜΕΤΡΩΝ*. том. 3,
Ο.Ε.Δ.Β. ΑΘΗΝΑ.

Χριστιανίδης, Γ. (2003). *Θέματα από την ιστορία των Μαθηματικών*,
Πανεπιστημιακές Εκδόσεις Κρήτης.