

Μάιος 2015

Ατομική Διπλωματική Εργασία

**ΑΝΑΠΑΡΑΣΤΑΣΗ ΚΟΙΝΗΣ ΓΝΩΣΗΣ ΓΙΑ ΚΑΤΑΝΟΗΣΗ
ΙΣΤΟΡΙΩΝ**

Χριστίνα Αντωνίου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2015

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αναπαράσταση κοινής γνώσης για κατανόηση ιστοριών

Χριστίνα Αντωνίου

Επιβλέπων Καθηγητής

Αντώνης Κάκας

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2015

Ευχαριστίες

Για την εκπόνηση της διπλωματικής εργασίας μου θέλω να ευχαριστήσω τον επιβλέποντα μου, Καθηγητή κ. Αντώνη Κάκα, για την καθοδήγηση και τη βοήθεια του.

Τη διατριβή μου την αφιερώνω στους γονείς μου, στα αδέρφια μου, στη γιαγιά μου και ιδιαίτερα στη μητέρα μου για τη συνεχή ενθάρρυνση και υποστήριξη της.

Περίληψη

Ο σκοπός της διπλωματικής εργασίας είναι η αξιοποίηση των υπολογιστών για την αναπαράσταση ιστοριών και εκπαιδευτικών κειμένων για να διερευνήσουμε σε ποιο βαθμό ο ηλεκτρονικός υπολογιστής είναι σε θέση να αναπαραστήσει, να κατανοήσει τα περιεχόμενα των ιστοριών και των εκπαιδευτικών κειμένων και να απαντήσει σε ερωτήσεις κατανόησης.

Το εργαλείο Story Understanding Machine χρησιμοποιήθηκε για την αναπαράσταση ιστοριών. Η γλώσσα που εφαρμόζει το συγκεκριμένο σύστημα είναι η RAC. Το εργαλείο διαβάζει μια ιστορία, κατασκευάζει το μοντέλο κατανόησης του κειμένου και είναι σε θέση να απαντήσει σε ερωτήσεις.

Συμπεραίνουμε ότι η γλώσσα που εφαρμόζει το συγκεκριμένο σύστημα μας επιτρέπει να αναπαραστήσουμε τις ιστορίες και τα εκπαιδευτικά κείμενα. Είμαστε σε θέση να αξιοποιήσουμε τους υπολογιστές για την αναπαράσταση ιστοριών και εκπαιδευτικών κειμένων.

Καταλήγουμε ότι σε μεγάλο βαθμό ο ηλεκτρονικός υπολογιστής μπορεί να κατανοήσει τα περιεχόμενα των ιστοριών και των εκπαιδευτικών κειμένων. Τέλος το σύστημα δεν είναι ικανό να απαντήσει σε όλες τις ερωτήσεις των εκπαιδευτικών κειμένων.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
Κεφάλαιο 2	Αναπαράσταση, Συνοχή και Αποτέλεσμα	4
	2.1 Γενικά	4
	2.2 Κατανόηση Ιστορίας	5
	2.3 Χαρακτηριστικά της ανθρώπινης αναπαράστασης	7
	2.4 Πώς μετράμε την ποιότητα της αναπαράστασης	9
	2.5 Τα είδη των αναπαραστάσεων	10
	2.5.1 Γραμμική αναπαράσταση – σενάρια και causal chain	10
	2.5.2 Heterarchical representations –causal network	12
	2.5.2.1 Είδη αιτιοκρατικών σχέσεων	14
	2.5.3 Hierarchical representations-story trees	15
	2.6 .Συνοχή	15
	2.6.1 Τοπική και καθολική συνοχή (local and global coherence)	16
	2.6.1.1 Τοπική συνοχή (local coherence)	17
	2.6.1.2 Καθολική συνοχή (global coherence)	17
	2.6.2 Τα είδη συνοχής	19
	2.6.3 Πώς οι αναγνώστες δημιουργούν τη συνοχή	20
	2.7 Κατανόηση Ιστορίας και Συμπεράσματα	20
	2.7.1 Είδη συμπερασμάτων	22
	2.7.2 Backward συμπεράσματα	23
	2.7.3 Forward συμπεράσματα	23
Κεφάλαιο 3	Η λειτουργία του συστήματος Story Understanding Machine	29
	3.1 Γενικά	29
	3.2 Περιγραφή session	29
	3.3 Περιγραφή Ιδιοτήτων	30
	3.4 Περιγραφή Pre-session	31
	3.5 Narrative Sessions	32
	3.6 Ερωτήσεις	33
	3.7 Κοινή Γνώση	34

3.8 Προτεραιότητες μεταξύ επιχειρημάτων-κανόνων	36
3.9 Εμφάνιση των απαντήσεων	37
Κεφάλαιο 4 Αναπαράσταση Ιστοριών	44
4.1 Γενικά	42
4.2 Η πόρτα (door.pl)	42
4.3 Ασθενοφόρο	46
4.4 Πυροσβεστική	47
4.5 Παγωτό	48
4.5.1 Παγωτό (παραλλαγή)	48
4.6 Πονόδοντος	49
4.6.1 Πονόδοντος (παραλλαγή)	49
4.6.2 Πονόδοντος (παραλλαγή)	50
Κεφάλαιο 5 Αναπαράσταση Εκπαιδευτικών Κειμένων	57
5.1 Γενικά	52
5.2 Carly's Family	52
5.3 Spider Webs	57
5.4 A new pet shop	61
5.5 Anna's new Friend	67
5.6 Περιγραφή Δυσκολιών	71
ΚΕΦΑΛΑΙΟ 6 Συμπεράσματα	73
Βιβλιογραφία	76

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή

1

1.1 Εισαγωγή

Οι ψυχολόγοι εμπνεύστηκαν από την λειτουργία του υπολογιστή και παρομοίασαν τη δομή του ανθρώπινου εγκεφάλου κατά τη διαδικασία της μάθησης και κατάκτησης γνώσης με αυτόν. Η επεξεργασία πληροφοριών είναι χαρακτηριστικό και του ηλεκτρονικού υπολογιστή και του ανθρώπινου εγκεφάλου. Τρεις φάσεις παρατηρούνται και στις δυο περιπτώσεις (Πόρποδας, 1993):

1. πρόσληψη πληροφοριών
2. επεξεργασία πληροφοριών
3. εξαγωγή συμπερασμάτων

Η πρώτη φάση στον άνθρωπο γίνεται μέσω των αισθήσεων ενώ στον υπολογιστή π.χ. μέσω πληκτρολογίου. Η δεύτερη φάση είναι επεξεργασία των πληροφοριών και αφορά την κωδικοποίηση των πληροφοριών, τη σύνδεση με τις υπάρχουσες πληροφορίες (διαμορφώνονται οι νέες με τις παλιές) και την αποθήκευση τους. Αυτή η φάση στον άνθρωπο πραγματοποιείται μέσα από τα εγκεφαλικά συστήματα των νευρώνων του. Τέλος στη τρίτη φάση ο άνθρωπος εξάγει-μεταδίδει τις γνώσεις του στο περιβάλλον μέσα από ειδικά συστήματα όπως η γλώσσα, οι κινήσεις κ.τ.λ. (Loftus & Loftus, 1976) όπως αναφέρεται στο (Πόρποδα, 1993). [2]

Κατά τον Πόρποδα (1993) κρίθηκε αναγκαίο να γίνει ο διαχωρισμός μεταξύ της «οργανικής δομής» (hardware) και της λειτουργική δομής (software) για την επέκταση της αντιστοιχίας υπολογιστή και ανθρώπινου εγκεφάλου. Στον άνθρωπο η «οργανική δομή» σχετίζεται με τα οργανικά συστήματα τα οποία αποτελούν αντικείμενο μελέτης για τις επιστήμες της βιολογίας, ιατρικής, βιοχημείας (κ.ά.). Η «οργανική δομή» στους υπολογιστές σχετίζεται με το υλικό του υπολογιστή, δηλαδή τη κατασκευή του υπολογιστή, το οποίο αποτελεί αντικείμενο εργασίας για τους μηχανικούς. Η λειτουργική δομή έχει να κάνει με τον τρόπο επεξεργασίας των πληροφοριών. Το πρόγραμμα είναι υπεύθυνο για την επεξεργασία πληροφοριών και οι δημιουργοί αυτών των προγραμμάτων είναι οι προγραμματιστές. Στον ανθρώπινο εγκέφαλο υπάρχουν αντίστοιχα «προγράμματα». Για παράδειγμα (Πόρποδα, 1993) η αριθμητική πράξη της πρόσθεσης δεν παρατηρείται στον άνθρωπο από τους πρώτους μήνες της ζωής του αλλά μαθαίνει να προσθέτει σε ένα μεταγενέστερο στάδιο της ζωής του. Σύμφωνα με το Πόρποδα (1993) «ένα πρόγραμμα διεκπεραίωσης της πρόσθεσης αποκτήθηκε από το άτομο». [2]

Η επιστήμη της ψυχολογίας εμπνεύστηκε από την λειτουργία των υπολογιστών και ερμήνευσε την λειτουργία του ανθρώπινου εγκεφάλου κατά τη διαδικασία μάθησης και κατάκτησης γνώσης με βάση τη λειτουργία του υπολογιστή. Σύμφωνα με τον Πόρποδα καθώς ο άνθρωπος προσλαμβάνει πληροφορίες μέσω των αισθήσεων παρατηρείται μια αλληλεπίδραση μεταξύ των πληροφοριών και των γνωστικών δομών που έχουν σχηματιστεί.» [2]

Η Τεχνητή Νοημοσύνη ως πεδίο της Επιστήμης της Πληροφορικής έδωσε μεγάλη βαρύτητα στην κατανόηση κείμενου. Η ανάπτυξη της επιστήμης της πληροφορικής και πιο συγκεκριμένα η ανάπτυξη του διαδικτύου τοποθέτησε την κατανόηση κείμενου σε ακόμα πιο σημαντική θέση καθώς τα δεδομένα στις ιστοσελίδες είναι σε μορφή κειμένου. Η αλληλεπίδραση ανθρώπου-υπολογιστή θα μπορούσε να εκμεταλλευτεί τις δυνατότητες της κατανόησης κειμένου όπως και άλλες μορφές επεξεργασίας της φυσικής γλώσσας. Οι δραστηριότητες σε αυτόν τον τομέα είναι πολλές από τον κλάδο της Τεχνητής Νοημοσύνης. [3]

Η διπλωματική εργασία έχει σκοπό την αξιοποίηση των υπολογιστών για την αναπαράσταση ιστοριών και εκπαιδευτικών κειμένων για να διερευνήσουμε σε ποιο βαθμό ο ηλεκτρονικός υπολογιστής είναι σε θέση να αναπαραστήσει, να κατανοήσει τα περιεχόμενα των ιστοριών

και των εκπαιδευτικών κειμένων και να απαντήσει στις ερωτήσεις που θέτουμε για τα συγκεκριμένα κείμενα.

Στα επόμενα κεφάλαια παρουσιάζεται το θεωρητικό υπόβαθρο για την αναπαράσταση των ιστοριών, έπειτα περιγράφεται ο τρόπος λειτουργίας του συστήματος Story Understanding Machine (STAR) μέσω του οποίου μοντελοποιούμε τον κόσμο των ιστοριών και απαντάει τις ερωτήσεις που εισάγουμε σε αυτό. Στα επόμενα κεφάλαια γίνεται η αναπαράσταση ιστοριών, αλλά και εκπαιδευτικών κειμένων στο STAR σύστημα. Στο τελευταίο κεφάλαιο της διπλωματικής εργασίας αναφέρονται τα συμπεράσματα σχετικά με την αναπαράσταση των ιστοριών και των εκπαιδευτικών κειμένων.

Κεφάλαιο 2

Αναπαράσταση, Συνοχή και Αποτέλεσμα

2.1 Γενικά	15
2.2 Κατανόηση Ιστορίας	18
2.3 Χαρακτηριστικά της ανθρώπινης αναπαράστασης	20
2.4 Πώς μετράμε την ποιότητα της αναπαράστασης	22
2.5 Τα είδη των αναπαραστάσεων	24
2.5.1 Γραμμική αναπαράσταση – σενάρια και causal chain	24
2.5.2 Heterarchical representations –causal network	24
2.5.2.1 Είδη αιτιοκρατικών σχέσεων	24
2.5.3 Hierarchical representations-story trees	24
2.6 Συνοχή	24
2.6.1 Τοπική και καθολική συνοχή (local and global coherence)	24
2.6.1.1 Τοπική συνοχή (local coherence)	24
2.6.1.2 Καθολική συνοχή (global coherence)	24
2.6.2 Τα είδη συνοχής	24
2.6.3 Πώς οι αναγνώστες δημιουργούν τη συνοχή	24
2.7 Κατανόηση Ιστορίας και Συμπεράσματα	24
2.7.1 Είδη συμπερασμάτων	25
2.7.2 Backward συμπεράσματα	25
2.7.3 Forward συμπεράσματα	25
2.7.4 Σκοπός της κατανόησης	25
2.8 Ψυχολογικό υπόβαθρο	20

2.1 Γενικά

Σε αυτό το κεφάλαιο εξετάζονται τα ακόλουθα χαρακτηριστικά:

1. Τα είδη των αναπαραστάσεων που δημιουργούνται κατά τη διάρκεια της κατανόησης.

2. Η συνοχή είναι αναπόσπαστο κομμάτι στη δημιουργία των αναπαραστάσεων.
3. Τα συμπεράσματα που εξάγονται για την ιστορία είναι η βάση για να υπάρξει συνοχή στην αναπαράσταση.

2.2 Κατανόηση Ιστορίας

Πρόσφατη έρευνα (Smith & Hancox, 2001) έδειξε ότι η κατανόηση των ιστοριών είναι επεξηγηματική (explanation driven) (Grasser et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001). Δηλαδή συνήθως οι αναγνώστες για να κατανοήσουν ένα κείμενο απαντούν σε ερωτήσεις του τύπου 'Γιατί;', και όχι σε ερωτήσεις του τύπου 'Τι θα συμβεί μετά;', 'Πώς;', 'Που;', ή 'Πότε;'. Με αυτόν τον τρόπο εξάγουν ένα σύνολο από επεξηγήσεις για στοιχεία που αναφέρονται στην ιστορία. Οι επεξηγήσεις είναι προσανατολισμένες «προς τα πίσω» (backward oriented) μέσα στο χρόνο της αφήγησης και δημιουργούνται από τις πληροφορίες του κειμένου και από την από την ενεργοποίηση της προηγούμενης γνώσης του αναγνώστη και χρησιμεύουν για να αποδώσουν τον κεντρικό νόημα της ιστορίας (Trabasso and Magliano, 1996) όπως αναφέρεται στο (Smith & Hancox, 2001). Οι επεξηγήσεις μπορεί να είναι γεγονότα που αναφέρονται ρητά στο κείμενο ή υποθέσεις, οι οποίες εξηγούν πως τα γεγονότα της ιστορίας σχετίζονται μεταξύ τους. [4]

Είναι προφανές ότι η κατανόηση ακολουθεί τη διαδρομή της (explanation driven) επεξήγησης (Smith & Hancox, 2001) αλλά άλλοι ερευνητές προτείνουν ότι η κατανόηση ακολουθεί τη διαδρομή της πρόβλεψης (expectation-driven). Η κατανόηση βασίζεται στην πρόβλεψη συμπερασμάτων. Ο αναγνώστης συμπληρώνει εκείνα τα κομμάτια της ιστορίας που δεν αναφέρονται ρητά στο κείμενο αλλά είναι γνωστά (για παράδειγμα 'ο Γιώργος άργησε στο σχολείο', ο αναγνώστης είναι σε θέση να κάνει την ακόλουθη πρόβλεψη: 'ο Γιώργος είναι μαθητής'). Καθώς ανακαλούμε από τη μνήμη γεγονότα μπορούμε να συμπληρώσουμε τη πληροφορία που λείπει από το κείμενο. Στο συγκεκριμένο μοντέλο οι προβλέψεις που κάνει ο αναγνώστης για την ιστορία είτε υποστηρίζονται από το κείμενο είτε απορρίπτονται σύμφωνα με το κείμενο. Ωστόσο η επιστήμη της ψυχολογίας αναφέρει ότι η πρόβλεψη συμπερασμάτων έχει περιορισμένη εμβέλεια, ιδιαίτερα περιορίζεται από το περιεχόμενο του κειμένου που έχει διαβάσει μέχρι τώρα ο αναγνώστης (Tabasso and Magliano, 1996) όπως αναφέρεται στο (Smith & Hancox, 2001). [4]

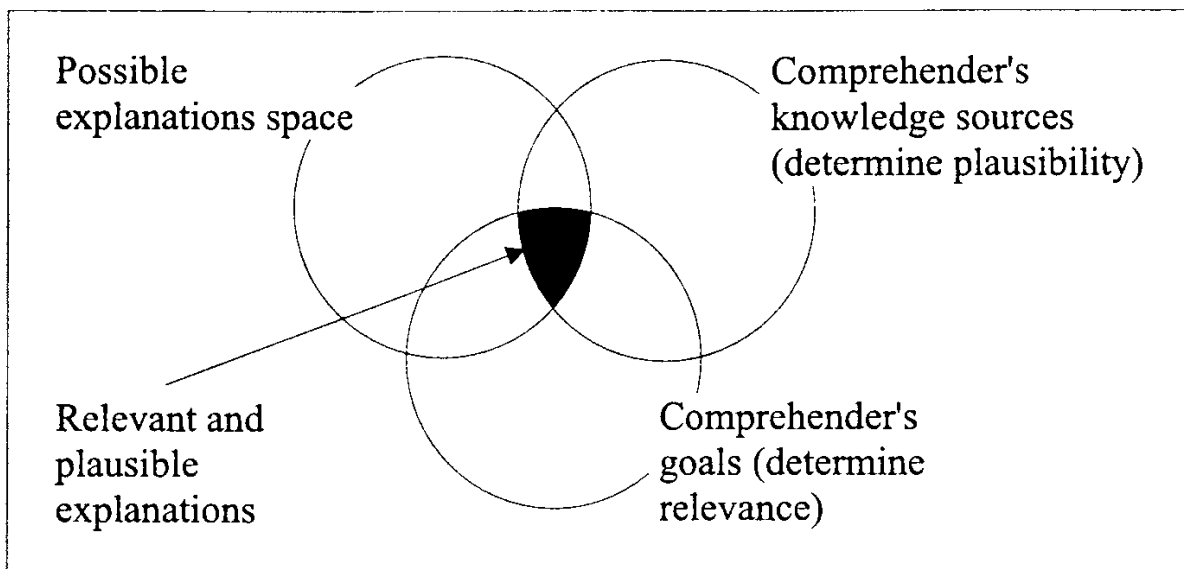
Οι επεξηγήσεις που δίνουμε για την ιστορία σε συνδυασμό με τη σημασιολογική κατανόηση των πληροφοριών που περιγράφονται στην ιστορία, αποθηκεύονται ως αναπαράσταση στην επεισοδιακή μνήμη. [4]

Αξίζει να αναφερθεί ότι πολλές επεξηγήσεις μπορεί να δημιουργηθούν για μια ιστορία ωστόσο η αναπαράσταση δεν είναι δυνατό να περιέχει όλες αυτές τις πληροφορίες. Από τη διαδικασία της κατανόησης δημιουργείται μια αναπαράσταση που περιλαμβάνει ένα υποσύνολο των επεξηγήσεων.

Υπάρχουν δύο βασικοί μηχανισμοί που βοηθούν τον αναγνώστη να κρατήσει μόνο τις επεξηγήσεις που θεωρεί λογικές και σχετικές και τις υπόλοιπες να τις απορρίψει (Norvig, 1989) όπως αναφέρεται στο (Smith & Hancox, 2001) :

- Εφαρμογή γνώσεων (application of knowledge source)
Για την κατανόηση του κειμένου, για την εξαγωγή συμπερασμάτων και επεξηγήσεων χρησιμοποιείται η προηγούμενη γνώση του αναγνώστη (Crystal, 1971) όπως αναφέρεται στο (Smith & Hancox, 2001).
- Consultation of comprehension goals
Ο αναγνώστης θέτει κάποιους στόχους και σύμφωνα με αυτούς βγάζει αξιόπιστες ή λογικές επεξηγήσεις. Για παράδειγμα (Smith & Hancox, 2001) αν απαιτείται μια λεπτομερή αναπαράσταση τότε ο στόχος του είναι να συμπεριλάβει πολλές πιθανές επεξηγήσεις. [4]

Ένα τυπικό υπόδειγμα αποκοπής-απόρριψης πιθανών επεξηγήσεων δίνεται στην παρακάτω εικόνα (Smith & Hancox, 2001).



Εικόνα 1 (Smith & Hancox, 2001)

Οι Smith και Hancox (2001) αναφέρουν ότι μια αναπαράσταση θεωρείται ένα υποσύνολο των πιθανών επεξηγήσεων. «Η ικανότητα να συνοψίζεις είναι βασική ιδιότητα της νοημοσύνης» Alterman (1991) όπως αναφέρεται στο (Smith & Hancox, 2001). Σύμφωνα με αυτήν την ιδιότητα ο αναγνώστης διατηρεί μόνο τις λογικές επεξηγήσεις στην αναπαράσταση του. Αυτή η ιδιότητα υπονοείται στα περισσότερα συστήματα κατανόησης και σπάνια δηλώνεται ρητά. (Δηλαδή τα υπολογιστικά συστήματα κατακλύζονται από άσχετες επεξηγήσεις.)

Ο άνθρωπος έχει περιορισμένη επεξεργασία και χωρητικότητα μνήμης, οι μηχανισμοί application of knowledge source και consultation of comprehension goals δίνουν τη δυνατότητα στον αναγνώστη να κρατήσει μόνο εκείνες τις πληροφορίες που θεωρεί σημαντικές από το σύνολο των πιθανών επεξηγήσεων. [4]

2.3 Χαρακτηριστικά της ανθρώπινης αναπαράστασης

Σύμφωνα με τους Smith & Hancox (2001) οι δυο μηχανισμοί που αναφέρθηκαν πιο πάνω, οι γνώσεις και οι στόχοι του αναγνώστη καθορίζουν ποιες είναι οι πιθανές επεξηγήσεις που θα απορρίψει. Η απόρριψη πιθανών επεξηγήσεων και η διάτρηση των λογικών επεξηγήσεων έχει ως αποτέλεσμα την αναπαράσταση, η οποία δεν μπορεί να χαρακτηριστεί ως ένα αντίγραφο των πληροφοριών που περιέχει η ιστορία. Οι μηχανισμοί που χρησιμοποιούνται για την απόρριψη πιθανών επεξηγήσεων προκύπτουν από την εμφάνιση των ακόλουθων χαρακτηριστικών:

- Πρόσθεση, αφαίρεση και αναδιοργάνωση

Οι άνθρωποι προσθέτουν γεγονότα και καταστάσεις στις αναπαραστάσεις του, τα γεγονότα και οι καταστάσεις δεν αναφέρονται ρητά στο κείμενο (Mandler and Johnson, 1977) όπως αναφέρεται στο (Smith & Hancox, 2001). Αυτά τα γεγονότα μπορεί να είναι στο μικρό-επίπεδο ή στο μακρο-επίπεδο (η προσθήκη νέων μακρο-δηλώσεων) (see van Dijk, 1977) όπως αναφέρεται στο (Smith & Hancox, 2001). Έκτος από την πρόσθεση γεγονότων και καταστάσεων, οι άνθρωποι διαγράφουν πληροφορίες που θεωρούν ότι είναι άσχετες. [4]

Η προϋπάρχουσα γνώση του αναγνώστη βοηθάει στην αναδιοργάνωση της πληροφορίας του κειμένου και τελικά η πληροφορία στην αναπαράσταση που περιέχεται έχει αναδιοργανωθεί σύμφωνα με την γνώση του αναγνώστη (για παράδειγμα όταν πάμε σινεμά πηγαίνουμε βγάζουμε εισιτήριο, μπαίνουμε στην αίθουσα, ψάχνουμε για θέση και καθόμαστε, και φεύγουμε όταν τελειώσει η ταινία, υπάρχει μια συγκεκριμένη ακολουθία γεγονότων). Αν μια ιστορία φαίνεται να ανήκει σε μια γενική κατηγορία (όπως ένα παραμύθι) τότε και η ακολουθία γεγονότων της ιστορίας είναι συγκεκριμένη, αυτό έχει ως αποτέλεσμα και η αναπαράσταση να αναδιοργανώνεται σύμφωνα με την συγκεκριμένη ακολουθία γεγονότων (Mandler and Johnson, 1977; Thorndyke, 1977) όπως αναφέρεται στο (Smith & Hancox, 2001). [4]

- Explicated causal relations

Οι άνθρωποι συνδέουν τα γεγονότα μεταξύ τους λόγω αιτίας. Ο αναγνώστης συνδέει τα γεγονότα από την αρχή μέχρι το τέλος της ιστορίας. Οι σχέσεις αυτές είναι σχέσεις αιτίας. Το αποτέλεσμα είναι να αποκτήσει συνοχή η αναπαράσταση. Για παράδειγμα οι δράσεις μπορούν να εξηγηθούν από τους σκοπούς των χαρακτήρων (Graesser et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001) . [4]

- Variable-depth

Ένας αναγνώστης μπορεί να δημιουργήσει μια αναπαράσταση που είναι διαφορετική από την αναπαράσταση ενός άλλου αναγνώστη. Αυτό συμβαίνει γιατί υπάρχουν παράγοντες όπως η χωρητικότητα της εργαζόμενης μνήμης (working memory capacity) (Whitney et al., 1991) όπως αναφέρεται στο (Smith & Hancox, 2001), η περιοχή γνώσης (Noordman and Vonk, 1992) όπως αναφέρεται στο (Smith & Hancox, 2001) και οι στόχοι του αναγνώστη (Graesser et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001) που επηρεάζουν την αναπαράσταση του αναγνώστη. [4]

2.4 Πώς μετράμε την ποιότητα της αναπαράστασης

Κατά τους Smith και Hancox (2001) υπάρχουν διαφορές ανάμεσα σε αυτά που διαβάζουν οι αναγνώστες και σε αυτά που θυμούνται από αυτά που διάβασαν. Το ερώτημα είναι το εξής σύμφωνα με τους Smith και Hancox (2001): πώς μπορεί ένας αναγνώστης να διακρίνει ποιες είναι οι πληροφορίες που πρέπει να κρατήσει και ποιες είναι αυτές που δεν χρειάζονται. Οι Smith και Hancox (2001) αναφέρουν τις επιθυμητές ιδιότητες (Alterman, 1991) που πρέπει να έχει μια περίληψη. Αυτές οι ιδιότητες εφαρμόζονται σε κάθε τύπο αναπαράστασης.

Μια αναπαράσταση θα πρέπει:

- Να περιέχει τα σημαντικά γεγονότα

Μια αναπαράσταση είναι χρήσιμη (σε μέγεθος) όταν περιέχει μόνο τα σημαντικά γεγονότα της ιστορίας. Τα πιο σημαντικά γεγονότα είναι εκείνα τα γεγονότα τα οποία δημιουργούν την πιο αφηρημένη έκδοση της αναπαράστασης. (Για παράδειγμα στην ιστορία A New Pet Shop κρατάμε τις πληροφορίες ότι υπάρχουν γατάκια, πουλιά και ψάρια δεν διατηρούμε πληροφορίες που έχουν να κάνουν με την περιγραφή αυτών των ζώων.) [4]

- Να έχει συνοχή

Μια αναπαράσταση θα πρέπει να έχει (hold together and make sense) νόημα (Alterman and Bookman, 1990) όπως αναφέρεται στο (Smith & Hancox, 2001). Οι Smith και Hancox (2001) περιγράφουν δύο τρόπους για να δημιουργηθεί η συνοχή στην αναπαράσταση. Ο πρώτος τρόπος είναι να συνδέσουμε τις ακολουθίες γεγονότων μέσα από μια ποικιλία σχέσεων. Οι αιτιοκρατικές σχέσεις (σχέσεις αιτίας) θεωρούνται οι πιο σημαντικές. Και ο δεύτερος τρόπος είναι να δώσουμε την κεντρική ιδέα της ιστορίας, δηλαδή να βρούμε ένα ή περισσότερα γεγονότα τα οποία ανήκουν στο μακρο-επίπεδο (ψηλά στην ιεραρχική δομή). [4]

- Να "καλύπτει" το περιεχόμενο

Η αναπαράσταση θα πρέπει να περιέχει είτε άμεσα είτε έμμεσα όλα τα γεγονότα της ιστορίας. Αυτό σημαίνει ότι για τα γεγονότα που αναφέρονται ρητά στο κείμενο αλλά δεν βρίσκονται στην αναπαράσταση αυτά θα πρέπει να προκύπτουν ως συμπέρασμα.

[4]

Σύμφωνα με τους Smith και Hancox, (2001) για να δημιουργήσουμε μια ιδανική αναπαράσταση τα παραπάνω χαρακτηριστικά θα μας βοηθήσουν να το πετύχουμε καθώς αναζητούμε αυτές τις πληροφορίες ενώ διαβάζουμε μια ιστορία. Οι αναγνώστες πρέπει να δημιουργήσουν μια αναπαράσταση της ιστορίας που έχει συνοχή και να καλύψουν και τα κύρια γεγονότα μια ιστορίας και τα δευτερεύοντα (έμμεσα) γεγονότα. Τα χαρακτηριστικά που απαιτούνται για να δημιουργηθεί μια ιδανική αναπαράσταση μπορούν να βοηθήσουν όταν υπάρχει δίλλημα για το ποια αναπαράσταση θα επικρατήσει. [4]

2.5 Τα είδη των αναπαραστάσεων

Οι Smith και Hancox (2001) υποστηρίζουν ότι η αναπαράσταση είναι ένα σύνολο από συνδεδεμένους κόμβους. Κάθε κόμβος της αναπαράστασης αντιπροσωπεύει ένα γεγονός που μπορεί να αναφέρεται ρητά ή έμμεσα στην ιστορία. Οι σύνδεσμοι ανάμεσα στους κόμβους χωρίζονται σε δύο κατηγορίες. Αυτοί που βρίσκονται χαμηλά στην ιεραρχία και ονομάζονται τοπικοί. Αυτοί δημιουργούνται ανάμεσα σε δύο κόμβους. Στην δεύτερη κατηγορία ανήκουν οι σύνδεσμοι που βρίσκονται ψηλά στην ιεραρχία και ονομάζονται καθολικοί. Καθολική σύνδεση δημιουργείται όταν ένα γεγονός-κόμβος συνδέεται με πολλούς κόμβους (clusters of nodes), οι οποίοι ανήκουν στα κατώτερα επίπεδα. Το είδος των συνδέσμων ανάμεσα στους κόμβους καθορίζει το είδος της συνοχής. Σύμφωνα με τη βιβλιογραφία από το κλάδο της ψυχολογίας μια αναπαράσταση αποτελείται από ένα σύνολο κόμβων και ένα σύνολο συνδέσμων που συχνά ονομάζεται «propositional textbase» (Kintsch and van Dijk; Garnham, 1987) όπως αναφέρεται στο (Smith & Hancox, 2001). [4]

Σε μια αναπαράσταση δημιουργούνται καθολικοί και τοπικοί σύνδεσμοι. Μια αναπαράσταση μπορεί να περιλαμβάνει περισσότερο καθολικούς συνδέσμους γιατί επιθυμεί να υπάρχει μια γενικότερη συνοχή. Υπάρχουν θεωρίες που καθορίζουν τη δομή της αναπαράστασης, αυτό σημαίνει ότι καθορίζουν και το είδος των συνδέσμων που δημιουργούνται στην συγκεκριμένη αναπαράσταση. Για παράδειγμα (Smith & Hancox, 2001) στην αναπαράσταση μέσω δικτύων ή τα αιτιοκρατικά δίκτυα (causal network) εντοπίζουμε καθολικές συνδέσεις. Υπάρχουν τρεις βασικοί τύποι αναπαραστάσεων για τις ιστορίες: η γραμμική (linear), heterarchical-causal network και η ιεραρχική. [4]

2.5.1 Γραμμική αναπαράσταση – σενάρια και causal chain

Η πιο γνωστή αναπαράσταση είναι η αιτιώδη αλυσίδα (causal chain). Η αιτιώδη αλυσίδα δημιουργείται από το script-based processing (Schank and Abelson, 1978) όπως αναφέρεται στο (Smith & Hancox, 2001). Τη συγκεκριμένη μέθοδο την χρησιμοποιούν διάφορα μοντέλα, και υλοποιείται όπως ένα πρόγραμμα (Lehnert et al., 1983: Schank and Abelson, 1978) όπως αναφέρεται στο (Smith & Hancox, 2001).

Μια αιτιώδη αλυσίδα είναι μια ακολουθία γεγονότων δηλαδή μια ακολουθία κόμβων, η οποία είναι σταθερή και υπάρχουν σχέσεις αιτίας ανάμεσα στους κόμβους. Για παράδειγμα η αγορά ενός εισιτηρίου επιτρέπει να περάσεις το εμπόδιο, το οποίο με τη σειρά του επιτρέπει να επιβιβαστείς στο τρένο το οποίο με τη σειρά του επιτρέπει να μετακινηθείς από το ένα μέρος στο άλλο (Dyer, 1992) όπως αναφέρεται στο (Smith & Hancox, 2001). Πολλοί ερευνητές (Smith & Hancox, 2001) εντοπίζουν μόνο έναν πρόγονο και ένα απόγονο για κάθε γεγονός.

Σύμφωνα με τη Βακάλη (1995) οι Shank και Abelson αναφέρουν ότι οι γνώσεις που έχουμε για τις καθημερινές δραστηριότητες όπως πηγαίνω στο εστιατόριο ή παρακολουθώ μια διάλεξη οργανώνονται σε σενάρια.

- Τα σενάρια βοηθούν να διατυπώσουμε προβλέψεις για το τι πρόκειται να γίνει.
- Είναι χρήσιμα για την κατανόηση γραπτών ή προφορικών ιστοριών.
- Παρέχουν βοήθεια για την επαναφορά γεγονότων από το παρελθόν.

Το σενάριο «πηγαίνω στο εστιατόριο» έχει έναν τίτλο. Μέσα σε ένα σενάριο μπορούμε να βρούμε πληροφορίες σχετικά με τους λόγους που άρχισε το σενάριο, το τι χρειάζεται για να εκτελεστεί το συγκεκριμένο σενάριο, τα αντικείμενα που χρησιμοποιούνται, οι ρόλοι που εντοπίζονται για τα άτομα που συμμετέχουν σε αυτήν την δραστηριότητας και ποιο είναι το τέλος που αναμένουμε από ένα τέτοιο σενάριο. [1]

Το σενάριο χωρίζεται σε επεισόδια (σκηνές ή πράξεις) για παράδειγμα (Βακάλη, 1995) μπαίνω στο εστιατόριο, παραγγέλνω, τρώω, πληρώνω και φεύγω από το εστιατόριο. Κάθε επεισόδιο αποτελείται από ενέργειες για παράδειγμα (Βακάλη, 1995) η σκηνή της εισόδου περιλαμβάνει τα εξής: μπαίνω στο εστιατόριο, ψάχνω για τραπέζι, επιλέγω σε ποιο τραπέζι θα καθίσω, πάω στο τραπέζι και κάθομαι. «Όλα αυτά συνιστούν μια οργανωμένη, διαδοχική στο χρόνο και χώρο αναπαράσταση της εννοιολογικής γνώσης σε σχέση με μια

δραστηριότητα, η οποία περικλείει διάφορες επιμέρους πράξεις που συνδέονται μεταξύ τους (Βακάλη, 1995 σελ. 186).» [1]

Το σενάριο θεωρείται ένα γενικό πλαίσιο μέσα στο σενάριο υπάρχουν μεταβλητές και παίρνουν διάφορες τιμές. Για παράδειγμα ο πελάτης μπορεί να είναι ευγενικός ή να είναι αγενής. Όταν η μεταβλητή συγκεκριμενοποιείται σημαίνει ότι έχει πάρει μια συγκεκριμένη τιμή (Βακάλη, 1995). [1]

Σύμφωνα με τη Βακάλη (1995) είναι δυνατό το σενάριο της «νυκτερινής διασκέδασης» μπορεί να περιέχει το σενάριο της «ταβέρνας» και αυτό με τη σειρά του να περιέχει το σενάριο του «φαγητού» αυτό μπορεί να συμβεί και αντίστροφα. [1]

Οι Smith και Hancox (2001) αναφέρουν ότι Schank και Abelson διακρίνουν πέντε είδη συνδέσμων που χρησιμοποιούνται για να δημιουργηθεί η αιτιοκρατική αλυσίδα:

1. Οι δράσεις μπορεί να αλλάξουν τις καταστάσεις.
2. Οι καταστάσεις ενεργοποιούν δράσεις.
3. Οι καταστάσεις απενεργοποιούν δράσεις
4. Οι καταστάσεις ή δράσεις μπορεί να «αρχικοποιήσουν» τα νοητικά μοντέλα (mental states)
5. Τα νοητικά μοντέλα (mental states) μπορεί να δημιουργήσουν δράσεις (λειτουργούν ως κίνητρα για δράσεις)

2.5.2. Heterarchical representations –causal network

Αυτή η μορφή αναπαράστασης δημιουργήθηκε για να καλύψει τα μειονεκτήματα της αιτιώδης αλυσίδας (Trabasso and van den Broek, 1985) όπως αναφέρεται στο (Smith & Hancox, 2001). Για την συγκεκριμένη μορφή αναπαράστασης ένα μοναδικό γεγονός μπορεί να διαθέτει πολλούς προγόνους και απογόνους σε αντίθεση με την αιτιώδη αλυσίδα που κάθε κόμβος διαθέτει μόνο έναν απόγονο και έναν πρόγονο. Μη γειτονικά γεγονότα μπορούν να συνδεθούν μεταξύ τους και από ένα γεγονός μπορεί να ξεκινούν πολλές ακμές ή να καταλήγουν πολλές ακμές. [4]

Η συγκεκριμένη μορφή αναπαράστασης αγνοεί τα ασήμαντα γεγονότα, περισσότερο από την αιτιώδη αλυσίδα. Οι κόμβοι ενός δικτύου αναπαριστούν μεμονωμένα γεγονότα όπως για παράδειγμα η κατάσταση του χαρακτήρα ή δράσεις. Υπάρχουν σχέσεις αιτίας ανάμεσα στους κόμβους του δικτύου. Για παράδειγμα (Smith & Hancox, 2001) ένα ζευγάρι από κόμβους (A και B) ερμηνεύεται ως εξής : ο A προκαλεί τον B. [4]

Δεν είναι πάντα σταθερή και δεν έχει απόλυτη δύναμη η σύνδεση ανάμεσα στους κόμβους των γεγονότων. Υπάρχουν διάφοροι παράγοντες που είναι υπεύθυνοι για να δημιουργηθεί μια σχέση αιτίας για δύο γεγονότα A και B (Trabasso and van den Broek, 1985: Trabasso et al., 1989: Trabasso and Magliano, 1996: van den Broek and Trabasso, 1986: van den Broek, 1990a: van den Broek, 1990b: van den Broek, 1994) όπως αναφέρεται στο (Smith & Hancox, 2001) :

- Χρονική προτεραιότητα – temporal priority

Η αιτία προηγείται της συνέπειας. «Το γεγονός που προκάλεσε να συμβεί κάτι βρίσκεται πάντα πριν το αποτέλεσμα» ή «η αιτία ποτέ δεν συμβαίνει μετά τη συνέπεια» (van den Broek, 1990b) όπως αναφέρεται στο (Smith & Hancox, 2001). (Παράδειγμα: Το σπίτι έπιασε φωτιά. Ήρθε η πυροσβεστική.)

- Λειτουργικότητα – operativity

Η αιτία πρέπει να είναι ενεργοποιημένη (operative) τη στιγμή που συμβαίνει το αποτέλεσμα. 'Για παράδειγμα 'ο Γιάννης συνάντησε την Μαρία' αυτό είχε ως αποτέλεσμα: 'ο Γιάννης παντρεύτηκε τη Μαρία'. 'ο Γιάννης γνώρισε την Μαρία' αυτή πρόταση προήλθε από την πρόταση 'ο Γιάννης συνάντησε την Μαρία' αυτή πρέπει να είναι ενεργοποιημένη (operative) πριν την πρόταση 'ο Γιάννης παντρεύτηκε την Μαρία' (Smith & Hancox, 2001).

(Παράδειγμα: Ο Γιάννης θέλει παγωτό. Ο Γιάννης απολαμβάνει το παγωτό στο πάρκο.)

- Απαραίτητο –Αναγκαίο –necessity

Αν το A δεν συμβεί (σύμφωνα με το κείμενο) τότε δεν μπορεί να συμβεί και το B. Για παράδειγμα 'John picked up a knife' είναι αναγκαία συνθήκη για το εξής: 'John stabbed Peter with Knife' (αν ο John δεν σήκωνε το μαχαίρι τότε δεν θα μπορούσε να συμβεί η δεύτερη πρόταση) (Smith & Hancox, 2001).

(Ο Γιάννης έπεσε από τις σκάλες. Η Μαίρη τηλεφώνησε για να έρθει ασθενοφόρο.)

- Επάρκεια – sufficiency

Ο περιορισμός της επάρκειας είναι πιο ασθενέστερος από τον περιορισμό της αναγκαιότητας, αν το A συμβεί (σύμφωνα με το κείμενο) τότε είναι πιθανόν να συμβεί το B (van den Broek, 1994). Για παράδειγμα, ‘John struck a match’ αυτό είναι μια δυνατή (επαρκής) συνθήκη για την επόμενη πρόταση ‘John lit his cigarette’. Δύο γεγονότα τα οποία έγιναν σχεδόν την ίδια στιγμή (Smith & Hancox, 2001).

(Παράδειγμα: Η Κατερίνα πήρε τη χτένα. Η Κατερίνα χτενίζει τα μαλλιά της.

Ο Γιάννης τρώει γλυκά. Ο Γιάννης πήγε στον οδοντίατρο)

2.5.2.1 Είδη αιτιοκρατικών σχέσεων

Το είδος της αιτιώδης αλυσίδας ανάμεσα στα γεγονότα καθορίζεται από τους παράγοντες που αναφέρθηκαν παραπάνω. Οι Smith & Hancox (2001) αναφέρουν ότι οι Trabasso, et al. υποστηρίζουν ότι υπάρχουν τέσσερα είδη αιτιοκρατικής σύνδεσης: φυσικά αίτια (φ) (physically causes), ψυχολογικά αίτια (ψ) (psychological causes), motivates (M) και enables (E). Το είδος που ανατίθεται σε κάθε σύνδεση εξαρτάται από το πώς οι περιορισμοί αλληλεπιδρούν με τα γεγονότα.

Παράδειγμα (Smith & Hancox, 2001) για τη δημιουργία αιτιοκρατικής σύνδεσης: A motivates B [M(A,B)] αν το A προηγείται χρονικά του B και το A είναι ενεργοποιημένο όταν το B συμβαίνει και το A είναι αναγκαίο για το B και το A περιέχει μια γενική πληροφορία.

Παράδειγμα (Smith & Hancox, 2001) : (A) Ο Βασίλης ήθελε ένα ποδήλατο. (B) Αυτός πήγε στο μαγαζί για να αγοράσει ποδήλατο. M(A,B)

Η αναπαράσταση μέσω δικτύων αναφέρεται συχνά ως εναλλακτική αιτιώδη αλυσίδα. Στην αναπαράσταση μέσω δικτύων είναι δυνατό ένας κόμβος-γεγονός να έχει πολλούς προγόνους και απογόνους όμως στις αιτιώδεις αλυσίδες δεν συμβαίνει αυτό (van den Broek, 1994) όπως αναφέρεται στο (Smith & Hancox, 2001). Κατά τους Smith και Hancox (2001) στο βιβλίο των Schank και Abelson υπάρχει ένα παράδειγμα αιτιοκρατικής αλυσίδας όπου μια μοναδική κατάσταση έχει πολλές συνέπειες: η αλλαγή της κατάστασης της καρέκλας προκαλεί τα εξής: κάποιος να πέσει από αυτήν, η μύρα να χυθεί και να δημιουργηθεί θόρυβος (Schank and Abelson, 1978) όπως αναφέρεται στο (Smith & Hancox, 2001). [4]

2.5.3 Hierarchical representations-story trees

Η ιεραρχική αναπαράσταση είναι ένας συνηθισμένος τρόπος αναπαράστασης όπως αναφέρουν οι ερευνητές (Smith & Hancox, 2001). Τα story grammars είναι τα πρώτα μοντέλα που υποστήριξαν την ιεραρχική δομή (Rumelhart, 1975) όπως αναφέρεται στο (Smith & Hancox, 2001). Στην ιεραρχική αναπαράσταση συναντούμε ζευγάρια, όπου υπάρχουν γραμμικοί σύνδεσμοι ανάμεσα στους κόμβους όπως συμβαίνει στην heterarchical αναπαράσταση. Σύμφωνα με τους Smith και Hancox (2001) η αναπαράσταση είναι τοπικά γραμμική (locally linear). Όμως το δίκτυο μπορεί να επεκταθεί και προς τα πάνω. Αυτό μπορεί να συμβεί με την προσθήκη κόμβων, οι οποίοι συνδέονται με την τους κόμβους που βρίσκονται στα χαμηλά επίπεδα. Με απλά λόγια είναι μια συλλογή από δίκτυα, σχηματίζοντας πυραμίδα. Τα υψηλότερα επίπεδα είναι πιο μικρά καθώς περιέχουν πιο λίγους κόμβους. Αυτοί οι κόμβοι γενικεύουν τις πληροφορίες που υπάρχουν στα χαμηλά επίπεδα (van Dijk, 1977) όπως αναφέρεται στο (Smith & Hancox, 2001). [4]

Η μακρο-δομή(macrostructure) είναι ένας δεύτερος ορισμός της ιεραρχικής δομής. Η μακρο-δομή περιγράφει τα «global levels descriptions» (δηλαδή τους κόμβους που έχουν καθολικούς συνδέσμους, και αυτοί οι κόμβοι μπορεί να βρίσκονται σε διαφορετικά επίπεδα) (van Dijk, 1977) όπως αναφέρεται στο (Smith & Hancox, 2001). Η μακρο-δομή (macrostructure) ορίζει «τα πιο σημαντικά γεγονότα τα οποία εξάγονται από μια ακολουθία προτάσεων» (van Dijk, 1977) όπως αναφέρεται στο (Smith & Hancox, 2001). (Δηλαδή αυτός ο κόμβος συνοψίζει τις πληροφορίες των κόμβων που ανήκουν σε αυτόν.) Οι Smith & Hancox (2001) αναφέρουν στο μοντέλο (Coreira,1980) η μακρο-δομή ταυτίζεται με τους κόμβους ενός story grammar tree. Ουσιαστικά η μακρο-δομή είναι ένας κόμβος σε ένα story tree του οποίου οι άμεσοι απόγονοι του περιέχουν δευτερεύουσες πληροφορίες (ή πληροφορίες με περισσότερες λεπτομέρειες) αυτού του κόμβου. Περιλαμβάνει τις πληροφορίες του κειμένου που αναφέρονται ρητά και τις σχέσεις μεταξύ αυτών των στοιχείων και όλα αυτά τα αναπαριστά ως μια γενική πληροφορία. [4]

2.6 .Συνοχή

Οι διάφοροι τρόποι αναπαράστασης εξαρτώνται από τα διαφορετικά είδη συνοχής που δημιουργούνται. Σύμφωνα με το κλάδο της ψυχολογίας η συνοχή μπορεί να είναι τοπική ή καθολική (Grasser et al., 1994 όπως αναφέρεται στο (Smith & Hancox, 2001) . Η τοπική

συνοχή εντοπίζεται στην heterarchical αναπαράσταση και η γενική συνοχή στην ιεραρχική αναπαράσταση. [4]

Αρχικά χαρακτηριζόταν ως «ιδιοκτησία» της ιστορίας επειδή διαδοχικές προτάσεις αναφέρονται στην ίδιες οντότητες (Hobbs, 1979) όπως αναφέρεται στο (Smith & Hancox, 2001). Στους Kintsch και van Dijk (1978) όπως αναφέρεται στο (Smith & Hancox, 2001) βρίσκουμε ένα παράδειγμα για αυτόν τον χαρακτηρισμό της συνοχής. Για να υπάρχει συνοχή ανάμεσα σε διαδοχικές προτάσεις ενός κειμένου θα πρέπει αυτές οι προτάσεις να προσδιορίζουν ή να αναφέρονται σε μια οντότητα, αυτή η οντότητα είναι ένα γεγονός ή ένας χαρακτήρας.

Παράδειγμα (Smith & Hancox, 2001) : John took a train to Paris. He arrived ten minutes late.

Οι δύο προτάσεις έχουν συνοχή μεταξύ τους επειδή αναφέρονται και οι δυο στη λέξη 'John'. Ο ορισμός είναι ανεπαρκής (Hobbs,1979) όπως αναφέρεται στο (Smith & Hancox, 2001).

John took a train from Paris to Istanbul. He likes spinach (Hobbs,1979) όπως αναφέρεται στο (Smith & Hancox, 2001) .

Είναι σαφές ότι οι περισσότεροι αναγνώστες θα θεωρήσουν ότι το πιο πάνω κείμενο δεν έχει συνοχή. Η συνοχή είναι σαν μια συνέχεια (on a continuum) (Hobbs,1979) όπως αναφέρεται στο (Smith & Hancox, 2001), οι ιστορίες στις οποίες συνδέσεις μεταξύ των στοιχείων της ιστορίας δύσκολα δημιουργούνται είναι λιγότερο συνεκτικές από ιστορίες στις οποίες είναι εύκολο να δημιουργηθούν συνδέσεις μεταξύ των στοιχείων της ιστορίας. Αυτό σημαίνει ότι για να έχει ένα κείμενο συνοχή οι σχέσεις που δημιουργούνται ανάμεσα στα συστατικά στοιχεία μιας ιστορίας, και παράγονται ως συμπέρασμα από τον αναγνώστη, θα πρέπει να είναι πολλές (πυκνές) και να επικαλύπτουν όλα (σχεδόν) τα γεγονότα της ιστορίας. (Επίσης αν έναν αναγνώστη είναι περισσότερο συγκεντρωμένος παράδειγμα έχει επίπεδο συγκέντρωσης A τότε η αναπαράσταση που θα δημιουργήσει θα έχει περισσότερη συνοχή από έναν αναγνώστη που έχει επίπεδο συγκέντρωσης B). [4]

2.6.1 Τοπική και καθολική συνοχή (local and global coherence)

Τα είδη συνοχής δημιουργούνται από την προϋπάρχουσα γνώση του αναγνώστη σε συνδυασμό με τους στόχους του αναγνώστη. [4]

2.6.1.1 Τοπική συνοχή (local coherence)

Η τοπική συνοχή εντοπίζεται στην γραμμική (linear) αλλά και στην heterarchical αναπαράσταση. «Αναφέρεται σε δομές και διαδικασίες που οργανώνουν στοιχεία, συστατικά γειτονικών ή μικρών προτάσεων» (Graesser et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001).

Για να δημιουργηθεί η τοπική συνοχή χρησιμοποιούνται οι δομές γνώσεις, η τοπική συνοχή χωρίζεται σε δύο κατηγορίες:

- Χρονικές σχέσεις - temporal relations
(Ένα στοιχείο μιας ιστορίας προηγείται και μετά ακολουθεί το επόμενο. Παράδειγμα ‘Το σπίτι καίγεται. (A) Ο Γιάννης έτρεξε έξω από το σπίτι’ (B). Για τα δυο γεγονότα A και B το γεγονός B μπορεί να ξεκινήσει πριν το τερματισμό του γεγονότος A.)
- Causal relations
(για παράδειγμα οι σχέσεις αυτές μπορεί να δημιουργούνται από φυσικά αίτια)

2.6.1.2 Καθολική συνοχή (global coherence)

Στην ιεραρχική αναπαράσταση εντοπίζουμε την καθολική συνοχή και δημιουργείται όταν οι περισσότερες ή όλες οι πληροφορίες του κειμένου συνδέονται σε ένα ή περισσότερα βασικά θέματα (Graessaer et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001). Η εύρεση αυτών των θεμάτων καθορίζει το επίπεδο συνοχής, αυτά τα θέματα βρίσκονται ψηλά στην ιεραρχική δομή. Αυτά τα θέματα συνδέονται με κόμβους, στους οποίους υπάρχει τοπική συνοχή ανάμεσα τους (Graessaer et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001). Ο αριθμός της τοπικής συνοχής καθορίζει τον αριθμό της καθολικής συνοχής. [4]

Τα σχήματα (schemas) είναι οι δομές που χρησιμοποιούνται για να δημιουργήσουν καθολική συνοχή (Mandler, 1984; Smith, 1997) όπως αναφέρεται στο (Smith & Hancox, 2001). Τα σχήματα χωρίζονται σε δύο κατηγορίες τα : τα world-based-schemas (είναι οι «στερεότυπες» γνώσεις, οι προϋπάρχουσες γνώσεις) και τα narrative-based-schemas (για παράδειγμα οι άξονες μιας ιστορίας: όπως ο τόπος που διαδραματίζονται τα γεγονότα, οι χαρακτήρες που πρωταγωνιστούν και ο χρόνος). [4]

Τα σχήματα έχουν τα εξής χαρακτηριστικά (Βακάλη, 1995). :

Τα σχήματα είναι πληροφορίες (σύνολα πληροφοριών), οι οποίες έχουν μια σταθερή βάση και υπάρχουν κάποια στοιχεία που δεν είναι σταθερά αλλά μεταβάλλονται. Για παράδειγμα (Βακάλη, 1995) το σχήμα «αγοράζω κάτι σε ένα κατάστημα»

Η σταθερή βάση είναι για το συγκεκριμένο παράδειγμα είναι: η ανταλλαγή αγαθών και χρημάτων. Τα στοιχεία που δεν είναι σταθερά είναι τα εξής: το είδος των αγαθών, το κόστος των αγαθών, τα καταστήματα, ο χρόνος αγοράς, οι πωλητές, οι καταναλωτές κ.ά. (Βακάλη, 1995). [1]

Όταν για τα στοιχεία που μεταβάλλονται δεν έχουν οριστεί συγκεκριμένες τιμές τότε είναι δυνατό να δώσουμε εμείς τιμές σε αυτές τι μεταβλητές, επιλέγοντας πάντα εκείνες την τιμές που θεωρούμε πιο κατάλληλες. Η τιμή που δίνουμε στη συγκεκριμένη μεταβλητή ονομάζεται «τιμή που δίνεται ερήμην» - εξ ορισμού (default value). Για παράδειγμα αν μας πει κάποιος ότι ξύπνησε νωρίς τότε υποθέτουμε ότι ξύπνησε μεταξύ επτά και οκτώ (Βακάλη, 1995). [1]

Τα σχήματα δηλαδή τα σύνολα πληροφοριών είναι δυνατό να περιέχουν και άλλα σχήματα (Βακάλη, 1995). Οι Rumelhart και Norman, όπως αναφέρονται στο (Βακάλη, 1995) δίνουν ως παράδειγμα το σχήμα της κεφαλής, στο οποίο υπάρχουν και άλλα σχήματα όπως του προσώπου, των ματιών, της μύτης, του στόματος, των αυτιών. Με τη σειρά του το σχήμα του στόματος περιέχει τα χείλη, τα δόντια (κ.ά.). [1]

Το σχήματα μπορούν να αποδώσουν διάφορες πληροφορίες από συγκεκριμένα αντικείμενα όπως καρέκλα, ψυγείο, υπολογιστής μέχρι ιδεολογίες και αφηρημένες έννοιες όπως δικαιοσύνη, ελευθερία. Τα σχήματα είναι δυνατό να περιέχουν πληροφορίες για συγκεκριμένες ομάδες ανθρώπων όπως φιλελεύθεροι, συντηρητικοί ή για συγκεκριμένα επαγγέλματα ή πρόσωπα όπως αστυνόμος, πυροσβέστης, ναυτικός. [1]

Τα σχήματα αναπαριστούν τις εμπειρίες και τις γνώσεις που έχουμε αποκτήσει από το περιβάλλον μας και τον πραγματικό κόσμο. Δεν θεωρούνται κανόνες ούτε ορισμοί. [1]

Όλοι οι άνθρωποι δεν έχουν τις ίδιες γνώσεις και εμπειρίες για το λόγο αυτό τα επιμέρους στοιχεία των σχημάτων μπορεί να είναι διαφορετικά. Επίσης μπορεί να περιλαμβάνουν διαφορετικά επιμέρους σχήματα και να δίνουν διαφορετικές τιμές στις μεταβλητές. [1]

Καθώς οι πληροφορίες εισέρχονται στον άνθρωπο μέσω των αισθήσεων τότε ενεργοποιείται το σχετικό σχήμα και μπορούμε να κατανοήσουμε και να ερμηνεύσουμε αυτές τις πληροφορίες. Τα σχήματα είναι ιδιαίτερα χρήσιμα στην κατανόηση του κόσμου. Όταν στην ιστορία δεν αναφέρονται ρητά κάποιες πληροφορίες τότε ενεργοποιείται το σχετικό σχήμα και είμαστε σε θέση να κατανοήσουμε αυτό το «κενό» της ιστορίας. [1]

Τα σχήματα είναι χρήσιμα για την κατανόηση του κόσμου αλλά βοηθούν στην πρόβλεψη, στην εξαγωγή συμπερασμάτων (Βακάλη, 1995).

2.6.2 Τα είδη συνοχής

Κάποιοι ερευνητές πιστεύουν ότι κατά τη διάρκεια της ανάγνωσης μόνο τοπική συνοχή δημιουργείται (McKoon and Ratcliff, 1992) όπως αναφέρεται στο (Smith & Hancox, 2001). Άλλοι πάλι ερευνητές αναφέρουν «ότι οι αναγνώστες προσπαθούν να κατασκευάσουν μια πιο γενική αναπαράσταση η οποία δημιουργείται με βάση το κείμενο και τις γνώσεις του αναγνώστη» (Graessaer et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001).

Ο Singer (Smith & Hancox, 2001) ταξινομεί τα μοντέλα συνοχής:

- Strong minimal hypothesis

Οι αναγνώστες δημιουργούν μόνο τοπική συνοχή. Συνδέουν πληροφορίες του κειμένου σε τοπικό επίπεδο και δεν γίνεται καμία προσπάθεια για να συνδέσουν αυτές τις πληροφορίες με άλλες μεγαλύτερες δομές. (Οι περισσότεροι ερευνητές δεν έχει αποδεχτεί πλήρως αυτήν την άποψη. Διάφοροι ερευνητές (όπως οι McKoon and Ratcliff, 1992) όπως αναφέρεται οι (Smith & Hancox, 2001) κλίνουν προς αυτήν την άποψη.) [4]

- Weak minimal hypothesis

Οι αναγνώστες συνδέουν τις πληροφορίες του κειμένου σε τοπικό επίπεδο. Είναι δυνατό να δημιουργήσουν συνδέσεις ανάμεσα σε τμήματα του κειμένου, τα οποία βρίσκονται σε διαφορετικές θέσεις του κειμένου (αν για παράδειγμα για ένα γεγονός που βρίσκεται στην αρχή του κειμένου ο αναγνώστης δεν μπορεί να βρει κάποιο γεγονός για να το συνδέσει σε τοπικό επίπεδο τότε ο αναγνώστης αναζητεί για γεγονότα που βρίσκονται πιο κάτω στο κείμενο). Αυτή η θεωρία υποστηρίζει ότι οι αναγνώστες δεν δημιουργούν καθολική συνοχή εκτός και αν οι στόχοι τους διευκολύνουν για αυτό το είδος συνοχής (McKoon and Ratcliff, 1992) όπως αναφέρεται στο (Smith & Hancox, 2001). [4]

- Constructionist hypothesis

Οι αναγνώστες δημιουργούν αναπαράσταση με καθολική συνοχή (Graessaer et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001). Οι αναγνώστες κατανοούν το κείμενο, δημιουργώντας συνδέσεις ανάμεσα σε τοπικά στοιχεία και σε στοιχεία που βρίσκονται σε διαφορετική θέση του κειμένου. (Εδώ παρατηρούμε ότι ο αναγνώστης δημιουργεί καθολική συνοχή επειδή ενδιαφέρεται να αναζητήσει συνδέσεις μεταξύ τοπικών και μη στοιχείων και όχι από ανάγκη όπως συμβαίνει στο Weak minimal hypothesis.) [4]

2.6.3 Πώς οι αναγνώστες δημιουργούν τη συνοχή

«Η έννοια της κατανόησης είναι το αποτέλεσμα μιας περίπλοκης διαδικασίας επίλυσης προβλημάτων στην οποία ο αναγνώστης δημιουργεί σχέσεις ανάμεσα σε ιδέες, γεγονότα και καταστάσεις του κειμένου» (van den Broeken, 1994) όπως αναφέρεται στο (Smith & Hancox, 2001). Κατά τη διάρκεια ανάγνωσης ενός κειμένου ο αναγνώστης προσπαθεί να βγάλει κάποια συμπεράσματα, αυτή η διαδικασία έχει ως αποτέλεσμα η αναπαράσταση που δημιουργεί ο αναγνώστης να έχει συνοχή. [4]

Τα συμπεράσματα που βγάζει ο αναγνώστης προκειμένου να κατανοήσει το κείμενο προκύπτουν καθώς συσχετίζει τις πληροφορίες του κειμένου. Για παράδειγμα (Smith & Hancox, 2001) ο αναγνώστης μπορεί να διαπιστώσει ότι δύο γεγονότα συνδέονται μεταξύ τους λόγω τοπικής χρονικής σχέσης, αυτή η χρονική σχέση δημιουργεί μια αιτιώδη σύνδεση σύμφωνα με τον κανόνα της χρονικής προτεραιότητας (temporal priority). Είναι δυνατό να εντοπιστούν σε ένα κείμενο διάφορες αιτιοκρατικές συνδέσεις μεταξύ γεγονότων, αυτό μπορεί να έχει ως επακόλουθο την προσθήκη ενός νέου 'γενικού' κόμβου στον οποίο συμπεριλαμβάνονται αυτά τα γεγονότα (δηλαδή δημιουργείται ένας νέος κόμβος-γεγονός στον οποίο συμπεριλαμβάνονται αυτά τα γεγονότα και αυτός κόμβος περιέχει έναν γενικό τίτλο για γεγονότα που περιλαμβάνει). [4]

2.7 Κατανόηση Ιστορίας και Συμπεράσματα

Συμπέρασμα είναι: «κάθε ισχυρισμός που πιστεύει ο αναγνώστης ότι είναι αληθής και είναι αποτέλεσμα της ανάγνωσης του κειμένου, αλλά πριν ο αναγνώστης δεν το πίστευε και δεν αναφέρεται αυτός ο ισχυρισμός ρητά στο κείμενο (Norving, 1980)» όπως αναφέρεται στο (Smith & Hancox, 2001).

Ωστόσο οι Smith & Hancox (2001) υποστηρίζουν ότι ο ορισμός αυτός είναι ανεπαρκής σε δύο σημεία. Σύμφωνα με τους Smith και Hancox (2001) θα πρέπει να γνωρίζουμε τι μορφή έχουν οι ισχυρισμοί. Μια απάντηση «είναι ότι ένα συμπέρασμα περιέχει επιχειρήματα και προτάσεις, οι οποίες δεν αναφέρονται ρητά στο κείμενο» (Singer et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001). Οι (Singer et al.,1994) όπως αναφέρεται στο (Smith & Hancox, 2001) δίνουν ένα παράδειγμα:

Julie soaked the bonfire. It went out.

Οι κόμβοι που αναπαριστούν τις ρητές πληροφορίες του κειμένου είναι οι ακόλουθοι (adapted from Singer et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001) :

(e1,douse,(agt(Julie),obj(bonfire)))

(e2,go-out,(exp(bonfire)))

(e1,douse,(agt(Julie),obj(bonfire), inst(water))))

(e2,go-out,(exp(bonfire)))

(I1,cause(e1,e2))

Ένα ακόμα στοιχείο προστέθηκε στο πρώτο γεγονός (water) και ένας σύνδεσμος (I1) ανάμεσα για τα δύο γεγονότα (Smith & Hancox, 2001). Ο αναγνώστης πιστεύει κάποιους ισχυρισμούς, όμως δεν θεωρούνται όλοι οι ισχυρισμοί συμπεράσματα (Smith & Hancox, 2001). Ο αναγνώστης βγάζει συμπεράσματα, αυτά μπορεί να διατηρούνται μόνιμα στην μνήμη (κωδικοποίηση-encoding) ή να χρησιμοποιούνται ως βοήθεια για την κατανόηση του κειμένου (activation). Τα συμπεράσματα χωρίζονται σε encoded και activated, αυτός ο διαχωρισμός ανάμεσα στα encoded και activated συμπεράσματα αναφέρονται στην επιστήμη της ψυχολογίας (van den Broek, 1994; Kintsch, 1988; Singer et al., 1994) όπως αναφέρεται στο (Smith & Hancox, 2001). Είναι δυνατό ο αναγνώστης να βγάζει ένα συμπέρασμα, το οποίο βασίζεται σε επαρκή στοιχεία αλλά στο τέλος επειδή δεν υποστηρίζεται από την ιστορία δεν μπορεί να ενσωματωθεί στην αναπαράσταση. [4]

Αυτός ο διαχωρισμός σχεδόν ποτέ δεν γίνεται από τα πρόσφατα μοντέλα για την κατανόηση ιστοριών (Smith & Hancox, 2001). Αυτά τα μοντέλα εξάγουν πολλές πιθανές επεξηγήσεις οι

οποίες μετά αξιολογούνται. Θεωρούνται *activated* επεξηγήσεις και όχι *encoded*, οι επεξηγήσεις που απορρίπτονται και δεν ενσωματώνονται στην αναπαράσταση, . [4]

2.7.1 Είδη συμπερασμάτων

Κατά τη διάρκεια της κατανόησης δημιουργούνται πολλά είδη συμπερασμάτων. Οι Graesser et al. (1994) όπως αναφέρεται στο (Smith & Hancox, 2001) διακρίνουν 13 διαφορετικά είδη, τέσσερα από αυτά σχετίζονται με την καθολική και τοπική συνοχή:

- Causal antecedent
Αυτός ο τύπος συμπεράσματος βρίσκει μια αιτία-πρόγονο (*causal antecedent*) για ένα γεγονός αυτό συχνά είναι ένα νέο γεγονός (προστίθεται ένα γεγονός που δεν υπήρχε πριν). Τα 'Causal antecedent συμπεράσματα' συντελούν στην τοπική συνοχή. [4]
- Superordinate goal
Αυτός ο τύπος συμπεράσματος καθιερώνει έναν σκοπό ο οποίος ενεργοποιεί μια δράση. Για παράδειγμα οι Smith & Hancox (2001) αναφέρουν την επόμενη ιστορία «'John was hungry. He went into the dinner.' Ένας αναγνώστης θα μπορούσε να συμπεράνει το εξής: John had the goal of eating'.» Τα «superordinate goal» συμπεράσματα συμβάλλουν και στα δύο είδη συνοχής. [4]
- Thematic
Τα 'thematic' συμπεράσματα δίνουν την κεντρική ιδέα ή ένα ηθικό δίδαγμα της ιστορίας. Για παράδειγμα (Smith & Hancox, 2001) όταν ο αναγνώστης διαβάζει τα παραμύθια του Αισώπου βγάζει ως συμπέρασμα το μήνυμα που θέλει να δώσει ο συγγραφέας του κειμένου. Αυτά τα συμπεράσματα βοηθούν στην καθολική συνοχή. [4]
- Character emotional reaction
Τα γεγονότα που αναφέρονται ρητά στο κείμενο δημιουργούν αυτά τα συμπεράσματα. Αυτά δίνουν τα συναισθήματα (τη συναισθηματική κατάσταση) του χαρακτήρα. Για παράδειγμα (Smith & Hancox, 2001) 'Mary's balloon burst. She cried and cried.' ένας αναγνώστης μπορεί να βγάλει το συμπέρασμα ότι η Mary είναι ανήσυχη. Αυτά τα συμπεράσματα συμβάλλουν στη καθολική συνοχή. [4]

Τα παραπάνω συμπεράσματα μπορούν να υποδιαιρεθούν στα backward συμπεράσματα τα οποία σχετίζονται με την εξέλιξη της αφήγησης (explanatory inferences, connecting) και στα forward (predictive inferences). [4]

2.7.2 Backward συμπεράσματα

Τα backward συμπεράσματα επίσης ονομάζονται επεξηγηματικά ή ενοποιητικά ('explanatory' ή 'bridging' in van den Broek, 1990a and Singer et al., 1994) όπως αναφέρεται στους (Smith & Hancox, 2001) εξηγούν 'γιατί συμβαίνει κάτι' (Trabasso and Magliano, 1996). Δίνουν απαντήσεις στις ερωτήσεις τύπου 'Γιατί'.

Τα backward συμπεράσματα είναι ικανά να δημιουργήσουν μια αιτιώδη σχέση ανάμεσα σε ένα γεγονός της ιστορίας και σε έναν νέο κόμβο, ο οποίος βρίσκεται πριν από το γεγονός της ιστορίας. Πολλές φορές χρειάζεται να δημιουργηθεί ένας ενοποιητικός κόμβος (bridging). Αυτός ο κόμβος είναι μια υπόθεση, η οποία προστίθεται στη βάση δεδομένων του συστήματος για να διατυπώσουμε μια πιθανή εξήγηση. [4]

Παράδειγμα backward συμπεράσμα από την ιστορία Ασθενοφόρο:

c(11) :: fell_down(Person,the_stairs) causes hurt(Person).

c(12) :: hurt(Person), person(Person), person(Other) causes phoned(Other,an_ambulance).

Το κατηγορημα causes hurt(Person) λειτουργεί ως ενοποιητικός κόμβος αλλά και ως επεξηγηματικός κόμβος (εξηγεί γιατί η Μαίρη τηλεφώνησε να έρθει ασθενοφόρο).

2.7.3 Forward συμπεράσματα

Τα forward συμπεράσματα δεν πρέπει να δημιουργούνται από τον αναγνώστη επειδή: (Smith & Hancox, 2001).

- Δεν χρειάζονται. Πολλές φορές τα forward συμπεράσματα αναφέρονται στο κείμενο.
- Είναι ανεπαρκή. Είναι πιθανό τα συμπεράσματα να είναι λανθασμένα.
- Δεν συμβάλλουν στην συνοχή του κειμένου.

Τα forward συμπεράσματα δημιουργούνται αλλά γρήγορα μετακινούνται από την εργαζόμενη μνήμη εκτός και αν επιβεβαιώνονται από την ιστορία (Keefe and McDaniel,

1993) όπως αναφέρεται στο (Smith & Hancox, 2001). Αν δεν μπορεί να δημιουργηθεί μια σύνδεση ανάμεσα σε δύο γεγονότα τα forward συμπεράσματα μπορεί να συμβάλλουν στην συνοχή .

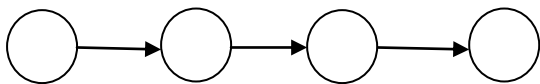
Για παράδειγμα (adapted from Murray et al., 1993) όπως αναφέρεται στο (Smith & Hancox, 2001) :

The angry waitress was totally fed up with her job. When a rude customer her, she lifted a plate of spaghetti above his head.

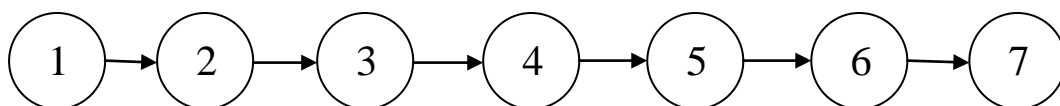
Σε αυτές τις δυο προτάσεις θα γίνει πιο κατανοητό το νόημα αν προστεθεί το εξής συμπέρασμα: ‘She poured the spaghetti over the customer’s head’(Smith & Hancox, 2001) .
[4]

Παρακάτω παρουσιάζονται οι γενικές (ενδεικτικές) μορφές των αναπαραστάσεων.

Causal chain



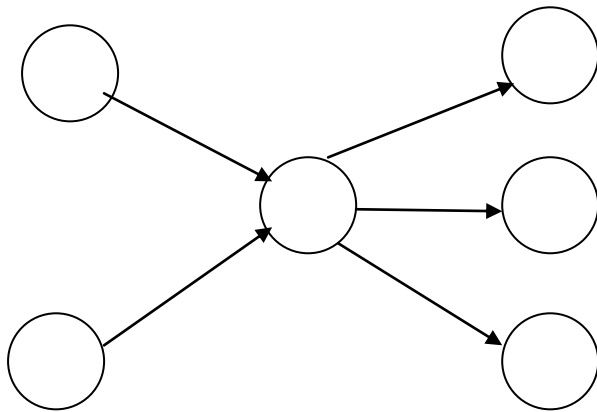
Παράδειγμα causal chain:



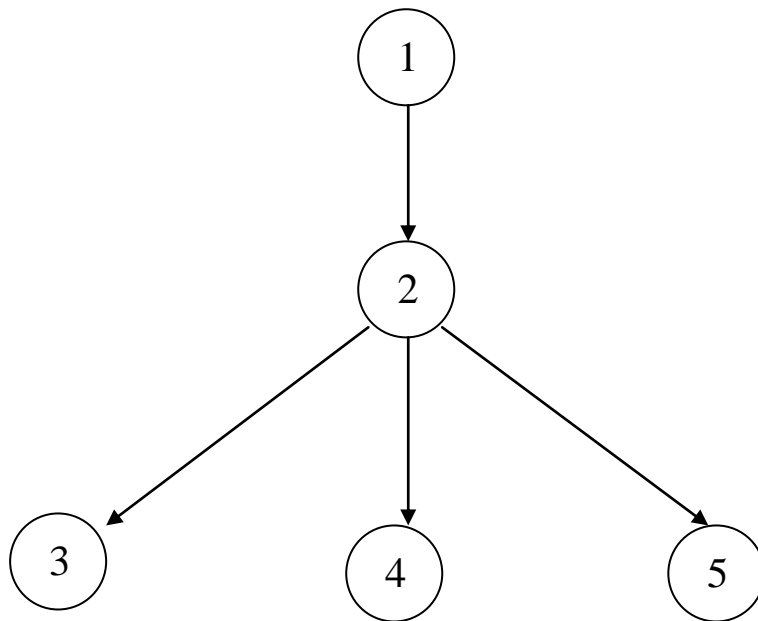
1. Αγοράζω πατάτες
2. Καθαρίζω τις πατάτες
3. Πλένω τις πατάτες

4. Κόβω τις πατάτες
5. Τηγανίζω τις πατάτες
6. Σερβίρω τις πατάτες
7. Τρώω τις πατάτες

Causal network

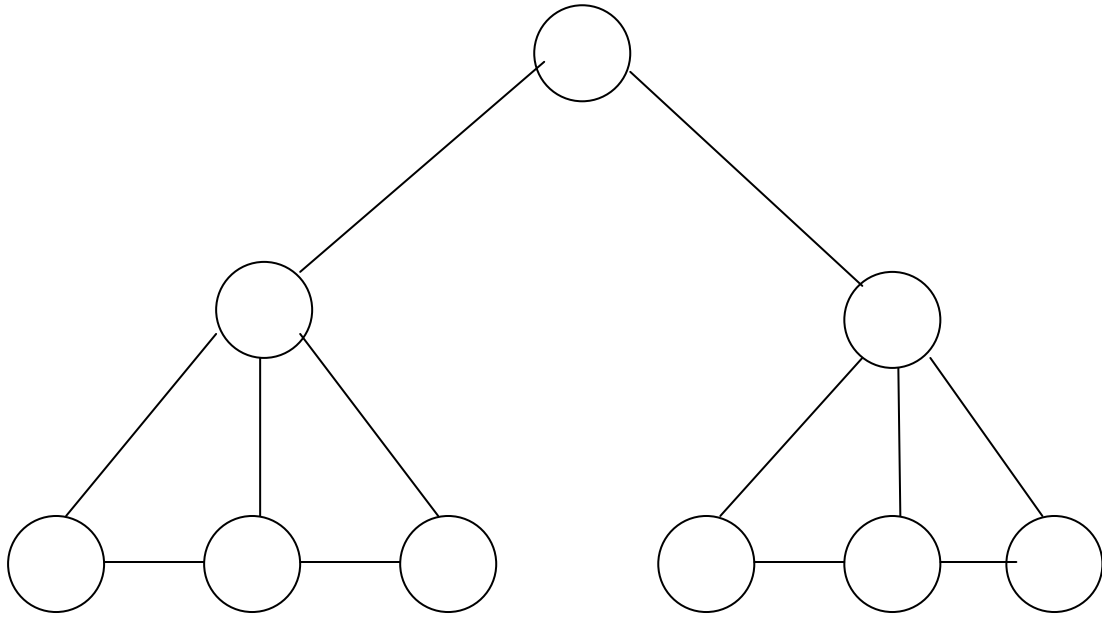


Παράδειγμα causal network:



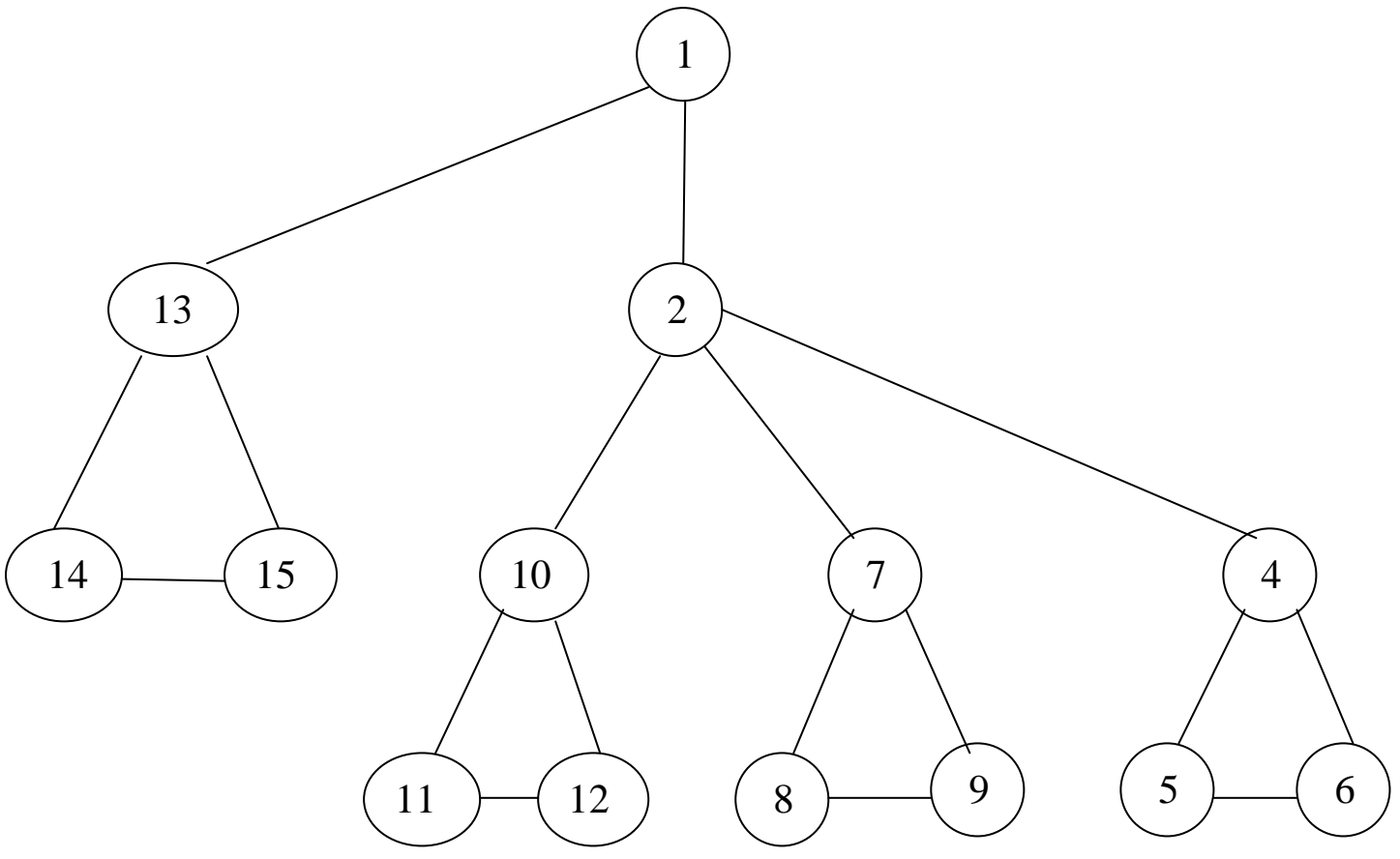
1. Βλάβη στον πάροχο ηλεκτρικού ρεύματος
2. Διακοπή του ρεύματος
3. Ο Γιώργος έπεσε από τις σκάλες
4. Η Μαρία κλείστηκε στο ασανσέρ
5. Διακόπηκε η διαδικτυακή συνομιλία μεταξύ της Χριστίνας και της Ειρήνης

Story trees



Παράδειγμα story trees από το κείμενο Carly's Family:

1. Carly has a large family.
2. She lives with four people.
3. Carly also has two pets.
4. Carly's mom is a doctor.
5. Carly's mom works at the hospital.
6. Carly's mom helps people who are sick.
7. Carly's dad works at home.
8. Carly's dad cooks for the family.
9. Carly's dad drives the kids to soccer practice.
10. Carly has two brothers.
11. James is ten years old.
12. Scott is fourteen years old.
13. Carly has two pets.
14. Jinx is a small, black cat.
15. Diego is a large, brown dog.
16. Carly loves her family!



Κεφάλαιο 3

Η λειτουργία του συστήματος Story Understanding Machine

3.1 Γενικά	29
3.2 Περιγραφή session	29
3.3 Περιγραφή Ιδιοτήτων	30
3.4 Περιγραφή Pre-session	31
3.5 Narrative Sessions	32
3.6 Ερωτήσεις	33
3.7 Κοινή Γνώση	34
3.8 Προτεραιότητες μεταξύ επιχειρημάτων-κανόνων	36
3.9 Εμφάνιση των απαντήσεων	37

3.1 Γενικά

Για την αναπαράσταση ιστοριών χρησιμοποιήθηκε το εργαλείο Story Understanding Machine. Η γλώσσα που εφαρμόζει το συγκεκριμένο σύστημα είναι η RAC. Έχει τη δυνατότητα να τρέξει σε πλατφόρμες που υποστηρίζουν τη SWI-Prolog όπως Windows, Linux, Unix και Mac. Το εργαλείο διαβάζει μια ιστορία, κατασκευάζει το μοντέλο κατανόησης του κειμένου και είναι σε θέση να απαντήσει σε ερωτήσεις. Πρέπει να δημιουργήσουμε το domain file, το οποίο είναι ένα SWI-Prolog αρχείο, για το οποίο θα κατασκευαστεί το μοντέλο κατανόησης από το STAR (Story comprehension through Argumentation) σύστημα.

3.2 Περιγραφή session

Τα sessions περιλαμβάνουν τις σκηνές, τις ερωτήσεις που πρέπει να απαντηθούν μέχρι τη συγκεκριμένη σκηνή και μια λίστα κατηγορημάτων, τα οποία χρησιμοποιούνται για να δείξουν πως το μοντέλο κατέληξε στην απάντηση του.

Το πρώτο μέρος περιέχει μια σειρά από sessions. Ένα session ορίζεται από μια σειρά δηλώσεων της μορφής: `session(s(Index),ListOfQuestions,ListOfVisibleConcepts)`.

Η πρώτη παράμετρος `s(Index)` δηλώνει τη σκηνή της αφήγησης και το `Index` είναι ένας φυσικός αριθμός.

Η δεύτερη παράμετρος (`ListOfQuestions`) είναι μια λίστα από ερωτήσεις της μορφής `q(Number)`, οι οποίες πρέπει να απαντηθούν από το σύστημα κατά τη διάρκεια του συγκεκριμένου session.

Η τρίτη παράμετρος (`ListOfVisibleConcepts`) είναι μια λίστα κατηγορημάτων που θέλουμε να εμφανίζονται στην οθόνη καθώς το σύστημα κατασκευάζει το μοντέλο κατανόησης. Το `ListOfVisibleConcepts` μπορεί να πάρει την τιμή «all» τότε όλα τα κατηγορήματα εμφανίζονται στο μοντέλο κατανόησης. Όταν η λίστα κατηγορημάτων (`ListOfVisibleConcepts`) είναι κενή δεν εμφανίζεται στην οθόνη πως το σύστημα κατέληξε στην απάντηση για την συγκεκριμένη ερώτηση.

Παραδείγματα session:

```
session(s(1),[q(1)],[has(_),is(_)]).
```

```
session(s(2),[q(2),q(3)],[]).
```

```
session(s(3),[q(4)],all).
```

3.3 Περιγραφή Ιδιοτήτων

Τα fluent είναι ιδιότητες που παραμένουν εξαιτίας της αδράνειας. Οι ιδιότητες των fluents που περιλαμβάνονται στη λίστα εξακολουθούν να ισχύουν μέσα στο χρόνο ώσπου να βρεθεί κάποια δράση που θα τις ανατρέψει και τελικά θα ακυρώσει τις ιδιότητες τους. Όταν ακυρωθεί η ιδιότητα ενός fluent τότε ακυρώνονται και όλες οι συνέπειες που προκύπτουν από αυτό. Αυτά δηλώνονται ως εξής: `flunets([a (_,_), b (_,_), c (_,_)])` με `a`, `b`, `c` τα ονόματα των κατηγορημάτων και μέσα στις παρενθέσεις δηλώνονται οι παράμετροι των κατηγορημάτων που παίρνουν ως είσοδο. Όσοι είναι οι παράμετροι τόσα είναι και τα σύμβολα ‘_’. Όλα τα υπόλοιπα κατηγορήματα αντιμετωπίζονται ως στιγμιαία (*instantaneously*) και δεν ισχύουν μέσα στο χρόνο.

Παράδειγμα (fluent) από την ιστορία door:

Το κατηγορήμα `sleep(Person)` στην ιστορία `door` έχει οριστεί ως `fluent` όμως η δράση `wake_up(Person)` προκαλεί τον τερματισμό του κατηγορήματος `sleep(Person)`.

`c(20) :: wake_up(Person) causes -sleep(Person)`.

Παράδειγμα: `([has (_,_), is(_), works(_,_)])`.

Παραδείγματα fluents από τις ιστορίες `door` και `Carly's Family` αντίστοιχα:

```
fluents([
  tired(_),
  sleep(_),
  afraid(_),
  it_is(_),
  wants_to_know(_,_),
  watch(_,_),
  is(_,_),
  thinks_close_door(_),
  blow(_),
]).
```

```
fluents([
  has(_,_),
  lives(_,_),
  is(_,_),
  works(_,_),
  helps(_,_),
  drives(_,_),
  loves(_,_),
  people_carlys_family_are(_),
]).
```

3.4 Περιγραφή Pre-session

Συνήθως αρχίζουμε με ένα `session` που έχει την ακόλουθη μορφή: `session(s(0),[],all)`

Αυτό είναι ένα pre-session που στη σκηνή με τον αριθμό μηδέν (0) δίνονται κάποιες πληροφορίες για την ιστορία και δεν αλλάζουν με το πέρασμα του χρόνου.

Παράδειγμα pre-session από την ιστορία door:

Session 0 και Story Scene 0:

session(s(0),[],all).

s(0) :: person(mary) at always.

Άλλο παράδειγμα pre-session από την ιστορία Carly's:

s(0):: person(carly) at always.

s(0):: person(carlys_mom) at always.

s(0):: person(carlys_dad) at always.

s(0):: person(james) at always.

s(0):: person(scott) at always.

Οι δηλώσεις ισχύουν κάθε στιγμή αλλά δεν μπορούν να χαρακτηριστούν σαν flunets, χαρακτηρίζονται ως σταθερές και είναι διαθέσιμες σε κάθε σημείο της ιστορίας.

3.5 Narrative Sessions

Διαχωρίζουμε την ιστορία σε έναν αριθμό από sessions όπως εμείς επιθυμούμε. Είναι δυνατό να δημιουργήσουμε μόνο ένα session, σε αυτήν την περίπτωση θα πρέπει να δώσουμε ολόκληρη την ιστορία για να γίνει η ανάγνωση και η κατανόηση από το σύστημα ώστε να απαντήσει τις ερωτήσεις. Εναλλακτικά θα μπορούσαμε να δημιουργήσουμε ένα session για κάθε πρόταση της ιστορίας όπου το σύστημα διαβάζει την ιστορία πρόταση-πρόταση (δηλαδή session-session) και απαντάει τις ερωτήσεις για κάθε session ξεχωριστά αφού δημιουργήσει το μοντέλο κατανόησης.

Παραδείγματα narrative sessions από την ιστορία door:

s(1) :: tired(mary) at 1.

s(2) :: in_bedroom(mary) at 2.

s(2) :: sleep(mary) at 2.
s(3) :: suddenly_close(door) at 4.
s(3) :: wake_up(mary) at 5.
s(3) :: afraid(mary) at 6.
s(4) :: get_up(mary,bed) at 7.
s(4) :: walk_to(mary,kitchen) at 8.
s(4) :: see(mary,open_window) at 9.
s(4) :: blow(wind) at 9.

Οι ιδιότητες/δράσεις τοποθετούνται η μια κάτω από την άλλη με χρονολογική σειρά όπως πραγματοποιούνται στην ιστορία. Όπως παρατηρούμε έχουν την εξής μορφή: s(Index) :: name_concept(x,y) at time. Index: δηλώνει τον αριθμό της σκηνής. Το name_concept είναι το όνομα της ιδιότητας/δράσης. Τα x και y είναι οι παράμετροι που παίρνει το κατηγορημα και το time ένας φυσικός αριθμός (0,1,2,3...) που δηλώνει την χρονική στιγμή που έγινε η ιδιότητα/δράση.

3.6 Ερωτήσεις

Οι ερωτήσεις ορίζονται με την ακόλουθη μορφή στο domain file:

```
q(Index)?? name(x,y) at time;  
           name(x,z) at time.
```

Το Index είναι ένας φυσικός αριθμός (0,1,2,3) και αντιπροσωπεύει τον αριθμό της ερώτησης. Το name είναι το όνομα της ιδιότητας/δράσης για το οποίο θέλουμε να ελέγξουμε αν αληθεύει τη δεδομένη χρονική στιγμή και τα x, y και z είναι οι παράμετροι της ιδιότητας δράσης. Οι ερωτήσεις μπορεί να είναι ερωτήσεις σωστού λάθους ή πολλαπλής επιλογής. Στις ερωτήσεις πολλαπλής επιλογής σε κάθε πιθανή απάντηση βάζουμε το σύμβολο ‘;’ και στην τελευταία πιθανή απάντηση της ίδιας ερώτησης τοποθετούμε το σύμβολο ‘.’ (τελεία) για να δείξουμε το τέλος της ερώτησης. Η ίδια ερώτηση μπορεί να γίνει σε διαφορετικά session. Αυτό συνήθως γίνεται για να ελέγξουμε αν το μοντέλο κατανόησης έχει αναθεωρήσει. Για κάθε επιλογή της ερώτησης το σύστημα επιστρέφει τα ακόλουθα:

- “accepted” σημαίνει ότι το σύστημα δέχεται αυτήν την απάντηση, αυτή η επιλογή ισχύει στο μοντέλο κατανόησης

- “rejected” σημαίνει ότι σύστημα απορρίπτει αυτήν την επιλογή, αυτή η επιλογή δεν ισχύει στο μοντέλο κατανόησης
- “possible” σημαίνει ότι ούτε δέχεται ούτε απορρίπτει αυτήν την επιλογή

Παραδείγματα ερωτήσεων από την ιστορία door και Carly’s Family αντίστοιχα:

q(1) ?? work(mary) at 1;
 go_to_vacation(mary)at 1.

q(2) ?? it_is(night) at 2;
 it_is(morning) at 2.

q(1) ?? people_carlys_family_are(four) at 0;
 people_carlys_family_are(five) at 0;
 people_carlys_family_are(six) at 0.

q(2) ?? works(carlys_mom,at_the_restaurant) at 1;
 works(carlys_mom,at_the_mall) at 1;
 works(carlys_mom,at_the_hospital) at 1.

3.7 Κοινή Γνώση

Η κοινή γνώση χρησιμοποιείται για την κατανόηση της ιστορίας. Οι σχέσεις ανάμεσα στα κατηγορήματα αποτελούν την κοινή γνώση. Υπάρχουν τρία είδη σχέσεων: property, causal, preclusion. Τα επιχειρήματα της κοινής γνώσης ορίζονται το ένα κάτω από το άλλο και έχουν την εξής μορφή:

- p(number) :: name(x,y) implies name_b(z), αν το επιχείρημα είναι property
- c(number) :: name(x,y) causal name_b(z), αν το επιχείρημα είναι causal

Το number είναι πάντα ένας αριθμός που δίνουμε στο επιχείρημα. Κάθε επιχείρημα έχει το δικό του μοναδικό αριθμό για να είναι συνεπής η γνώση. Το name είναι το όνομα της ιδιότητας/δράσης δηλαδή η συνθήκη του επιχειρήματος για να συμπεράνουμε ότι ισχύει η ιδιότητα name_b. Τα x, y, z είναι οι παράμετροι των κατηγορημάτων. Το επιχείρημα πρέπει πάντα να τελειώνει με το σύμβολο της τελείας ‘.’. Στην περίπτωση που το επιχείρημα είναι property ισχύει το εξής: αν η προ-συνθήκη συμβαίνει στο χρόνο t τότε και το κατηγορήμα

name_b συμβαίνει στον ίδιο χρόνο t. Ενώ αν το επιχείρημα είναι causal ισχύει το εξής: αν η προ-συνθήκη, name, συμβαίνει στο χρόνο t τότε το κατηγορημα name_b συμβαίνει στο χρόνο t+1. Η διαφορά σε αυτά τα δυο είδη κανόνων είναι ότι οι property κανόνες δείχνουν τη σχέση ανάμεσα στις ιδιότητες ενώ οι casual κανόνες προκαλούν μια ιδιότητα.

Παραδείγματα κοινής γνώσης από την ιστορία door

p(11) :: tired(Person) implies work(Person).

p(12) :: go_to_vacation(Person) implies -tired(Person).

p(13) :: work(Person) implies -go_to_vacation(Person).

p(14) :: go_to_vacation(Person) implies -work(Person).

Παρατηρούμε ότι όταν μια ιδιότητα συσχετίζεται με κάποια άλλη ιδιότητα τότε είναι δυνατό να ορίζει το ξεκίνημα ή τον τερματισμό της ιδιότητας.

Παράδειγματα:

c(11) :: wants(Person,ice_cream) causes buy(Person,ice_cream).

c(12) :: buy(Person,ice_cream) causes eat(Person,ice_cream).

c(20) :: wake_up(Person) causes -sleep(Person).

p(33) :: get_up(Person,bed), afraid(Person) implies wants_to_know(Person,what_happened).

Παρατηρούμε ότι μπορούν να δημιουργηθούν νέες ιδιότητες στην ιστορία όπως οι ιδιότητες wants_to_know(Person,what_happened) και watch(Person,tv). Σε αυτήν την περίπτωση πρέπει να ελέγξουμε αν οι ιδιότητες παραμένουν σταθερές (μέχρι να βρεθεί κάτι να τις ανατρέψει) καθώς εξελίσσεται η ιστορία για να τις κατατάξουμε στα fluents.

Παραδείγματα από την ιστορία door:

p(33) :: get_up(Person,bed), afraid(Person) implies wants_to_know(Person,what_happened).

p(34) :: walk_to(Person) implies -watch(Person,tv).

p(35) :: wants_to_know(Person,what_happened) implies -watch(Person,tv).

p(36) :: watch(Person,tv) implies -wants_to_know(Person,what_happened).

p(33)>>p(36).

Για προγραμματιστικούς λόγους οι μεταβλητές σε έναν κανόνα πρέπει να εμφανίζονται δύο ή περισσότερες φορές.

Παραδείγματα από την ιστορία Carly's Family και door:

p(31) :: is(X,small),is(X,black), is(X,cat) implies is(X,pet).

p(32) :: is(X,pet) implies is_part_of_family(X).

p(33) :: is(Y,large),is(diego,brown),is(Y,dog) implies is(Y,pet).

p(34) :: is(Y,pet) implies is_part_of_family(Y).

p(35) :: is_part_of_family(X), is_part_of_family(Y) implies are(X,Y,part_of_carlys_family).

p(36) :: are(X,Y,part_of_carlys_family), person(Person), is(X,dog), is(Y, dog) implies - is(Person,the_best_soccer_player_in_the_family).

%p(37) :: is(Person,the_best_soccer_player_in_the_family) implies - are(X,Y,part_of_carlys_family).

p(37) :: is(Person,doctor) implies -is(Person,coaches_the_soccer_team).

c(11) :: wanted(Person,ice_cream) causes buy(Person,ice_cream).

c(12) :: buy(Person,ice_cream) causes enjoyed_eating(Person,ice_cream).

c(13) :: wanted(Person,ice_cream) causes make(Person,ice_cream).

3.8 Προτεραιότητες μεταξύ επιχειρημάτων-κανόνων

Στην κοινή γνώση είναι δυνατό να υπάρχουν προτεραιότητες μεταξύ των επιχειρημάτων. Οι προτεραιότητες τοποθετούνται με βάση το περιεχόμενο της κάθε ιστορίας και δεν είναι σταθερές αλλά αλλάζουν σύμφωνα με τις συνθήκες της ιστορίας. Γενικά τα causal επιχειρήματα έχουν μεγαλύτερη προτεραιότητα από την αδράνεια και η αδράνεια έχει μεγαλύτερη προτεραιότητα από τα property επιχειρήματα. Οι προτεραιότητες δηλώνονται ως εξής: $a(N1) \gg b(N2)$ όπου a και b είναι οι τύποι των επιχειρημάτων και N1 και N2 είναι ο αριθμός των επιχειρημάτων. Το σύμβολο της προτεραιότητας δηλώνει ότι το πρώτο επιχείρημα είναι πιο δυνατό από το δεύτερο επιχείρημα.

Η προτεραιότητα πρέπει πάντα να τελειώνει με το σύμβολο της τελείας ‘.’ διαφορετικά εμφανίζεται στην οθόνη συντακτικό λάθος. Οι προτεραιότητες εφαρμόζονται στην κεφαλή (head) ενός επιχειρήματος.

Παραδείγματα:

c(20) :: wake_up(Person) causes -sleep(Person).

c(21) :: suddenly_close(door), wake_up(Person) causes afraid(Person).

p(22) :: suddenly_close(door), afraid(Person), person(Person) implies
thinks_close_door(thief).

p(24) :: thinks_close_door(wind) implies -thinks_close_door(thief).

p(25) :: thinks_close_door(thief) implies -thinks_close_door(wind).

p(24)>>p(22).

3.9 Εμφάνιση των απαντήσεων

Για κάθε session υπολογίζονται και οι αντίστοιχες ερωτήσεις. Για να ξεκινήσει η εκτέλεση του συστήματος πληκτρολογούμε “start”. Το σύστημα ρωτάει αν θέλουμε να συνεχίσουμε στην επόμενη σκηνή. Πληκτρολογώντας ‘y’ δηλώνουμε ότι θέλουμε να συνεχίσουμε ενώ πληκτρολογώντας ‘n’ δηλώνουμε ότι θέλουμε να σταματήσουμε. Στην περίπτωση που δώσουμε την τελευταία επιλογή δεν έχουμε δυνατότητα να συνεχίσουμε από εκεί που σταματήσαμε αλλά ξεκινάμε από την αρχή.

Παραδείγματα απαντήσεων:

```

ERROR: c:/users/user/desktop/star system 1.0.018/star system 1.0.018/engine.pl:14:
       source_sink `domains/doorbell' does not exist
Warning: c:/users/user/desktop/star system 1.0.018/star system 1.0.018/engine.pl:14:
         Goal (directive) failed: user: ([config],domain(_G1327), (multifile (>>)/2),[_
% c:/Users/user/Desktop/STAR System 1.0.018/STAR System 1.0.018/engine.pl compiled 0.0
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 6.6.6)
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

```

For help, use ?- help(Topic). or ?- apropos(Word).

```

1 ?-
% c:/Users/user/Desktop/STAR System 1.0.018/STAR System 1.0.018/domains/door 1.pl comp
1 ?- start.

```

```

=====
>>> Reading story up to scene s(0)
=====

```

```

>>> Universal argument...
>>> Acceptable argument...

```

```

>>> Comprehension model:

```

```

s(0) : person(mary) at always
      0: < person(mary)>

```

Continue (y/n)? y.

```

=====
>>> Reading story up to scene s(1)
=====

```

```

>>> Universal argument...
>>> Acceptable argument...
p(13) : pro(go_to_vacation(mary)#true,[work(mary)=true]) @ 0 -f-> (go_to_vacation(mary)
p(14) : pro(work(mary)#true,[go_to_vacation(mary)=true]) @ 0 -b-> (go_to_vacation(mary)
p(11) : pro(work(mary)=true,[tired(mary)=true]) @ 0 -f-> (work(mary)=true,0,p(11))
p(12) : pro(tired(mary)#true,[go_to_vacation(mary)=true]) @ 0 -b-> (go_to_vacation(mar
p(13) : pro(go_to_vacation(mary)#true,[work(mary)=true]) @ 1 -f-> (go_to_vacation(mary)
p(14) : pro(work(mary)#true,[go_to_vacation(mary)=true]) @ 1 -b-> (go_to_vacation(mary)
i(0)  : per(tired(mary)=true,[tired(mary)=true]) @ 1 -f-> (tired(mary)=true,1,per(o(0))
p(11) : pro(work(mary)=true,[tired(mary)=true]) @ 1 -f-> (work(mary)=true,1,p(11))
i(0)  : per(tired(mary)#true,[tired(mary)#true]) @ 1 -b-> (tired(mary)=true,0,per(o(0))

```

```

>>> Comprehension model:

```

```

s(0) : person(mary) at always
s(1) : tired(mary) at 1

```

```

      0: -go_to_vacation(mary) < person(mary)> tired(mary) work(mary)
      1: -go_to_vacation(mary) < person(mary)> < tired(mary)> work(mary)

```

```

>>> Answering question q(1):
+ accepted choice: ,[work(mary) at 1]
- rejected choice: ,[go_to_vacation(mary) at 1]

```

Continue (y/n)?

Εικόνα 2

```

i(0) : per(it_is(morning)#true,[it_is(morning)#true]) @ 1 -f-> (it_is(morning)#true,1,
i(0) : per(it_is(morning)#true,[it_is(morning)#true]) @ 1 -f-> (it_is(morning)#true,1,
p(16) : pro(it_is(morning)#true,[it_is(night)=true]) @ 0 -f-> (it_is(morning)#true,0,p
p(17) : pro(it_is(night)#true,[it_is(morning)=true]) @ 0 -b-> (it_is(morning)#true,0,p
i(0) : per(it_is(night)=true,[it_is(night)=true]) @ 1 -f-> (it_is(night)=true,1,per(p
i(0) : per(it_is(morning)=true,[it_is(morning)=true]) @ 1 -b-> (it_is(morning)#true,0,
i(0) : per(it_is(morning)=true,[it_is(morning)=true]) @ 1 -b-> (it_is(morning)#true,0,
i(0) : per(it_is(morning)#true,[it_is(morning)#true]) @ 2 -f-> (it_is(morning)#true,2,
i(0) : per(it_is(morning)#true,[it_is(morning)#true]) @ 2 -f-> (it_is(morning)#true,2,
p(13) : pro(go_to_vacation(mary)#true,[work(mary)=true]) @ 0 -f-> (go_to_vacation(mary
p(13) : pro(go_to_vacation(mary)#true,[work(mary)=true]) @ 2 -f-> (go_to_vacation(mary
p(14) : pro(work(mary)#true,[go_to_vacation(mary)=true]) @ 0 -b-> (go_to_vacation(mary
p(14) : pro(work(mary)#true,[go_to_vacation(mary)=true]) @ 2 -b-> (go_to_vacation(mary
p(16) : pro(it_is(morning)#true,[it_is(night)=true]) @ 1 -f-> (it_is(morning)#true,1,p
p(17) : pro(it_is(night)#true,[it_is(morning)=true]) @ 1 -b-> (it_is(morning)#true,1,p
i(0) : per(sleep(mary)=true,[sleep(mary)=true]) @ 1 -f-> (sleep(mary)=true,1,per(o(0))
i(0) : per(it_is(night)=true,[it_is(night)=true]) @ 2 -f-> (it_is(night)=true,2,per(p(
i(0) : per(it_is(morning)=true,[it_is(morning)=true]) @ 2 -b-> (it_is(morning)#true,1,
i(0) : per(it_is(morning)=true,[it_is(morning)=true]) @ 2 -b-> (it_is(morning)#true,1,
i(0) : per(it_is(night)#true,[it_is(night)#true]) @ 1 -b-> (it_is(night)=true,0,per(p(
c(20) : cau(sleep(mary)#true,[wake_up(mary)=true]) @ 1 -b-> (wake_up(mary)#true,0,c(20
p(11) : pro(work(mary)=true,[tired(mary)=true]) @ 0 -f-> (work(mary)=true,0,p(11))
p(11) : pro(work(mary)=true,[tired(mary)=true]) @ 2 -f-> (work(mary)=true,2,p(11))
p(12) : pro(tired(mary)#true,[go_to_vacation(mary)=true]) @ 0 -b-> (go_to_vacation(mar
p(13) : pro(go_to_vacation(mary)#true,[work(mary)=true]) @ 1 -f-> (go_to_vacation(mary
p(14) : pro(work(mary)#true,[go_to_vacation(mary)=true]) @ 1 -b-> (go_to_vacation(mary
p(16) : pro(it_is(morning)#true,[it_is(night)=true]) @ 2 -f-> (it_is(morning)#true,2,p
p(17) : pro(it_is(night)#true,[it_is(morning)=true]) @ 2 -b-> (it_is(morning)#true,2,p
i(0) : per(tired(mary)=true,[tired(mary)=true]) @ 1 -f-> (tired(mary)=true,1,per(o(0))
i(0) : per(tired(mary)#true,[tired(mary)#true]) @ 2 -b-> (tired(mary)=true,1,per(o(0))
i(0) : per(sleep(mary)=true,[sleep(mary)=true]) @ 2 -f-> (sleep(mary)=true,2,per(o(0))
i(0) : per(sleep(mary)#true,[sleep(mary)#true]) @ 1 -b-> (sleep(mary)=true,0,per(o(0))
i(0) : per(it_is(night)#true,[it_is(night)#true]) @ 2 -b-> (it_is(night)=true,1,per(p(
c(20) : cau(sleep(mary)#true,[wake_up(mary)=true]) @ 2 -b-> (wake_up(mary)#true,1,c(20
p(11) : pro(work(mary)=true,[tired(mary)=true]) @ 1 -f-> (work(mary)=true,1,p(11))
p(15) : pro(it_is(night)=true,[in_bedroom(mary)=true,sleep(mary)=true]) @ 2 -f-> (it_i
i(0) : per(tired(mary)=true,[tired(mary)=true]) @ 2 -f-> (tired(mary)=true,2,per(o(0))
i(0) : per(tired(mary)#true,[tired(mary)#true]) @ 1 -b-> (tired(mary)=true,0,per(o(0))
i(0) : per(sleep(mary)#true,[sleep(mary)#true]) @ 2 -b-> (sleep(mary)=true,1,per(o(0))

```

>>> Comprehension model:

```

s(0) : person(mary) at always
s(1) : tired(mary) at 1
s(2) : in_bedroom(mary) at 2
s(2) : sleep(mary) at 2

```

```

0:   -go_to_vacation(mary)                                -it_is(morning)      it_is(night)
work(mary)
1:   -go_to_vacation(mary)                                -it_is(morning)      it_is(night)
work(mary)
2:   -go_to_vacation(mary) < in_bedroom(mary) >         -it_is(morning)      it_is(night)
work(mary)

```

```

>>> Answering question q(2):
+ accepted choice: ,[it_is(night)at 2]
- rejected choice: ,[it_is(morning)at 2]

```

Continue (y/n)?

Εικόνα 3

```

c(20) : cau(sleep(mary)#true, [wake_up(mary)=true]) @ 3 -b-> (wake_up(mary)#true,2,c(20)
p(11) : pro(work(mary)=true, [tired(mary)=true]) @ 0 -f-> (work(mary)=true,0,p(11))
p(11) : pro(work(mary)=true, [tired(mary)=true]) @ 2 -f-> (work(mary)=true,2,p(11))
p(12) : pro(tired(mary)#true, [go_to_vacation(mary)=true]) @ 0 -b-> (go_to_vacation(mar
p(13) : pro(go_to_vacation(mary)#true, [work(mary)=true]) @ 1 -f-> (go_to_vacation(mary)
p(14) : pro(work(mary)#true, [go_to_vacation(mary)=true]) @ 1 -b-> (go_to_vacation(mary)
p(16) : pro(it_is(morning)#true, [it_is(night)=true]) @ 2 -f-> (it_is(morning)#true,2,p
p(17) : pro(it_is(night)#true, [it_is(morning)=true]) @ 2 -b-> (it_is(morning)#true,2,p
i(0) : per(tired(mary)=true, [tired(mary)=true]) @ 1 -f-> (tired(mary)=true,1,per(o(0))
i(0) : per(tired(mary)=true, [tired(mary)=true]) @ 3 -f-> (tired(mary)=true,3,per(o(0))
i(0) : per(tired(mary)#true, [tired(mary)#true]) @ 2 -b-> (tired(mary)=true,1,per(o(0))
i(0) : per(sleep(mary)=true, [sleep(mary)=true]) @ 2 -f-> (sleep(mary)=true,2,per(o(0))
i(0) : per(sleep(mary)=true, [sleep(mary)=true]) @ 4 -f-> (sleep(mary)=true,4,per(o(0))
i(0) : per(sleep(mary)#true, [sleep(mary)#true]) @ 1 -b-> (sleep(mary)=true,0,per(o(0))
i(0) : per(sleep(mary)#true, [sleep(mary)#true]) @ 3 -b-> (sleep(mary)=true,2,per(o(0))
i(0) : per(afraid(mary)=true, [afraid(mary)=true]) @ 6 -f-> (afraid(mary)=true,6,per(o(
i(0) : per(afraid(mary)#true, [afraid(mary)#true]) @ 5 -b-> (afraid(mary)=true,4,per(o(
i(0) : per(it_is(night)=true, [it_is(night)=true]) @ 3 -f-> (it_is(night)=true,3,per(p
i(0) : per(it_is(night)#true, [it_is(night)#true]) @ 2 -b-> (it_is(night)=true,1,per(p
c(20) : cau(sleep(mary)#true, [wake_up(mary)=true]) @ 2 -b-> (wake_up(mary)#true,1,c(20)
c(20) : cau(sleep(mary)#true, [wake_up(mary)=true]) @ 6 -f-> (sleep(mary)#true,6,c(20))
p(11) : pro(work(mary)=true, [tired(mary)=true]) @ 1 -f-> (work(mary)=true,1,p(11))
p(15) : pro(it_is(night)=true, [in_bedroom(mary)=true, sleep(mary)=true]) @ 2 -f-> (it_i
i(0) : per(tired(mary)=true, [tired(mary)=true]) @ 2 -f-> (tired(mary)=true,2,per(o(0))
i(0) : per(tired(mary)#true, [tired(mary)#true]) @ 1 -b-> (tired(mary)=true,0,per(o(0))
i(0) : per(sleep(mary)=true, [sleep(mary)=true]) @ 3 -f-> (sleep(mary)=true,3,per(o(0))
i(0) : per(sleep(mary)#true, [sleep(mary)#true]) @ 2 -b-> (sleep(mary)=true,1,per(o(0))
i(0) : per(afraid(mary)#true, [afraid(mary)#true]) @ 6 -b-> (afraid(mary)=true,5,per(o(

```

>>> Comprehension model:

```

s(0) : person(mary) at always
s(1) : tired(mary) at 1
s(2) : in_bedroom(mary) at 2
s(2) : sleep(mary) at 2
s(3) : suddenly_close(door) at 4
s(3) : wake_up(mary) at 5
s(3) : afraid(mary) at 6

```

```

0:   afraid(mary)      -go_to_vacation(mary)                -it_is(morning)
    thinks_close_door(thief)  -thinks_close_door(wind)      tired(mary)      -wake_
1:   afraid(mary)      -go_to_vacation(mary)                -it_is(morning)
    thinks_close_door(thief)  -thinks_close_door(wind)      < tired(mary) >  -wake_
2:   afraid(mary)      -go_to_vacation(mary)      < in_bedroom(mary) >  -it_is(morning)
    thinks_close_door(thief)  -thinks_close_door(wind)      tired(mary)      -wake_
3:   afraid(mary)      -go_to_vacation(mary)                -it_is(morning)
    thinks_close_door(thief)  -thinks_close_door(wind)      tired(mary)      -wake_
4:   afraid(mary)      -go_to_vacation(mary)                -it_is(morning)
oor> thinks_close_door(thief)  -thinks_close_door(wind)      tired(mary)      -wake_
5:   afraid(mary)      -go_to_vacation(mary)                -it_is(morning)
    thinks_close_door(thief)  -thinks_close_door(wind)      tired(mary)      < wake_
6:   < afraid(mary) >  -go_to_vacation(mary)                -it_is(morning)
    thinks_close_door(thief)  -thinks_close_door(wind)      tired(mary)

```

>>> Answering question q(3):

```

+ accepted choice: , [thinks_close_door(thief) at 3]
- rejected choice: , [thinks_close_door(wind) at 3]

```

Continue (y/n)?

Εικόνα 4

```

i(0) : per(sleep(mary)#true,[sleep(mary)#true]) @ 2 -b-> (sleep(mary)=true,1,per(o(0)))
i(0) : per(afraid(mary)=true,[afraid(mary)=true]) @ 7 -f-> (afraid(mary)=true,7,per(o(0)))
i(0) : per(afraid(mary)#true,[afraid(mary)#true]) @ 6 -b-> (afraid(mary)=true,5,per(o(0)))
i(0) : per(throw(wind)#true,[throw(wind)#true]) @ 9 -b-> (throw(wind)=true,8,per(o(0)))

```

>>> Comprehension model:

```

s(0) : person(mary) at always
s(1) : tired(mary) at 1
s(2) : in_bedroom(mary) at 2
s(2) : sleep(mary) at 2
s(3) : suddenly_close(door) at 4
s(3) : wake_up(mary) at 5
s(3) : afraid(mary) at 6
s(4) : get_up(mary,bed) at 7
s(4) : walk_to(mary,kitchen) at 8
s(4) : see(mary,open_window) at 9
s(4) : blow(wind) at 9

```

```

0:      afraid(mary)      blow(wind)      -go_to_vacation(mary)      -i
suddenly_close(door)      -thinks_close_door(thief)      tired(s
is open      wants_to_know(mary,what
1:      afraid(mary)      blow(wind)      -go_to_vacation(mary)      -i
      -thinks_close_door(thief)      thinks_close_door(wind)      < tired(s
is open      wants_to_know(mary,what
2:      afraid(mary)      blow(wind)      -go_to_vacation(mary)      < in_bedroom(mary) > -i
      -thinks_close_door(thief)      thinks_close_door(wind)      tired(s
w is open      wants_to_know(mary,wha
3:      afraid(mary)      blow(wind)      -go_to_vacation(mary)      -i
      -thinks_close_door(thief)      thinks_close_door(wind)      tired(s
is open      wants_to_know(mary,what
4:      afraid(mary)      blow(wind)      -go_to_vacation(mary)      -i
suddenly_close(door) >      -thinks_close_door(thief)      thinks_close_door(wind)      tired(s
w is open      wants_to_know(mary,wha
5:      afraid(mary)      blow(wind)      -go_to_vacation(mary)      -i
      -thinks_close_door(thief)      thinks_close_door(wind)      tired(s
is open      wants_to_know(mary,what
6:      < afraid(mary) >      blow(wind)      -go_to_vacation(mary)      -i
      -thinks_close_door(thief)      thinks_close_door(wind)      tired(s
is open      wants_to_know(mary,what
7:      afraid(mary)      blow(wind)      -go_to_vacation(mary)      -i
      -thinks_close_door(thief)      thinks_close_door(wind)      tired(s
is open      wants_to_know(mary,what
8:      afraid(mary)      blow(wind)      -go_to_vacation(mary)      -i
      -thinks_close_door(thief)      thinks_close_door(wind)      tired(s
is open      < walk_to(mary,kitchen) >      wants_to_know(mary,what
9:      afraid(mary)      < blow(wind) >      -go_to_vacation(mary)      -i
      -thinks_close_door(thief)      thinks_close_door(wind)      tired(s
is open      < see(mary,open_window) >      wants_to_know(mary,what

```

>>> Answering question q(4):

+ accepted choice: ,[wants_to_know(mary,what_happened) at 8]

- rejected choice: ,[watch(mary,tv) at 8]

>>> Answering question q(5):

- rejected choice: ,[thinks_close_door(thief) at 3]

+ accepted choice: ,[thinks_close_door(wind) at 3]

Continue (y/n)? n.

true ■

Εικόνα 5

Κεφάλαιο 4

Αναπαράσταση Ιστοριών

4.1 Γενικά	42
4.2 Η πόρτα (door.pl)	42
4.3 Ασθενοφόρο	46
4.4 Πυροσβεστική	46
4.5 Παγωτό	47
4.5.1 Παγωτό (παραλλαγή)	48
4.6 Πονόδοντος	48
4.6.1 Πονόδοντος (παραλλαγή)	49
4.6.2 Πονόδοντος (παραλλαγή)	50

4.1 Γενικά

Η ευελιξία της γλώσσας του συστήματος μας επιτρέπει να μοντελοποιήσουμε τις δράσεις της ιστορίας και τις καταστάσεις των χαρακτήρων με χρονολογική σειρά όπως ακριβώς αναφέρονται μέσα στην ιστορία. Όταν δηλώνουμε τις δράσεις της ιστορίας ταυτόχρονα ορίζουμε και την χρονική στιγμή της. Είναι λοιπόν σημαντικό να διακρίνουμε ποιες από τις ιδιότητες παραμένουν σταθερές, έως ότου βρεθεί κάτι που θα τις αλλάξει, για να τις ορίσουμε σαν fluent. Συνήθως η ιστορία ξεκινά από την χρονική στιγμή 0 αλλά δεν είναι υποχρεωτικό.

4.2 Η πόρτα (door.pl)

Αφήγηση της Ιστορίας σε φυσική γλώσσα.

Η Μαίρη ήταν κουρασμένη. Κοιμόταν στο δωμάτιο της. Ξαφνικά έκλεισε η πόρτα της κουζίνας και ξύπνησε. Φοβήθηκε. Σηκώθηκε από το κρεβάτι της και περπάτησε προς την κουζίνα. Η Μαίρη είδε ότι το παράθυρο της κουζίνας ήταν ανοιχτό και κατάλαβε ότι ο αέρας έκλεισε την πόρτα.

Αναπαράσταση της ιστορίας

session(s(0),[],all).

session(s(1),[q(1)],all).

session(s(2),[q(2)],all).

session(s(3),[q(3)],all).

session(s(4),[q(4),q(5)],all).

s(0) :: person(mary) at always.

s(1) :: tired(mary) at 1.

s(2) :: in_bedroom(mary) at 2.

s(2) :: sleep(mary) at 2.

s(3) :: suddenly_close(door) at 4.

s(3) :: wake_up(mary) at 5.

s(3) :: afraid(mary) at 6.

s(4) :: get_up(mary,bed) at 7.

s(4) :: walk_to(mary,kitchen) at 8.

s(4) :: see(mary,open_window) at 9.

s(4) :: blow(wind) at 9.

%%% Γιατί η Μαίρη είναι κουρασμένη; %%% %%% %%% %%% %%% %%% %%% %%%

q(1) ?? work(mary) at 1;

 go_to_vacation(mary) at 1.

%%% Πότε κοιμόταν η Μαίρη; %%% %%%

q(2) ?? it_is(night) at 2;

 it_is(morning) at 2.

%%% %%% Τι έκανε τη στιγμή 8; %%% %%%

%Η Μαίρη πιστεύει ότι ο κλέφτης έκλεισε την πόρτα; η Μαίρη πιστεύει ότι ο αέρας έκλεισε την πόρτα; %%% %%% %%% %%% %%% %%%

q(3) ?? thinks_close_door(thief) at 3;

thinks_close_door(wind) at 3.

%%%%% Τι έκανε τη στιγμή 8; %%%

%% Τη στιγμή 8 η Μαίρη ήθελε να μάθει τι συνέβη; %%%%

%%% Τη στιγμή 8 η Μαίρη έβλεπε τηλεόραση; %%%%

q(4) ?? wants_to_know(mary,what_happened) at 8;

watch(mary,tv) at 8.

%%% Ποιος πιστεύει ότι έκλεισε την πόρτα; %%%%

%%% Η Μαίρη πιστεύει ότι Ο κλέφτης έκλεισε την πόρτα; Η Μαίρη πιστεύει ότι ο αέρας έκλεισε την πόρτα; %%%

q(5) ?? thinks_close_door(thief) at 3;

thinks_close_door(wind) at 3.

%%%%% BACKGROUND / WORLD KNOWLEDGE %%%%

```
fluents([
  tired(_),
  sleep(_),
  afraid(_),
  it_is(_),
  wants_to_know(_,_),
  watch(_,_),
  is(_,_),
  thinks_close_door(_),
  blow(_),
]).
```

%%% BACKGROUND/WORLD KNOWLEDGE %%%%

%%%%% ΕΡΩΤΗΣΗ1 %%%%

%% Συνήθως οι άνθρωποι κουράζονται όταν εργάζονται %%%%

p(11) :: tired(Person) implies work(Person).

%% Συνήθως όταν ο άνθρωπος κάνει διακοπές δεν είναι κουρασμένος
%%%%%%%%%

p(12) :: go_to_vacation(Person) implies -tired(Person).

p(13) :: work(Person) implies -go_to_vacation(Person).

p(14) :: go_to_vacation(Person) implies -work(Person).

%%% ΕΡΩΤΗΣΗ2 %%%

%% Συνήθως οι άνθρωποι κοιμούνται το βράδυ %%%%%%%%%

p(15) :: in_bedroom(Person), sleep(Person) implies it_is(night).

p(16) :: it_is(night) implies -it_is(morning).

p(17) :: it_is(morning) implies -it_is(night).

%% ΕΡΩΤΗΣΗ3 %%%%%%%%%

c(20) :: wake_up(Person), sleep(Person) causes -sleep(Person).

c(21) :: suddenly_close(door), wake_up(Person) causes afraid(Person).

p(22) :: suddenly_close(door), afraid(Person), person(Person) implies
thinks_close_door(thief).

p(24) :: thinks_close_door(wind) implies -thinks_close_door(thief).

p(25) :: thinks_close_door(thief) implies -thinks_close_door(wind).

p(24)>>p(22).

%%% ΕΡΩΤΗΣΗ4 %%

p(33) :: get_up(Person,bed), afraid(Person) implies wants_to_know(Person,what_happened).

p(34) :: walk_to(Person) implies -watch(Person,tv).

p(35) :: wants_to_know(Person,what_happened) implies -watch(Person,tv).

p(36) :: watch(Person,tv) implies -wants_to_know(Person,what_happened).

p(33)>>p(36).

%% ΕΡΩΤΗΣΗ5 %%%

%%% Αν είναι το παράθυρο είναι ανοικτό και φυσάει τότε η πόρτα θα κλείσει από τον αέρα
%%

p(40) :: see(mary,open_window) implies is(window,open).

c(41) :: is(window,open), blow(wind) causes thinks_close_door(wind).

c(41)>>p(25).

4.3 Ασθενοφόρο

%John fell down the stairs. Mary phoned for an ambulance.

session(s(0),[q(1)],all).

s(0) :: person(john) at always.

s(0) :: person(mary) at always.

s(0) :: fell_down(john,the_stairs) at 0.

s(0) :: phoned(mary,an_ambulance) at 1.

% why did Mary phone the ambulance?

q(1) ?? hurt(john) at 1.

fluents([hurt(_)]).

c(11) :: fell_down(Person,the_stairs) causes hurt(Person).

c(12) :: hurt(Person), person(Person), person(Other) causes phoned(Other,an_ambulance).

4.4 Πυροσβεστική

%John run out of the house. He saw fire engine approaching.

session(s(0),[q(1)],all).

s(0) :: person(john) at always.

s(0) :: run_out(john,of_the_house) at 1.

s(0) :: saw(john,approaching_fire_engine) at 2.

% why did John run out?

q(1) ?? on_fire(house) at 1;

earthquake(happened) at 1;
in_house(thief) at 1.

fluents([on_fire(_),saw(_,_)]).

p(11) :: run_out(Person,of_the_house), saw(Person,approaching_fire_engine) implies
on_fire(house).

p(12) :: on_fire(house) implies -in_house(thief).

p(13) :: in_house(thief) implies -on_fire(house).

p(14) :: on_fire(house) implies -earthquake(happened).

p(15) :: earthquake(happened) implies -on_fire(house).

p(16) :: earthquake(happened) implies -in_house(thief).

p(17) :: in_house(thief) implies -earthquake(happened).

4.5 Παγωτό

% Ο Γιάννης θέλει παγωτό. Ο Γιάννης τρώει παγωτό.

session(s(0),[q(1)],all).

s(0) :: person(john) at always.

s(0) :: wants(john,ice_cream) at 0.

s(0) :: eat(john,ice_cream) at 2.

% what did John buy ?

q(1) ?? buy(john,ice_cream) at 1.

fluents([wants(_,_),eat(_,_)]).

c(11) :: wants(Person,ice_cream) causes buy(Person,ice_cream).

c(12) :: buy(Person,ice_cream) causes eat(Person,ice_cream).

4.5.1 Παγωτό (παραλλαγή)

%John wanted an ice-cream. John enjoyed eating ice-cream in the park.

session(s(0),[q(1)],all).

s(0) :: person(john) at always.

s(0) :: wanted(john,ice_cream) at 0.

s(1) :: enjoyed_eating(john,ice_cream) at 2.

q(1) ?? buy(john,ice_cream) at 1;

 make(john,ice_cream) at 1.

fluents([wanted(_,_),enjoyed_eating(_,_),make(_,_)]).

c(11) :: wanted(Person,ice_cream) causes buy(Person,ice_cream).

c(12) :: buy(Person,ice_cream) causes enjoyed_eating(Person,ice_cream).

c(13) :: wanted(Person,ice_cream) causes make(Person,ice_cream).

%c(11)>>c(13) δεν είναι έτοιμο το σύστημα.

p(14) :: buy(Person,ice_cream) implies -make(Person,ice_cream).

p(15) :: make(Person,ice_cream) implies -buy(Person,ice_cream).

p(14)>>c(13).

c(11)>>p(15).

4.6 Πονόδοντος

% ΟΝίκος τρώει πολλά γλυκά. Ο Νίκος επισκέφτηκε τον οδοντίατρο.

session(s(0),[q(1)],all).

s(0) :: person(nick) at always.

s(0) :: eats(nick,a_lot_of_sweets) at 0.

s(0) :: went_to(nick,the_dentist) at 2.

q(1) ?? has(nick,toothache) at 1.

fluents([eats(_,_),has(_,_),went_to(_,_)]).

c(11) :: eats(Person,a_lot_of_sweets) causes has(Person,toothache).

c(12) :: has(Person,toothache) causes went_to(Person,the_dentist).

4.6.1 Πονόδοντος (παραλλαγή)

%John eats a lot of sweets. John went to the dentist.

session(s(0),[q(1)],all).

session(s(1),[q(2)],all).

s(0) :: person(john) at always.

s(0) :: eats(john,a_lot_of_sweets) at 0.

s(1) :: went_to(john,the_dentist) at 2.

q(1) ?? has(john,toothache) at 1;

 has(john,diabete) at 1;

 gain(john,weight) at 1.

q(2) ?? has(john,toothache) at 1;

 check(john) at 1;

 pay(john) at 1.

fluents([eats(_,_),has(_,_)]).

p(1) :: person(Person) implies -has(Person,toothache).

p(2) :: person(Person) implies -has(Person,diabete).

p(3) :: person(Person) implies -gain(Person,weight).

c(11) :: eats(Person,a_lot_of_sweets) causes has(Person,toothache).

c(12) :: has(Person,toothache) causes went_to(Person,the_dentist).

p(13) :: went_to(Person,the_dentist) implies has(Person,toothache).

c(14) :: eats(Person,a_lot_of_sweets) causes gain(Person,weight).

c(15) :: eats(Person,a_lot_of_sweets) causes has(Person,diabete).

p(16) :: went_to(Person,the_dentist) implies check(Person).

p(17) :: went_to(Person,the_dentist) implies pay(Person).

c(11)>>p(1).

p(18) :: has(Person,toothache) implies -pay(Person).

4.6.2 Πονόδοντος (παραλλαγή)

%John eats a lot of sweets.John went to the dentist.

session(s(0),[q(1)],all).

s(0) :: person(john) at always.

s(0) :: eats(john,a_lot_of_sweets) at 0.

s(1) :: went_to(john,the_dentist) at 2.

q(1) ?? possible_has(john,toothache) at 1;

possible_has(john,diabete) at 1;

possible_gain(john,weight) at 1.

fluents([eats(_,_),has(_,_),happy(_)]).

c(11) :: eats(Person,a_lot_of_sweets) causes possible_has(Person,toothache).

c(12) :: possible_has(Person,toothache) causes possible_went_to(Person,the_dentist).

c(14) :: eats(Person,a_lot_of_sweets) causes possible_gain(Person,weight).

c(15) :: eats(Person,a_lot_of_sweets) causes possible_has(Person,diabete).

c(16) :: eats(Person,a_lot_of_sweets) causes possible_happy(Person).

Κεφάλαιο 5

Αναπαράσταση Εκπαιδευτικών Κειμένων

5.1 Γενικά	52
5.2 Carly's Family	52
5.3 Spider Webs	57
5.4 A new pet shop	61
5.5 Anna's new Friend	67
5.6 Περιγραφή Δυσκολιών	71

5.1 Γενικά

Σε αυτό το κεφάλαιο χρησιμοποιήσαμε τα παρακάτω εκπαιδευτικά κείμενα Carly's Family, Spider Webs, A new pet shop, Anna's new Friend, τα οποία είναι διαθέσιμα από: <http://englishforeveryone.org/Topics/Reading-Comprehension.htm>. Στόχος του κεφαλαίου είναι οι αναπαραστάσεις των εκπαιδευτικών ιστοριών στο STAR σύστημα και η περιγραφή των δυσκολιών για τα συγκεκριμένα εκπαιδευτικά κείμενα.

5.2 Carly's Family

Παρουσιάζεται το κείμενο και οι ερωτήσεις του στη φυσική γλώσσα.

Carly has a large family. She lives with four people. Carly also has two pets.

Carly's mom is a doctor. Carly's mom works at the hospital. Carly's mom helps people who are sick.

Carly's dad works at home. Carly's dad cooks for the family. Carly's dad drives the kids to soccer practice.

Carly has two brothers. James is ten years old. Scott is fourteen years old.

Carly has two pets. Jinx is a small, black cat. Diego is a large, brown dog.

Carly loves her family!

Ερωτήσεις:

1) How many people are in Carly's family?

- A. four
- B. five
- C. six

2) Carly's mom works at the

- A. restaurant
- B. mall
- C. hospital

3) This passage is mostly about Carly's

- A. family
- B. pets
- C. soccer team

4) Which of the following is most likely true?

- A. Carly's mom coaches the soccer team.
- B. James is the best soccer player in the family.
- C. Jinx and Diego are part of Carly's family.

5) The oldest brother in Carly's family is

- A. James

B. Scott

C. Diego

6) How is your family the same as Carly's family? How is it different? Explain.

Παρακάτω παρουσιάζεται η αναπαράσταση της ιστορίας στο STAR σύστημα.

session(s(0),[q(1)],all).

session(s(1),[q(2)],all).

session(s(2),[],all).

session(s(3),[],all).

session(s(4),[q(3),q(4)],all).

s(0):: person(carly) at always.

s(0):: person(carlys_mom) at always.

s(0):: person(carlys_dad) at always.

s(0):: person(james) at always.

s(0):: person(scott) at always.

s(0):: has(carly,large_family) at 0.

s(0):: lives(carly,with_four_people) at 0.

s(0):: has(carly,two_pets)at 0.

s(1):: is(carlys_mom,doctor) at 1.

s(1):: works(carlys_mom,at_the_hospital) at 1.

s(1):: helps(carlys_mom,people_who_are(sick)) at 1.

s(2):: works(carlys_dad,at_home) at 2.

s(2):: cooks(carlys_dad,for_the_family)at 2.

s(2):: drives(carlys_dad,the_kids_to_soccer_practice) at 2.

s(3):: has(carly,two_brothers)at 3.

s(3):: is(james,ten_years_old)at 3.

s(3):: is(scott,fourteen_years_old)at 3.

s(4):: has(carly,two_pets)at 4.

s(4):: is(jinx,small)at 4.

s(4):: is(jinx,black)at 4.

s(4):: is(jinx,cat)at 4.

s(4):: is(diego,large)at 4.

s(4):: is(diego,brown)at 4.

s(4):: is(diego,dog)at 4.

%How many people are in Carly's family?

q(1) ?? people_carlys_family_are(four) at 0;

people_carlys_family_are(five) at 0;

people_carlys_family_are(six) at 0.

%Carly's mom works at the

q(2) ?? works(carlys_mom,at_the_restaurant) at 1;

works(carlys_mom,at_the_mall) at 1;

works(carlys_mom,at_the_hospital) at 1.

%The oldest brother in Carly's family is

q(3) ?? is(james,oldest_brother_in_carlys_family) at 3;

is(scott,oldest_brother_in_carlys_family) at 3;

is(diego,oldest_brother_in_carlys_family) at 3.

% Which of the following is most likely true?

q(4) ?? is(carlys_mom,coaches_the_soccer_team) at 4;

is(james,the_best_soccer_player_in_the_family)at 4;

are(diego,jinx,part_of_carlys_family) at 4.

fluents([

has(_,_),

lives(_,_),

is(_,_),

works(_,_),

helps(_,_),

drives(_,_),

loves(_,_),

people_carlys_family_are(_)

]).

%% BACKGROUND /WORLD KNOWLEDGE

%% BACKGROUND FOR THE FIRST QUESTION%%

p(11) :: person(Person), lives(Person,with_four_people) implies
people_carlys_family_are(five).

p(17) :: people_carlys_family_are(six) implies -people_carlys_family_are(five).

p(18) :: people_carlys_family_are(six) implies -people_carlys_family_are(four).

p(19) :: people_carlys_family_are(five) implies -people_carlys_family_are(four).

p(20) :: people_carlys_family_are(five) implies -people_carlys_family_are(six).

p(19) :: people_carlys_family_are(four) implies -people_carlys_family_are(five).

p(20) :: people_carlys_family_are(four) implies -people_carlys_family_are(six).

%% BACKGROUND FOR THE SECOND QUESTION%% % % %

%p(21) :: is(Person,doctor), helps(Person,people_who_are(sick)) implies
works(Person,at_the_hospital).

p(22) :: works(Person,at_the_hospital) implies -works(Person,at_the_mall).

p(23) :: works(Person,at_the_hospital) implies -works(Person,at_the_restaurant).

p(24) :: works(Person,at_the_mall) implies -works(Person,at_the_hospital).

p(25) :: works(Person,at_the_restaurant) implies -works(Person,at_the_hospital).

% BACKGROUND FOR THE FOURTH QUESTION%%

% Τα κατοικίδια είναι μέρος της οικογένειας % % % % %

p(31) :: is(X,small),is(X,black), is(X,cat) implies is(X,pet).

p(32) :: is(X,pet) implies is_part_of_family(X).

p(33) :: is(Y,large),is(diego,brown),is(Y,dog) implies is(Y,pet).

p(34) :: is(Y,pet) implies is_part_of_family(Y).

p(35) :: is_part_of_family(X), is_part_of_family(Y) implies are(X,Y,part_of_carlys_family).

p(36) :: are(X,Y,part_of_carlys_family), person(Person), is(X,dog), is(Y, dog) implies -
is(Person,the_best_soccer_player_in_the_family).

%p(37) :: is(Person,the_best_soccer_player_in_the_family) implies -
are(X,Y,part_of_carlys_family).

p(37) :: is(Person,doctor) implies -is(Person,coaches_the_soccer_team).

% BACKGROUND FOR THE FIFTH QUESTION%%

p(40) :: is(Person,ten_years_old), person(Person), is(Other,fourteen_years_old) implies is(Other,oldest_brother_in_carllys_family).

p(41) :: is(Person,ten_years_old), person(Person), person(Other), is(Other,fourteen_years_old) implies -is(Person,oldest_brother_in_carllys_family).

p(44) :: is(X,dog) implies -is(X,oldest_brother_in_carllys_family).

5.3 Spider Webs

Παρουσιάζεται το κείμενο και οι ερωτήσεις του στη φυσική γλώσσα.

All spiders spin webs. That's because webs help spiders. Webs help spiders do three things. Webs help spiders hold eggs. Webs help spiders hide. And webs help spiders catch food. Webs help spiders hold eggs. Many spiders like to lay their eggs in their webs. The webs help keep the eggs together. Webs help spiders keep their eggs safe.s

Webs help spiders hide. Most spiders are dark. They are brown, grey, or black. But spider webs are light. They are white and cloudy. When spiders hide in their webs, they are harder to see.

Webs help spiders catch food. Spider webs are sticky. When a bug flies into the web, it gets stuck. It moves around. It tries to get out. But it can't. It is trapped! Spiders can tell that the bug is trapped. That's because spiders feel the web move. And the spider is hungry. The spider goes to get the bug.

As you can see, webs help spiders hold eggs. Webs help spiders hide. And webs help spiders catch food. Without webs, spiders would not be able to live like they do. Spiders need their webs to survive!

Ερωτήσεις

1) This passage is mostly about

A. spider colors

- B. spider webs
- C. spider eggs

2) Spider webs help spiders

- I. hold eggs
- II. catch food
- III. find water

- A. I only
- B. I and II only
- C. I, II, and III

3) As used in paragraph 4, the word trapped most nearly means

- A. stuck
- B. hidden
- C. eaten

4) How can spiders tell when something is trapped in their web?

- A. They hear it.
- B. They smell it.
- C. They feel it.

5) As used in the last sentence of the passage, the word survive means to stay

- A. alive
- B. hidden
- C. caught

6) The passage lists three reasons why spiders spin webs. Of these reasons, which do you think is the most important? How come?

7) Do you like spiders? Why or why not? Have you ever been scared by a spider? Have you ever been hurt by one? Have you ever helped one? Explain.

8) Are spiders good? Do we need them in our world? Why or why not?

Παρακάτω παρουσιάζεται η αναπαράσταση της ιστορίας στο STAR σύστημα.

session(s(0),[q(1)],all).

session(s(1),[],all).

session(s(2),[],all).

session(s(3),[q(2)],all).

session(s(4),[],all).

s(0):: spin(all_spiders,webs) at 0.

s(0):: thats_because_help(webs,spiders_do_three_thihgs) at 0.

s(0):: help(webs,spiders_hold_eggs) at 0.

s(0):: help(webs,spiders_hide) at 0.

s(0):: help(webs,spiders_catch_food) at 0.

s(1):: like(many_spiders,to_lay_their_eggs_in_their_webs) at 1.

s(1):: help(webs,keep_the_eggs_together) at 1.

s(1):: help(webs,keep_the_eggs_safe) at 1.

s(2):: help(webs,spiders_hide) at 2.

s(2):: are(most_spiders,dark) at 2.

s(2):: are(spiders,brown) at 2.

s(2):: are(spiders,greys) at 2.

s(2):: are(spiders,black) at 2.

s(2):: are(spiders,white_and_cloudy) at 2.

s(2):: when_hide_in_their_webs(spiders,they_are_harder_to_see) at 2.

s(3):: are(spider_webs,sticky) at 3.

s(3):: gets_stuck(when_a_bug_flies_into_the_web) at 3.

s(3):: move_around(byg) at 4.

s(3):: tries(bug,get_out) at 5.

s(3):: cant(bug,get_out) at 6.

s(3):: trapped(bug) at 6.

s(3):: can_tell(spiders,trapped(bug)) at 7.

s(3):: thats_because_feel(spiders,the_web_move) at 7.

s(3):: is(spider,hungry) at 8.

s(3):: eat(spider,bug) at 8.

s(4):: dont_live(spiders,without_webs) at 9.

s(4):: need(spiders,web_to_survive) at 9.

% Spider webs help spiders

q(1) ?? help(webs,spiders_hold_eggs) at 3;

 help(webs,spiders_catch_food) at 3;

 help(webs,spiders_find_water) at 3.

%How can spiders tell when something is trapped in their web?

q(2) ?? thats_because_hear(spiders,the_web_move) at 7;

 thats_because_smell(spiders,the_web_move) at 7;

 thats_because_feel(spiders,the_web_move) at 7.

fluents([

 help(_,_),

 like(_,_),

 are(_)

]).

%%% BACKGROUND / WORLD KNOWLEDGE %%% %%% %%%

p(11) :: help(webs,spiders_find_water) implies -help(webs,spiders_hold_eggs).

p(12) :: help(webs,spiders_hold_eggs) implies -help(webs,spiders_find_water).

p(13) :: thats_because_feel(spiders,the_web_move) implies -
thats_because_hear(spiders,the_web_move).

p(14) :: thats_because_feel(spiders,the_web_move) implies -
thats_because_smell(spiders,the_web_move).

5.4 A new pet shop

Παρουσιάζεται το κείμενο και οι ερωτήσεις του στη φυσική γλώσσα.

A new pet shop opened up on Main Street.

Martha and Sam went to look at the new shop.

There was a brightly colored bird in the window.

The bird had red and blue feathers.

The bird had a long beak.

Inside the shop, there was a large fish tank.

The tank had many fish.

The fish were many shapes, sizes, and colors.

In the back of the shop, there was a small cage.

There were furry, gray kittens in the cage.

The kittens were very tiny.

Martha decided to buy a kitten.

Sam decided to buy a fish.

The two friends had a great day at the new pet shop.

Ερωτήσεις

1) The new pet shop is

A. full of people

B. on Main Street

C. small and busy

2) What is the best title for this passage?

A. A Day at the Pet Shop

B. My New Fish Tank

C. The Best Store in Town

3) The pet shop owner most likely put the bird in the window because

- A. the fish could not be around the bird
- B. he wanted to attract new customers
- C. the bird needed enough sunlight

4) The fish were located in the

- A. large tank
- B. back room
- C. front window

5) Which sentence from the passage best describes what was in the back of the pet shop?

- A. "A brightly colored bird was perched on a stand."
- B. "Many different colored fish swam in a large tank."
- C. "Small, furry, gray colored kittens were in a cage."

6) Which animal was NOT in the pet shop?

- A. kitten
- B. puppy
- C. fish

7) Sam decided to buy

- A. a bird
- B. a fish
- C. a kitten

8) If you went to the pet shop with Martha and Sam, what kind of pet would you want to get?
Why?

9) Do pet shops usually have other types of animals that the passage does not talk about?
What other animals are usually at a pet shop?

Παρακάτω παρουσιάζεται η αναπαράσταση της ιστορίας στο STAR σύστημα.

session(s(0),[],all).

session(s(1),[q(1)],all).

session(s(2),[q(2)],all).

session(s(3),[q(3)],all).

session(s(4),[q(4)],all).

session(s(5),[q(5),q(6)],all).

s(0):: person(martha) at always.

s(0):: person(sam) at always.

s(1):: opened_up(a_new_pet_shop,on_Main_Street) at 1.

s(1):: went_to_look(martha_and_sam,at_the_new_shop)at 2.

s(2):: there_was(a_brightly_colored_bird,in_the_window) at 3.

s(2):: had(the_bird,red_and_blue_feathers) at 3.

s(2):: had(the_bird,a_long_beak) at 3.

s(3):: there_was(a_large_fish_tank,inside_the_shop) at 4.

s(3):: had(the_tank,many_fish)at 4.

s(3):: were(fish,many_shapes_sizes_and_colors) at 4.

s(4):: there_was(a_small_cage,in_the_back_of_the_shop) at 5.

s(4):: there_were(furry_gray_kittens,in_the_cage) at 5.

s(4):: were(kittens,very_tiny) at 5.

s(5):: decided_to_buy(martha,a_kitten) at 6.

s(5):: decided_to_buy(sam,a_fish) at 6.

s(5):: had(the_two_friends,a_great_day_at_the_new_pet_shop) at 6.

%%the new pet shop is

q(1) ?? is(the_new_petPshop,full_of_people) at 1;

is(the_new_petPshop,on_Main_Street) at 1;

is(the_new_petPshop,small_and_busy) at 1.

%%The pet shop owner most likely put the bird in the window because

q(2) ??

the_pet_owner_put_the_bird_in_window_because(the_fish_could_not_be_round_the_bird)
at 3;

the_pet_owner_put_the_bird_in_window_because(he_wanted_to_attract_new_customers) at
3;

the_pet_owner_put_the_bird_in_window_because(the_bird_needed_enough_sunlight)
at 3.

%%The fish were located in the

q(3) ?? the_fish_were_located_in_the(large_tank) at 4;

the_fish_were_located_in_the(back_room) at 4;

the_fish_were_located_in_the(front_window) at 4.

%%Which sentence from the passage best describes what was in the back of the pet shop?

q(4) ??

there_was(a_brightly_colored_bird_was_perched_on_a_stand,in_the_back_of_the_shop)at 5;

there_was(many_different_colored_fish_swam_in_a_large_tank,in_the_back_of_the_shop)
at 5;

there_was(small_furry_gray_colored_kittens_were_in_a_cage,in_the_back_of_the_shop) at
5.

%%Sam decided to buy

q(5) ?? decided_to_buy(sam,a_bird) at 6;

decided_to_buy(sam,a_fish) at 6;

decided_to_buy(sam,a_kitten) at 6.

%% which animal was not in the pet shop?

q(6) ?? was_not(kitten,in_the_pet_shop) at 6;
was_not(puppy,in_the_pet_shop) at 6;
was_not(fish,in_the_pet_shop) at 6.

fluents([
opened_up(_,_),
went_to_look(_,_),
there_was(_,_),
had(_,_),
there_were(_,_),
were(_,_)
]).

%% %%

p(11) :: opened_up(a_new_pet_shop,on_Main_Street) implies
is(the_new_petPshop,on_Main_Street).

p(12) :: is(the_new_petPshop,on_Main_Street) implies -
is(the_new_petPshop,full_of_people).

p(13) :: is(the_new_petPshop,on_Main_Street) implies -
is(the_new_petPshop,small_and_busy).

p(14) :: there_was(a_brightly_colored_bird,in_the_window) implies
attract(the_bright_colors_our_attention).

p(15) :: attract(the_bright_colors_our_attention) implies
the_pet_owner_put_the_bird_in_window_because(he_wanted_to_attract_new_customers).

p(16) ::
the_pet_owner_put_the_bird_in_window_because(he_wanted_to_attract_new_customers)
implies -
the_pet_owner_put_the_bird_in_window_because(the_fish_could_not_be_round_the_bird).

p(17) ::
the_pet_owner_put_the_bird_in_window_because(he_wanted_to_attract_new_customers)
implies -
the_pet_owner_put_the_bird_in_window_because(the_bird_needed_enough_sunlight).

%p(16) :: attract(the_bright_colors_our_attention) implies -
the_pet_owner_put_the_bird_in_window_because(the_fish_could_not_be_round_the_bird).
%p(17) :: attract(the_bright_colors_our_attention) implies -
the_pet_owner_put_the_bird_in_window_because(the_bird_needed_enough_sunlight).

p(18) :: there_was(a_large_fish_tank,inside_the_shop)implies
the_fish_were_located_in_the(large_tank).

p(19) :: the_fish_were_located_in_the(large_tank) implies -
the_fish_were_located_in_the(back_room).

p(20) :: the_fish_were_located_in_the(large_tank) implies -
the_fish_were_located_in_the(front_window).

p(21) ::
there_was(a_small_cage,in_the_back_of_the_shop),there_were(furry_gray_kittens,in_the_ca
ge) implies
there_was(small_furry_gray_colored_kittens_were_in_a_cage,in_the_back_of_the_shop).

p(22) ::
there_was(small_furry_gray_colored_kittens_were_in_a_cage,in_the_back_of_the_shop)
implies -
there_was(a_brightly_colored_bird_was_perched_on_a_stand,in_the_back_of_the_shop).

p(23) ::
there_was(small_furry_gray_colored_kittens_were_in_a_cage,in_the_back_of_the_shop)
implies -
there_was(many_different_colored_fish_swam_in_a_large_tank,in_the_back_of_the_shop).

p(24) ::decided_to_buy(sam,a_fish) implies -decided_to_buy(sam,a_bird).

p(25) ::decided_to_buy(sam,a_fish) implies -decided_to_buy(sam,a_kitten).

p(26) :: there_were(furry_gray_kittens,in_the_cage) implies -
was_not(kitten,in_the_pet_shop).

p(27) :: there_was(a_large_fish_tank,inside_the_shop) implies -
was_not(fish,in_the_pet_shop).

5.5 Anna's New Friend

Παρουσιάζεται το κείμενο και οι ερωτήσεις του στη φυσική γλώσσα.

It is Thursday. It is raining today. It is a rainy day. Anna is inside the house. Anna is watching TV. Anna is watching TV inside the house.

Anna cannot go outside. It is raining outside. Anna cannot go outside because it is raining outside.

Anna is bored.

Wait!

Anna hears someone at the door. Someone is at the door of her house. Anna opens the door. What does Anna see?

Anna sees a dog. The dog is small. Anna sees a small dog. The dog is wet. The dog is wet from the rain.

"Awww! You are all wet," Anna says to the dog. "You are very cute!"

"Mom!" Anna says.

"Yes, dear?" asks Anna's mother.

"There is a dog here. There is a dog at the door," Anna says.

"What?" asks Anna's mother. "A dog?"

Anna's mother comes to the door. Anna's mother sees the wet dog. The wet dog looks cute.

"Can we keep it?" asks Anna.

"Well," says Anna's mother.

"Please?" asks Anna. "He really needs a home."

"Okay," says Anna's mother.

Both Anna and the dog are happy.

Παρακάτω παρουσιάζεται η αναπαράσταση της ιστορίας στο STAR σύστημα.

session(s(0),[],all).

session(s(1),[q(1)],all).

session(s(2),[],all).

session(s(3),[q(2)],all).

session(s(4),[q(3)],[is(,_)]).

session(s(5),[q(4),q(5)],[sees(,_),asks(,_),say(,_)]).

s(0):: person(anna) at always.

s(0):: person(annas_mother) at always.

s(1):: it_is(thursday) at 1.

s(1):: it_is(raining,today) at 1.

s(1):: it_is(a_rainy,day) at 1.

s(1):: is(anna,inside_the_house) at 1.

s(1):: is_watching(anna,tv) at 2.

s(1):: is_watching(anna,tv_inside_the_house) at 2.

s(2):: cannot_go(anna,outside) at 2.

s(2):: it_is(raining,outside) at 1.

s(2):: cannot_go(anna,because_it_is(raining,outside)) at 2.

s(2):: is_bored(anna) at 3.

s(3):: hears(anna,somenone_at_the_door) at 4.

s(3):: is(someone,at_the_door_of_her_house) at 4.

s(3):: opens(anna,the_door) at 5.

s(3):: what_does_see(anna) at 6.

s(3):: sees(anna,a_dog) at 6.
s(3):: is(the_dog,small) at 7.
s(3):: sees(anna,a_small_dog) at 7.
s(3):: is(the_dog,wet) at 7.
s(3):: is_wet(the_dog,from_the_rain) at 7.

s(4):: says(anna,you_are_all_wet) at 8.
s(4):: says(anna,you_are_very_cute) at 9.
s(4):: says(anna,mom) at 10.
s(4):: asks(annas_mother,yes_dear) at 11.
s(4):: says(anna,there_is(a_dog,here)) at 12.
s(4):: says(anna,there_is(a_dog,at_the_door)) at 13.
s(4):: aks(annas_mother,what) at 14.
s(4):: aks(annas_mother,a_dog) at 15.
s(4):: comes(annas_mother,to_the_door) at 16.
s(4):: sees(annas_mother,to_wet_dog) at 16.
s(4):: looks(the_wet_dog,cute) at 17.
s(4):: asks(anna,can_we_keep_it) at 18.
s(4):: says(annas_mother,well) at 19.
s(4):: asks(anna,please) at 19.
s(4):: really_needs(he,a_home) at 20.
s(4):: says(annas_mother,okay) at 21.
s(4):: are(both_anna_and_the_dog,happy) at 22.

fluents([
 it_is(_),
 it_is(_,_),
 is_watching(_,_),
 is_bored(_,_),
 is_wet(_,_),
 is(_,_),
 sees(_,_),
 are(_,_)
]).

%%What day of the week is it in the passage?

q(1) ?? it_is(tuesday) at 1;

it_is(wednesday) at 1;

it_is(thursday) at 1.

%%Who comes to visit Anna's house?

q(2) ?? comes_to_visit_annas_house(a_man) at 6;

comes_to_visit_annas_house(a_cat) at 6;

comes_to_visit_annas_house(a_dog) at 6.

%%How does the dog look?

q(3) ?? is(the_dog,wet) at 9;

is(the_dog,cute) at 9;

is(the_dog,strong)at 9.

%%What happens at the end of the passage?

q(4) ?? keeps(anna,the_dog) at 21;

takes(anna,the_dog_to_the_hospital) at 21;

dries(anna,the_dog_with_a_towel) at 21.

%%Anna is

q(5) ?? is(anna,old) at 22;

is(anna,kind) at 22;

is(anna,angry) at 22.

p(11) :: it_is(thursday) implies -it_is(wednesday).

p(12) :: it_is(thursday) implies -it_is(tuesday).

c(13) :: is(someone,at_the_door_of_her_house) causes opens(anna,the_door).

c(14) :: opens(anna,the_door), sees(anna,a_dog) causes comes_to_visit_annas_house(a_dog).

p(15) :: comes_to_visit_annas_house(a_dog) implies -comes_to_visit_annas_house(a_cat).

p(16) :: comes_to_visit_annas_house(a_dog) implies -comes_to_visit_annas_house(a_man).

c(17) :: sees(annas_mother,to_wet_dog), asks(anna,can_we_keep_it) causes
says(annas_mother,okay).

p(18) :: says(annas_mother,okay) implies keeps(anna,the_dog).

p(19) :: keeps(anna,the_dog) implies -takes(anna,the_dog_to_the_hospital).

p(20) :: keeps(anna,the_dog) implies -dries(anna,the_dog_with_a_towel).

p(21) :: says(anna,you_are_very_cute) implies is(the_dog,cute).

p(22) :: is(the_dog,wet) implies -is(the_dog,strong).

p(23) :: is(the_dog,cute) implies -is(the_dog,strong).

p(24) :: asks(anna,can_we_keep_it) implies is(anna,kind).

p(25) :: is(anna,kind) implies -is(anna,old).

p(26) :: are(both_anna_and_the_dog,happy) implies is(anna,happy).

p(27) :: is(anna,happy) implies -is(anna,angry).

5.6 Περιγραφή Δυσκολιών

Σκοπός του κεφαλαίου ήταν να μοντελοποιήσουμε τον κόσμο που περιγράφεται στα εκπαιδευτικά κείμενα, να απεικονίσουμε τους χαρακτήρες, τις δράσεις που διαδραματίζονται κατά τη διάρκεια του κειμένου και τις γνώσεις που μας προσφέρει ένα εκπαιδευτικό κείμενο όπως για παράδειγμα το κείμενο 'Spider Webs'. Τέλος να αναπαραστήσουμε τις ερωτήσεις των εκπαιδευτικών κειμένων στο σύστημα STAR.

Η ευελιξία της γλώσσας RAC μας επιτρέπει να μοντελοποιήσουμε τον κόσμο των εκπαιδευτικών κειμένων. Σε κάθε κείμενο υπάρχουν ερωτήσεις πολλαπλής επιλογής και ερωτήσεις ανοικτού τύπου. Στο σύστημα ενσωματώσαμε τις ερωτήσεις πολλαπλής επιλογής. Για κάποιες ερωτήσεις το σύστημα δεν είναι ικανό να δημιουργήσει το μοντέλο κατανόησης για απάντησει. Για παράδειγμα η τρίτη ερώτηση του κειμένου Carly's Family.

3) Το κείμενο αναφέρεται περισσότερο για :

A. Την οικογένεια της Carly's

- B. Τα κατοικίδια της Carly's
- C. Την ομάδα soccer της Carly's

Ο συγγραφέας σε αυτό το κείμενο περιγράφει τους ανθρώπους και τα κατοικίδια της οικογένειας. Ο συγγραφέας δίνει μόνο μια πληροφορία για την ομάδα σόκερ. Στην πρώτη παράγραφο αναφέρει ότι είναι μια μεγάλη οικογένεια, το κορίτσι ζει με άλλα τέσσερα άτομα και υπάρχουν δύο κατοικίδια. Ο τίτλος του κειμένου μας δίνει την απάντηση. Ο αναγνώστης αναζητεί σε ποιο θέμα αναφέρεται περισσότερο ο συγγραφέας του κειμένου για να απαντήσει την ερώτηση. Το σύστημα δεν έχει την ικανότητα δηλαδή να αναζητήσει μέσα στο κείμενο σε ποιο θέμα αναφέρεται περισσότερο ο συγγραφέας για να απαντήσει. Αυτό ισχύει και για την ερώτηση δύο του κειμένου 'A new pet shop'.

Η ερώτηση έξι του κειμένου 'A new pet shop' :

6) Which animal was NOT in the pet shop?

- A. kitten
- B. puppy
- C. fish

Σύμφωνα με το κείμενο η σωστή απάντηση είναι η B. Ο συγγραφέας το κείμενου δεν αναφέρει πουθενά για τα κουταβάκια. Το σύστημα θα απαντήσει στην επιλογή B: possible και στις επιλογές A και C θα δώσει reject. Στο κείμενο δεν υπάρχουν πληροφορίες για την επιλογή B, το σύστημα δεν μπορεί να αποδεχτεί ούτε να απορρίψει την επιλογή και καταλήγει να απαντήσει με την επιλογή possible.

Από την άλλη ο αναγνώστης είναι σε θέση να απορρίψει τις επιλογές A και C γιατί ο συγγραφέας αναφέρει για ψάρια και γατάκια στο κείμενο. Συγκεκριμένα στην τρίτη παράγραφο γράφει για τα ψάρια και στην τέταρτη για τα γατάκια. Ο αναγνώστης διαπιστώνει ότι ο συγγραφέας δεν κάνει αναφορά για κουταβάκια οπότε καταλήγει στο συμπέρασμα ότι η επιλογή B είναι η σωστή. Το σύστημα καταλήγει στο συμπέρασμα ότι μια επιλογή είναι πιθανή, όταν δεν υπάρχουν πληροφορίες είτε από το κείμενο είτε από την κοινή γνώση όπως συμβαίνει στη συγκεκριμένη ερώτηση. Ο αναγνώστης και το σύστημα δεν βρίσκουν πληροφορίες για τα κουταβάκια. Ο αναγνώστης δέχεται ως σωστή την επιλογή B ενώ το σύστημα θεωρεί την επιλογή B ως πιθανή και οι δυο απορρίπτουν τις επιλογές A και C.

Κεφάλαιο 6

Συμπεράσματα

6.1 Συμπεράσματα

73

6.1 Συμπεράσματα

Η διπλωματική εργασία είχε σκοπό την αξιοποίηση των υπολογιστών για την αναπαράσταση ιστοριών και εκπαιδευτικών κειμένων. Να διερευνήσουμε σε ποιο βαθμό ο ηλεκτρονικός υπολογιστής είναι σε θέση να κατανοήσει τα περιεχόμενα των ιστοριών και των εκπαιδευτικών κειμένων.

Στην πρώτη κατηγορία ιστοριών παρατηρούμε ότι η ευελιξία της γλώσσας RAC, η οποία βασίζεται στη θεωρία της επιχειρηματολογίας, μας δίνει τη δυνατότητα να απεικονίσουμε τους χαρακτήρες της ιστορίας και τις δράσεις με χρονολογική σειρά όπως ακριβώς αναφέρονται μέσα στην ιστορία.

Στην κοινή γνώση είναι δυνατό να υπάρχουν προτεραιότητες μεταξύ των επιχειρημάτων. Οι προτεραιότητες τοποθετούνται με βάση το περιεχόμενο της κάθε ιστορίας, δεν είναι σταθερές αλλάζουν σύμφωνα με τις συνθήκες και το περιεχόμενο της ιστορίας. Αφού δημιουργήσουμε τους κανόνες για την ιστορία πρέπει να δηλώσουμε τις προτεραιότητες μεταξύ των κανόνων. Οι προτεραιότητες χρησιμεύουν για την συνέπεια της θεωρίας και για να εξάγει το μοντέλο κατανόησης σωστά συμπεράσματα. Οι προτεραιότητες μπορεί να προκύψουν είτε από τις πληροφορίες που αναφέρονται ρητά μέσα στο κείμενο είτε από την κοινή γνώση που έχουμε για τον κόσμο.

Παρατηρούμε όμως ότι το σύστημα δεν είναι σε θέση να υποστηρίξει προτεραιότητες όπως αυτή που δηλώνεται μέσα στα σχόλια στην παρακάτω ιστορία.

%John wanted an ice-cream. John enjoyed eating ice-cream in the park.

session(s(0),[q(1)],all).

s(0) :: person(john) at always.

s(0) :: wanted(john,ice_cream) at 0.

s(1) :: enjoyed_eating(john,ice_cream) at 2.

q(1) ?? buy(john,ice_cream) at 1;

 make(john,ice_cream) at 1.

fluents([wanted(_,_),enjoyed_eating(_,_),make(_,_)]).

c(11) :: wanted(Person,ice_cream) causes buy(Person,ice_cream).

c(12) :: buy(Person,ice_cream) causes enjoyed_eating(Person,ice_cream).

c(13) :: wanted(Person,ice_cream) causes make(Person,ice_cream).

%c(11)>>c(13) δεν είναι έτοιμο το σύστημα.

p(14) :: buy(Person,ice_cream) implies -make(Person,ice_cream).

p(15) :: make(Person,ice_cream) implies -buy(Person,ice_cream).

p(14)>>c(13).

c(11)>>p(15).

Συνήθως όταν οι άνθρωποι θέλουν παγωτό αγοράζουν ή φτιάχνουν. Η πρόταση c(11) :: wanted(Person,ice_cream) causes buy(Person,ice_cream) έχει μεγαλύτερη προτεραιότητα από την πρόταση c(13) :: wanted(Person,ice_cream) causes make(Person,ice_cream) γιατί γνωρίζουμε ότι από την κοινή γνώση οι περισσότεροι άνθρωποι που θέλουν παγωτό το αγοράζουν και δεν το φτιάχνουν.

Σε ότι αφορά την απεικόνιση του πραγματικού χρόνου, η γλώσσα δεν είναι σε θέση να απεικονίσει τους πραγματικούς χρόνους που συνέβησαν τα γεγονότα στην ιστορία. Αυτός ο περιορισμός δεν αποτελεί πρόβλημα στην αναπαράσταση των ιστοριών. Οι χρονικές στιγμές των δράσεων/ιδιοτήτων αποτυπώνονται στο σύστημα εικονικά.

Σε ερωτήματα όπως π.χ. ‘πόσα χρόνια πέρασαν από τότε που απέκτησε γιο η Μαρία;’ το σύστημα δεν είναι δυνατό να κάνει υπολογισμούς για να δώσει τη σωστή απάντηση όπως κάνει ο άνθρωπος.

Συμπεραίνουμε ότι η γλώσσα που εφαρμόζει το συγκεκριμένο σύστημα μας επιτρέπει να αναπαραστήσουμε τις ιστορίες και τα εκπαιδευτικά κείμενα. Είμαστε σε θέση να αξιοποιήσουμε τους υπολογιστές για την αναπαράσταση ιστοριών και εκπαιδευτικών κειμένων.

Καταλήγουμε ότι σε μεγάλο βαθμό ο ηλεκτρονικός υπολογιστής μπορεί να κατανοήσει τα περιεχόμενα των ιστοριών και των εκπαιδευτικών κειμένων. Τέλος το σύστημα δεν είναι ικανό να απαντήσει σε όλες τις ερωτήσεις των εκπαιδευτικών κειμένων.

Βιβλιογραφία

- [1] Μάνιου-Βακάλη, Μ. (1995), *Μάθηση Μνήμη Λήθη*, Θεσσαλονίκη: Γραφικές Τέχνες
- [2] Πόρποδας, Δ. Κ. (1993), *Η Διαδικασία της Μάθησης*, Αθήνα
- [3] Diakidou, I. A., & Kakas, A., & Michael, L., & Miller R. Narrative Text Comprehension: From Psychology to AI
- [4] Smith, E., & Hancox, P. (2001), Representation, Coherence and Inference, *Artificial Intelligence Review* 15, 295–323.
- [5] englishforeveryone.org *Reading Comprehension Worksheets* Διαθέσιμο από:
<http://englishforeveryone.org/Topics/Reading-Comprehension.htm>
[τελευταία πρόσβαση: 14/05/2015]