

Ατομική Διπλωματική Εργασία

**ΑΛΓΟΡΙΘΜΟΙ ΕΚΜΑΘΗΣΗΣ ΘΕΩΡΙΩΝ
ΕΠΙΧΕΙΡΗΜΑΤΟΛΟΓΙΑΣ**

Έλενα Αρτεμίου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2016

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αλγόριθμοι εκμάθησης θεωριών επιχειρηματολογίας

Έλενα Αρτεμίου

Επιβλέπων Καθηγητής

Γιάννης Δημόπουλος

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

Μάιος 2016

Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, σηματοδοτείται και το τέλος της πορείας μου στο Πανεπιστήμιο Κύπρου, ως προπτυχιακή φοιτήτρια του τμήματος Πληροφορικής και ταυτόχρονα ενός σημαντικού κεφαλαίου τη ζωής μου.

Νιώθω την υποχρέωση να ευχαριστήσω θερμά κάποια άτομα που ήταν κοντά μου αυτά τα τέσσερα χρόνια των σπουδών μου.

Θα ήθελα πρωτίστως να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Γιάννη Δημόπουλο για την συνεχή καθοδήγηση, καθώς και για την πολύτιμη βοήθεια και στήριξη που μου έχει προσφέρει στα πλαίσια εκπλήρωσης της διπλωματικής μου εργασίας.

Επιπλέον, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στην οικογένεια μου που με ενθάρρυναν καθημερινά και με στήριζαν στο αγώνα μου για την εκπλήρωση ενός από τους πιο σημαντικούς και δύσκολους στόχους της ζωής μου. Ακόμη θα ήθελα να τους ευχαριστήσω για την απεριόριστη δύναμη, κατανόηση και συμπαράσταση που μου έχουν προσφέρει όλα αυτά τα χρόνια. Ιδιαίτερες ευχαριστίες θα ήθελα να εκφράσω στους γονείς μου, που μου έχουν μάθει να βάζω στόχους στη ζωή μου και να τους κυνηγώ.

Τέλος θα ήταν παράλειψη μου να μην εκφράσω ένα μεγάλο ευχαριστώ στους φίλους μου που ήταν δίπλα μου καθ' όλη την διάρκεια των σπουδών μου και για την κατανόηση που μου έχουν δείξει.

Ευχαριστώ,

Έλενα Αρτεμίου

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με διάφορες μεθόδους εκμάθησης θεωριών επιχειρηματολογίας. Σε γενικές γραμμές συσχετίζεται με την εύρεση διαφορών αλγορίθμων, οι οποίοι έχουν σαν στόχο την εκμάθηση θεωριών επιχειρηματολογίας, την υλοποίηση αυτών των αλγορίθμων και την εξαγωγή κάποιων συμπερασμάτων μέσα από την εκτέλεση κάποιων πειραμάτων. Για να γίνει όμως αυτό εφικτό έπρεπε να πραγματοποιηθεί η μελέτη διαφορών άρθρων, τα οποία σχετίζονται με την επιχειρηματολογία και η κατανόηση κάποιων βασικών εννοιών οι οποίες χρησιμοποιούνται στη συνέχεια αυτού του εγγράφου.

Στην παρούσα μελέτη παρουσιάζονται τέσσερις αλγόριθμοι. Ο πρώτος αλγόριθμος εντοπίζει όλες τις ευσταθείς επεκτάσεις μιας θεωρίας της επιχειρηματολογίας. Επίσης μεταβάλλοντας τη θεωρία, συγκεκριμένα αφαιρώντας της κάποιο ή κάποια από τα επιχειρήματα της, πάλι ο αλγόριθμος είναι σε θέση να βρει τις καινούργιες ευσταθείς επεκτάσεις που προκύπτουν. Ο δεύτερος αλγόριθμος δοθέντος κάποιων ευσταθών επεκτάσεων βρίσκει μία ελάχιστη θεωρία, η οποία σίγουρα θα έχει σαν ευσταθείς επεκτάσεις της τις επεκτάσεις που δόθηκαν σαν είσοδος στον αλγόριθμο. Με τον όρο ελάχιστη θεωρία αναφερόμαστε στη θεωρία με τις λιγότερες σχέσεις επίθεσης. Οι επόμενοι δύο αλγόριθμοι συγκρίνουν δύο θεωρίες μεταξύ τους και εντοπίζουν κατά πόσο είναι όμοιες, συγκεκριμένα βρίσκουν το πόσο διαφέρουν οι δύο θεωρίες. Η διαφορά των δύο αλγορίθμων εναπόκειται στον τρόπο σύγκρισης των θεωριών.

Αφού προηγήθηκε η μελέτη και ο εντοπισμός των αλγορίθμων προχωρήσαμε στην υλοποίηση τους, η οποία έγινε σε γλώσσα προγραμματισμού C. Για την υλοποίηση χρειάστηκε να κάνουμε χρήση ενός προτασιακού επιλυτή (SAT solver) και ενός προτασιακού επιλυτή που επιλύει προβλήματα σταθμικής προτασιακής βελτιστότητας (MaxSat solver). Συγκεκριμένα χρησιμοποιήθηκαν οι lingeling (SAT solver) και maxino (MaxSat solver).

Τελικό στάδιο της διπλωματικής ήταν η πειραματική αξιολόγηση, όπου έγιναν κάποια πειράματα που αφορούσαν τη διαφορά που υπήρχε μεταξύ των θεωριών. Οι παράγοντες οι οποίοι άλλαζαν ήταν ο αριθμός των επιχειρημάτων που αποτελούσαν τη θεωρία, η

πυκνότητα των θεωριών και το πλήθος των υποσυνόλων που αφαιρούνταν κάθε φορά από τις θεωρίες. Τα υποσύνολα αποτελούνταν από επιχειρήματα.

Καταλήξαμε σε αρκετά συμπεράσματα από την πειραματική αξιολόγηση που έχει γίνει. Το πιο βασικό όμως συμπέρασμα ήταν ότι καθώς αυξάνεται το πλήθος των υποσυνόλων που αφαιρούνται από μία θεωρία, τότε η διαφορά που υπολογίζεται γι' αυτή τη θεωρία παρουσιάζει μείωση.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Σκοπός διπλωματικής εργασίας	1
	1.2 Δομή διπλωματικής εργασίας	1
Κεφάλαιο 2	Επιχειρηματολογία.....	3
	2.1 Τι είναι επιχειρηματολογία	3
	2.2 Αφηρημένο σύστημα επιχειρηματολογίας αλά Dung	6
	2.3 Σημασιολογία με βάση τις ετικέτες	9
	2.4 Σημασιολογία με βάση τις επεκτάσεις	14
Κεφάλαιο 3	Προτασιακή Ικανοποιησιμότητα και αλγόριθμοι εκμάθησης θεωριών επιχειρηματολογίας.....	25
	3.1 Προτασιακή Ικανοποιησιμότητα	25
	3.2 Υπολογισμός ευσταθών επεκτάσεων μιας θεωρίας	27
	3.3 Υπολογισμός ελάχιστης θεωρίας	39
	3.4 Σύγκριση επιχειρηματικών πλαισίων με βάση τις ευσταθείς επεκτάσεις	45
	3.5 Σύγκριση επιχειρηματικών πλαισίων με βάση τις δυαδικές σχέσεις επίθεσης	48
Κεφάλαιο 4	Σύστημα Υλοποίησης και Πειραματική Αξιολόγηση.....	52
	4.1 Επεξήγηση συστήματος υλοποίησης	52
	4.2 Πειραματική αξιολόγηση	57
Κεφάλαιο 5	Συμπεράσματα	76
	5.1 Συμπεράσματα μελέτης	76
	5.2 Μελλοντικές επεκτάσεις	78
	Βιβλιογραφία	79

Π α ρ ά ρ τ η μ α Α..... Α-1

Π α ρ ά ρ τ η μ α Β..... Β-1

Π α ρ ά ρ τ η μ α Γ..... Γ-1

Π α ρ ά ρ τ η μ α Δ..... Δ-1

Π α ρ ά ρ τ η μ α Ε..... Ε-1

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός διπλωματικής εργασίας	1
1.2 Δομή διπλωματικής εργασίας	1

1.1 Σκοπός Διπλωματικής Εργασίας

Στόχος της παρούσας διπλωματικής εργασίας ήταν η μελέτη μεθόδων που στόχευαν στην εκμάθησης θεωριών της επιχειρηματολογίας. Η γνώση η οποία μπορούμε να αντλήσουμε από μία δεδομένη θεωρία μπορεί να συσχετίζεται με διάφορους παράγοντες. Κάποιοι από αυτούς τους παράγοντες μπορεί να είναι οι δυαδικές σχέσεις επίθεσης που υπάρχουν σε αυτήν ή κάποιοι κανόνες που διέπουν ένα σύστημα επιχειρηματολογίας. Συγκεκριμένα οι αλγόριθμοι που έπρεπε να μελετηθούν για την παρούσα διπλωματική στηρίζονται στους κανόνες, οι οποίοι καθορίζουν τις ευσταθείς επεκτάσεις μιας θεωρίας.

1.2 Δομή διπλωματικής εργασίας

Στο κεφάλαιο 2 της παρούσας διπλωματικής εργασίας αναφερόμαστε στο τι ακριβώς είναι η μελέτη της επιχειρηματολογίας. Επιπλέον, παρουσιάζονται κάποιες βασικές έννοιες που σχετίζονται με την επιχειρηματολογία και κάποια επεξηγηματικά παραδείγματα για αυτές τις έννοιες. Ήταν αναγκαία η αναφορά σε αυτές τις έννοιες έτσι ώστε να γίνει κατανοητό το πρόβλημα το οποίο μελετήσαμε.

Στο κεφάλαιο 3, αρχικά γίνεται μια μικρή αναφορά στην έννοια της προτασιακής ικανοποίησης, αφού υπάρχει συσχετισμός της έννοιας αυτής με την επιχειρηματολογία. Ακολούθως παρουσιάζονται οι αλγόριθμοι στους οποίους καταλήξαμε μετά από έρευνα και τους οποίους έχουμε υλοποιήσει. Πιο συγκεκριμένα οι αλγόριθμοι αυτοί είναι: αλγόριθμος εντοπισμού ευσταθών επεκτάσεων δοθείσας μίας θεωρίας της επιχειρηματολογίας, ο αλγόριθμος εύρεσης μίας ελάχιστης θεωρίας της επιχειρηματολογίας δοθέντων κάποιων ευσταθών επεκτάσεων, και δύο διαφορετικοί αλγόριθμοι οι οποίοι συγκρίνουν δύο θεωρίες της επιχειρηματολογίας και εντοπίζουν την διαφορά που υπάρχει μεταξύ των δύο αυτών θεωριών. Κατά την περιγραφή των αλγορίθμων παρουσιάζεται η συσχέτιση που υπάρχει μεταξύ της προτασιακής ικανοποίησης και της επιχειρηματολογίας.

Στο κεφάλαιο 4 γίνεται αναφορά στις λεπτομέρειες υλοποίησης των αλγορίθμων που παρουσιάστηκαν στο κεφάλαιο 3. Επίσης παρουσιάζονται η πειραματική αξιολόγηση του συστήματος υλοποίησης.

Τέλος, στο κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα που πάρθηκαν από τα πειράματα που έχουν πραγματοποιηθεί καθώς και τα συμπεράσματα – αποτελέσματα στα οποία έχουμε καταλήξει. Ακόμη γίνεται αναφορά σε κάποιες μελλοντικές προεκτάσεις που θα μπορούσαν να γίνουν στην παρούσα διπλωματική εργασία.

Κεφάλαιο 2

Επιχειρηματολογία

2.1 Τι είναι επιχειρηματολογία	3
2.2 Αφηρημένο σύστημα επιχειρηματολογίας αλά Dung	6
2.3 Σημασιολογία με βάση τις ετικέτες	9
2.4 Σημασιολογία με βάση τις επεκτάσεις	14

Στην αρχή αυτού του κεφαλαίου θα εισαχθεί η έννοια της επιχειρηματολογίας. Συγκεκριμένα θα ορίσουμε τι είναι η επιχειρηματολογία και θα παρουσιάσουμε ένα παράδειγμα για να γίνει πιο κατανοητή η έννοια αυτή. Ακολούθως θα ορίσουμε τι είναι το αφηρημένο σύστημα επιχειρηματολογίας αλά Dung. Επιπλέον θα επικεντρωθούμε στη σημασιολογία των θεωριών της επιχειρηματολογίας, η οποία μπορεί να γίνει με βάση τις ετικέτες ή τις επεκτάσεις.

2.1 Τι είναι επιχειρηματολογία

Η επιχειρηματολογία είναι η διεπιστημονική μελέτη του πώς οι άνθρωποι μπορούν να καταλήξουν σε συμπεράσματα μέσα από τη λογική σκέψη. Είναι η γλώσσα η οποία μπορεί να επηρεάσει τον δέκτη, αφού κάνοντας χρήση επιχειρημάτων και τεκμηρίων μπορεί να υποστηρίξει μία θέση με απώτερο σκοπό την μεταβολή της γνώμης του αποδέχτη απέναντι στο υπό συζήτηση θέμα. Με άλλα λόγια η επιχειρηματολογία είναι ένα ισχυρό όπλο μέσω του οποίου μπορούμε να μεταδώσουμε την δική μας άποψη.

Βασικό χαρακτηριστικό της επιχειρηματολογίας είναι το «επιχείρημα». Ορίζοντας την έννοια «επιχείρημα» μπορούμε να πούμε ότι είναι οποιοδήποτε μέσο που μπορεί να μας μεταφέρει σε αποδεκτά συμπεράσματα, όπως για παράδειγμα οι προτάσεις και οι

αποδείξεις. Στη φυσική γλώσσα ένα επιχείρημα αποτελείται από τις προϋποθέσεις που είναι μία ή περισσότερες προτάσεις, και το συμπέρασμα που είναι και αυτό μία πρόταση. Τόσο οι προϋποθέσεις όσο και το συμπέρασμα είναι προτάσεις που μπορούν να είναι αληθείς ή ψευδείς, ενώ ολόκληρο το επιχείρημα χαρακτηρίζεται ως έγκυρο ή άκυρο.

Επιπλέον, η επιχειρηματολογία μπορεί να θεωρηθεί και ως μια μορφή τέχνης, ίσως και επιστήμης, η οποία περιλαμβάνει τον εριστικό διάλογο. Με τον όρο «εριστικό διάλογο» αναφερόμαστε στον διάλογο, ο οποίος έχει σαν στόχο την αντιπαράθεση και τη νίκη απέναντι στον αντίπαλο. Όπως αντιλαμβανόμαστε είναι μία μορφή τέχνης μέσω της οποίας οι άνθρωποι μπορούν να προστατεύσουν τις πεποιθήσεις και τα πιστεύω τους.

Ακόμη η επιχειρηματολογία είναι ένα βασικό συστατικό της ανθρώπινης νοημοσύνης. Η ικανότητα του ανθρώπου να ασχολείται με επιχειρήματα και να κάνει σωστούς συλλογισμούς τον βοηθά στην κατανόηση νέων προβλημάτων και στην επίλυση τους, καθώς επίσης και στην αιτιολόγηση και διευκρίνιση των απόψεών του. Αξίζει να σημειωθεί ότι η έννοια της επιχειρηματολογίας εισέρχεται από πολύ νωρίς στα σχολεία, αφού θεωρείται ως βασική έννοια που πρέπει να γνωρίζουν τα παιδιά.

Εκτός από την καθημερινότητα μας, η επιχειρηματολογία εμφανίζεται και σε μία πλειάδα από επαγγέλματα. Ας ρίξουμε μια ματιά σε κάποια από αυτά τα επαγγέλματα. Αρχικά θα εξετάσουμε το επάγγελμα του δικηγόρου. Μπορούμε να πούμε πως η βάση του συγκεκριμένου επαγγέλματος είναι η επιχειρηματολογία. Γενικά ένας δικηγόρος έχει σαν στόχο να αποδείξει την αθωότητα του πελάτη του στον δικαστή. Για την πραγμάτωση αυτού του γεγονότος είναι αναγκαία η κατάλληλη προετοιμασία των επιχειρημάτων που θα υποβληθούν στο δικαστήριο, καθώς και ο έλεγχος της εγκυρότητας ορισμένων αποδεικτικών στοιχείων. Ένα άλλο παράδειγμα επαγγέλματος που σχετίζεται με την επιχειρηματολογία είναι αυτό του πωλητή. Ο πωλητής πρέπει να πείσει τον πελάτη του να προβεί στην αγορά του προϊόντος που πωλεί. Για να επιτύχει αυτό το στόχο, ο πωλητής πρέπει να παρουσιάσει στον πελάτη τα κατάλληλα επιχειρήματα ανάλογα με το είδος του προϊόντος που θέλει να πωλήσει.

Σε αυτό το σημείο, ας δούμε ένα απλό παράδειγμα (παράδειγμα 2.1) επιχειρηματολογίας μεταξύ 3 ατόμων.

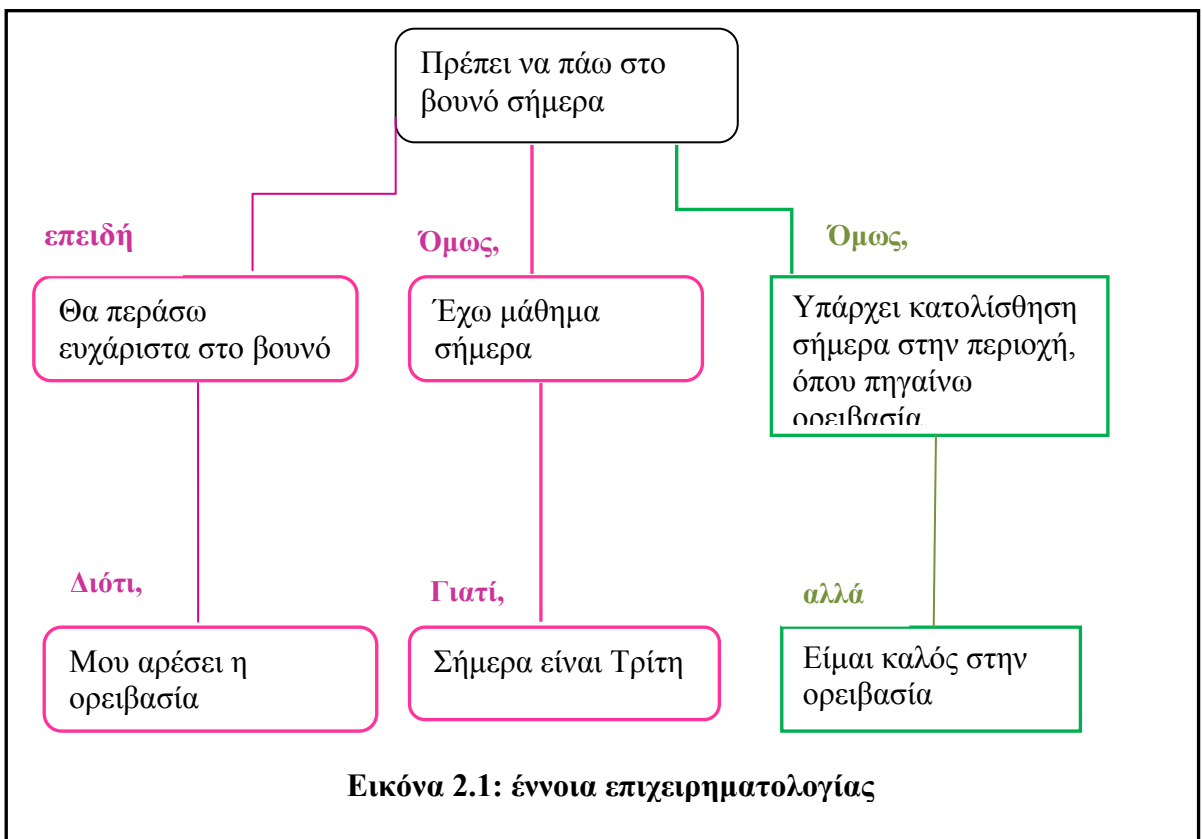
Παράδειγμα κατανόησης 2.1 :

- a: Ο Γιώργος θα πάει με τα πόδια στη δουλειά, γιατί είναι κοντά στο σπίτι του.
- b: Ο Γιώργος δεν θα πάει με τα πόδια στην δουλειά, επειδή βρέχει .
- c: Ο Γιώργος έχει ομπρέλα.

Στο πιο πάνω παράδειγμα έχουμε τρία άτομα, τους a, b και c που επιχειρηματολογούν. Τα άτομα αυτά ονομάζονται πράκτορες (agents) και τα λόγια που λένε είναι τα επιχειρήματα.

Παρατηρούμε ότι ο a υποστηρίζει μία θέση την οποία δεν υποστηρίζει ο b. Συγκεκριμένα ο b υποστηρίζει το αντίθετο από αυτό που υποστηρίζει ο a και γι' αυτό το λόγο λέμε ότι ο b επιτίθεται στον a. Από την άλλη βλέπουμε ότι ο c με το επιχείρημα του στηρίζει τον a λέγοντας ότι ο Γιώργος έχει ομπρέλα, άρα ο Γιώργος μπορεί να πάει στη δουλειά με τα πόδια και έτσι το c επιτίθεται στον b, εννοώντας ότι το ότι βρέχει δεν αποτελεί εμπόδιο στο να πάει με τα πόδια στη δουλειά.

Πιο κάτω παρουσιάζεται μια εικόνα μέσα από την οποία θα προσπαθήσουμε να κάνουμε πιο κατανοητή την έννοια της επιχειρηματολογίας.



Στην πιο πάνω εικόνα (εικόνα 2.1), φαίνεται το αρχικό επιχείρημα το οποίο είναι το «Πρέπει να πάω βουνό σήμερα». Ο πράκτορας ενισχύει το επιχείρημα αυτό λέγοντας ότι πρέπει να πάει βουνό σήμερα, γιατί θα περάσει ευχάριστα. Ακολούθως το επιχείρημα αυτό του πράκτορα ενισχύεται με το επιχείρημα ότι θα περάσει ευχάριστα, διότι του αρέσει η ορειβασία. Με αυτό τον τρόπο το επιχείρημα «Μου αρέσει η ορειβασία», υπερασπίζεται τα δύο άλλα επιχειρήματα που προαναφέρθηκαν.

Από την άλλη πλευρά, όμως ένα άλλο επιχείρημα εναντιώνεται στο αρχικό επιχείρημα λέγοντας ότι δεν πρέπει να πάει σήμερα βουνό, γιατί έχει μάθημα. Μετέπειτα ένα άλλο επιχείρημα υπερασπίζεται το επιχείρημα ότι έχει μάθημα, λέγοντας ότι σήμερα είναι Τρίτη. Με αυτό τον τρόπο το επιχείρημα αυτό καταφέρνει να υποστηρίξει το επιχείρημα «Έχω μάθημα σήμερα» και να εναντιωθεί στο επιχείρημα «Σήμερα πρέπει να πάω βουνό».

Στην τελευταία περίπτωση όπου το επιχείρημα λέει ότι «Υπάρχει κατολίσθηση σήμερα στην περιοχή, όπου πηγαίνω για ορειβασία» εναντιώνεται στο αρχικό επιχείρημα, υποστηρίζοντας ότι δεν πρέπει να πάει στο βουνό σήμερα. Από την άλλη, το επιχείρημα «είμαι καλός στην ορειβασία», υπερασπίζεται το αρχικό επιχείρημα και εναντιώνεται στο επιχείρημα «Υπάρχει κατολίσθηση σήμερα στην περιοχή, όπου πηγαίνω για ορειβασία».

Στο πιο πάνω παράδειγμα κάθε πράκτορας εκφράζει την δική του άποψη μέσα από επιχειρήματα τα οποία παρουσιάζει και προσπαθεί να πείσει τους υπόλοιπους πράκτορες για το αν πρέπει να πάει ή όχι στο βουνό.

2.2 Αφηρημένο σύστημα επιχειρηματολογίας αλά Dung.

Η έννοια του αφηρημένου συστήματος της επιχειρηματολογίας ή αλλιώς το επιχειρηματικό πλαίσιο (Argumentation Framework, AF) εισάχθηκε από τον επιστήμονα Phan Minh Dung[1]. Σύμφωνα με τον Dung λοιπόν, ένα σύστημα επιχειρηματολογίας ορίζεται ως ένα ζεύγος $\langle A, R \rangle$ όπου A είναι πεπερασμένο ή μη-πεπερασμένο σύνολο επιχειρημάτων και R το σύνολο δυαδικών σχέσεων επίθεσης

πάνω στο σύνολο A όπου $R \subseteq A \times A$. Δηλαδή λέμε ότι ο a επιτίθεται στο επιχείρημα b , αν και μόνο αν $(a,b) \in R$.

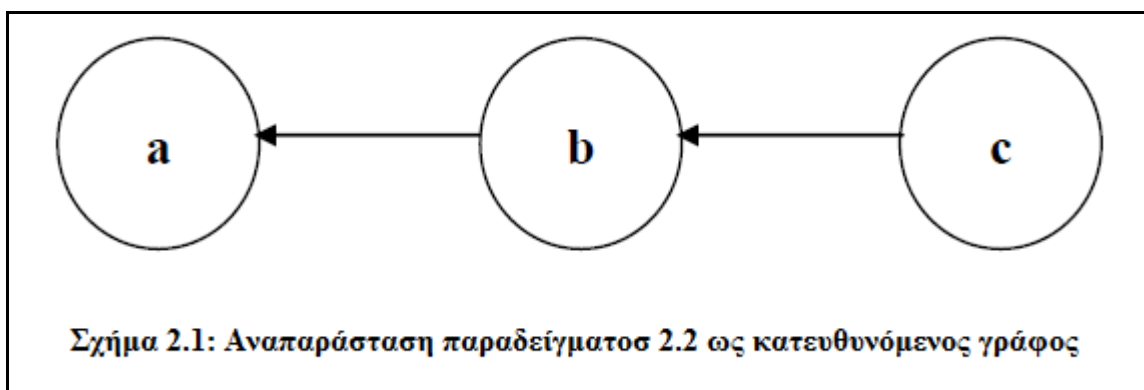
Το σύστημα επιχειρηματολογίας μπορεί να αναπαρασταθεί υπό την μορφή ενός κατευθυνόμενου γράφου, όπου οι κόμβοι του θα αντιπροσωπεύουν τα επιχειρήματα και οι ακμές τις σχέσεις επίθεσης μεταξύ των επιχειρημάτων. Η κατεύθυνση της ακμής είναι αυτή που καθορίζει ποιο επιχείρημα κάνει την επίθεση και ποιο την δέχεται. Συγκεκριμένα η ακμή ξεκινά από το επιχείρημα – κόμβο που κάνει την επίθεση και καταλήγει στο επιχείρημα- κόμβο που δέχεται την επίθεση. Η αναπαράσταση αυτή του συστήματος επιχειρηματολογίας σε γράφο λέμε ότι αποτελεί μια θεωρία.

Ας χρησιμοποιήσουμε το παράδειγμα 2.1 που αναφερθήκαμε προηγουμένως, και να προσπαθήσουμε να κάνουμε την αναπαράσταση με ένα κατευθυνόμενο γράφο.

Παράδειγμα κατανόησης 2.2:

- a: Ο Γιώργος θα πάει με τα πόδια στη δουλειά, γιατί είναι κοντά στο σπίτι του.
- b: Ο Γιώργος δεν θα πάει με τα πόδια στην δουλειά, επειδή βρέχει .
- c: Ο Γιώργος έχει ομπρέλα.

Στο παράδειγμα 2.2 έχουμε τρεις πράκτορες, όπου ο κάθε ένας από αυτούς έχει το δικό του επιχείρημα, άρα ο γράφος μας θα έχει τρεις κόμβους και δύο ακμές, αφού όπως αναφέραμε προηγουμένως ο b επιτίθεται στον a και ο c επιτίθεται στον b .



Το αφηρημένο σύστημα επιχειρηματολογίας που αντιστοιχεί στο παράδειγμα 2.2 είναι το εξής:

$$AF = \langle \underbrace{\{a, b, c\}}_A, \underbrace{\{(b, a), (c, b)\}}_R \rangle$$

A: Σύνολο επιχειρημάτων, R: Δυναδικές σχέσεις επίθεσης

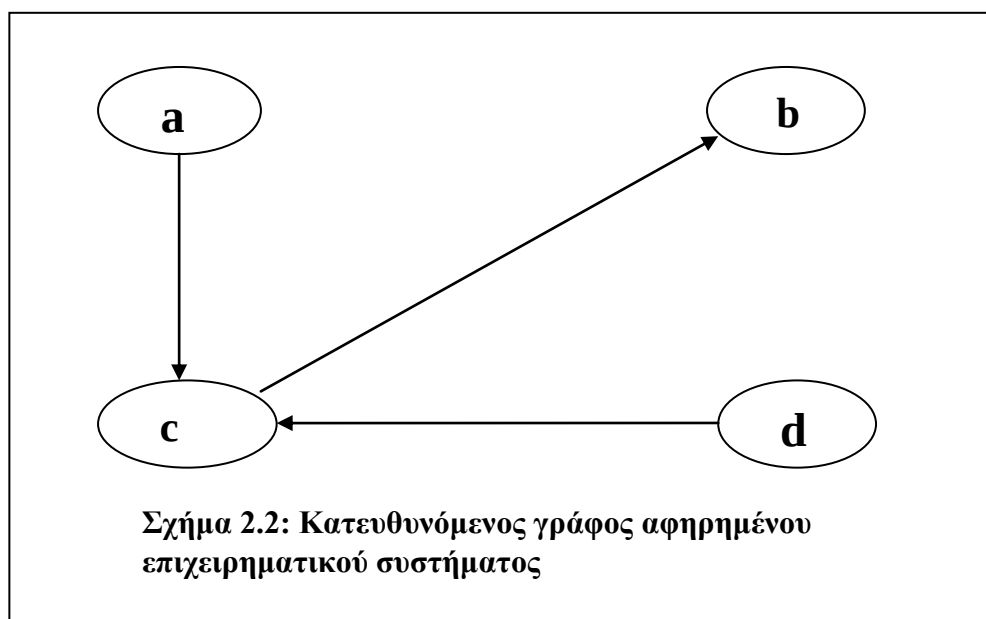
Τώρα ας δούμε ένα παράδειγμα (παράδειγμα 2.3) στο οποίο δίνεται το αφηρημένο σύστημα επιχειρηματολογίας και πρέπει να δημιουργηθεί ο κατευθυνόμενος γράφος.

Παράδειγμα κατανόησης 2.3:

Έστω ότι : $AF = \langle \{a, b, c, d\}, \{(a, c), (d, c), (c, b)\} \rangle$

Παρατηρώντας το πιο πάνω AF (αφηρημένο σύστημα επιχειρηματολογίας) αντιλαμβανόμαστε ότι ο κατευθυνόμενος γράφος που θα προκύψει θα αποτελείται από τέσσερις κόμβους, όσα είναι δηλαδή και τα επιχειρήματα., καθώς επίσης και από τρεις κατευθυνόμενες ακμές, όσες δηλαδή είναι και οι σχέσεις επιθέσεις μεταξύ των επιχειρημάτων.

Ο κατευθυνόμενος γράφος ο οποίος αντιστοιχεί στο συγκεκριμένο σύστημα επιχειρηματολογίας παρουσιάζεται στο σχήμα 2.2.



Κοιτάζοντας τα πιο πάνω παραδείγματα μας δημιουργείται το ερώτημα κατά πόσο ποια επιχειρήματα είναι αποδεχτά για τις συγκεκριμένες θεωρίες. Με τον όρο αποδεχτό (acceptable) επιχείρημα αναφερόμαστε στο επιχείρημα που μπορεί να ισχύει ακόμα και μετά από την επίθεση ή τις επιθέσεις που τυχόν να δέχεται. Η διαδικασία αξιολόγησης των επιχειρημάτων, δηλαδή η διαδικασία που θα αποφασίσει αν ένα επιχείρημα είναι αποδεχτό ή όχι ονομάζεται σημασιολογία της αφηρημένης επιχειρηματολογίας και είναι απαραίτητη γιατί σε ένα σύστημα επιχειρηματολογίας δεν μπορούν να έχουν όλα τα επιχειρήματα υπόσταση, δηλαδή δεν μπορούν όλα τα επιχειρήματα να είναι ευσταθή λόγω της δυνατότητας που έχουν το ένα να επιτίθεται στο άλλο [2].

Υπάρχουν δύο τρόποι με τους οποίους μπορούμε να αξιολογήσουμε τα επιχειρήματα και να καθορίσουμε κατά πόσα αυτά είναι αποδεχτά ή όχι. Οι δύο αυτοί τρόποι είναι: η σημασιολογία με βάση τις ετικέτες (labeling-based) και η σημασιολογία με βάση τις επεκτάσεις (extensions-based). Ακολουθεί ανάλυση των δύο αυτών τρόπων στα αμέσως δύο επόμενα υποκεφάλαια.

2.3 Σημασιολογία με βάση τις ετικέτες

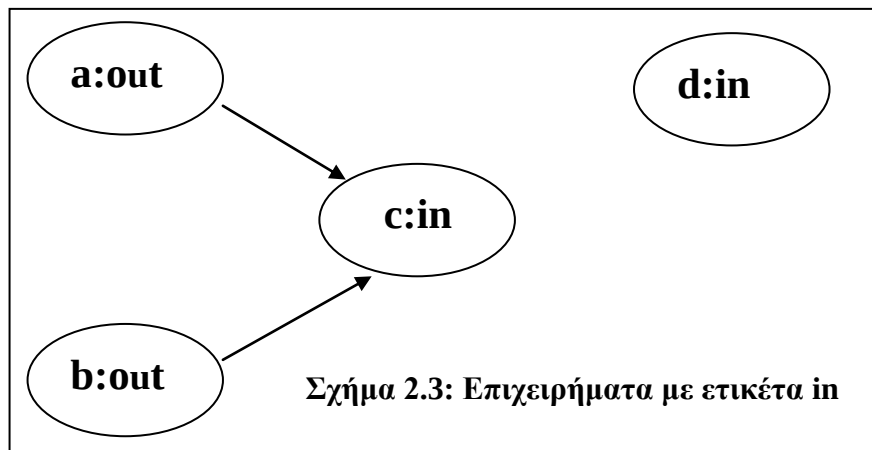
Η ετικετοποίηση δεν είναι μια και μοναδική σε κάθε θεωρία, αλλά σίγουρα υπάρχει μία. Υπάρχουν τρεις πιθανές ετικέτες που μπορούν να ανατεθούν σε κάποιο επιχείρημα. Οι ετικέτες αυτές είναι οι ακόλουθες: in, out και undec (undecided) [3].

Στο σημείο αυτό θα εξετάσουμε τους κανόνες βάση των οποίων γίνεται η ανάθεση των ετικετών στα επιχειρήματα.

Ετικέτα in:

Ένα επιχείρημα παίρνει την ετικέτα in αν μπορεί να «απαντήσει» στις επιθέσεις που δέχεται, δηλαδή αν και μόνο αν όλοι όσοι του επιτίθενται (defeaters) έχουν ετικέτα out (βλέπε σχήμα 2.3).

Ακόμη ένα επιχείρημα μπορεί να πάρει την ετικέτα in όταν κανείς δεν του επιτίθεται (βλέπε σχήμα 2.3).

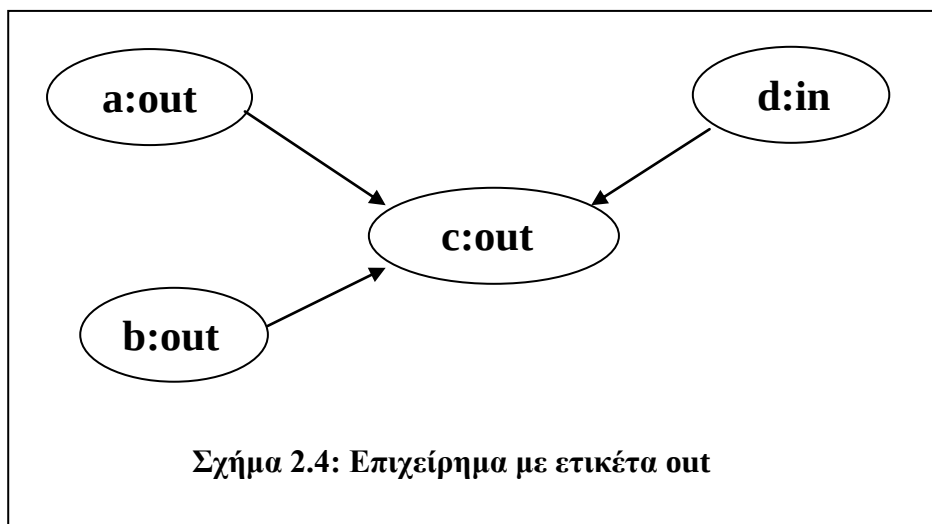


Το επιχείρημα c παίρνει την ετικέτα in γιατί τα επιχειρήματα a και b που του επιτίθενται έχουν την ετικέτα out.

Επίσης το επιχείρημα d παίρνει και αυτό την ετικέτα in, αφού δεν δέχεται επίθεση από κανένα άλλο επιχείρημα.

Ετικέτα out:

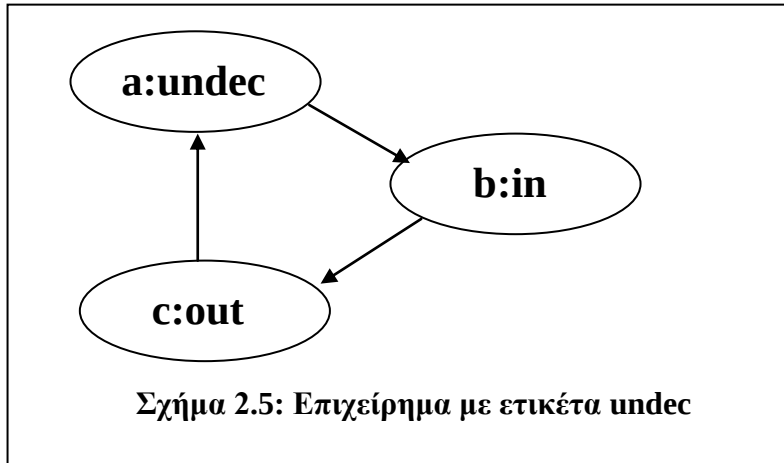
Ένα επιχείρημα παίρνει ετικέτα out εάν και μόνο εάν τουλάχιστον ένας από αυτούς που του επιτίθενται (defeaters) έχει ετικέτα in (βλέπε σχήμα 2.4).



Το επιχείρημα c παίρνει ετικέτα out, αφού δέχεται επίθεση από τουλάχιστον ένα επιχείρημα που έχει ετικέτα out. Συγκεκριμένα το επιχείρημα c δέχεται επίθεση από το επιχείρημα d.

Ετικέτα undec:

Το επιχείρημα παίρνει την ετικέτα undec (αναποφάσιστο) αν δεν υπάρχει κάποια θέση για το αν είναι αποδεκτό ή όχι. (βλέπε σχήμα 2.5).



Το επιχείρημα a παίρνει την ετικέτα undec, γιατί δεν ξέρουμε ποια ετικέτα πρέπει να πάρει. Έστω ότι το επιχείρημα a παίρνει την ετικέτα in. Τότε πρέπει όλα τα επιχειρήματα που του επιτίθενται να έχουν την ετικέτα out. Μόνο το επιχείρημα c επιτίθεται στο a, άρα το c έχει την ετικέτα out. Όμως ως συνέπεια αυτού το c πρέπει να δέχεται επίθεση από τουλάχιστον ένα επιχείρημα που να έχει ετικέτα in. Και αφού το c δέχεται επίθεση μόνο από το b, τότε το b πρέπει να έχει ετικέτα in. Αφού το b έχει ετικέτα in, πρέπει όλα τα επιχειρήματα που του επιτίθενται να έχουν ετικέτα out. Το b δέχεται επίθεση μόνο από το a, άρα το a πρέπει να πάρει την ετικέτα out. Καταλήγουμε σε αντίφαση, αφού υποθέσαμε ότι το a έχει ετικέτα in. Αφού δεν μπορούμε να προσδιορίσουμε αν το επιχείρημα a θα πάρει in ή out, τότε του αναθέτουμε την ετικέτα undec.

Ανεπίσημα μπορούμε να πούμε ότι όταν ένα επιχείρημα έχει πάρει την ετικέτα in τότε σημαίνει ότι κάποιος έχει αποδεχτεί το επιχείρημα, ενώ όταν ένα επιχείρημα έχει πάρει την ετικέτα out σημαίνει ότι κάποιος το έχει απορρίψει. Επιχείρημα με την ετικέτα undec σημαίνει ότι υπάρχει κάποια έλλειψη για να αποφασίσουμε, αν το επιχείρημα απορρίπτεται ή αν γίνεται αποδεκτό.

Ένας πιο αυστηρός ορισμός για την ετικοποίηση είναι ο ακόλουθος:

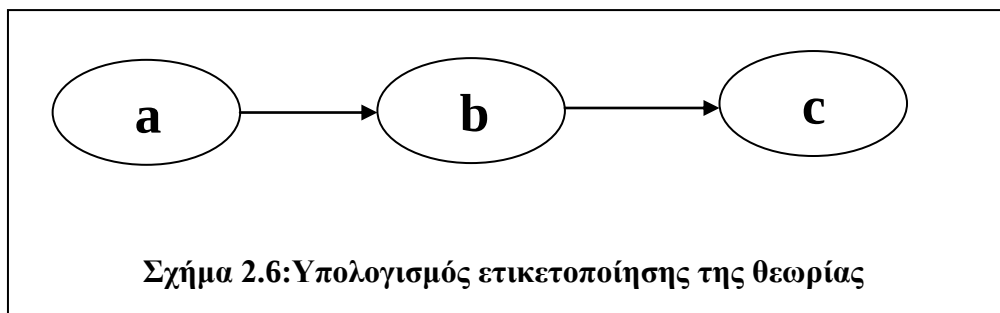
Έστω αφηρημένο σύστημα επιχειρηματολογίας $AF=(A,R)$ (όπου A είναι τα επιχειρήματα του συστήματος και R οι σχέσεις επίθεσης) και η συνάρτηση ανάθεσης ετικετών $Lab: A \rightarrow \{in,out,undec\}$. Τα επιχειρήματα παίρνουν ετικέτες σύμφωνα με τους πιο κάτω κανόνες:

- $\forall a \in A: (Lab(a) = out \equiv \exists b \in A : (b \text{ def } a \wedge Lab(b) = in))$
- $\forall a \in A : (Lab(a) = in \equiv \forall b \in A : (b \text{ def } a \supset Lab(b) = out))$

Ακολουθούν κάποια παραδείγματα με θεωρίες στις οποίες γίνεται ετικετοποίηση στα επιχειρήματα.

Παράδειγμα κατανόησης 2.4:

Έστω ότι μας δίνεται η θεωρία $AF=<\{a,b,c\},\{(a,b),(b,c)\}>$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα.



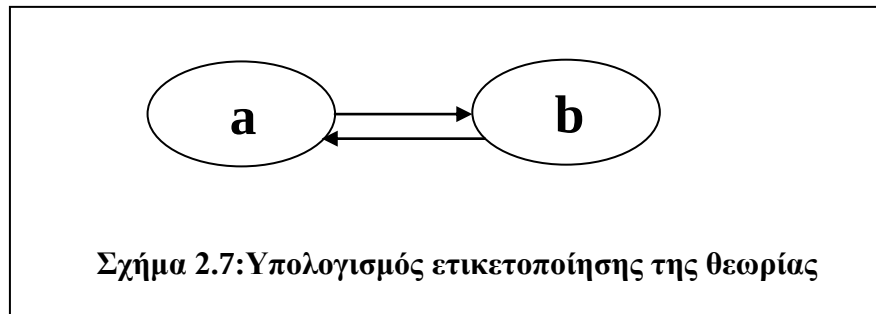
Ετικετοποίηση: $Lab(a) = in, Lab(b) = out, Lab(c) = in$

Επεξήγηση ετικετοποίησης παραδείγματος κατανόησης 2.4:

Ο τρόπος με τον οποίο προκύπτει αυτή η ετικετοποίηση είναι ο εξής: το επιχείρημα a δεν δέχεται επίθεση από κάποιο επιχείρημα, άρα θα πάρει ετικέτα in . Και αφού το επιχείρημα b δέχεται επίθεση μόνο από το επιχείρημα a , τότε το b θα έχει για ετικέτα out . Ακόμη το c και δέχεται επίθεση μόνο από το b επιχείρημα, τότε το c θα έχει ετικέτα in .

Παράδειγμα κατανόησης 2.5:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a, b\}, \{(a, b), (b, a)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα.



Ετικετοποίηση:

Η θεωρία αυτή έχει περισσότερες από μία ετικετοποιήσεις.

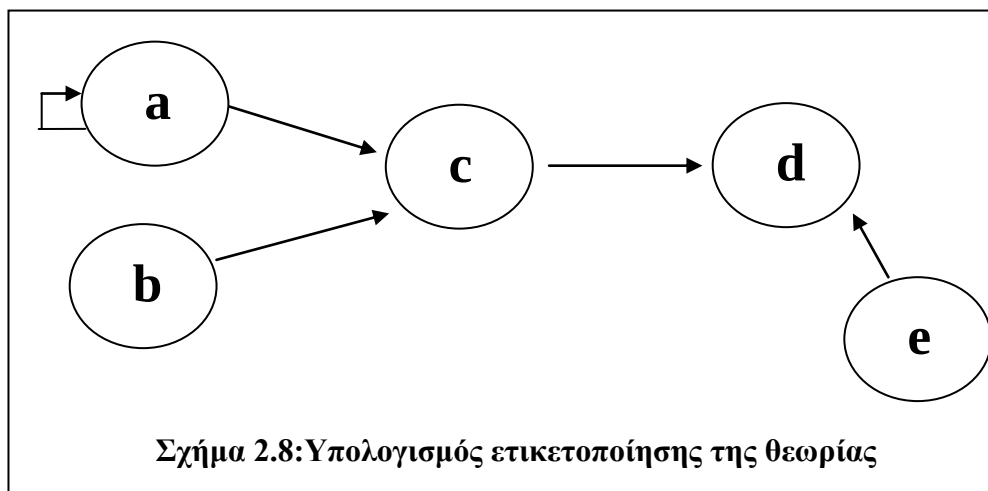
- Ετικετοποίηση 1: $Lab(a) = in, Lab(b) = out$
- Ετικετοποίηση 2: $Lab(a) = out, Lab(b) = in$
- Ετικετοποίηση 3: $Lab(a) = undec, Lab(b) = undec$

Επεξήγηση ετικετοποίησης παραδείγματος κατανόησης 2.5:

Παρατηρούμε πως για το πιο πάνω παράδειγμα υπάρχει περισσότερη από μία ετικετοποίηση. Η ετικέτα που θα δώσουμε στο επιχείρημα a (πρώτο επιχείρημα) θα καθορίσει την ετικέτα που θα δώσουμε στο επιχείρημα b . Με άλλα λόγια βάση της ετικέτας αυτής θα καθοριστεί το τελικό αποτέλεσμα. Και πάλι η ανάθεση ετικετών στα επιχειρήματα θα γίνει με τους κανόνες που περιγράφηκαν σε αυτό το υποκεφάλαιο.

Παράδειγμα κατανόησης 2.6:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a, b, c, d, e\}, \{(a, a), (a, c), (b, c), (c, d), (e, d)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα.



Ετικετοποίηση: $\text{Lab}(a) = \text{undec}$, $\text{Lab}(b) = \text{in}$, $\text{Lab}(c) = \text{out}$, $\text{Lab}(d) = \text{out}$, $\text{Lab}(e) = \text{in}$

Επεξήγηση ετικετοποίησης παραδείγματος κατανόησης 2.6:

Τα επιχειρήματα b και c παίρνουν ετικέτα in , αφού δεν δέχονται από κάποιο άλλο επιχείρημα επίθεση. Όσον αφορά το επιχείρημα a δεν μπορούμε να προσδιορίσουμε αν είναι αποδεχτό ή όχι. Αν το κοιτάξουμε πιο βαθιά θα παρατηρήσουμε το εξής: έστω ότι το επιχείρημα a έχει ετικέτα in . Τότε θα πρέπει να δέχεται επίθεση από κάποιο άλλο επιχείρημα που είναι out . Όπως βλέπουμε όμως το a δεν δέχεται επίθεση από κάποιο άλλο επιχείρημα, αλλά επιτίθεται στο ίδιο του τον εαυτό. Από αυτό αντιλαμβανόμαστε ότι τα a πρέπει να έχει ετικέτα out και έτσι οδηγούμαστε σε αντίφαση με την αρχική υπόθεση που κάναμε. Με την ίδια λογική μπορούμε να δούμε πως πάλι θα οδηγηθούμε σε αντίφαση αν υποθέσουμε ότι το επιχείρημα a έχει ετικέτα out . Αν κοιτάξουμε το επιχείρημα c , θα δούμε ότι δέχεται επίθεση από κάποιο επιχείρημα με ετικέτα in και άρα το c θα έχει ετικέτα out . Τέλος, το επιχείρημα d παίρνει ετικέτα out , γιατί δέχεται επίθεση από το επιχείρημα e που έχει ετικέτα in .

2.4 Σημασιολογία με βάση τις επεκτάσεις

Η σημασιολογία με βάση τις επεκτάσεις ορίζει το σύνολο των επεκτάσεων από το σύστημα επιχειρηματολογίας $\langle A, R \rangle$ όπου A είναι τα επιχειρήματα και R οι σχέσεις επίθεσης, το οποίο αντιπροσωπεύει ένα σύνολο επιχειρημάτων που ανήκουν στο A και μπορούν να επιβιώσουν από τις επιθέσεις άλλων επιχειρημάτων. Ένα επιχείρημα χαρακτηρίζεται ως αιτιολογημένο, αν ανήκει στο σύνολο αυτών των επεκτάσεων.

Αυστηρός ορισμός:

Δεδομένου ενός επιχειρηματικού πλαισίου AF και μιας σημασιολογίας S , μπορούμε να αιτιολογήσουμε ένα επιχείρημα a σε ένα επιχειρηματικό πλαίσιο με δύο τρόπους. Οι τρόποι αυτοί είναι οι ακόλουθοι [4]:

1. Σκεπτικά αιτιολογούμενο (Skeptically justified):

Ένα επιχείρημα ανήκει σε αυτή την κατηγορία αν ανήκει σε κάθε επέκταση του συνόλου των επεκτάσεων του επιχειρηματικού πλαισίου.

Άρα ένα επιχείρημα a είναι σκεπτικά αιτιολογούμενο αν και μόνο αν $\forall E \in Es$ $(AF) a \in E$.

2. Εύπιστα αιτιολογούμενο (Credulously justified):

Ένα επιχείρημα ανήκει σε αυτή την κατηγορία αν ανήκει σε κάποια επέκταση του συνόλου των επεκτάσεων του επιχειρηματικού πλαισίου.

Άρα ένα επιχείρημα a είναι εύπιστα αιτιολογούμενο αν και μόνο αν $\exists E \in Es$ $(AF) a \in E$.

Στο σημείο αυτό θα αναφερθούμε και θα εξηγήσουμε κάποιες βασικούς ορισμούς – αρχές που διέπουν την σημασιολογία βασισμένη σε επεκτάσεις [2].

• Απουσία συγκρούσεων (Conflict free) [5] :

Ένα σύνολο επιχειρημάτων S έχει την ιδιότητα απουσίας συγκρούσεων αν και μόνο αν στο σύνολο δεν περιέχονται επιχειρήματα στα οποία να υπάρχει οποιαδήποτε σχέση επίθεσης μεταξύ τους. Δηλαδή αν το επιχείρημα a επιτίθεται στο b ή το b επιτίθεται στο a τα επιχειρήματα a και b δεν μπορούν να υπάρχουν στο σύνολο.

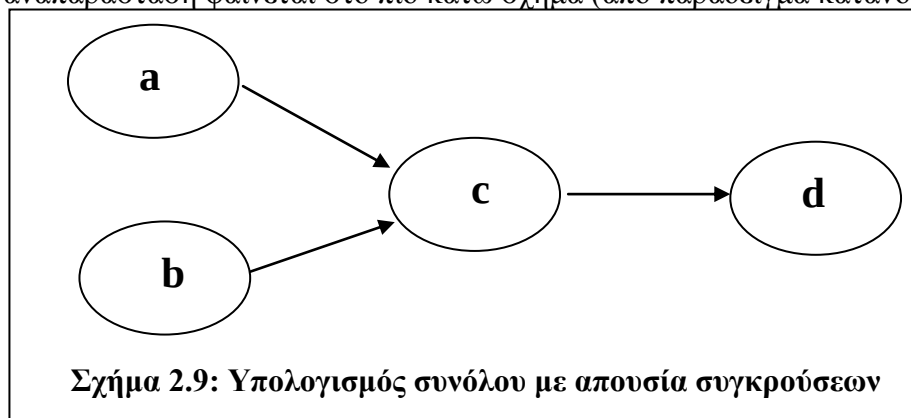
Αυστηρός ορισμός:

Έστω η θεωρία $AF = \langle A, R \rangle$, το σύνολο $S \subseteq A$ έχει την ιδιότητα απουσίας συγκρούσεων αν και μόνο αν $\nexists a, b \in S$ και $(a, b) \in R$.

Ένα σύνολο ενός επιχειρηματικού πλαισίου που υποστηρίζει αυτή την αρχή συμβολίζεται με cf .

Παράδειγμα κατανόησης 2.7:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a, b, c, d\}, \{(a, b), (b, c), (c, d)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα (από παράδειγμα κατανόησης 2.3).



Θα πάρουμε δύο τυχαία σύνολα από το σύνολο των επιχειρημάτων και θα αποφασίσουμε αν ικανοποιούν την ιδιότητα απουσίας συγκρούσεων. Έστω τα σύνολα $\{c,d\}$ και $\{a,b,d\}$.

➤ Σύνολο $\{c,d\}$

Το σύνολο επιχειρημάτων $\{c,d\}$ δεν ικανοποιεί την ιδιότητα απουσίας συγκρούσεων, γιατί όπως βλέπουμε από το σύστημα επιχειρηματολογίας αλλά και από τον γράφο, υπάρχει σχέση επίθεσης μεταξύ των επιχειρημάτων c και d . Άρα δεν μπορούν να ανήκουν και τα δύο επιχειρήματα σε ένα σύνολο που θα έχει την ιδιότητα απουσίας συγκρούσεων.

➤ Σύνολο $\{a,b,d\}$

Το σύνολο επιχειρημάτων $\{a,b,d\}$ ικανοποιεί την ιδιότητα απουσίας συγκρούσεων, γιατί όπως παρατηρούμε από τη θεωρία δεν υπάρχει καμία σχέση επίθεσης μεταξύ των επιχειρημάτων a , b και d .

Παράδειγμα κατανόησης 2.8:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a\}, \{(a,a)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα.



Σχήμα 2.10: Υπολογισμός συνόλου με απουσία συγκρούσεων

Σε αυτή την περίπτωση βλέπουμε ότι έχουμε μόνο ένα επιχειρήμα και μία σχέση επίθεσης του επιχειρήματος στον εαυτό του. Σε αυτή την περίπτωση το σύνολο $\{a\}$ δεν ικανοποιεί την ιδιότητα απουσίας συγκρούσεων.

- Αποδεχτό επιχείρημα (Acceptable argument)

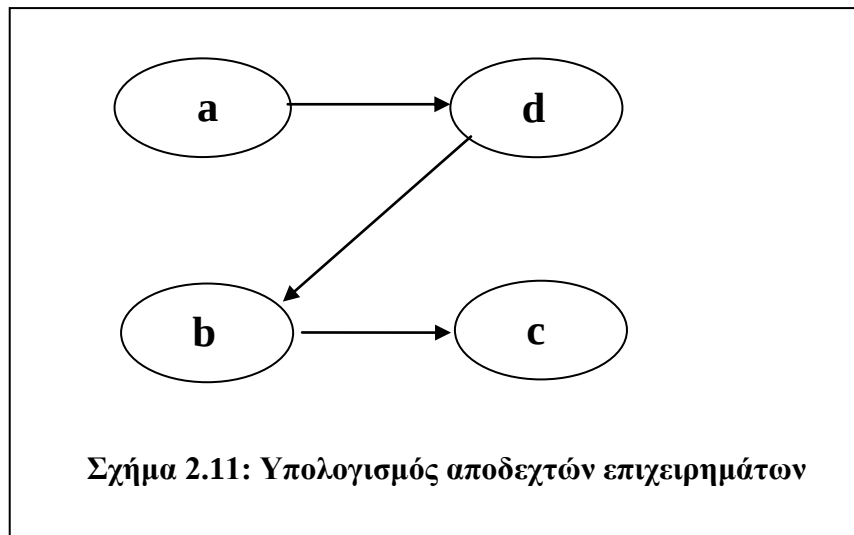
Έστω ένα σύνολο επιχειρημάτων S και ένα σύνολο επιχειρημάτων A όπως το ορίσαμε στο αφηρημένο σύστημα επιχειρηματολογίας αλά Dung. Και έστω ένα επιχείρημα a που ανήκει στο σύνολο S . Το επιχείρημα a είναι αποδεκτό αν και μόνο αν κάθε επιχείρημα $b \in A$: αν το b επιτίθεται στο a , τότε το b δέχεται επίθεση από κάποιο επιχείρημα του S . Δηλαδή το επιχείρημα a δέχεται υπεράσπιση.

Αυστηρός ορισμός:

Έστω η θεωρία $AF = \langle A, R \rangle$, το επιχείρημα $a \in S$ είναι αποδεκτό στο σύνολο S όπου $S \subseteq A$, αν και μόνο αν $\forall b \in A$ όπου $(b, a) \in R$, τότε $(c, b) \in R$ και $c \in S$. Η συνάρτηση που δεδομένου ενός συνόλου $S \subseteq A$ επιστρέφει το σύνολο των αποδεχτών επιχειρημάτων λέγεται χαρακτηριστική συνάρτηση του πλαισίου επιχειρηματολογίας και συμβολίζεται με F_{AF} .

Παράδειγμα κατανόησης 2.9:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a, b, c, d\}, \{(a, d), (d, b), (b, c)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα (βλέπε σχήμα 2.10).



Θα πάρουμε ένα τυχαίο σύνολο S από το σύνολο των επιχειρημάτων και θα αποφασίσουμε αν τα επιχειρήματα που ανήκουν στο σύνολο S είναι αποδεκτά ως προς το S .

Έστω το σύνολο $S=\{d,c\}$. Θα εξετάσουμε ξεχωριστά τα δύο επιχειρήματα d και c που ανήκουν στο S .

➤ Επιχείρημα c:

Το επιχείρημα c δέχεται επίθεση από το επιχείρημα b το οποίο δεν ανήκει στο σύνολο S . Για να είναι αποδεχτό το επιχείρημα c θα πρέπει κάποιο άλλο επιχείρημα που ανήκει στο σύνολο S να κάνει επίθεση στο b . Παρατηρούμε πως αυτή η συνθήκη ικανοποιείται, αφού υπάρχει η σχέση επίθεσης (d,c) στην θεωρία. Άρα το επιχείρημα c είναι αποδεχτό στο σύνολο S .

➤ Επιχείρημα d:

Το επιχείρημα d δέχεται επίθεση από το επιχείρημα a το οποίο δεν ανήκει στο σύνολο S . Για να είναι αποδεχτό το επιχείρημα d θα πρέπει κάποιο άλλο επιχείρημα που ανήκει στο σύνολο S να κάνει επίθεση στο a . Παρατηρούμε πως αυτή η συνθήκη δεν ικανοποιείται, αφού δεν υπάρχει επιχείρημα που να επιτίθεται στο επιχείρημα a . Άρα το επιχείρημα d δεν είναι αποδεχτό στο σύνολο S .

- Παραδεχτό σύνολο (Admissible set) :

Ένα σύνολο επιχειρημάτων S είναι παραδεκτό αν και μόνο αν ικανοποιεί την ιδιότητα απουσίας συγκρούσεων (δηλαδή δεν υπάρχουν συγκρούσεις στο σύνολο) και κάθε επιχείρημα που ανήκει στο S είναι αποδεκτό σε σχέση με το S (δηλαδή το σύνολο υπερασπίζεται όλα του τα επιχειρήματα). Το κενό σύνολο είναι παραδεκτό για κάθε σύστημα επιχειρηματολογίας γιατί είναι απαλλαγμένο από συγκρούσεις και εξ ορισμού υπερασπίζεται όλα του τα επιχειρήματα, που στην ουσία δεν υπάρχουν.

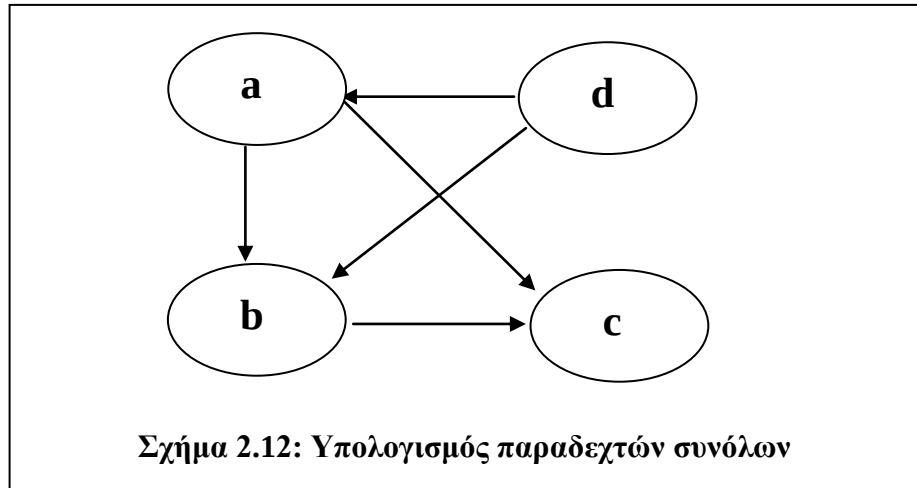
Αυστηρός ορισμός:

Έστω η θεωρία $AF=<A,R>$, το σύνολο $S\subseteq A$ είναι παραδεκτό αν και μόνο αν το S έχει

την ιδιότητα απουσίας συγκρούσεων και $\forall a \in S$ το a είναι αποδεκτό. Το σύνολο όλων των παραδεχτών συνόλων ενός επιχειρηματικού πλαισίου AF έχει τον εξής συμβολισμό: $AS(AF)$.

Παράδειγμα κατανόησης 2.10:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a,b,c,d\}, \{(a,b), (b,c), (a,c), (d,a), (d,b)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα.



Θα πάρουμε δύο τυχαία σύνολα από το σύνολο των επιχειρημάτων και θα αποφασίσουμε αν είναι παραδεκτά σύνολα. Έστω τα σύνολα $\{a,b\}$ και $\{d,c\}$.

➤ Σύνολο $\{a,b\}$:

Δεν είναι παραδεκτό σύνολο, γιατί υπάρχει σχέση επίθεσης μεταξύ των επιχειρημάτων a και b και συνεπώς δεν ικανοποιείται η ιδιότητα απουσίας συγκρούσεων. Ακόμη το επιχείρημα b δεν είναι αποδεκτό στο σύνολο $\{a,b\}$ αφού δέχεται επίθεση από το επιχείρημα d και δεν υπάρχει κάποιο επιχείρημα στο σύνολο $\{a,b\}$ το οποίο να επιτίθεται στο d . Το ίδιο ισχύει και για το επιχείρημα a , αφού δέχεται επίθεση από το επιχείρημα d και δεν υπάρχει κάποιο επιχείρημα στο σύνολο $\{a,b\}$ το οποίο να επιτίθεται στο d .

➤ Σύνολο $\{c,d\}$:

Είναι παραδεκτό σύνολο, γιατί ικανοποιείται η συνθήκη απουσίας συγκρούσεων αφού δεν υπάρχουν συγκρούσεις μεταξύ των c και d . και τα επιχειρήματα c και d είναι αποδεκτά στο σύνολο $\{c,d\}$, αφού το επιχείρημα c δέχεται επίθεση από το επιχείρημα b αλλά το επιχείρημα d κάνει επίθεση στο επιχείρημα b .

Ακολουθώς θα επικεντρωθούμε στις τέσσερις βασικές επεκτάσεις που είναι οι εξής:

1. Πλήρης Επέκταση (Complete Extensions)
2. Προτιμώμενη Επέκταση (Preferred Extensions)
3. Ευσταθής Επέκταση (Stable Extensions)
4. Στηριγμένη Επέκταση (Grounded Extensions)

1.Πλήρης Επέκταση (Complete Extensions) [2]:

Ένα σύνολο επιχειρημάτων S ονομάζεται πλήρης επέκταση αν και μόνο αν είναι παραδεκτό σύνολο και κάθε επιχείρημα που είναι αποδεκτό ανήκει στο S . Δηλαδή με τον όρο πλήρης επέκταση αναφερόμαστε σ'ένα σύνολο που μπορεί να υπερασπιστεί τον εαυτό του και που περιλαμβάνει όλα τα επιχειρήματα που υποστηρίζει.

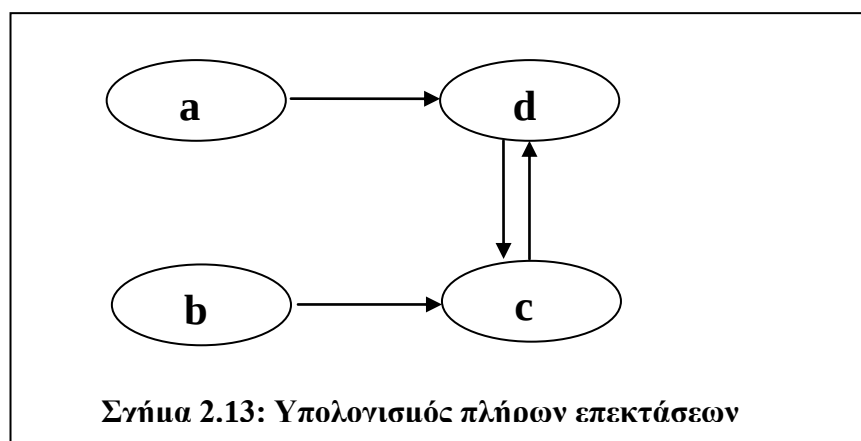
Ο συμβολισμός της πλήρης επέκτασης είναι ο ακόλουθος: E_{co} .

Αυστηρός ορισμός είναι ο εξής:

Έστω η θεωρία $AF = \langle A, R \rangle$, το σύνολο $S \subseteq A$ είναι πλήρης επέκταση αν και μόνο αν το S είναι παραδεκτό σύνολο και $\forall b \in A$ όπου b είναι αποδεκτό επιχείρημα, τότε το $b \in S$.

Παράδειγμα Κατανόησης 2.11:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a, b, c, d\}, \{(a, d), (b, c), (c, d), (d, c)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα (σχήμα 2.13).



Οι πλήρεις επεκτάσεις του πιο πάνω παραδείγματος είναι τα ακόλουθα σύνολα: $E_{co} = \{\emptyset, \{a, c\}, \{b, d\}\}$.

2. Προτιμώμενη Επέκταση (Preferred Extensions) [2]:

Ένα σύνολο επιχειρημάτων S ονομάζεται προτιμώμενη επέκταση ενός επιχειρηματικού πλαισίου (AF) αν είναι το μέγιστο παραδεχτό σύνολο του επιχειρηματικού πλαισίου. Με άλλα λόγια η προτιμώμενη επέκταση είναι το μεγαλύτερο σύνολο που μπορεί να υπερασπιστεί τον εαυτό του από επιθέσεις. Υπάρχει πάντα μία προτιμώμενη επέκταση σε μια θεωρία γιατί σε όλες τις θεωρίες υπάρχει το κενό σύνολο ($\emptyset$), το οποίο είναι παραδεχτό σύνολο.

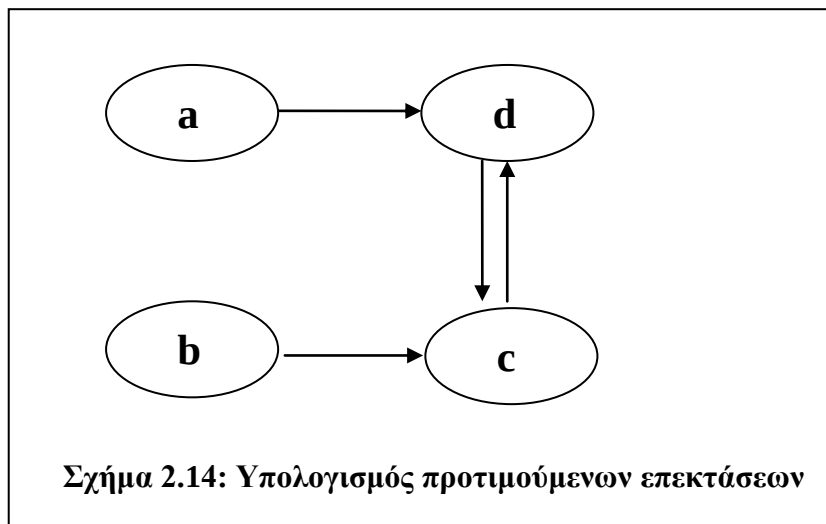
Ο συμβολισμός της προτιμώμενης επέκτασης είναι ο ακόλουθος: E_{PR} .

Αυστηρός ορισμός:

Έστω η θεωρία $AF = \langle A, R \rangle$, το σύνολο $S \subseteq A$ είναι προτιμώμενη επέκταση αν και μόνο αν το S είναι το μέγιστο στοιχείο του $AS(AF)$.

Παράδειγμα κατανόησης 2.12 :

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a, b, c, d\}, \{(a, d), (b, c), (c, d), (d, c)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα (σχήμα 2.14).



Οι προτιμώμενες επεκτάσεις του πιο πάνω παραδείγματος είναι τα ακόλουθα σύνολα:

$$E_{PR} = \{\{a, c\}, \{b, d\}\}.$$

3.Ευσταθής Επέκταση (Stable Extensions) [2]:

Ένα σύνολο επιχειρημάτων S ονομάζεται ευσταθής επέκταση αν και μόνο αν ισχύουν οι δύο πιο κάτω ιδιότητες:

- i. Ιδιότητα απουσίας συγκρούσεων στο σύνολο(conflict-free)
- ii. Ιδιότητα συλλογικής άμυνας (collective defence). Σύμφωνα με αυτή την ιδιότητα κάποιο επιχείρημα που ανήκει στο σύνολο S πρέπει να επιτίθεται σε κάθε άλλο επιχείρημα που δεν ανήκει στο σύνολο S .

Σε μία θεωρία μπορεί να μην υπάρχουν ή να υπάρχουν περισσότερες από μία ευσταθείς επεκτάσεις.

Ο συμβολισμός της ευσταθούς επέκτασης είναι ο ακόλουθος: E_{ST} .

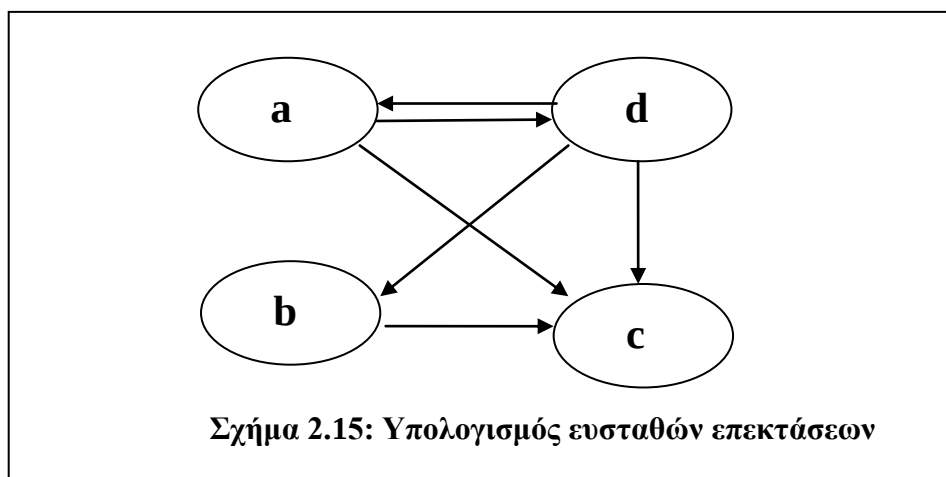
Αυστηρός ορισμός:

Έστω η θεωρία $AF = \langle A, R \rangle$, το σύνολο $S \subseteq A$ είναι ευσταθής επέκταση αν και μόνο αν το S

ικανοποιεί την ιδιότητα της απουσίας συγκρούσεων (conflict-free) και $\forall b \notin S, \exists a \in S$ τέτοιο ώστε $(a, b) \in R$.

Παράδειγμα κατανόησης 2.13:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a, b, c, d\}, \{(a, b), (a, c), (b, a), (b, c), (c, d)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα (σχήμα 2.15).



Στο πιο πάνω παράδειγμα παρατηρούμε πω μόνο μία ευσταθής επέκταση υπάρχει θεωρία, η οποία είναι: $E_{ST} = \{\{a, b\}, \{d\}\}$.

4. Στηριγμένη Επέκταση (Grounded Extensions) [2]:

Ένα σύνολο επιχειρημάτων S ονομάζεται στηριγμένη επέκταση αν είναι το ελάχιστο σύνολο που είναι πλήρης επέκταση.

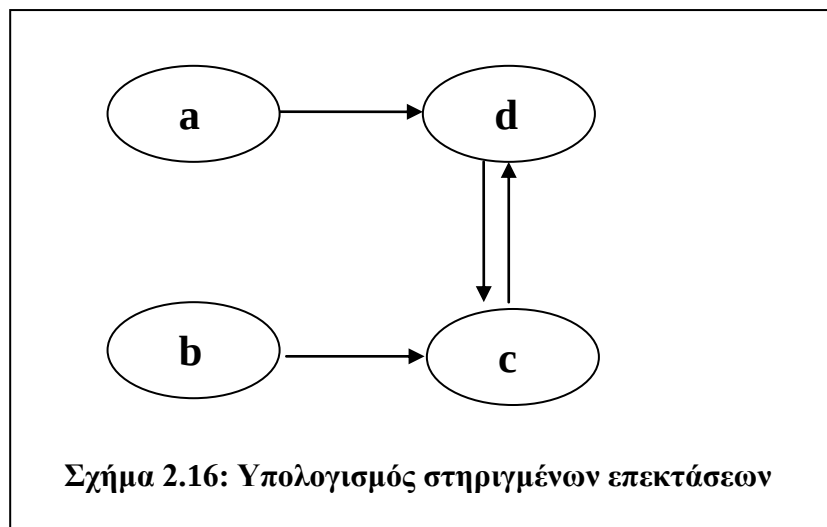
Ο συμβολισμός της στηριγμένης επέκτασης είναι ο ακόλουθος: E_{GR} .

Αυστηρός ορισμός:

Έστω η θεωρία $AF = \langle A, R \rangle$, το σύνολο $S \subseteq A$ είναι στηριγμένη επέκταση αν και μόνο αν το S είναι το ελάχιστο παραδεκτό σύνολο και $\forall b \in A$ όπου b είναι αποδεχτό επιχείρημα, τότε το $b \in S$.

Παράδειγμα κατανόησης 2.14:

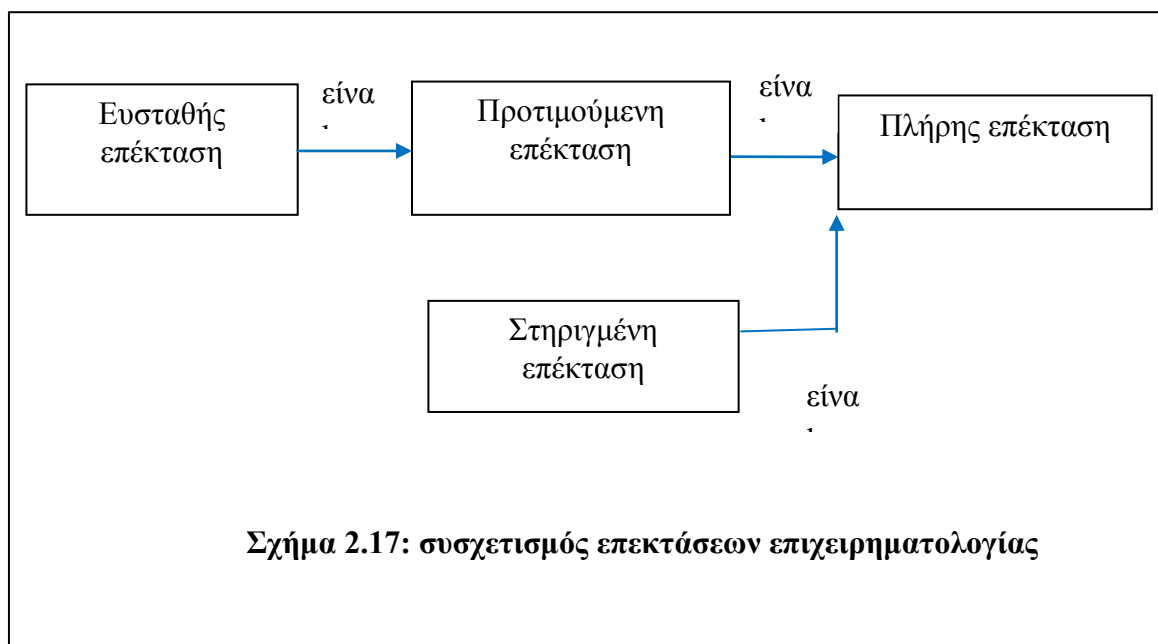
Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a, b, c, d\}, \{(a, d), (b, c), (c, d), (d, c)\} \rangle$ της οποίας η αναπαράσταση φαίνεται στο πιο κάτω σχήμα (σχήμα 2.16).



Η στηριγμένη επέκταση του πιο πάνω παραδείγματος είναι το ακόλουθο σύνολο: $E_{GR} = \{ \emptyset \}$.

Αν ρίξουμε μια πιο προσεχτική ματιά στις πιο πάνω περιγραφές που έχουν δοθεί για τις βασικές επεκτάσεις, οι οποίες διέπουν ένα σύστημα επιχειρηματολογίας, θα καταλήξουμε στο συμπέρασμα ότι υπάρχει κάποιος συσχετισμός μεταξύ των

επεκτάσεων αυτών. Ο συσχετισμός αυτός αναπαρίστανται στο πιο κάτω σχήμα (βλέπε σχήμα 2.17).



Κεφάλαιο 3

Προτασιακή ικανοποιησιμότητα και αλγόριθμοι εκμάθησης θεωριών της επιχειρηματολογίας

3.1 Προτασιακή ικανοποιησιμότητα	25
3.2 Υπολογισμός ευσταθών επεκτάσεων μιας θεωρίας	27
3.3 Υπολογισμός ελάχιστης θεωρίας	39
3.4 Σύγκριση επιχειρηματικών πλαισίων με βάση τις ευσταθείς επεκτάσεις	45
3.5 Σύγκριση επιχειρηματικών πλαισίων με βάση τις δυαδικές σχέσεις επίθεσης	48

Σε αυτό το κεφάλαιο αρχικά θα ορίσουμε την έννοια της προτασιακής ικανοποιησιμότητας. Ακολούθως θα παρουσιαστούν οι αλγόριθμοι που χρειάστηκαν να εξεταστούν για την παρούσα διπλωματική. Μέσα από την περιγραφή αυτών των αλγορίθμων θα προκύψει η συσχέτιση της προτασιακής ικανοποιησιμότητας με την επιχειρηματολογία.

3.1 Προτασιακή Ικανοποιησιμότητα

Το πρόβλημα της προτασιακής ικανοποιησιμότητας είναι γνωστό και ως SAT πρόβλημα. Αποτελεί αντικείμενο μελέτης για τους επιστήμονες εδώ και πάρα πολλά χρόνια και είναι ένα από τα πιο βασικά προβλήματα στην θεωρία υπολογισμού, καθώς επίσης και στην μαθηματική λογική. Επιπλέον ανήκει στην κατηγορία των NP-complete προβλημάτων, δηλαδή προβλήματα τα οποία δεν γνωρίζουμε ακόμη αν μπορούν να επιλυθούν σε πολυωνυμικό χρόνο.

Η προτασιακή ικανοποιησιμότητα χρησιμοποιείται για να μοντελοποιήσουμε ένα μεγάλο σύνολο προβλημάτων στο οποίο πρέπει να οριστούν οι μεταβλητές απόφασης, έτσι ώστε να ικανοποιούν ένα δεδομένο σύνολο περιορισμών. Συγκεκριμενοποιώντας, έχοντας ως δεδομένο ένα σύνολο από δυαδικές μεταβλητές (δηλαδή μεταβλητές οι οποίες παίρνουν τιμή 0 ή 1) και ένα σύνολο από περιορισμούς το οποίο αναπαρίστανται σε κανονική συζευκτική μορφή (Conjunctive Normal Form), στόχος είναι να ελεγχθεί αν υπάρχει κάποια ανάθεση τιμών στις μεταβλητές, τέτοια ώστε να ικανοποιεί όλους τους περιορισμούς. Σε περίπτωση που ισχύει κάτι τέτοιο τότε το πρόβλημα είναι επιλύσιμο, αλλιώς το πρόβλημα δεν έχει λύση [5].

Ας εξετάσουμε τώρα τη μορφή με την οποία αναπαρίστανται οι περιορισμοί σε ένα πρόβλημα προτασιακής ικανοποιησιμότητας. Όπως έχει προαναφερθεί οι περιορισμοί αναπαρίστανται στη μορφή του λογικού τύπου της συζευκτικής κανονικής μορφής (CNF). Αν θέλουμε να γίνουμε πιο συγκεκριμένοι, ένας περιορισμός είναι ένα σύνολο από συζευκτικούς όρους, όπου κάθε όρος αποτελείται από διαζεύξεις. Με τον όρο σύζευξη αναφερόμαστε στο λογικό AND, το οποίο συμβολίζεται με « $\wedge$ » και με τον όρο διάζευξη αναφερόμαστε στο λογικό OR, το οποίο συμβολίζεται με « $\vee$ ». Τα διάφορα στοιχεία στα οποία εφαρμόζεται το λογικό OR ονομάζονται λεκτικά (literals) και το καθένα από αυτά αντιπροσωπεύουν μια λογική μεταβλητή και κάθε διαζευκτικός όρος ονομάζεται clause. Ένα λεκτικό μπορεί να είναι μια μεταβλητή, όπως για παράδειγμα η μεταβλητή x , ή η άρνηση της μεταβλητής, όπως για παράδειγμα η μεταβλητή $\bar{x}$ ($\bar{x}$ = άρνηση της μεταβλητής x , δηλαδή είναι το $\text{not}(x)$).

Πιο κάτω ακολουθεί ένα παράδειγμα περιορισμού σε μορφή CNF:

Παράδειγμα κατανόησης 3.1:

$$(x \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (y \vee \bar{z}) \wedge (z \vee \bar{x}) \wedge (\bar{x} \vee \bar{y} \vee \bar{z})$$

Σχήμα 3.1:Περιορισμός σε CNF μορφή

Στο πιο πάνω παράδειγμα κάνουμε τις εξής παρατηρήσεις:

Λεκτικά (literals) είναι: $x, y, z, \bar{x}, \bar{y}, \bar{z}$

Όροι (clauses) είναι: $(x \vee y \vee z), (x \vee \bar{y}), (y \vee \bar{z}), (z \vee \bar{x}), (\bar{x} \vee \bar{y} \vee \bar{z})$

Οι όροι (clauses) μπορεί να μην περιέχουν κάποια μεταβλητή και αυτά οι όροι ονομάζονται «κενοί» όροι (empty clauses). Ακόμη μπορεί να περιέχουν μόνο ένα λεκτικό (literal) και σε αυτή την περίπτωση ονομάζονται «μοναδιαίοι» όροι (unity clause) και τέλος μπορούν να περιέχουν περισσότερα από ένα λεκτικά.

Τα λεκτικά (literals) μπορούν να πάρουν την τιμή 0 που είναι η αντίστοιχη αποτίμηση της μεταβλητής σε ψευδής (false) ή την τιμή 1 που είναι η αντίστοιχη αποτίμηση της μεταβλητής σε αληθής (true).

Ένας όρος (clause) είναι αληθής και ικανοποιήσιμος, αν υπάρχει τουλάχιστον ένα λεκτικό του, στο οποίο ανατίθεται η τιμή 1, ενώ δεν είναι ικανοποιήσιμος αν όλα τα λεκτικά του πάρουν την τιμή 0.

Ένα περιορισμός καθορίζεται ως αληθής και ικανοποιήσιμος αν όλοι οι όροι (clauses) που το αποτελούν προκύψουν να είναι αληθής.

Στα μετέπειτα υποκεφάλαια θα δούμε πως συσχετίζεται το πρόβλημα της προτασιακής ικανοποίησης με την επιχειρηματολογία.

3.2 Υπολογισμός ευσταθών επεκτάσεων μιας θεωρίας

Ο πρώτος αλγόριθμός που είχαμε να μελετήσουμε αφορούσε τον εντοπισμό όλων των ευσταθών επεκτάσεων ενός επιχειρηματικού πλαισίου, δηλαδή μιας θεωρίας. Ακόμη μεταβάλλοντας την θεωρία αυτή αφαιρώντας της κάποιο ή κάποια υποσύνολα επιχειρημάτων, δηλαδή αφαιρώντας από αυτήν οποιαδήποτε από τα επιχειρήματα της, πάλι ο αλγόριθμος θα πρέπει να εντοπίζει όλες τις ευσταθείς επεκτάσεις της.

Στο σημείο αυτό ας θυμηθούμε τι ακριβώς εννοούμε με τους όρους «θεωρία» και «ευσταθείς επεκτάσεις».

Αρχικά, όπως έχει παρουσιαστεί και στο υποκεφάλαιο 2.2, με τον όρο θεωρία ορίζεται ένα ζεύγος $\langle A, R \rangle$ όπου A είναι το σύνολο των επιχειρημάτων και R το σύνολο

δυναδικών σχέσεων επίθεσης πάνω στο σύνολο A όπου $R \subseteq A \times A$. Δηλαδή λέμε ότι το επιχείρημα a επιτίθεται στο επιχείρημα b αν και μόνο αν $(a,b) \in R$.

Επιπλέον, όπως φαίνεται και στο υποκεφάλαιο 2.3, για να χαρακτηριστεί ένα σύνολο ως ευσταθή επέκταση μιας θεωρίας πρέπει να ικανοποιούνται οι ακόλουθες ιδιότητες:

1. Ιδιότητα απουσίας συγκρούσεων μέσα στο σύνολο (Conflict-free)
2. Συλλογική άμυνα (Collective Defense)

Επομένως, αρκεί να μοντελοποιήσουμε τις δύο πιο πάνω ιδιότητες για να βρούμε τις ευσταθείς επεκτάσεις μιας θεωρίας, η οποία θα μας δίνεται σαν είσοδος στο πρόβλημα.

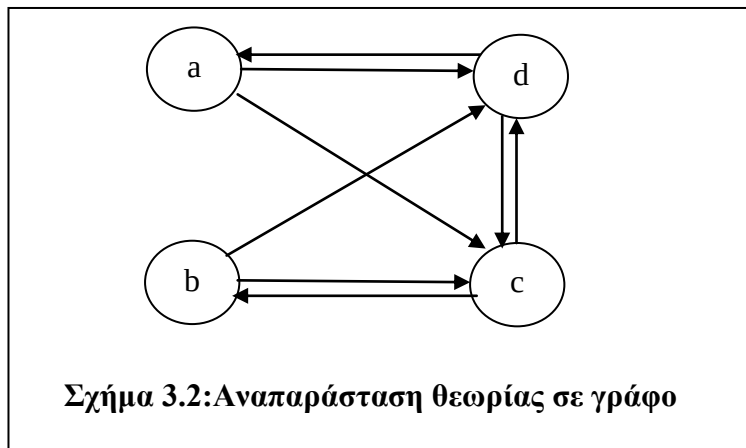
Η θεωρία που θα δίνεται σαν είσοδος στο πρόβλημα θα αποθηκεύεται σ'ένα δισδιάστατο πίνακα γειτνίασης με αριθμό γραμμών και στηλών όσα είναι και τα επιχειρήματα της θεωρίας. Οι σχέσεις των επιθέσεων που υπάρχουν στην θεωρία θα αντιπροσωπεύονται από τις τιμές που θα υπάρχουν μέσα στον πίνακα. Οι τιμές που μπορούν να ανατεθούν σε κάθε θέση του πίνακα είναι 0, το οποίο αντιστοιχεί στο γεγονός ότι δεν υπάρχει ακμή, δηλαδή δεν υπάρχει σχέση επίθεσης, ενώ η άλλη τιμή που μπορεί να ανατεθεί είναι η 1, το οποίο αντιστοιχεί στο γεγονός ότι υπάρχει ακμή, δηλαδή υπάρχει σχέση επίθεσης. Πιο αναλυτικά αν υπάρχει σχέση επίθεσης από το επιχείρημα A στο επιχείρημα B , τότε στη θέση του πίνακα $[A,B]$ θα ισχύει το ακόλουθο $\text{πίνακα}[A,B] = 1$, διαφορετικά αν δεν υπάρχει ακμή τότε θα ισχύει το εξής $\text{πίνακα}[A,B] = 0$.

Πιο κάτω παρουσιάζεται ένα παράδειγμα αναπαράστασης μιας θεωρίας στο αντίστοιχο γράφο και στον αντίστοιχο δισδιάστατο πίνακα γειτνίασης της.

Παράδειγμα κατανόηση 3.2:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a,b,c,d\}, \{(a,d), (b,c), (c,d), (d,c), (a,c), (b,d), (d,a), (c,b)\} \rangle$. Οι αναπαραστάσεις της οποίας φαίνονται στα πιο κάτω σχήματα (βλέπε σχήματα 3.2 και 3.3).

Αναπαράσταση θεωρίας που δόθηκε στο παράδειγμα κατανόησης 3.2 σε γράφο:



Αναπαράσταση θεωρίας που δόθηκε στο παράδειγμα κατανόησης 3.2 σε πίνακα γειτνίασης:

ΕΠΙΧΕΙΡΗΜΑΤΑ	a	b	c	d
a	0	0	1	1
b	0	0	1	1
c	0	1	0	1
d	1	0	1	0

Σχήμα 3.3: Αποθήκευση θεωρίας σε δισδιάστατο πίνακα γειτνίασης

Ας δούμε τώρα πως μπορούμε να μοντελοποιήσουμε τις ιδιότητες των ευσταθών επεκτάσεων.

Απουσία συγκρούσεων μέσα στο σύνολο (Conflict-free):

Με βάση αυτή την ιδιότητα πρέπει να αποτρέψουμε επιχειρήματα τα οποία γειτονεύουν να μπουν μέσα στην ίδια ευσταθή επέκταση. Με αυτό τον τρόπο αποτρέπουμε επιχειρήματα που τους ενώνει μια σχέση επίθεσης μεταξύ τους, να αποτελούν στοιχεία της ίδιας ευσταθής επέκτασης. Για το σκοπό αυτό θα πρέπει για κάθε σχέση επίθεσης που ανήκει στο σύνολο R να υπάρχει τουλάχιστον ένα επιχείρημα της κάθε σχέσης που δεν θα ανήκει στην ευσταθή επέκταση. Επομένως, έστω μία δεδομένη σχέση επίθεσης

(A,B), δηλαδή το επιχείρημα A κάνει επίθεση στο επιχείρημα B, τότε πρέπει τουλάχιστον ένα από τα δύο αυτά επιχειρήματα να μην ανήκει στην ευσταθή επέκταση, δηλαδή είτε δεν θα ανήκει το A, είτε δεν θα ανήκει το B, είτε δεν θα ανήκει κανένα από τα δύο. Κάνοντας αυτή την ενέργεια για όλες τις σχέσεις επίθεσης που υπάρχουν στην θεωρία αποτρέπουμε τη δημιουργία συγκρούσεων μέσα στην ευσταθή επέκταση.

Συλλογική άμυνα (Collective Defense):

Σύμφωνα με αυτή την ιδιότητα όλα τα επιχειρήματα που δεν θα ανήκουν στο σύνολο το οποίο θα αποτελεί την ευσταθής επέκταση, θα πρέπει να δέχονται επίθεση από κάποιο ή κάποια επιχειρήματα που ανήκουν στην ευσταθής επέκταση. Επομένως μπορούμε να κάνουμε την εξής ενέργεια για να μοντελοποιήσουμε αυτή την ιδιότητα: για κάθε επιχείρημα το οποίο δεν βάζουμε μέσα στο σύνολο της ευσταθούς επέκτασης πρέπει να βάλουμε αναγκαστικά στο συγκεκριμένο σύνολο κάποιο άλλο επιχείρημα το οποίο επιτίθεται στο επιχείρημα που αφήσαμε έξω κάθε φορά. Για παράδειγμα αν έχουμε μία σχέση επίθεσης (A,B) και επιλέξουμε να αφήσουμε εκτός του συνόλου της ευσταθής επέκτασης το επιχείρημα B, τότε αναγκαστικά πρέπει να προσθέσουμε στο σύνολο της ευσταθούς επέκτασης το επιχείρημα A. Κάνοντας αυτή την διαδικασία ικανοποιούμε και την δεύτερη ιδιότητα των ευσταθών επεκτάσεων.

Πιο κάτω παρουσιάζεται η εφαρμογή των πιο πάνω κανόνων στο παράδειγμα κατανόησης 3.2.

Παράδειγμα κατανόηση 3.3:

Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a,b,c,d\}, \{(a,d), (b,c), (c,d), (d,c), (a,c), (b,d), (d,a), (c,b)\} \rangle$. Η θεωρία αυτή αποτελείται από τέσσερα επιχειρήματα και οχτώ σχέσεις επίθεσης μεταξύ αυτών των επιχειρημάτων.

Μοντελοποίηση απουσίας συγκρούσεων:

Όταν υπάρχει σχέση επίθεσης μεταξύ δύο επιχειρημάτων A, B αντίστοιχα, τότε δεν μπορούν και τα δύο αυτά επιχειρήματα να μπουν μέσα στην ίδια ευσταθή επέκταση.

Επομένως ο κανόνας που προκύπτει είναι ο εξής:

Η ουσία είναι ότι εάν A, B είναι γείτονες (υπάρχει επίθεση), δεν μπορούν και τα δύο να μουν μέσα στην ευσταθή επέκταση. Επομένως πρέπει να ισχύει ο κανόνας $\neg (A \wedge B)$. Εφαρμόζοντας τον κανόνα De Morgan (σύμφωνα με τον κανόνα De Morgan ισχύει το ακόλουθο: $(\neg (X \wedge Y)) \equiv (\neg X \vee \neg Y)$) στον κανόνα αυτό θα προκύψει $(\neg A \vee \neg B)$. Με το σύμβολο « $\neg$ », συμβολίζεται η άρνηση εισαγωγής του επιχειρήματος στην ευσταθή επέκταση).

Για την πιο πάνω θεωρία προκύπτουν τα εξής:

$\neg a \vee \neg d$: είτε το επιχείρημα a είτε το επιχείρημα d θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα a επιτίθεται στο επιχείρημα d

$\neg b \vee \neg c$: είτε το επιχείρημα b είτε το επιχείρημα c θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα b επιτίθεται στο επιχείρημα c

$\neg c \vee \neg d$: είτε το επιχείρημα c είτε το επιχείρημα d θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα c επιτίθεται στο επιχείρημα d

$\neg d \vee \neg c$: είτε το επιχείρημα d είτε το επιχείρημα c θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα d επιτίθεται στο επιχείρημα c

$\neg a \vee \neg c$: είτε το επιχείρημα a είτε το επιχείρημα c θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα a επιτίθεται στο επιχείρημα c

$\neg b \vee \neg d$: είτε το επιχείρημα b είτε το επιχείρημα d θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα b επιτίθεται στο επιχείρημα d

$\neg d \vee \neg a$: είτε το επιχείρημα d είτε το επιχείρημα a θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα d επιτίθεται στο επιχείρημα a

$\neg c \vee \neg d$: είτε το επιχείρημα c είτε το επιχείρημα d θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα c επιτίθεται στο επιχείρημα d

Μοντελοποίηση συλλογικής άμυνας:

Όπως έχουμε προαναφέρει για την ιδιότητα αυτή πρέπει για κάθε επιχείρημα που μένει εκτός του συνόλου της ευσταθούς επέκτασης, να προστεθεί στο σύνολο αυτό κάποιο άλλο επιχείρημα που του επιτίθεται. Έτσι εξετάζουμε ένα προς ένα τα επιχειρήματα της θεωρίας.

Για την πιο πάνω θεωρία θα έχουμε τους εξής κανόνες:

$\neg a \rightarrow d$: αν δεν συμπεριλάβουμε το a στην ευσταθή επέκταση, τότε πρέπει να προσθέσουμε σε αυτήν το d , αφού είναι το μοναδικό επιχείρημα από το οποίο δέχεται το a επίθεση. Εφαρμόζοντας τους κανόνες : $\neg X \vee Y \equiv X \rightarrow Y$ και $\neg(\neg X) = X$, προκύπτει ότι $\neg(\neg a) \vee d = a \vee d$.

Εκτελώντας την ίδια λογική και στα υπόλοιπα επιχειρήματα θα έχουμε τους εξής κανόνες:

$b \vee c, c \vee a \vee b \vee d$ και $d \vee a \vee b \vee c$.

Τώρα ας δούμε πως θα αλλάξει η πιο πάνω διαδικασία μοντελοποίησης των ιδιοτήτων των ευσταθών επεκτάσεων, αν αφαιρέσουμε ένα υποσύνολο επιχειρημάτων το οποίο αποτελείται από το επιχείρημα a , δηλαδή μεταβάλλοντας την θεωρία που παρουσιάστηκε στο παράδειγμα κατανόησης 3.2 αφαιρώντας της το επιχείρημα a .

Παράδειγμα κατανόησης 3.4:

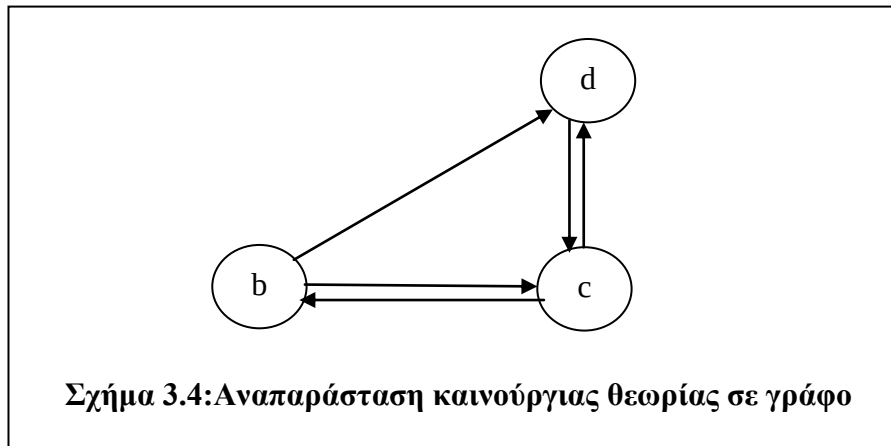
Έστω ότι μας δίνεται η θεωρία $AF = \langle \{a, b, c, d\}, \{(a, d), (b, c), (c, d), (d, c), (a, c), (b, d), (d, a), (c, b)\} \rangle$. Η θεωρία αυτή αποτελείται από τέσσερα επιχειρήματα και οχτώ σχέσεις επίθεσης μεταξύ αυτών των επιχειρημάτων. Και έστω ότι αφαιρούμε το επιχείρημα a από την θεωρία.

Η θεωρία μεταβάλλεται, κάθε φορά που αφαιρείται κάποιο από τα επιχειρήματα της, αφού μαζί με τα επιχειρήματα που φεύγουν, πρέπει να φύγουν και οι σχέσεις επίθεσης που κάνουν και δέχονται τα συγκεκριμένα επιχειρήματα.

Έτσι, αν αφαιρέσουμε το επιχείρημα a από την θεωρία που παρουσιάζεται στο παράδειγμα κατανόησης 3.4 θα προκύψει η εξής καινούργια θεωρία: $AF = \langle \{b, c, d\}, \{(b, c), (c, d), (d, c), (b, d), (c, b)\} \rangle$.

Στα πιο κάτω σχήματα παρουσιάζονται οι αναπαραστάσεις σε γράφο και πίνακα γειτνίασης της καινούργιας θεωρίας που προκύπτει από την αφαίρεση του επιχειρήματος a .

Αναπαράσταση σε γράφο της προκύπτων θεωρίας από την αφαίρεση του επιχειρήματος a από την αρχική θεωρία:



Αναπαράσταση σε πίνακα γειτνίασης της προκύπτων θεωρίας από την αφαίρεση του επιχειρήματος a από την αρχική θεωρία:

ΕΠΙΧΕΙΡΗΜΑΤΑ	b	c	d
b	0	1	1
c	1	0	1
d	0	1	0

Σχήμα 3.5: Αποθήκευση καινούργιας θεωρίας σε δισδιάστατο πίνακα

Μοντελοποίηση απουσίας συγκρούσεων:

Ο τρόπος μοντελοποίησης δεν μεταβάλλεται. Αυτό που θα αλλάξει είναι σε ποιες σχέσεις επίθεσης θα εφαρμοστεί αυτή η μοντελοποίηση. Συγκεκριμένα η διαδικασία αυτή θα εφαρμοστεί σε όλες τις σχέσεις επίθεσης της καινούργιας θεωρίας, δηλαδή δεν θα εφαρμοστεί στις σχέσεις επίθεσης της αρχικής θεωρίας οι οποίες συμπεριλαμβάνουν το επιχείρημα που αφαιρέθηκε.

Για το πιο πάνω παράδειγμα θα προκύψουν οι εξής κανόνες:

$\neg b \vee \neg c$: είτε το επιχείρημα b είτε το επιχείρημα c θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα b επιτίθεται στο επιχείρημα c

$\neg c \vee \neg d$: είτε το επιχείρημα c είτε το επιχείρημα d θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα c επιτίθεται στο επιχείρημα d

$\neg d \vee \neg c$: είτε το επιχείρημα d είτε το επιχείρημα c θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα d επιτίθεται στο επιχείρημα c

$\neg b \vee \neg d$: είτε το επιχείρημα b είτε το επιχείρημα d θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα b επιτίθεται στο επιχείρημα d

$\neg c \vee \neg d$: είτε το επιχείρημα c είτε το επιχείρημα d θα ανήκει στην ευσταθές επέκταση, αφού το επιχείρημα c επιτίθεται στο επιχείρημα d

Μοντελοποίηση συλλογικής άμυνας:

Και πάλι ο τρόπος μοντελοποίησης δεν αλλάζει. Αυτό που αλλάζει είναι το σύνολο των επιχειρημάτων στα οποία θα εφαρμοστεί αυτή η διαδικασία. Όπως ήδη γνωρίζουμε η διαδικασία αυτή εξετάζει ένα προς ένα τα επιχειρήματα. Συγκεκριμένα για κάθε επιχείρημα, το οποίο ανήκει στη θεωρία, εξετάζουμε τις επιθέσεις που δέχεται. Όπως αντιλαμβανόμαστε η μοντελοποίηση της συλλογικής άμυνας δεν θα εφαρμοστεί στα επιχειρήματα τα οποία αφαιρούμε από την θεωρία, θα εφαρμοστεί μόνο στα επιχειρήματα της καινούργιας θεωρίας.

Για την πιο πάνω θεωρία θα έχουμε τους εξής κανόνες:

$\neg b \rightarrow c$: αν δεν συμπεριλάβουμε το b στην ευσταθή επέκταση, τότε πρέπει να προσθέσουμε σε αυτήν το c , αφού είναι το μοναδικό επιχείρημα από το οποίο δέχεται το b επίθεση. Εφαρμόζοντας τους κανόνες : $\neg X \vee Y \equiv X \rightarrow Y$ και $\neg(\neg X) = X$, προκύπτει ότι $\neg(\neg b) \vee c = b \vee c$.

Εκτελώντας την ίδια λογική και στα υπόλοιπα επιχειρήματα θα έχουμε τους εξής κανόνες:

$c \vee b \vee d$ και $d \vee b \vee c$.

Στο σημείο αυτό ας δούμε πόσα συνολικά υποσύνολα επιχειρημάτων μπορούν να αφαιρεθούν από μία θεωρία. Έστω ότι το πλήθος των επιχειρημάτων που απαρτίζουν μία θεωρία είναι ίσο με 3, τότε θα μπορούσαν να αφαιρεθούν από αυτήν 2^3 υποσύνολα. Γενικά ο μέγιστος αριθμός υποσυνόλων που μπορούν να αφαιρεθούν από μία θεωρία

που αποτελείται από n -επιχειρήματα είναι: 2^n . Ο τρόπος με τον οποίο αποφασίζονται ποια επιχειρήματα θα αφαιρεθούν από τη θεωρία κάθε φορά και ποια όχι στηρίζεται σε ένα διάνυσμα μεγέθους n , δηλαδή όσα είναι και το πλήθος των επιχειρημάτων που αποτελούν τη θεωρία. Κάθε θέση του διανύσματος αυτού παίρνει τιμή 0 ή 1 και αντιστοιχίζεται με ένα και μοναδικό επιχείρημα της θεωρίας. Κάθε ένα από τα επιχειρήματα της θεωρίας αντιστοιχείται με μόνο μία θέση του διανύσματος και η θέση αυτή είναι μοναδική, δηλαδή δεν μπορούν διαφορετικά επιχειρήματα να αντιστοιχίζονται με την ίδια θέση του διανύσματος και το ανάποδο. Αν η τιμή κάποιας θέσης του διανύσματος είναι ίση με 1, τότε αυτό σημαίνει ότι το επιχείρημα που αντιστοιχείται με αυτή τη θέση πρέπει να αφαιρεθεί από τη θεωρία. Σε αντίθετη περίπτωση, δηλαδή αν η τιμή της θέσης του διανύσματος είναι ίση με 0, τότε το επιχείρημα που αντιπροσωπεύεται από αυτή τη θέση δεν πρέπει να αφαιρεθεί από τη θεωρία. Έτσι είναι εύκολο να αντιληφθούμε ότι ο συνολικός αριθμός υποσυνόλων που μπορούν να αφαιρεθούν από μία θεωρία είναι ίσος με τους δυνατούς συνδυασμούς από μηδέν και ένα που μπορούν να γίνουν σε ένα διάνυσμα μεγέθους n , όπου το n μεταβάλλεται ανάλογα με το πλήθος των επιχειρημάτων που έχει η θεωρία κάθε φορά.

Πιο κάτω ακολουθεί ένα παράδειγμα όπου παρουσιάζονται τα πιθανά υποσύνολα που μπορούν να αφαιρεθούν από μία θεωρία.

Παράδειγμα Κατανόησης 3.5:

Έστω ότι μας δίνεται μία θεωρία $AF = \langle \{a,b\}, \{(a,b), (b,a)\} \rangle$. Η θεωρία αυτή αποτελείται από δύο επιχειρήματα.

Συνολικά μπορούν να της αφαιρεθούν $2^2 = 4$ υποσύνολα.

Ας δούμε ποια είναι αυτά τα 4 πιθανά υποσύνολα επιχειρημάτων που μπορούν να αφαιρεθούν από τη θεωρία κάνοντας χρήση ενός διανύσματος μεγέθους 2, δηλαδή όσο είναι και το πλήθος των επιχειρημάτων που έχει η θεωρία. Το διάνυσμα αυτό μπορούμε να πούμε ότι είναι ένα bit vector, γιατί οι τιμές που μπορούν να ανατεθούν στις θέσεις του είναι 0 ή 1. Η πρώτη θέση του bit vector που θα χρησιμοποιηθεί αντιπροσωπεύει το

επιχείρημα a, ενώ η δεύτερη θέση του αντιπροσωπεύει το επιχείρημα b. Ας δούμε πως αναπαρίστανται το bit vector ανάλογα με το υποσύνολο κάθε φορά.

Bit vector πρώτου υποσυνόλου:

A	b
0	0

Το πιο πάνω bit vector αντιστοιχεί στο πρώτο πιθανό υποσύνολο επιχειρημάτων που μπορεί να αφαιρεθεί από τη θεωρία. Συγκεκριμένα το υποσύνολο αυτό είναι το κενό, αφού δεν αφαιρείται κανένα από τα επιχειρήματα της θεωρίας. Όπως αντιλαμβανόμαστε δεν παρουσιάζεται κάποια μεταβολή στην αρχική θεωρία, αν της αφαιρέσουμε το κενό υποσύνολο. Άρα η γνώση που θα λάβουμε σε αυτή την περίπτωση θα είναι η ίδια γνώση με αυτή που θα λαμβάναμε αν δεν αφαιρούσαμε το κενό υποσύνολο. Πιο ειδικά, αφού η γνώση που λαμβάνουμε στο συγκεκριμένο αλγόριθμο σχετίζεται με τις ευσταθείς επεκτάσεις των θεωριών, τότε θα παρατηρήσουμε ότι οι ευσταθείς επεκτάσεις που θα μας επιστραφούν δοθέντος αυτών των δύο θεωριών θα είναι οι ίδιες.

Bit vector δεύτερου υποσυνόλου:

A	b
0	1

Από την πιο πάνω αναπαράσταση του διανύσματος αντιλαμβανόμαστε ότι το υποσύνολο που θα αφαιρεθεί αποτελείται από μόνο ένα επιχείρημα το οποίο είναι το επιχείρημα b. Στην περίπτωση αυτή θα εκλάβουμε διαφορετική γνώση, αφού οι δύο θεωρίες θα είναι διαφορετικές.

Bit vector τρίτου υποσυνόλου:

A	b
1	0

Από την πιο πάνω αναπαράσταση του διανύσματος αντιλαμβανόμαστε ότι το υποσύνολο που θα αφαιρεθεί αποτελείται και πάλι από μόνο ένα επιχείρημα το οποίο

είναι το επιχείρημα a. Στην περίπτωση αυτή θα εκλάβουμε διαφορετική γνώση, αφού οι δύο θεωρίες θα είναι διαφορετικές.

Bit vector τέταρτου υποσυνόλου:

A	b
1	1

Από την πιο πάνω αναπαράσταση του διανύσματος αντιλαμβανόμαστε ότι το υποσύνολο που θα αφαιρεθεί αποτελείται από δύο επιχειρήματα τα οποία είναι το επιχείρημα a και το επιχείρημα b. Στην περίπτωση αυτή παρατηρούμε ότι αφαιρούνται όλα τα επιχειρήματα που απαρτίζουν τη θεωρία, με αποτέλεσμα να δημιουργείται μία θεωρία «κενή». Με τον όρο «κενή» θεωρία αναφερόμαστε σε μία θεωρία η οποία δεν έχει κάποιο επιχείρημα και ως συνέπεια δεν έχει ούτε σχέσεις επίθεσης. Άρα από μία τέτοια θεωρία είναι εύκολα να αντιληφθούμε ότι δεν μπορούμε να πάρουμε κάποια γνώση.

Ας μη ξεχνάμε όμως ότι ο αλγόριθμος πρέπει να μας επιστρέφει όλες τις ευσταθείς επεκτάσεις μιας θεωρίας. Για τον σκοπό αυτό πρέπει να εφαρμόσουμε ένα ακόμα επιπλέον κανόνα, εκτός από τους κανόνες απουσίας συγκρούσεων και συλλογικής άμυνας. Με τον κανόνα αυτό θα προσπαθήσουμε να μην παίρνουμε την ίδια λύση πολλές φορές. Συγκεκριμένα, λύση η οποία μας έχει ήδη δοθεί ως ευσταθής επέκταση, δεν πρέπει να μας ξανά επιστραφεί ως λύση. Αυτό το επιτυγχάνουμε προσθέτοντας την άρνηση της λύσης που βρίσκουμε κάθε φορά και συνεχίζουμε με τον ίδιο τρόπο όπως και προηγούμενος για να βρούμε μία καινούργια ευσταθής επέκταση.

Ακολουθεί ένα παράδειγμα εφαρμογής του κανόνα που μόλις έχει περιγραφεί.

Παράδειγμα κατανόησης 3.6:

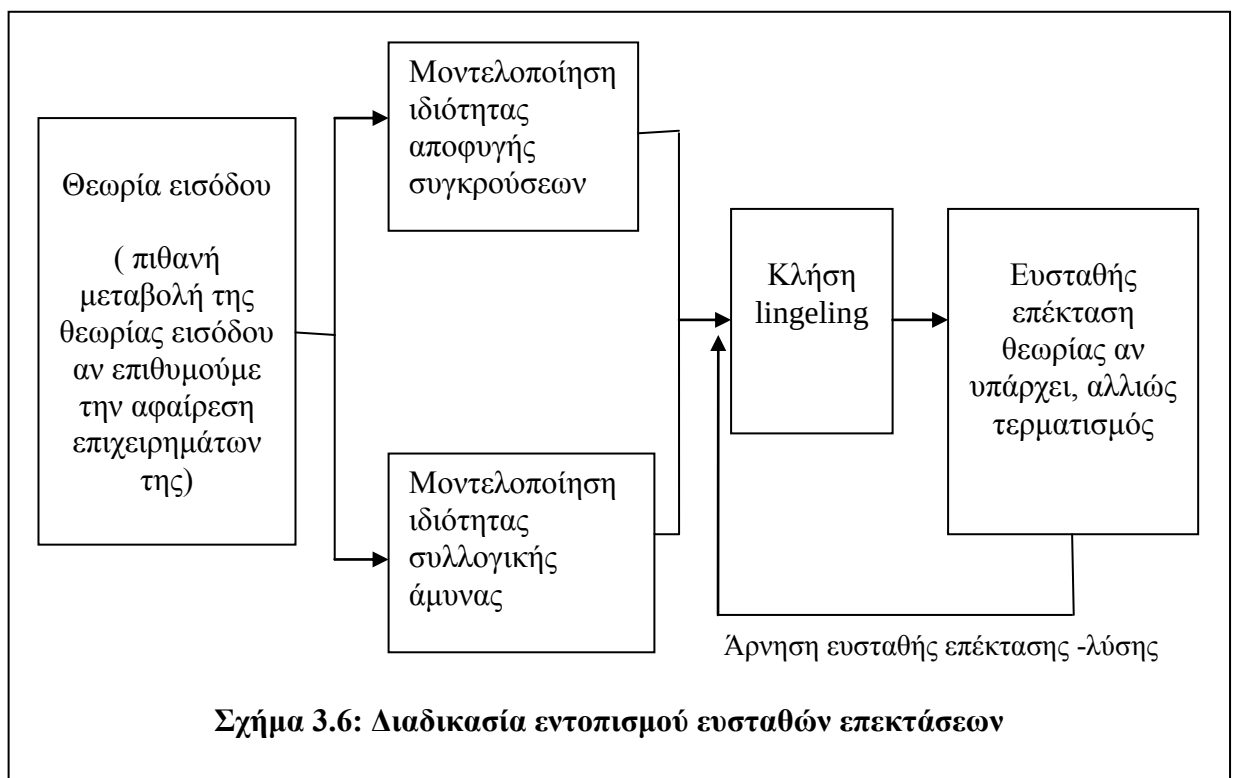
Έστω ότι μας δίνεται η εξής λύση ως ευσταθής επέκταση : $S\top_1 = \{a,b\}$

Εφαρμόζοντας τον πιο πάνω κανόνα, δηλαδή τον κανόνα που μας εγγυάται ότι δεν θα μας ξανά δοθεί ως λύση θα προκύψει το εξής: $\neg (a \wedge b)$.

Εφαρμόζοντας τον κανόνα De Morgan θα προκύψει: $(\neg a \vee \neg b)$.

Στο σημείο αυτό μπορούμε να διαπιστώσουμε ότι το αρχικό πρόβλημα μας, το οποίο ήταν ο εντοπισμός των ευσταθών επεκτάσεων μιας θεωρίας, μπορεί εύκολα να μετατραπεί σε ένα πρόβλημα προτασιακής ικανοποιησιμότητας (SAT problem), εκτελώντας απλά την πιο πάνω μεθοδολογία μοντελοποίησης των ιδιοτήτων των ευσταθών επεκτάσεων. Κάνοντας αυτή την μετατροπή εκφράζουμε τις ιδιότητες που θέλουμε να ισχύουν σε περιορισμούς. Με αυτό τον τρόπο μας δίνεται η ευχέρεια να χρησιμοποιήσουμε κάποιο προτασιακό επιλυτή (SAT solver), ο οποίος θα μας βοηθήσει στο πρόβλημα που έχουμε να επιλύσουμε. Έτσι μπορούμε να δώσουμε ως είσοδο στον προτασιακό επιλυτή αυτούς τους περιορισμούς. Σε αυτό το σημείο ερχόμαστε αντιμέτωποι με δύο περιπτώσεις. Οι δύο αυτές περιπτώσεις είναι: είτε δεν θα υπάρχει λύση στο πρόβλημα, έτσι ο επιλυτής απλά θα μας ενημερώσει ότι δεν υπάρχει λύση που να ικανοποιεί τους περιορισμούς, είτε θα υπάρχει λύση και ο επιλυτής θα μας ενημερώσει πως υπάρχει λύση και θα μας την επιστρέψει. Ο προτασιακός επιλυτής που θα χρησιμοποιήσουμε για τον σκοπό αυτό είναι ο lingeling, του οποίου η λειτουργία και χρήση περιγράφεται στο κεφάλαιο 4 υποκεφάλαιο 4.1.

Στο πιο κάτω σχήμα (σχήμα 3.6) παρουσιάζεται διαγραμματικά η διαδικασία εντοπισμού των ευσταθών επεκτάσεων μιας θεωρίας που δίνεται σαν είσοδος.



3.3 Υπολογισμός ελάχιστης θεωρίας

Ο αλγόριθμος που θα παρουσιαστεί σ' αυτό το υποκεφάλαιο δέχεται σαν είσοδο ένα σύνολο από ευσταθείς επεκτάσεις και στοχεύει στην δημιουργία μιας θεωρίας που να ικανοποιεί αυτές τις επεκτάσεις. Η θεωρία, όμως, που επιθυμούμε να βρούμε δεν είναι οποιαδήποτε θεωρία, αφού στόχος μας είναι να εντοπίσουμε την ελάχιστη θεωρία. Με τον όρο ελάχιστη θεωρία αναφερόμαστε στη θεωρία που αποτελείται από τις λιγότερες ακμές, δηλαδή από τις λιγότερες σχέσεις επίθεσης μεταξύ των επιχειρημάτων που την αποτελούν. Πιο συγκεκριμένα, ο αλγόριθμος αυτός επιστρέφει σαν αποτέλεσμα μια θεωρία με τις ελάχιστες ακμές, η οποία θα έχει σαν ευσταθείς επεκτάσεις τις τις επεκτάσεις που δέχεται σαν είσοδο του προβλήματος.

Αν το σκεφτούμε καλύτερα, θα προσέξουμε πως ο αλγόριθμος του υποκεφαλαίου αυτού κάνει ακριβώς την αντίθετη δουλειά με αυτήν που κάνει ο αλγόριθμος που περιγράφηκε στο πιο πάνω υποκεφάλαιο (υποκεφάλαιο 3.2). Ο αλγόριθμος του υποκεφαλαίου 3.2 δέχεται σαν είσοδο κάποια θεωρία και βρίσκει όλες τις ευσταθείς επεκτάσεις της, ενώ ο αλγόριθμος που θα παρουσιαστεί πιο κάτω βρίσκει μια θεωρία με βάση τις ευσταθείς επεκτάσεις που παίρνει σαν είσοδο. Επομένως, στόχος μας και σε αυτό τον αλγόριθμο είναι να μοντελοποιήσουμε τις ιδιότητες της απουσίας συγκρούσεων και συλλογικής άμυνας με στόχο αυτή την φορά την δημιουργία κάποιας θεωρίας.

Ας δούμε πως μπορούμε να κάνουμε αυτές τις μοντελοποιήσεις:

Μοντελοποίηση απουσίας συγκρούσεων:

Επιχειρήματα τα οποία ανήκουν στην ίδια ευσταθή επέκταση, δεν πρέπει να επιτίθεται το ένα στο άλλο, δηλαδή δεν πρέπει να υπάρχει οποιαδήποτε ακμή μεταξύ τους.

Επιπλέον, ένα σημαντικό σημείο στο οποίο πρέπει να εστιάσουμε και να δώσουμε την απαραίτητη προσοχή είναι η περίπτωση στην οποία πρέπει να είμαστε βέβαιοι ότι κάποιο επιχειρήμα δεν επιτίθεται στον εαυτό του, γιατί αν συνέβαινε κάτι τέτοιο, τότε η θεωρία δεν θα είχε καμία ευσταθή επέκταση. Άρα πρέπει να λάβουμε στα υπόψη μας και να αποτρέψουμε τέτοιες πιθανές επιθέσεις.

Μοντελοποίηση συλλογικής άμυνας:

Εξετάζουμε μία προς μία της ευσταθής επεκτάσεις. Σε κάθε περίπτωση τα επιχειρήματα που δεν ανήκουν στην ευσταθή επέκταση θα πρέπει να δέχονται πιθανές επιθέσεις από τα επιχειρήματα που ανήκουν στην επέκταση. Αν θέλαμε να το δούμε και από την άλλη μεριά, τότε θα πρέπει όλα ή κάποια από τα επιχειρήματα που ανήκουν στην επέκταση να επιτίθενται σ' αυτά που δεν ανήκουν. Άρα θα πρέπει να υπάρχουν κάποιες ή όλες οι ακμές από τα επιχειρήματα που ανήκουν στις επεκτάσεις στα επιχειρήματα που δεν ανήκουν.

Στο πιο κάτω παράδειγμα εφαρμόζονται οι πιο πάνω μοντελοποιήσεις.

Παράδειγμα κατανόησης 3.7:

Έστω ότι μας δίνονται οι δύο ακόλουθες ευσταθής επεκτάσεις: $ST_1 = \{1,5\}$ και $ST_2 = \{2,3,6\}$ και έστω ότι υπάρχουν συνολικά επτά επιχειρήματα.

Στο σημείο αυτό, ας ορίσουμε τη σημασιολογία του συμβόλου X_{ij} . Όταν γίνεται αναφορά σ' αυτό το σύμβολο τότε θα υπονοούμε ότι το επιχείρημα i κάνει επίθεση στο επιχείρημα j (δηλαδή ότι πρέπει να υπάρχει ακμή από το i στο j), ενώ όταν αναφερόμαστε στο $\neg X_{ij}$ θα υπονοούμε ότι το i δεν πρέπει να επιτίθεται στο j (δηλαδή ότι δεν πρέπει να υπάρχει ακμή από το i στο j).

Μοντελοποίηση απουσίας συγκρούσεων:

Δεν πρέπει να υπάρχουν συγκρούσεις μεταξύ των επιχειρημάτων 1 και 5, αφού ανήκουν στην ίδια ευσταθή επέκταση, καθώς επίσης ούτε μεταξύ των επιχειρημάτων 2,3 και 6.

Επομένως προκύπτουν τα εξής:

Για το σύνολο ST_1 : $\neg X_{1,1}$, $\neg X_{1,5}$, $\neg X_{5,5}$, $\neg X_{5,1}$

Για το σύνολο ST_2 : $\neg X_{2,2}$, $\neg X_{2,3}$, $\neg X_{2,6}$, $\neg X_{3,2}$, $\neg X_{3,3}$, $\neg X_{3,6}$, $\neg X_{6,2}$, $\neg X_{6,3}$,
 $\neg X_{6,6}$

Με αυτό τον τρόπο είμαστε σίγουροι ότι δεν θα μπουν ακμές που να παραβιάζουν την ιδιότητα της απουσίας συγκρούσεων.

Μοντελοποίηση συλλογικής άμυνας:

Κάθε επιχείρημα που δεν ανήκει σε κάθε ευσταθή επέκταση θα πρέπει να δέχεται επίθεση από κάποιο που ανήκει. Έτσι για το σύνολο $S\mathbb{I}_2$ τα επιχειρήματα 1,4,5 και 7 πρέπει να δέχονται επίθεση από τα επιχειρήματα 2,3 και 6. Επομένως θα έχουμε το εξής: $\neg X_{2,1} \rightarrow X_{3,1} \vee X_{6,1}$. Με αυτό σημαίνει ότι αν δεν μπει ακμή από το επιχείρημα 2 στο επιχείρημα 1, τότε αναγκαστικά πρέπει να μπει ακμή από το 3 στο 1 ή από το 6 στο 1. Εφαρμόζοντας τον κανόνα $\neg A \rightarrow B \leftrightarrow A \vee B$ προκύπτει το εξής:

$X_{2,1} \vee X_{3,1} \vee X_{6,1}$. Με τον ίδιο τρόπο προκύπτουν τα ακόλουθα: $X_{2,4} \vee X_{3,4} \vee X_{6,4}$, $X_{2,5} \vee X_{3,5} \vee X_{6,5}$ και $X_{2,7} \vee X_{3,7} \vee X_{6,7}$.

Για το σύνολο $S\mathbb{I}_1$ τα επιχειρήματα 2,3,4,6 και 7 πρέπει να δέχονται επίθεση από τα επιχειρήματα 1 και 5. Επομένως χρησιμοποιώντας την ίδια λογική με πιο πάνω πρέπει να προκύψει το ακόλουθο:

$X_{1,2} \vee X_{5,2}$, $X_{1,3} \vee X_{5,3}$, $X_{1,4} \vee X_{5,4}$, $X_{1,6} \vee X_{5,6}$, $X_{1,7} \vee X_{5,7}$.

Ας μην ξεχνάμε, όμως πως δεν επιθυμούμε μια οποιαδήποτε θεωρία, αλλά την ελάχιστη θεωρία, δηλαδή την θεωρία με τις λιγότερες ακμές. Για τον σκοπό αυτό θα κάνουμε χρήση ενός προτασιακού επιλυτή που επιλύει προβλήματα σταθμικής προτασιακής βελτιστότητας (weighted MaxSat problem-weighted Maximum Satisfiability problem). Συγκεκριμένα θα χρησιμοποιήσουμε το maxino επιλυτή.

Στο σημείο αυτό ας δούμε τι είναι το πρόβλημα προτασιακής βελτιστότητας (maxSat problem). Το συγκεκριμένο πρόβλημα αποτελεί γενίκευση του προβλήματος της προτασιακής ικανοποιησιμότητας (SAT problem), αφού σκοπός είναι δοθείσας μιας κανονικής συζευκτικής μορφής να ελεγχθεί αν υπάρχει κάποια ανάθεση τιμών στις μεταβλητές, τέτοια ώστε να ικανοποιεί τους περισσότερους όρους (clauses) της. Η προτασιακή βελτιστότητα μπορεί να γενικευτεί στην σταθμικά προτασιακή βελτιστότητα (weighted MaxSat), όπου εφαρμόζονται βάρη στους όρους (clauses). Το βάρος αυτό υποδεικνύει ότι η ικανοποίηση ενός όρου είναι πιο σημαντική από την

ικανοποίηση άλλων όρων. Στόχος σε αυτή την γενίκευση είναι να ελαχιστοποιηθεί το συνολικό βάρος των μη-ικανοποιήσιμων όρων[6].

Στο σημείο αυτό, ας κοιτάξουμε πιο επιγραμματικά τον maxino επιλυτή. Ο maxino επιλυτής επιλύει προβλήματα σταθμικής προτασιακής βελτιστότητας κάνοντας χρήση ελαστικών όρων (hard clauses) και αυστηρών όρων (soft clauses). Συγκεκριμένα ελαστικοί όροι είναι οι όροι που μη τυχόν ικανοποίηση τους δεν μας επηρεάζει, ενώ αυστηροί όροι είναι οι όροι που θέλουμε οπωσδήποτε να ικανοποιηθούν. Στόχος του επιλυτή είναι να μεγιστοποιήσει το άθροισμα των βαρών των ικανοποιήσιμων όρων. Με άλλα λόγια στόχος του είναι να ελαχιστοποιήσει το άθροισμα των βαρών των μη-ικανοποιήσιμων όρων. Όπως αντιλαμβανόμαστε οι ελαστικοί όροι θα πρέπει να έχουν βάρη μικρότερα από ότι οι αυστηροί όροι. Επιπλέον το άθροισμα των βαρών των ελαστικών όρων πρέπει να είναι μικρότερο από το άθροισμα των βαρών των αυστηρών όρων. Τα βάρη που ανατίθενται στους όρους πρέπει να είναι θετικές ακέραιες τιμές[7].

Στο σημείο αυτό, ας δούμε πως πρέπει να αλλάξουμε τους κανόνες μοντελοποιήσεις των ιδιοτήτων των ευσταθών επεκτάσεων, έτσι ώστε να μας επιστρέφεται η ελάχιστη θεωρία. Αρχικά πρέπει να εντοπίσουμε ποιοι όροι-κανόνες (clauses) πρέπει σίγουρα να ικανοποιηθούν και ποιοι κανόνες που μη τυχόν ικανοποίηση τους δεν μας επηρεάζει και τόσο. Είναι εύκολο να δούμε πως αφού μας ενδιαφέρει να βρούμε τις ακμές μεταξύ των επιχειρημάτων οι κανόνες που προσθέτουν αυτές τις ακμές θα έχουν μεγαλύτερο βάρος από τους κανόνες που αποτρέπουν πιθανές ακμές μεταξύ των επιχειρημάτων. Πιο συγκεκριμένα οι κανόνες που προκύπτουν από την μοντελοποίηση της ιδιότητας απουσίας συγκρούσεων πρέπει να έχουν μικρότερο βάρος από αυτές που προκύπτουν από την ιδιότητα συλλογικής άμυνας. Ακόμη πρέπει να προστεθούν επιπλέον κανόνες εκτός από τους κανόνες που προκύπτουν από τις δύο ιδιότητες των ευσταθών επεκτάσεων για να μας δοθούν οι ελάχιστες ακμές. Αυτοί οι επιπλέον κανόνες σχετίζονται με τους κανόνες της συλλογικής άμυνας. Συγκεκριμένα παίρνουμε ένα προς ένα τους κανόνες της συλλογικής άμυνας και για κάθε ένα από αυτούς δημιουργούμε τόσους επιπλέον κανόνες όσες είναι και οι μεταβλητές που τους απαρτίζουν και οι κανόνες αυτοί αποτελούνται από μία και μόνο μεταβλητή και συγκριμένα αποτελούνται από την άρνηση των συγκεκριμένων μεταβλητών. Στους κανόνες αυτούς βάζουμε μικρό βάρος. Με τον τρόπο αυτό, δηλαδή προσθέτοντας αυτούς τους κανόνες και

βάζοντας μικρό βάρος τόσο σε αυτούς όσο και στους κανόνες που προκύπτουν από την απουσία συγκρούσεων, κάνουμε όλους τους κανόνες που σχετίζονται με την απουσία ακμής να έχουν μικρότερη προτεραιότητα ικανοποίησης, έτσι ώστε σε περίπτωση που υπάρχει κάποιος κανόνας που προσθέτει κάποια ακμή και αποτελείται από μία μεταβλητή, η μεταβλητή αυτή να γίνεται αληθής, αφού ο κανόνας αυτός έχει μεγαλύτερο βάρος και έτσι θα προστίθεται η ακμή που αντιπροσωπεύεται από αυτή τη μεταβλητή. Ακόμη στην περίπτωση όπου κάποιος κανόνας που σχετίζεται με την πρόσθεση ακμών απαρτίζεται από πολλές μεταβλητές, δηλαδή αντιπροσωπεύει την πρόσθεση πολλών ακμών, μόνο μία μεταβλητή του κανόνα θα γίνει αληθής και αυτό γιατί ο maximo επιλυτής προσπαθεί να ικανοποιήσει όσους πιο πολλούς κανόνες μπορεί και έτσι πάλι μόνο μία ακμή θα προστεθεί. Θέλοντας να γίνουμε ποιο συγκεκριμένοι στο θέμα της ανάθεση των βαρών στους κανόνες αναφέρουμε το εξής: όλοι οι κανόνες που προκύπτουν από την ιδιότητα της απουσίας συγκρούσεων, καθώς επίσης και οι επιπλέον κανόνες που αναφέραμε πριν λίγο πρέπει να έχουν βάρος ίσο με ένα, ενώ οι κανόνες που προκύπτουν από την ιδιότητα συλλογικής άμυνας πρέπει να έχουν βάρος ίσο με το συνολικό αριθμό των όρων (clauses), δηλαδή όσο είναι και το πλήθος των όρων. Με αυτό τον τρόπο εξασφαλίζουμε ότι ισχύει και η συνθήκη όπου θέλει το άθροισμα των βαρών αυτών που δεν μας πειράζει αν ικανοποιηθούν ή όχι να είναι μικρότερο από το άθροισμα των βαρών αυτών που θέλω σίγουρα να ικανοποιηθούν.

Πιο κάτω παρουσιάζεται η εφαρμογή της προσαρμογής των βαρών.

Παράδειγμα κατανόησης 3.8:

Έστω ότι μας δίνεται η ακόλουθη ευσταθής επέκταση: $S \uparrow_1 = \{1,5\}$ και έστω ότι υπάρχουν συνολικά επτά επιχειρήματα.

Τώρα ας εξετάσουμε πως αλλάζει η κάθε μία από τις μεθόδους ~~μ~~^X μοντελοποίησης των κανόνων των ευσταθών επεκτάσεων για εντοπισμό της ελάχιστης θεωρίας, δηλαδή μιας θεωρίας που να αποτελείται από τις λιγότερες ακμές.

Μοντελοποίηση απουσίας συγκρούσεων:

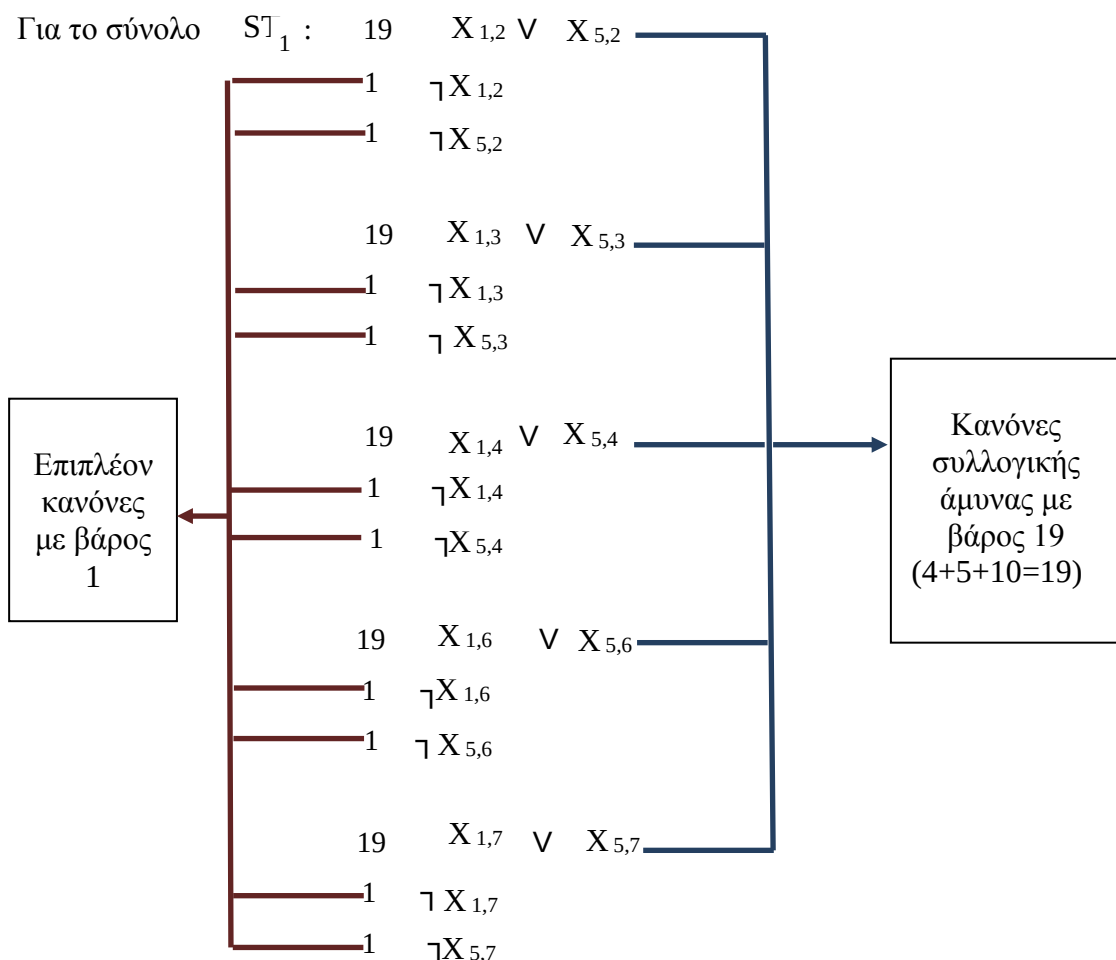
Δεν πρέπει να υπάρχουν συγκρούσεις μεταξύ των επιχειρημάτων 1 και 5, αφού ανήκουν στην ίδια ευσταθή επέκταση.

Επομένως προκύπτουν τα εξής:

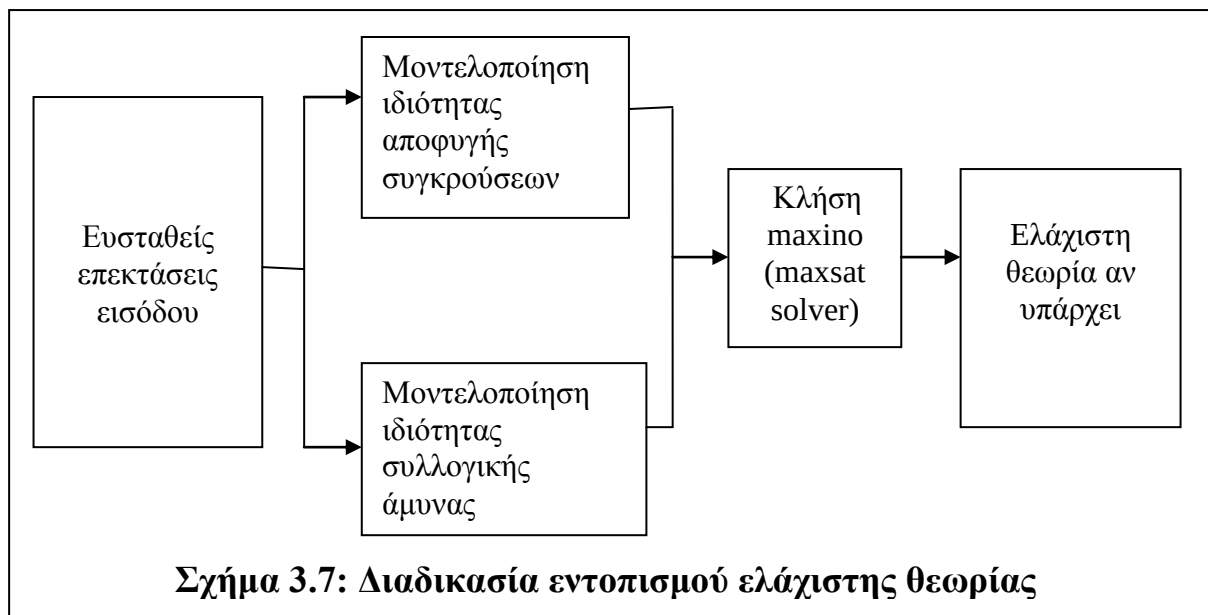
$$\text{Για το σύνολο } ST_1 : \left. \begin{array}{l} 1 \neg X_{1,1} \\ 1 \neg X_{1,5} \\ 1 \neg X_{5,1} \\ 1 \neg X_{5,5} \end{array} \right\} \begin{array}{l} \text{όπου 1 είναι το βάρος} \\ \text{των κανόνων} \end{array}$$

Μοντελοποίηση συλλογικής άμυνας:

Κάθε επιχείρημα που δεν ανήκει στην ευσταθή επέκταση πρέπει να δέχεται επίθεση από κάποιο που ανήκει στην ευσταθή επέκταση. Το βάρος των επιπλέον κανόνων πρέπει να είναι 1 και το βάρος των κανόνων της συλλογικής άμυνας να είναι ίσο με πλήθος των κανόνων του προβλήματος, δηλαδή πλήθος κανόνων απουσίας συγκρούσεων + πλήθος κανόνων συλλογικής άμυνας + επιπλέον κανόνες.



Στο σχήμα που ακολουθεί (βλέπε σχήμα 3.7) παρουσιάζεται διαγραμματικά η διαδικασία εύρεσης της ελάχιστης θεωρίας, δοθέντος κάποιων ευσταθών επεκτάσεων.



3.4 Σύγκριση επιχειρηματικών πλαισίων με βάση τις ευσταθείς επεκτάσεις

Ο αλγόριθμος αυτός έχει σαν στόχο να εντοπίσει τη διαφορά που υπάρχει μεταξύ δύο θεωριών. Βασικό κριτήριο με το οποίο θα γίνει αυτή η σύγκριση είναι οι ευσταθείς επεκτάσεις των θεωριών. Πιο συγκεκριμένα ο αλγόριθμος θα κοιτάζει να βρει πόσες διαφορετικές ευσταθείς επεκτάσεις υπάρχουν μεταξύ των δύο θεωριών. Το αποτέλεσμα που θα μας δίνει αυτός ο αλγόριθμος είναι ένας θετικός αριθμός, ο οποίος θα αντιπροσωπεύει τη διαφορά που υπάρχει μεταξύ των θεωριών. Αν ο αριθμός αυτός είναι ίσος με 0, τότε οι δύο θεωρίες είναι ίδιες, δηλαδή όσο πιο μικρός είναι ο αριθμός τόσο πιο όμοιες είναι οι δύο θεωρίες.

Υπάρχουν πολλοί διαφορετικοί αλγόριθμοι με τους οποίους μπορούμε να βρούμε την διαφορά μεταξύ των θεωριών. Ο καθένας από αυτούς έχει τον δικό του παράγοντα με τον οποίο μετρά αυτή την διαφορά. Ο συγκεκριμένος αλγόριθμος για παράδειγμα στηρίζεται στις ευσταθείς επεκτάσεις των θεωριών.

Στο σημείο αυτό ας κοιτάξουμε τον αλγόριθμο με περισσότερη λεπτομέρεια. Ο αλγόριθμος αυτός δέχεται σαν είσοδο τις ευσταθείς επεκτάσεις των δύο θεωριών που είναι επιθυμητή η σύγκριση τους για εντοπισμό της διαφοράς τους. Ο αλγόριθμος προβαίνει στην σύγκριση των δύο θεωριών με βάση αυτή την γνώση-είσοδο που δέχεται. Αν επιθυμούμε να ενισχύσουμε αυτή τη γνώση, έτσι ώστε να προσθέσουμε επιπλέον σιγουριά για τα αποτελέσματα μας, μπορούμε να βρούμε τις ευσταθείς επεκτάσεις των θεωριών που προκύπτουν μετά από την αφαίρεση κάποιων υποσύνολων από επιχειρήματα. Αυτά τα υποσύνολα δεν αποτελούνται από οποιαδήποτε επιχειρήματα, αλλά απαρτίζονται από επιχειρήματα που ανήκουν στις δύο θεωρίες. Τα υποσύνολα που αφαιρούνται κάθε φορά πρέπει να είναι τα ίδια και για τις δύο θεωρίες. Ο τρόπος με τον οποίο εντοπίζουμε τις ευσταθείς επεκτάσεις των θεωριών είναι ο ίδιος με αυτόν που περιγράφηκε στον υποκεφάλαιο 3.2. Σε αυτό το σημείο μπορούμε να κάνουμε την εξής παρατήρηση: ο αλγόριθμος που περιγράφουμε τώρα δέχεται σαν είσοδο του το αποτέλεσμα του αλγορίθμου που περιγράφηκε στο υποκεφάλαιο 3.2.

Συνεχίζοντας την περιγραφή του αλγορίθμου, ο αλγόριθμος έχοντας τη γνώση που χρειάζεται αρχίζει να την επεξεργάζεται, συγκρίνοντας τις ευσταθείς επεκτάσεις των δύο θεωριών. Πιο ειδικά εξετάζει αν κάθε ευσταθής επέκταση της πρώτης θεωρίας αποτελεί και ευσταθή επέκταση της δεύτερης θεωρίας. Συγκεκριμένα ο αλγόριθμος εξετάζει μία προς μία τις ευσταθείς επεκτάσεις της πρώτης θεωρίας και προσπαθεί να τις αντιστοιχίσει με κάποια από τις ευσταθείς επεκτάσεις της δεύτερης θεωρίας και το αντίστροφο. Η αντιστοίχιση υπάρχει μόνο σε περίπτωση που η ευσταθής επέκταση υπάρχει και στις δύο θεωρίες. Σε περίπτωση όπου κάποια ευσταθής επέκταση δεν αποτελεί ευσταθής επέκταση και για τις δύο θεωρίες τότε η διαφορά των δύο θεωριών αυξάνεται κατά ένα.

Ένα σημείο στο οποίο πρέπει να στρέψουμε την προσοχή μας είναι το γεγονός ότι πρέπει να είμαστε βέβαιοι πως κάθε φορά συγκρίνουμε τις αντίστοιχες ευσταθείς επεκτάσεις, δηλαδή συγκρίνουμε τις επεκτάσεις που προκύπτουν από την αφαίρεση του αντίστοιχου συνόλου από επιχειρήματα.

Ακολουθεί ένα παράδειγμα εφαρμογής του πιο πάνω αλγορίθμου.

Παράδειγμα κατανόησης 3.9:

Έστω ότι μας δίνονται τα ακόλουθα δεδομένα εισόδου για τον αλγόριθμο:

Δεδομένα σχετικά με την πρώτη θεωρία:

Ευσταθείς επεκτάσεις χωρίς την αφαίρεση
κάποιου επιχειρήματος:

$$ST1=\{1,2,3\}$$

$$ST2=\{1\}$$

Ευσταθείς επεκτάσεις με την αφαίρεση
αφαίρεση του συνόλου $A=\{1\}$:

$$ST3=\{2,3\}$$

Ευσταθείς επεκτάσεις με την αφαίρεση
του συνόλου $B=\{2,3\}$:

$$ST4=\{3,5\}$$

$$ST5=\{4\}$$

Δεδομένα σχετικά με την δεύτερη θεωρία:

Ευσταθείς επεκτάσεις χωρίς την
αφαίρεση κάποιου επιχειρήματος:

$$ST6=\{1,2,3\}$$

Ευσταθείς επεκτάσεις με την
αφαίρεση του συνόλου $A=\{1\}$:

$$ST7=\{2\}, \quad ST8=\{2,3\}$$

Ευσταθείς επεκτάσεις με την
αφαίρεση του συνόλου $B=\{2,3\}$:

$$ST9=\{3,5\}$$

$$ST10=\{4\}$$

$$ST11=\{1\}$$

Ας δούμε την εφαρμογή του αλγορίθμου στο πιο πάνω παράδειγμα. Αρχικά θα συγκριθούν μεταξύ τους οι ευσταθείς επεκτάσεις των θεωριών που προέκυψαν χωρίς την αφαίρεση κάποιου επιχειρήματος από τις θεωρίες. Θα συγκριθεί το $ST1$ μαζί με το $ST6$. Τα δύο αυτά σύνολα είναι τα ίδια, άρα η διαφορά δεν θα αυξηθεί θα παραμείνει 0 (μηδέν). Ακολούθως θα συγκριθεί το $ST2$ με το $ST6$. Τα δύο αυτά σύνολα δεν είναι τα ίδια, άρα η διαφορά θα αυξηθεί κατά ένα. Μετέπειτα θα συγκριθεί το $ST6$ με τα $ST1$ και $ST2$, όμως η διαφορά δεν θα αλλάξει αφού τα $ST6$ και $ST1$ είναι ίδια σύνολα. Ακολούθως θα συγκριθούν οι ευσταθείς επεκτάσεις που προέκυψαν από την αφαίρεση του συνόλου A (αφαίρεση επιχειρήματος 1). Δηλαδή θα συγκριθεί το $ST3$ με τα $ST7$ και $ST8$, αλλά η διαφορά θα παραμείνει 1 γιατί τα $ST3$ και $ST8$ είναι ίδια σύνολα. Ακόμη θα συγκριθούν τα $ST7$ με το $ST3$, και η διαφορά θα γίνει δύο (θα αυξηθεί κατά ένα αφού δεν είναι τα ίδια σύνολα) και θα συγκριθεί και το $ST8$ με το $ST3$ και η

διαφορά θα παραμείνει η ίδια. Ακολουθώντας με τον ίδιο τρόπο θα γίνουν και οι συγκρίσεις ανάμεσα στις ευσταθείς επεκτάσεις που προκύπτουν από την αφαίρεση του συνόλου B (αφαίρεση επιχειρημάτων 2 και 3. Στο τέλος θα δούμε πως η διαφορά είναι ίση με 3, αφού στις συγκρίσεις που θα γίνουν θα δούμε ότι το σύνολο $ST11$ δεν αντιστοιχείται με το $ST4$ ή το $ST5$.

Άρα οι δύο θεωρίες διαφέρουν κατά 3, δηλαδή έχουν συνολικά τρεις διαφορετικές ευσταθείς επεκτάσεις.

3.5 Σύγκριση επιχειρηματικών πλαισίων με βάση τις δυαδικές σχέσεις επίθεσης

Ο στόχος αυτού του αλγορίθμου είναι ο ίδιος με αυτό του αλγορίθμου που περιγράφηκε στο προηγούμενο υποκεφάλαιο (υποκεφάλαιο 3.4). Αυτός ο αλγόριθμος, λοιπόν, δοθέντος δύο θεωριών εντοπίζει τη διαφορά που υπάρχει μεταξύ των δύο θεωριών. Το αποτέλεσμα αυτού του αλγορίθμου είναι ένας δεκαδικός αριθμός. Αν ο αριθμός αυτός είναι 0, τότε οι δύο θεωρίες είναι όμοιες-ίδιες. Όσο πιο μικρός είναι ο αριθμός τόσο πιο όμοιες είναι οι δύο θεωρίες.

Υπάρχουν πολλοί τρόποι με τους οποίους μπορούμε να κάνουμε την σύγκριση αυτή των θεωριών που αποτελούνται από επιχειρήματα. Κάποιοι από αυτούς στηρίζονται στην σημασιολογία των θεωριών, (όταν λέμε σημασιολογία των θεωριών μπορούμε να αναφερόμαστε για παράδειγμα στις ευσταθείς επεκτάσεις των θεωριών), ενώ κάποιοι άλλοι στηρίζονται στην σύνταξη, δηλαδή την μορφή των θεωριών. Ο αλγόριθμος που θα παρουσιαστεί σε αυτό το υποκεφάλαιο στηρίζεται στην μορφή των θεωριών, αφού προσπαθεί να βρει κατά πόσο διαφέρουν οι δύο θεωρίες κοιτάζοντας τις δυαδικές σχέσεις επίθεσης των δύο αυτών θεωριών.

Στο σημείο αυτό ας θυμηθούμε και πάλι πως ακριβώς ορίζεται η έννοια της θεωρίας. Όπως έχει παρουσιαστεί και στο υποκεφάλαιο 2.2, με τον όρο θεωρία ορίζεται ένα ζεύγος $\langle A, R \rangle$ όπου A είναι το σύνολο των επιχειρημάτων και R το σύνολο δυαδικών σχέσεων επίθεσης πάνω στο σύνολο A όπου $R \subseteq A \times A$. Δηλαδή λέμε ότι το επιχείρημα a επιτίθεται στο επιχείρημα b αν και μόνο αν $(a, b) \in R$.

Τώρα, ας δούμε με περισσότερη λεπτομέρεια τον αλγόριθμο. Ο αλγόριθμος δέχεται σαν είσοδο δύο θεωρίες, οι οποίες απαρτίζονται από τον ίδιο αριθμό επιχειρημάτων. Οι δύο αυτές θεωρίες αποθηκεύονται σε δύο διαφορετικούς δισδιάστατους πίνακες γειτνίασης, με αριθμό γραμμών και στηλών όσα είναι και τα επιχειρήματα που απαρτίζουν τις δύο θεωρίες. Οι διαστάσεις των δύο αυτών πινάκων είναι ίδιες, αφού οι δύο θεωρίες έχουν τον ίδιο αριθμό επιχειρημάτων. Στους πίνακες γίνονται αναθέσεις τιμών με τον ίδιο τρόπο κοιτάζοντας κάθε φορά όμως την αντίστοιχη θεωρία. Δηλαδή για τον πρώτο πίνακα θα κάνουμε αναθέσεις με βάση την πρώτη θεωρία που δέχεται σαν είσοδο ο αλγόριθμος και για τον δεύτερο πίνακα θα στηριχτούμε στην δεύτερη θεωρία. Κάθε θέση του πίνακα παίρνει τιμή 0 ή 1. Συγκεκριμένα αν από το επιχείρημα A επιτίθεται στο επιχείρημα B, τότε στη θέση του πίνακα $[A,B]=1$, ενώ αν δεν υπήρχε κάποια επίθεση μεταξύ των δύο αυτών επιχειρημάτων θα είχαμε στη θέση του πίνακα $[A,B]=0$.

Στη συνέχεια αφού γίνει η πιο πάνω διαδικασία ο αλγόριθμος κοιτάζει να εντοπίσει τον αριθμό των διαφορετικών ζευγών που έχουν αυτές οι δυαδικές σχέσεις. Δηλαδή εντοπίζει πόσες διαφορετικές σχέσεις επίθεσης έχει η πρώτη θεωρία από την δεύτερη και το αντίστροφο. Ακολουθώντας, αφού εντοπίσει τις διαφορετικές σχέσεις επίθεσης των δύο θεωριών, ας ονομάσουμε αυτές τις διαφορετικές σχέσεις *difference*, θα υπολογίσει το μέγιστο αριθμό των πιθανών ζευγών που θα μπορούσε να είχε η θεωρία. Με την ορολογία «μέγιστο αριθμό πιθανών ζευγών» αναφερόμαστε σε όλες τις επιτρεπτές δυαδικές σχέσεις επίθεσης που μπορούν να απαρτίζουν τη θεωρία. Οι μη επιτρεπτές σχέσεις επίθεσης είναι αυτές που το επιχείρημα επιτίθεται είναι το ίδιο με το επιχείρημα το οποίο δέχεται αυτή την επίθεση. Με άλλα λόγια δεν επιτρέπεται ένα επιχείρημα να κάνει επίθεση στον εαυτό του. Αν θέλουμε να κοιτάξουμε πιο επιγραμματικά πως γίνεται αυτός ο υπολογισμός, τότε θα κάνουμε την εξής παρατήρηση: κάθε επιχείρημα μπορεί να επιτίθεται σε όλα τα άλλα επιχειρήματα εκτός από τον εαυτό του. Άρα αν ονομάσουμε τον αριθμό των επιχειρημάτων που αποτελούν την θεωρία *nodes*, τότε ένα επιχείρημα μπορεί να επιτίθεται σε $(nodes-1)$ άλλα επιχειρήματα. Έτσι θέλοντας να υπολογίσουμε όλες τις πιθανές σχέσεις επιθέσεων μια θεωρίας πρέπει να λάβουμε υπόψη μας όλες τις πιθανές επιθέσεις που μπορεί να κάνει το καθένα από τα επιχειρήματα που απαρτίζουν την θεωρία. Ας ονομάσουμε το μέγιστο αριθμό επιθέσεων μιας θεωρίας *attacks*, τότε θα είναι ίσος με $nodes*(nodes-1)$. Μετέπειτα ο αλγόριθμος διαιρεί το *difference* με το *attacks*. Το αποτέλεσμα που προκύπτει από αυτή την πράξη

αντιπροσωπεύει την διαφορά των δύο θεωριών. Όσο πιο μικρό είναι το αποτέλεσμα που βρίσκουμε τόσο πιο «κοντά», δηλαδή όμοιες, είναι οι δύο θεωρίες. Επεξηγώντας, αν το αποτέλεσμα είναι 0 (μηδέν), τότε οι δύο θεωρίες είναι τελείως ίδιες.

Πιο κάτω ακολουθεί ένα παράδειγμα κατανόησης αυτού του αλγορίθμου.

Παράδειγμα κατανόησης 3.10:

Έστω ότι μας δίνεται η θεωρία $AF1 = \langle \{a,b,c,d,e\}, \{(a,b), (b,c), (d,c), (d,e)\} \rangle$. Η θεωρία αυτή αποτελείται από πέντε επιχειρήματα και τέσσερις σχέσεις επίθεσης μεταξύ αυτών των επιχειρημάτων. Έστω ότι μας δίνεται η θεωρία $AF2 = \langle \{a,b,c,d,e\}, \{(a,b), (b,c), (d,c), (e,d), (a,d)\} \rangle$. Η θεωρία αυτή αποτελείται από πέντε επιχειρήματα και πέντε σχέσεις επίθεσης μεταξύ αυτών των επιχειρημάτων. Και έστω ότι μας δίνεται και η θεωρία $AF3 = \langle \{a,b,c,d,e\}, \{(a,b), (c,b), (c,d), (d,e), (a,d)\} \rangle$. Η θεωρία αυτή αποτελείται από πέντε επιχειρήματα και πέντε σχέσεις επίθεσης μεταξύ αυτών των επιχειρημάτων.

Για σκοπούς κατανόησης του αλγορίθμου θα γίνουν δύο συγκρίσεις μεταξύ των πιο πάνω θεωριών. Συγκεκριμένα θα συγκρίνουμε τις θεωρίες $AF1$ και $AF2$ μεταξύ τους, καθώς επίσης και τις θεωρίες $AF1$ και $AF3$ μεταξύ τους.

Σύγκριση θεωριών $AF1$ και $AF2$:

Βήμα 1: Εντοπισμός διαφορετικών σχέσεων επιθέσεων μεταξύ των δύο πιο πάνω θεωριών.

$AF1 - AF2 = \{(d,e)\}$ (η σχέση επίθεσης (d,e) υπάρχει στη θεωρία $AF1$, ενώ δεν υπάρχει στην $AF2$)

$AF2 - AF1 = \{(e,d), (a,d)\}$ (οι σχέσεις επίθεσης (e,d) και (a,d) υπάρχουν στη θεωρία $AF2$, ενώ στην θεωρία δεν υπάρχουν $AF1$)

Βήμα 2: Ακολουθώς προσθέτουμε το πλήθος των στοιχείων που βρήκαμε για να βρούμε τις συνολικές διαφορετικές σχέσεις επίθεσης των δύο θεωριών.

Άρα θα έχουμε: difference = $1+2=3$.

Βήμα 3: Εντοπίζουμε το μέγιστο αριθμό πιθανών ζευγών της θεωρίας

Άρα θα έχουμε $\text{attacks} = 4 \cdot 5 = 20$ (οι δύο θεωρίες αποτελούνται από 5 επιχειρήματα)

Βήμα 4: Διαιρούμε το συνολικό αριθμό διαφορετικών σχέσεων επίθεσης με το μέγιστο αριθμό πιθανών ζευγών της θεωρίας

Άρα θα έχουμε $\text{difference/attacks} = 3/20 = 0.15$

Άρα οι θεωρίες AF1 και AF2 διαφέρουν κατά 0,15.

Σύγκριση θεωριών AF1 και AF3:

Βήμα 1: Εντοπισμός διαφορετικών σχέσεων επιθέσεων μεταξύ των δύο πιο πάνω θεωριών.

$\text{AF1-AF3} = \{(b,c), (d,c)\}$ (η σχέση επίθεσης (b,c) και (d,c) υπάρχει στη θεωρία AF1, ενώ δεν υπάρχει στην AF3)

$\text{AF3-AF1} = \{(c,b), (c,d), (a,d)\}$. (οι σχέσεις επίθεσης (c,b), (c,d) και (a,d) υπάρχουν στη θεωρία AF3, ενώ στην θεωρία δεν υπάρχουν AF1)

Βήμα 2: Ακολουθώς προσθέτουμε το πλήθος των στοιχείων που βρήκαμε για να βρούμε τις συνολικές διαφορετικές σχέσεις επίθεσης των δύο θεωριών.

Άρα θα έχουμε $2+3=5$.

Βήμα 3 Εντοπίζουμε το μέγιστο αριθμό πιθανών ζευγών της θεωρίας

Άρα θα έχουμε $\text{attacks} = 4 \cdot 5 = 20$ (οι δύο θεωρίες αποτελούνται από 5 επιχειρήματα)

Βήμα 4: Διαιρούμε το συνολικό αριθμό διαφορετικών σχέσεων επίθεσης με το μέγιστο αριθμό πιθανών ζευγών της θεωρίας

Άρα θα έχουμε $\text{difference/attacks} = 5/20 = 0.25$

Άρα οι θεωρίες AF1 και AF3 διαφέρουν κατά 0,25.

Κεφάλαιο 4

Σύστημα Υλοποίησης και Πειραματική Αξιολόγηση

4.1 Επεξήγηση συστήματος υλοποίησης	52
4.2 Πειραματική Αξιολόγηση	57

Σε αυτό το κεφάλαιο αρχικά θα γίνει μια λεπτομερής αναφορά στην υλοποίηση του συστήματος, η οποία έχει γίνει σε γλώσσα προγραμματισμού C. Ακολούθως θα παρουσιαστούν τα διάφορα πειράματα που έχουν γίνει στο συγκεκριμένο σύστημα καθώς και τα αποτελέσματα και συμπεράσματα αυτών των πειραμάτων.

4.1 Επεξήγηση συστήματος υλοποίησης

Το σύστημα υλοποίησης αποτελείται από τους αλγόριθμους οι οποίοι παρουσιάστηκαν στο κεφάλαιο 3. Συγκεκριμένα έγινε υλοποίηση του αλγορίθμου εντοπισμού όλων των ευσταθών επεκτάσεων δοθέντος μια θεωρίας, του αλγορίθμου δημιουργίας ελάχιστης θεωρίας δοθέντος ενός συνόλου από ευσταθείς επεκτάσεις και των αλγορίθμων που στοχεύουν στον εντοπισμό της διαφοράς που υπάρχει μεταξύ δύο δοθέντων θεωριών της επιχειρηματολογίας.

Γενικά το πρόγραμμα έχει σαν στόχο δοθέντος μιας θεωρίας να αντλήσει όση πιο πολλή γνώση μπορεί από αυτήν. Η γνώση αυτή χαρακτηρίζει την συγκεκριμένη θεωρία, άρα με τον τρόπο αυτό πετυχαίνεται η εκμάθηση της θεωρίας, ο οποίος ήταν και ο βασικός στόχος της παρούσας διπλωματικής εργασίας. Για να το επιτύχει αυτό δημιουργεί μια τυχαία θεωρία και ακολούθως παίρνει γνώση από αυτή. Στη συνέχεια κάνοντας χρήση

αυτής της γνώσης που άντλησε δημιουργεί μία καινούργια θεωρία. Στο επόμενο στάδιο εντοπίζει κατά πόσο είναι όμοιες αυτές οι δύο θεωρίες εκτελώντας την σύγκριση μεταξύ τους.

Αναλυτικά το πρόγραμμα λειτουργεί ως εξής: Αρχικά το πρόγραμμα δέχεται σαν είσοδο από το χρήστη μέσω του πληκτρολογίου τον αριθμό των τυχαίων θεωριών-γραφημάτων που επιθυμεί να δημιουργηθούν, καθώς και το πλήθος των κόμβων που θα απαρτίζουν τα γραφήματα. Ο αριθμός των κόμβων του γράφου αντιστοιχεί στο πλήθος των επιχειρημάτων της θεωρίας, ενώ οι ακμές του γράφου αντιστοιχούν στις δυαδικές σχέσεις επίθεσης της θεωρίας. Επιπλέον, ο χρήστης καλείται να δώσει και την πυκνότητα του γράφου, η οποία πρέπει να είναι μία δεκαδική τιμή μικρότερη ίση από το ένα. Η πυκνότητα θα χρησιμοποιηθεί για να εντοπιστεί ο συνολικός αριθμός των ακμών που θα υπάρχουν στο γράφο. Συγκεκριμένα ο αριθμός των ακμών είναι ίσος με την πυκνότητα επί τον μέγιστο αριθμό πιθανών ζευγών της θεωρίας, δηλαδή τον αριθμό των ακμών του πλήρους γραφήματος. Όσο πιο μεγάλη είναι η πυκνότητα, τόσο πιο πολλές σχέσεις επίθεσης, δηλαδή ακμές, θα υπάρχουν στο γράφημα. Δηλαδή όταν η πυκνότητα είναι μηδέν, τότε δεν υπάρχουν ακμές στο γράφο, ενώ όταν η πυκνότητα είναι ένα, τότε υπάρχουν όλες οι δυνατές ακμές που μπορούν να υπάρχουν στο γράφο.

Ακολουθώντας έχοντας ως γνώση τον αριθμό των κόμβων και τον αριθμό των ακμών του γράφου υπολογίζεται η τυχαία θεωρία-γράφο. Για το σκοπό αυτό γίνεται χρήση ενός δισδιάστατου πίνακα γειτνίασης, όπου τοποθετούνται με τυχαίο τρόπο οι ακμές μεταξύ των επιχειρημάτων. Συγκεκριμένα, γίνεται χρήσης μιας έτοιμης συνάρτησης της γλώσσας C η οποία επιστρέφει τυχαίους αριθμούς και επιλέγονται δύο τυχαίοι αριθμοί από το μηδέν μέχρι και το πλήθος των κόμβων που αποτελούν τη θεωρία. Αυτοί οι δύο αριθμοί αντιστοιχούν σε μία συγκεκριμένη θέση του πίνακα γειτνίασης. Έστω ότι οι αριθμοί αυτοί είναι i και j . Τότε στη θέση $[i,j]$ θα έχει τιμή ίση με ένα, δηλαδή θα ισχύει $\text{θέση}[i,j]=1$. Αυτό σημαίνει ότι υπάρχει ακμή από το επιχείρημα i στο επιχείρημα j . Αν ισχύει $\text{θέση}[i,j]=0$ τότε αυτό σημαίνει ότι δεν υπάρχει ακμή μεταξύ του i και j . Η τυχαία θεωρία που δημιουργείται καταγράφεται σε ένα αρχείο `randomTheoria.txt`. και εμφανίζεται στην οθόνη.

Ακολουθώς το πρόγραμμα θα εντοπίσει τις ευσταθείς επεκτάσεις της θεωρίας. Στην περίπτωση όπου ο χρήστης επιθυμεί να εκλάβει περισσότερη γνώση, εκτός από τις ευσταθείς επεκτάσεις της αρχικής θεωρίας, πρέπει να δώσει ένα αριθμό, ο οποίος θα αντιπροσωπεύει τον αριθμό των φορών που θα τροποποιηθεί η αρχική θεωρία. Η αρχική θεωρία τροποποιείται, αφαιρώντας κάποια από τα επιχειρήματα της. Συγκεκριμένα αφαιρούνται από αυτήν υποσύνολα επιχειρημάτων που έχουν διαφορετικό πληθικό αριθμό. Γι' αυτό το λόγο ο χρήστης καλείται να δώσει το πλήθος των υποσυνόλων των κόμβων που επιθυμεί να αφαιρεθούν από την θεωρία-γράφο. Έτσι το πρόγραμμα θα του επιστρέψει και τις καινούργιες ευσταθείς επεκτάσεις που θα προκύψουν από την τροποποιημένη θεωρία. Για τον εντοπισμό των ευσταθών επεκτάσεων γίνεται η μοντελοποίηση των ιδιοτήτων των ευσταθών επεκτάσεων όπως περιγράφηκαν στο υποκεφάλαιο 3.2. Η μοντελοποίηση αυτή καταγράφεται σε ένα αρχείο το οποίο δίνεται σαν είσοδος στον προτασιακό επιλυτή, δηλαδή στο *lingeling*. Η λύση που δίνει ο επιλυτής είναι μία ευσταθής επέκταση της θεωρίας. Έστω ότι μία λύση που μας δίνεται από το *lingeling* είναι $\{1\ 2\ -3\}$. Τότε οι κόμβοι 1 και 2 αποτελούν μία ευσταθής επέκταση. Ακολουθώς θα προστεθεί η άρνηση της λύσης αυτής στο αρχείο μοντελοποίησης και θα δοθεί ξανά στο επιλυτή για να βρει μια καινούργια ευσταθή επέκταση. Αυτό γίνεται μέχρι να μας δώσει ο επιλυτής “UNSATISFALIBLE”. Με αυτό τον τρόπο παίρνουμε όλες τις ευσταθείς επεκτάσεις. Η άρνηση συμβολίζεται με «-». Οι επεκτάσεις που μας δίνονται ως λύση γράφονται σε ένα αρχείο *extensions.txt*.

Το αρχείο *extensions.txt* δίνεται σαν είσοδος στο πρόγραμμα που υπολογίζει την ελάχιστη θεωρία. Στο σημείο αυτό το πρόγραμμα ακολουθεί τους κανόνες μοντελοποίησης των ιδιοτήτων των ευσταθών επεκτάσεων, όπως παρουσιάστηκαν στο υποκεφάλαιο 3.3 και καταγράφει αυτή την μοντελοποίηση σε ένα αρχείο το οποίο δίνει σαν είσοδο στο *maximo* επιλυτή. Ακολουθώς ο επιλυτής θα δώσει την καινούργια θεωρία που θα προκύψει, η οποία θα υπακούει στις ευσταθείς επεκτάσεις της τυχαίας θεωρίας, αφού χτίστηκε με βάση αυτές. Αν βρέθηκε κάποια καινούργια θεωρία, τότε καταγράφεται σ' ένα αρχείο *newTheoria.txt* και εμφανίζεται στην οθόνη.

Στο σημείο αυτό, αφού υπολογίστηκαν οι δύο θεωρίες, τυχαία και καινούργια θεωρία που δημιουργήθηκε με βάση τις ευσταθείς επεκτάσεις της τυχαίας θεωρίας, θα πρέπει να

γίνει σύγκριση τους για να εντοπιστεί η διαφορά που υπάρχει μεταξύ τους. Αρχικά θα γίνει σύγκριση των δύο θεωριών με βάση τις ευσταθείς επεκτάσεις τους. Συγκεκριμένα αν είναι η πρώτη φορά που θα συγκριθεί η τυχαία θεωρία με κάποια καινούργια θεωρία, τότε ο χρήστης καλείται να δώσει το πλήθος των υποσυνόλων των επιχειρημάτων που επιθυμεί να αφαιρεθούν και από τις δύο θεωρίες. Τα υποσύνολα αυτά είναι τα ίδια και για τις δύο θεωρίες, έτσι ώστε να τους αφαιρεθούν τα ίδια επιχειρήματα. Τα υποσύνολα αυτά καταγράφονται σε ένα αρχείο iprosinola.txt, το οποίο θα χρησιμοποιηθεί για τις επόμενες συγκρίσεις που θα γίνουν μεταξύ της τυχαίας θεωρίας και των καινούργιων θεωριών που θα προκύψουν. Ακολούθως εντοπίζονται οι ευσταθείς επεκτάσεις των θεωριών με τον ίδιο τρόπο που υπολογίστηκαν και προηγουμένως οι ευσταθείς επεκτάσεις της τυχαίας θεωρίας. Για κάθε μία από τις δύο θεωρίες δημιουργείται ένα αρχείο όπου καταγράφονται οι επεκτάσεις τους που υπολογίστηκαν. Ακολούθως επεξεργάζονται τα δύο αυτά αρχεία και γίνεται η σύγκριση μεταξύ των επεκτάσεων των δύο θεωριών. Η διαδικασία που εκτελείται για αυτή την σύγκριση έχει ήδη περιγραφεί στο υποκεφάλαιο 3.4. Με το τέλος αυτής της διαδικασίας υπολογίζεται η διαφορά που υπάρχει μεταξύ των δύο θεωριών και η οποία παρουσιάζεται στην οθόνη.

Στη συνέχεια εκτελείται ακόμα μία σύγκριση μεταξύ των δύο θεωριών, η οποία, όμως στηρίζεται στις δυαδικές σχέσεις επίθεσης. Η διαδικασία αυτή επεξεργάζεται τους δύο γράφους, οι οποίοι αντιπροσωπεύουν τις δύο θεωρίες για να εντοπίσει τις ακμές που υπάρχουν μεταξύ των κόμβων κάθε γράφου, δηλαδή εντοπίζει τις επιθέσεις που υπάρχουν μεταξύ των επιχειρημάτων κάθε θεωρίας. Η διαδικασία που ακολουθείται για αυτή την σύγκριση έχει ήδη περιγραφεί στο υποκεφάλαιο 3.5. Με το τέλος αυτής της διαδικασίας και πάλι υπολογίζεται η διαφορά μεταξύ των δύο θεωριών και η οποία παρουσιάζεται στην οθόνη.

Τέλος σε ένα αρχείο data_output.txt καταγράφονται τα τελικά αποτελέσματα. Συγκεκριμένα για κάθε θεωρία που δημιουργείται καταγράφεται ο αριθμός των επιχειρηματιών που την απαρτίζουν, η πυκνότητα της, αν βρέθηκε η ελάχιστη θεωρία (1-αν βρέθηκε και 0-αν δεν βρέθηκε) και η διαφορά που υπάρχει μεταξύ των δύο θεωριών με βάση της δυαδικές σχέσεις και με βάση τις ευσταθείς επεκτάσεις.

Στην πιο κάτω εικόνα (βλέπε εικόνα 4.1) παρουσιάζεται μια πιθανή εκτέλεση του προγράμματος, όπου παρουσιάζεται η αλληλεπίδραση του χρήστη με το σύστημα.

Εικόνα 4.1: Πιθανή εκτέλεση του προγράμματος

```
Dwse ton arithmo twn thewriwn poy epithimeis na dimiourgithoun:
1
Dwse twn arithmo tvn epixeirimatwn pou tha apoteloun ti thewria:
10
Dwse tin piknotita tis thewrias:
0.67
Random thewria:
0 0 1 1 0 1 1 1 1 0
0 0 1 1 0 0 1 1 1 1
1 0 0 0 0 1 0 1 0 1
1 0 1 0 1 1 1 1 1 1
1 1 1 0 0 1 0 0 1 0
0 1 1 1 1 0 1 1 1 1
1 0 1 0 1 1 0 0 1 1
0 1 0 1 1 1 1 0 1 1
1 0 1 1 1 1 0 1 0 1
0 0 0 0 1 0 0 1 1 0
./createRandom
Dwse to plithos twn forwn pou epithimeis gia na ginei afairesi twn epixeirimatwn apo ti thewria:
1
Dwse to plithos twn iposinolwn pou epithimeis na afairethoun:
3
kaninourgia thewria:
0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 0 1 1 0
1 1 0 1 0 1 1 1 1 1
1 1 1 0 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 1 1 1 0
1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
Dwse to plithos twn iposinolwn pou tha afairethoun kai stis dio thewries:
1
./findExt_R_N_IPO
H diafora twn dio thewriwn me vasi tis eustathis epektaseis einai: 1
H diafora twn dio thewriwn me vasi tis diadikes sxeseis epitheseis einai: 0.588889
```

Επεξηγώντας την πιο πάνω πιθανή εκτέλεση βλέπουμε ότι ο χρήστης επιλέγει να δημιουργήσει μόνο μία θεωρία η οποία αποτελείται από 10 επιχειρήματα-nodes με πυκνότητα 0.67. Ακολούθως του παρουσιάζεται η τυχαία θεωρία που δημιουργήθηκε με βάση τα δεδομένα που έδωσε ο ίδιος. Μετέπειτα ο χρήστης επιλέγει να μεταβάλει την αρχική θεωρία μόνο μία φορά, δηλαδή θα αφαίρεση από αυτήν υποσύνολα μόνο μία φορά. Ακολούθως ο χρήστης επιλέγει να αφαιρέσει 3 υποσύνολα επιχειρημάτων. μετά από αυτό το σύστημα εντοπίζει τις ευσταθείς επεκτάσεις της αρχική και μεταβαλλόμενης θεωρίας η οποία προκύπτει από την αφαίρεση των υποσυνόλων και εμφανίζεται στον χρήστη η καινούργια θεωρία που υπολογίστηκε. Ακολούθως ο χρήστης επιλέγει να αφαιρεθεί ένα υποσύνολο και από τις δύο θεωρίες για να γίνει η

σύγκριση με βάση τις ευσταθείς επεκτάσεις. Τέλος υπολογίζεται η διαφορά και με τις δύο μεθόδους και εκτυπώνεται στην οθόνη.

4.2 Πειραματική Αξιολόγηση

Σε αυτό το υποκεφάλαιο θα γίνει μια σύντομη περιγραφή των πειραμάτων που έχουν γίνει. Επιπλέον θα παρουσιαστούν τα αποτελέσματα αυτών των πειραμάτων, καθώς και τα συμπεράσματα στα οποία καταλήξαμε με βάση αυτά τα αποτελέσματα.

Η πειραματική αξιολόγηση η οποία έχει εκτελεστεί αποτελείτο από εννέα διαφορετικές περιπτώσεις. Αυτό που έκανε τις περιπτώσεις να διαφέρουν μεταξύ τους ήταν το διαφορετικό πλήθος επιχειρημάτων που θα αποτελούσαν τις θεωρίες που θα δημιουργούνταν, καθώς και η διαφορετική πυκνότητα που δινόταν γι' αυτές. Για κάθε μία από αυτές τις εννέα περιπτώσεις δημιουργούσαμε 20 διαφορετικές θεωρίες με αριθμό επιχειρημάτων και πυκνότητα ανάλογα με την περίπτωση στην οποία ανήκαν. Οι θεωρίες αυτές ήταν οι αρχικές μας θεωρίες. Κάθε μία από αυτές τις αρχικές θεωρίες τις μεταβάλλαμε 11 φορές, αφαιρώντας από αυτές υποσύνολα επιχειρημάτων. Ο αριθμός των υποσυνόλων που αφαιρούσαμε δεν ήταν ο ίδιος, αλλά κάθε φορά αυξανόταν. Επίσης αρχικά ο πληθικός αριθμός των υποσυνόλων μπορούσε να είναι μεγαλύτερος ίσος με ένα, δηλαδή μπορούσαν να αφαιρεθούν από τις θεωρίες περισσότερα από ένα επιχειρήματα την ίδια στιγμή. Κάθε φορά που μεταβάλλαμε κάθε μία από αυτές τις αρχικές θεωρίες υπολογίζονταν οι ευσταθείς επεκτάσεις τους και ακολούθως δημιουργείτο για κάθε μία από αυτές η αντίστοιχη καινούργια θεωρία τους με βάση τις αντίστοιχες επεκτάσεις τους που υπολογίστηκαν. Μετέπειτα για κάθε αρχική θεωρία υπολογιζόταν η διαφορά που υπήρχε μεταξύ αυτής και των καινούργιων θεωριών που προέκυπταν κάθε φορά. Συνολικά είχαμε 11 τέτοιες συγκρίσεις για κάθε μία από τις αρχικές θεωρίες, αφού κάθε φορά που μεταβαλλόταν η αρχική θεωρία δημιουργείτο μία νέα θεωρία. Κάθε μία από τις αρχικές θεωρίες που δημιουργούνται επεξεργαζόταν ξεχωριστά. Ακόμη για κάθε μία από αυτές τις αρχικές θεωρίες αφαιρούσαμε υποσύνολα με πληθικό αριθμό ίσο με ένα, δηλαδή αφαιρούσαμε μόνο ένα επιχείρημα από την αρχική θεωρία κάθε φορά. Ακολούθως υπολογίζαμε την καινούργια θεωρία με βάση τις ευσταθείς επεκτάσεις που προέκυπταν από την μεταβολή αυτή και

υπολογίζαμε και πάλι την διαφορά που υπήρχε μεταξύ των θεωριών (αρχικής και καινούργιας).

Πριν παρουσιάσουμε τα αποτελέσματα που προέκυψαν από τα πιο πάνω πειράματα ας παραθέσουμε πως αναμέναμε να είναι τα αποτελέσματα που θα προέκυπταν. Η αναμενόμενη συμπεριφορά ήταν η μείωση της διαφοράς που θα υπολογιζόταν για κάθε μία από τις αρχικές θεωρίες, καθώς ο αριθμός των υποσυνόλων που θα αφαιρούνταν από αυτές θα αυξανόταν. Περιμέναμε αυτή τη συμπεριφορά, γιατί υποθέσαμε πως όσο πιο πολλά υποσύνολα της αφαιρούνταν, τόσο πιο πολλή γνώση θα παίρναμε από την αρχική θεωρία. Και έτσι αφού είχαμε πιο πολλή γνώση, τότε η θεωρία η οποία θα χτιζόταν με βάση αυτή την γνώση θα ήταν πολύ κοντά στην αρχική θεωρία. Θέλοντας να κοιτάξουμε μια πιο συγκεκριμένη περίπτωση, όπου η καινούργια θεωρία χτιζόταν με βάση την γνώση που προέκυπτε από την αφαίρεση όλων των πιθανών υποσυνόλων, ευελπιστούσαμε πως η διαφορά που θα υπολογιζόταν θα ήταν μηδενική. Επιπλέον αναμέναμε πως για κάθε θεωρία η διαφορά που θα προέκυπτε από την αφαίρεση υποσυνόλων με πληθικό αριθμό ίσο με ένα θα ήταν μεγαλύτερη από αυτήν που θα προέκυπτε αν τις αφαιρούνταν υποσύνολα με απροσδιόριστο πληθικό αριθμό, δηλαδή υπήρχε η δυνατότητα να αφαιρεθούν από αυτήν την ίδια στιγμή πολλά επιχειρήματα.

Στο σημείο αυτό, αφού έχουμε ορίσει πως περιμέναμε να είναι τα αποτελέσματα που θα προέκυπταν από τα προαναφερθέντα πειράματα, δεν μένει παρά να εξετάσουμε κατά πόσο αυτά ήταν και τα αποτελέσματα τα οποία πήραμε από την εκτέλεση των πειραμάτων.

Πιο κάτω παρουσιάζονται οι γραφικές παραστάσεις με τα αποτελέσματα τα οποία έχουν παρθεί. Όπως έχει αναφερθεί υπάρχουν εννέα διαφορετικές περιπτώσεις. Για κάθε περίπτωση παρατίθενται τρεις διαφορετικές γραφικές παραστάσεις. Η σειρά με την οποία θα δοθούν οι γραφικές παραστάσεις είναι η εξής:

1. Πρώτη γραφική παράσταση: αναπαρίστανται η διαφορά με βάση τις σχέσεις επίθεσης. Τα υποσύνολα που αφαιρούνται κάθε φορά από την αρχική θεωρία για να βρω την καινούργια θεωρία έχουν πληθικό αριθμό μεγαλύτερο από 0.
2. Δεύτερη γραφική παράσταση: αναπαρίστανται η διαφορά με βάση τις διαφορετικές ευσταθείς επεκτάσεις. Τα υποσύνολα που αφαιρούνται κάθε φορά από την αρχική θεωρία για να βρω την καινούργια θεωρία έχουν πληθικό αριθμό μεγαλύτερο από 0.

3. Τρίτη γραφική παράσταση = φαίνεται η διαφορά με βάση τις σχέσεις επίθεσης. Τα υποσύνολα που αφαιρούνται κάθε φορά από την αρχική θεωρία για να βρω την καινούργια θεωρία έχουν πληθικό αριθμό ίσο με 1.

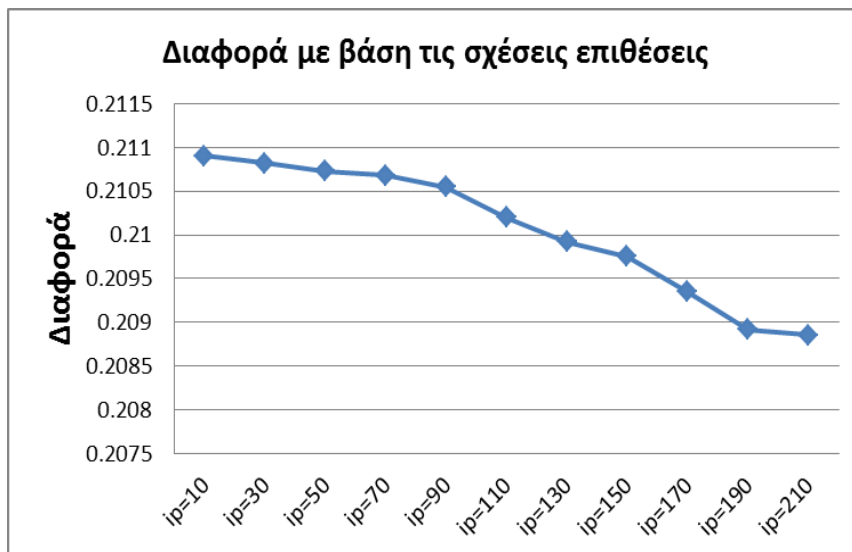
Και στις τρεις γραφικές παραστάσεις στον κάθετο άξονα αναπαρίστανται η υπολογιζόμενη διαφορά, ενώ στον οριζόντιο άξονα φαίνεται ο αριθμός των υποσυνόλων που αφαιρούνταν από την αρχική θεωρία.

Ακόμη πρέπει να αναφερθούμε στον τόπο που υπολογίστηκε η διαφορά σε κάθε περίπτωση. Όπως έχουμε ήδη αναφέρει έχουμε δημιουργήσει 20 διαφορετικές θεωρίες για κάθε περίπτωση. Άρα η διαφορά που υπολογίστηκε ήταν ο μέσος όρος των διαφορών που σημείωνε κάθε μία από αυτές τις 20 διαφορετικές θεωρίες.

Ας παραθέσουμε τις γραφικές παραστάσεις:

1^η Περίπτωση : αριθμός επιχειρημάτων=30, πυκνότητα=0.25

Γραφική Παράσταση 5.1:



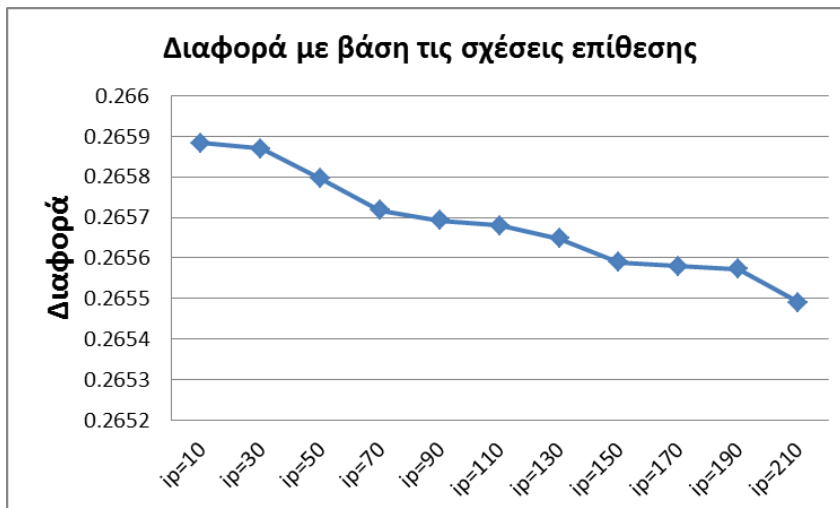
Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετρείται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 30 επιχειρήματα και πυκνότητα 0.25. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.2:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετρίεται με βάση τις ευσταθείς επεκτάσεις. Οι θεωρίες έχουν 30 επιχειρήματα και πυκνότητα 0.25. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.3:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετρίεται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 30 επιχειρήματα και πυκνότητα 0.25. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι ίσος με ένα.

2^η Περίπτωση: αριθμός επιχειρημάτων=30, πυκνότητα=0.50

Γραφική Παράσταση 5.4:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 30 επιχειρήματα και πυκνότητα 0.50. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.5:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις ευσταθείς επεκτάσεις. Οι θεωρίες έχουν 30

επιχειρήματα και πυκνότητα 0.50. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.6:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 30 επιχειρήματα και πυκνότητα 0.50. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι ίσος με ένα.

3^η Περίπτωση: αριθμός επιχειρημάτων=30, πυκνότητα=0.65

Γραφική Παράσταση 5.7:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 30

επιχειρήματα και πυκνότητα 0.65. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.8:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετريέται με βάση τις ευσταθείς επεκτάσεις. Οι θεωρίες έχουν 30 επιχειρήματα και πυκνότητα 0.65. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.9:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετريέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 30

επιχειρήματα και πυκνότητα 0.65. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι ίσος με ένα.

4^η Περίπτωση: αριθμός επιχειρημάτων=50, πυκνότητα=0.25

Γραφική Παράσταση 5.10:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 50 επιχειρήματα και πυκνότητα 0.25. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.11:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις ευσταθείς επεκτάσεις. Οι θεωρίες έχουν 50

επιχειρήματα και πυκνότητα 0.25. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

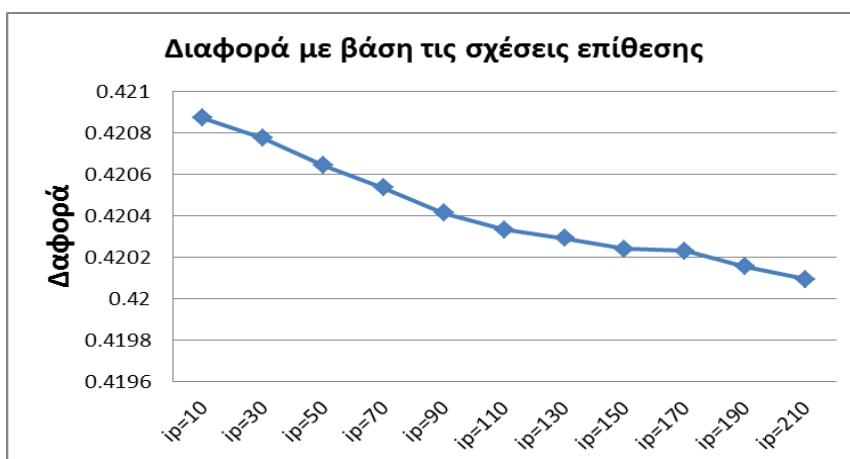
Γραφική Παράσταση 5.12:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 50 επιχειρήματα και πυκνότητα 0.25. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι ίσος με ένα.

5^η Περίπτωση: αριθμός επιχειρημάτων=50, πυκνότητα=0.50

Γραφική Παράσταση 5.13:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 50

επιχειρήματα και πυκνότητα 0.50. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.14:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις ευσταθείς επεκτάσεις. Οι θεωρίες έχουν 50 επιχειρήματα και πυκνότητα 0.50. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.15:

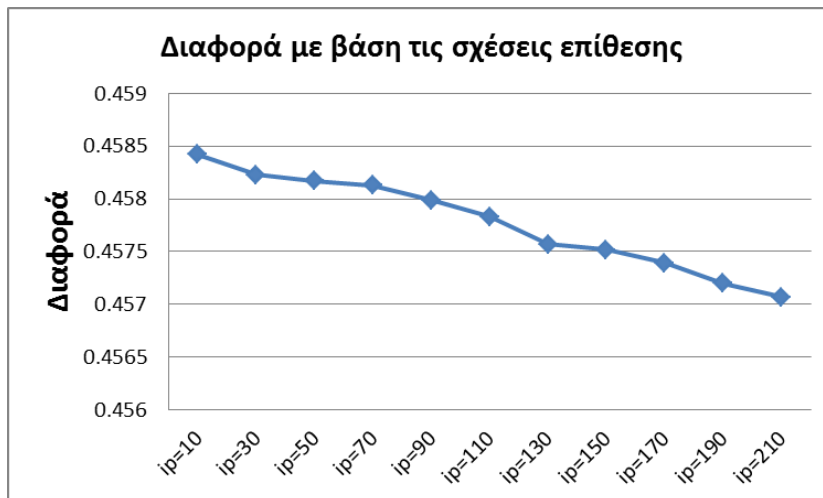


Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 50

επιχειρήματα και πυκνότητα 0.50. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι ίσος με ένα.

6^η Περίπτωση: αριθμός επιχειρημάτων=50, πυκνότητα=0.65

Γραφική Παράσταση 5.16:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετρίεται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 50 επιχειρήματα και πυκνότητα 0.65. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.17:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετρίεται με βάση τις ευσταθείς επεκτάσεις. Οι θεωρίες έχουν 50

επιχειρήματα και πυκνότητα 0.65. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.18:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 50 επιχειρήματα και πυκνότητα 0.65. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι ίσος με ένα.

7^η Περίπτωση: αριθμός επιχειρημάτων=70, πυκνότητα=0.25

Γραφική Παράσταση 5.19:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 70

επιχειρήματα και πυκνότητα 0.25. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 20:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετρείται με βάση τις ευσταθείς επεκτάσεις. Οι θεωρίες έχουν 70 επιχειρήματα και πυκνότητα 0.25. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.21:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετρείται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 70

επιχειρήματα και πυκνότητα 0.25. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι ίσος με ένα.

7^η Περίπτωση: αριθμός επιχειρημάτων=70, πυκνότητα=0.50

Γραφική Παράσταση 5.22:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετρείται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 70 επιχειρήματα και πυκνότητα 0.50. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.23:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετρείται με βάση τις ευσταθείς επεκτάσεις. Οι θεωρίες έχουν 70

επιχειρήματα και πυκνότητα 0.50. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

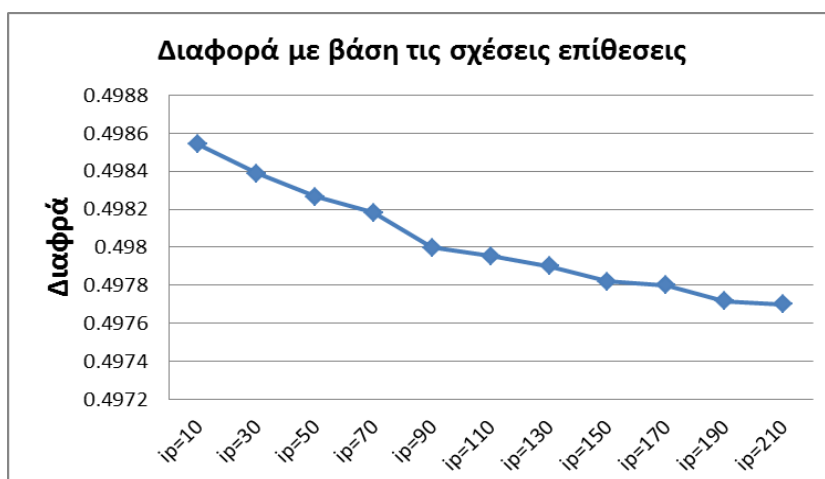
Γραφική Παράσταση 5.24:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετريέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 70 επιχειρήματα και πυκνότητα 0.50. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι ίσος με ένα.

9^η Περίπτωση: αριθμός επιχειρημάτων=70, πυκνότητα=0.65

Γραφική Παράσταση 5.25:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετريέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 70

επιχειρήματα και πυκνότητα 0.65. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.26:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις ευσταθείς επεκτάσεις. Οι θεωρίες έχουν 70 επιχειρήματα και πυκνότητα 0.65. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Γραφική Παράσταση 5.27:



Στην πιο πάνω γραφική παράσταση φαίνεται η διαφορά που σημειώνεται σε σχέση με τον αριθμό των υποσυνόλων που αφαιρούνται από τις αρχικές θεωρίες. Στην περίπτωση αυτή η διαφορά μετριέται με βάση τις δυαδικές σχέσεις επίθεσης. Οι θεωρίες έχουν 70

επιχειρήματα και πυκνότητα 0.65. Ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι μεγαλύτερος ίσος με ένα.

Κοιτάζοντας τις πιο πάνω γραφικές παραστάσεις οδηγούμαστε σε αρκετά συμπεράσματα. Ας παραθέσουμε ένα προς ένα τα συμπεράσματα αυτά.

Αρχικά ας εστιάσουμε την προσοχή μας στις γραφικές παραστάσεις στις οποίες παρουσιάζεται ο τρόπος με τον οποίο μεταβάλλεται η διαφορά που υπολογίζεται για κάθε μία αρχική θεωρία ξεχωριστά και που ο υπολογισμός αυτής της διαφοράς γίνεται με βάση τις σχέσεις επιθέσεις. Παρατηρώντας αυτές τις παραστάσεις οδηγούμαστε στα εξής συμπεράσματα:

- Υπάρχει μία γενική μορφή όσον αφορά τον τρόπο με τον οποίο αλλάζει η διαφορά, καθώς αυξάνεται το πλήθος των υποσυνόλων που αφαιρούνται από τις θεωρίες. Αυτή η μορφή είναι η ίδια και στην περίπτωση όπου αφαιρούνται υποσύνολα με πληθικό αριθμό μεγαλύτερο από μηδέν, αλλά και στην περίπτωση όπου αφαιρούνται υποσύνολα με πληθικό αριθμό ίσο με ένα. Πιο συγκεκριμένα βλέπουμε ότι η διαφορά μειώνεται κάθε φορά που αυξάνεται ο αριθμός των υποσυνόλων που αφαιρούνται. Όπως έχει ήδη αναφερθεί στο υποκεφάλαιο 4.2, αυτή η συμπεριφορά ήταν αναμενόμενη. Επιπλέον στο υποκεφάλαιο 4.2 παρουσιάζεται και ο λόγος για τον οποίο αναμέναμε αυτή την συμπεριφορά.
- Για κάθε μία από τις περιπτώσεις η διαφορά που σημειώνεται όταν ο πληθικός αριθμός των υποσυνόλων που αφαιρούνται είναι ίσος με ένα είναι πιο μεγάλη από την αντίστοιχη διαφορά που υπολογίζεται όταν τα υποσύνολα που αφαιρούνται έχουν πληθικό αριθμό μεγαλύτερο ίσο από ένα. Και πάλι αυτή ήταν μία από τις συμπεριφορές που περιμέναμε πως θα προέκυπτε. Ο λόγος που συμβαίνει αυτό πιθανόν να οφείλεται στο γεγονός ότι αφαιρώντας ταυτόχρονα πολλά επιχειρήματα από μία θεωρία αντλούμε πιο εξιδανικευμένη γνώση, αφού κοιτάζουμε πιο αποσπασματικά την θεωρία αυτή. Ο λόγος που συμβαίνει αυτό είναι γιατί αφαιρώντας πολλά επιχειρήματα μαζί από μία θεωρία τότε το

μέγεθος της θεωρίας που απομένει να εξεταστεί είναι πολύ πιο μικρό από ότι αν αφαιρούσαμε μόνο ένα επιχείρημα.

- Η διαφορά είναι ανάλογη της πυκνότητας που χαρακτηρίζει μία θεωρία. Πιο συγκεκριμένα μπορούμε να κατηγοριοποιήσουμε τις 9 περιπτώσεις του πειράματος σε 3 κατηγορίες με βάση το πλήθος των επιχειρημάτων, δηλαδή σε μία κατηγορία ανήκουν οι περιπτώσεις που έχουν ίδιο αριθμό επιχειρημάτων. Κοιτάζοντας την κάθε κατηγορία ξεχωριστά παρατηρούμε ότι καθώς αυξάνεται η πυκνότητα αυξάνεται και η διαφορά που υπολογίζεται. Πιο συγκεκριμένα σε μια «αραιή» θεωρία, δηλαδή σε μία θεωρία όπου υπάρχουν λίγες επιθέσεις, στην περίπτωση του πειράματος μας αναφερόμαστε στις θεωρίες που είχαν πυκνότητα 0.25, η διαφορά που σημειώνεται είναι μικρότερη από τη διαφορά που υπολογίζεται στην περίπτωση όπου η θεωρία είναι «πυκνή» ή «αραιή». Ακόμη η διαφορά που εντοπίζεται σε μία «μέτρια» θεωρία είναι μεγαλύτερη από αυτήν που υπολογίζεται σε μία «πυκνή» θεωρία. Με τον όρο «πυκνή» θεωρία αναφερόμαστε στις θεωρίες που αποτελούνται από πολλές σχέσεις επίθεσης. Στην περίπτωση του πειράματος μας όταν αναφερόμαστε στις «πυκνές» θεωρίες αναφερόμαστε στις θεωρίες που έχουν πυκνότητα ίση με 0.65, ενώ με τον όρο «αραιή» θεωρία αναφερόμαστε σε θεωρίες που δεν είναι ούτε «αραιές» αλλά ούτε και «πυκνές». Στην περίπτωση του πειράματος μας με τον όρο «αραιές» θεωρίες αναφερόμαστε στις θεωρίες με πυκνότητα ίση με 0.50.

Στο σημείο αυτό ας στρέψουμε την προσοχή μας στις γραφικές παραστάσεις στις οποίες παρουσιάζεται ο τρόπος με τον οποίο μεταβάλλεται η διαφορά που υπολογίζεται για κάθε μία αρχική θεωρία ξεχωριστά και που ο υπολογισμός αυτής της διαφοράς γίνεται με βάση τις ευσταθείς επεκτάσεις. Κοιτάζοντας αυτές τις παραστάσεις καταλήγουμε στην εξής παρατήρηση: δεν παρουσιάζεται κάποια σταθερή μορφή στον τρόπο μεταβολής της διαφοράς. Σε κάποιες περιπτώσεις η διαφορά αυξάνεται, σε κάποιες άλλες μειώνεται, ενώ σε κάποιες άλλες περιπτώσεις αυξομειώνεται. Αυτή την συμπεριφορά δεν την αναμέναμε. Η αναμενόμενη συμπεριφορά ήταν να σημειώνεται πτώση της διαφοράς, καθώς αυξανόταν το πλήθος των υποσυνόλων που αφαιρούνταν. Κοιτάζοντας όμως πιο επιγραμματικά τα δεδομένα τα οποία είχαμε καταλήξαμε ότι το

αποτέλεσμα αυτό ήταν λογικό. Αυτό το γεγονός συμβαίνει γιατί έστω ότι είχαμε δύο θεωρίες οι οποίες ήταν «σχεδόν» όμοιες, δηλαδή διέφεραν κατά μία μόνο σχέση επίθεσης, αυτή η σχέση επίθεσης ήταν ικανή να αλλάξει ολόκληρη τη σημασιολογία της θεωρίας με αποτέλεσμα να κάνει της δύο θεωρίες να διαφέρουν κατά πολύ. Όμως, αν συγκρίναμε τις δύο αυτές θεωρίες ως προς την «σύνταξη» τους, δηλαδή αν κοιτάζαμε πόσες διαφορετικές σχέσεις επίθεσης είχαν τότε η διαφορά που θα εντοπίζαμε θα ήταν μικρότερη από αυτή που στηριζόταν στην σημασιολογία των θεωριών. Άρα το τελικό συμπέρασμα είναι ότι ο τρόπος με τον οποίο υπολογιζόταν η διαφορά με βάση τις ευσταθείς επεκτάσεις δεν είναι και ο καταλληλότερος τρόπος για να κρίνουμε αν δύο θεωρίες είναι όμοιες.

Κεφάλαιο 5

Συμπεράσματα

5.1 Συμπεράσματα μελέτης	76
5.2 Μελλοντικές επεκτάσεις	78

Στο κεφάλαιο αυτό θα γίνει μια σύντομη σύνοψη της πορείας της παρούσας διπλωματικής εργασίας και ακολούθως θα παραθέσουμε τα συμπεράσματα στα οποία οδηγηθήκαμε από την μελέτη αυτή. Επιπλέον θα γίνει αναφορά σε κάποιους τρόπους με τους οποίους μπορεί η παρούσα εργασία να επεκταθεί εκτενέστερα.

5.1 Συμπεράσματα μελέτης

Κλείνοντας την παρούσα διπλωματική εργασία θα ήταν παράλειψη μας να μην δούμε κατά πόσο έχουμε επιτύχει τον σκοπό το οποίο κληθήκαμε να φέρουμε εις πέρας.

Στο σημείο αυτό ας θυμηθούμε τον σκοπό της παρούσας διπλωματικής, έτσι ώστε να κρίνουμε κατά πόσο τον έχουμε επιτύχει κοιτάζοντας την πορεία που έχουμε ακολουθήσει. Πρωταρχικός στόχος ήταν η μελέτη μεθόδων μέσω των οποίων θα αντλούσαμε γνώση από μια θεωρία της επιχειρηματολογίας. Με άλλα λόγια η μελέτη που έχει γίνει αφορούσε τον εντοπισμό τρόπων μέσω των οποίων θα παίρναμε πληροφορίες από μία θεωρία, οι οποίες χαρακτήριζαν τη συγκεκριμένη θεωρία.

Βλέποντας τα προηγούμενα κεφάλαια της παρούσας διπλωματικής μπορούμε να δούμε πως έχει γίνει μελέτη διαφόρων αλγορίθμων για την επίτευξη του στόχου που μόλις έχουμε αναφέρει. Συγκεκριμένα μελετήθηκαν και υλοποιήθηκαν τέσσερις αλγόριθμοι. Ο πρώτος αλγόριθμός αντλεί γνώση από μία δεδομένη θεωρία. Συγκεκριμένα, όπως

έχει ήδη ειπωθεί σε προηγούμενα κεφάλαια (βλέπε κεφάλαια 3 και 4), ο αλγόριθμος αυτός εντοπίζει όλες τις ευσταθείς επεκτάσεις μιας θεωρίας κάνοντας χρήση ενός προτασιακού επιλυτή. Ο αλγόριθμος αυτός μπορεί να εφαρμοστεί και σε τροποποιημένες θεωρίες, δηλαδή θεωρίες στις οποίες αφαιρούνται κάποια από τα επιχειρήματα τους. Ο επόμενος αλγόριθμος που μελετήσαμε εντοπίζει μια καινούργια θεωρία της επιχειρηματολογίας με βάση κάποιας γνώσης που δέχεται ως είσοδο. Πιο ειδικά ο αλγόριθμος αυτός παίρνει σαν είσοδο ένα σύνολο από ευσταθείς επεκτάσεις και ακολούθως κάνοντας χρήση ενός προτασιακού επιλυτή που επιλύει προβλήματα σταθμικής προτασιακής βελτιστότητας, εντοπίζει την καινούργια θεωρία. Η καινούργια θεωρία που προκύπτει υπακούει στις ευσταθείς επεκτάσεις με βάση τις οποίες χτίστηκε. Ακολούθως μελετήθηκαν δύο αλγόριθμοι οι οποίοι εντοπίζουν κατά πόσο δύο θεωρίες της επιχειρηματολογίας είναι όμοιες, δηλαδή υπολογίζουν τη διαφορά που υπάρχει μεταξύ των δύο θεωριών. Πιο ειδικά ο πρώτος αλγόριθμος υπολογίζει τη διαφορά με βάση τις ευσταθείς επεκτάσεις των δύο θεωριών, ενώ ο δεύτερος αλγόριθμος στηρίζεται στις δυαδικές σχέσεις επίθεσης των θεωριών.

Κοιτάζοντας τα πιο πάνω που έχουν ειπωθεί μπορούμε να αντιληφθούμε κατά πόσο έχουμε πετύχει τον στόχο της παρούσας διπλωματικής. Ο στόχος αυτός έχει επιτευχθεί, αφού έχουμε ήδη παραθέσει αλγορίθμους οι οποίοι μπορούν να μας επιστρέψουν κάποια γνώση που χαρακτηρίζει κάποια θεωρία της επιχειρηματολογίας.

Επιπλέον μπορούμε να εξάγουμε κάποιο γενικό συμπέρασμα μέσα από την πειραματική αξιολόγηση που έχει ήδη παρουσιαστεί στο υποκεφάλαιο 4.2. Βασική ιδέα των πειραμάτων που έχουν γίνει ήταν δοθείσας μιας αρχικής θεωρίας να αντλήσουμε γνώση από αυτή την θεωρία και με βάση αυτή την γνώση να δημιουργήσουμε μια καινούργια θεωρία. Μετέπειτα εντοπίζαμε κατά πόσο οι δύο αυτές θεωρίες ήταν όμοιες. Μέσα από τα συμπεράσματα που καταλήξαμε από την πειραματική αξιολόγηση προκύπτει ένα πιο γενικό συμπέρασμα. Συγκεκριμένα οδηγούμαστε στο εξής συμπέρασμα: η γνώση που αντλούμε από μια θεωρία είναι ανάλογη της δειγματοληψίας που κάνουμε. Για να γίνουμε πιο κατανοητοί μπορούμε να πούμε πως όσα πιο πολλά δείγματα, παίρνουμε από μια θεωρία τόσο πιο πολλή γνώση αντλούμε γι' αυτή την θεωρία.

5.2 Μελλοντικές επεκτάσεις

Υπάρχουν αρκετές μελλοντικές επεκτάσεις, καθώς και βελτιώσεις που μπορούν να γίνουν γι' αυτή την διπλωματική εργασία.

Αρχικά ας δούμε πως θα μπορούσαμε να βελτιώσουμε τον ήδη υπάρχον σύστημα. Για σκοπούς καλύτερης απόδοσης του συστήματος από πλευράς χρόνου εκτέλεσης αλλά και διαχείρισης μνήμης θα μπορούσαμε να κάνουμε χρήση διαφορετικών δομών.

Επιπλέον, για σκοπούς ελαχιστοποίησης του χρόνου εκτέλεσης θα μπορούσαμε να επιλέξουμε μία πλειάδα από προτασιακούς επιλυτές των οποίων θα συγκρίναμε τους χρόνους που χρειάζονταν για να βρουν λύση στο πρόβλημα που τους δινόταν και αναλόγως θα επιλέγαμε τους επιλυτές με το λιγότερο απαιτούμενο χρόνο.

Επιπλέον για σκοπούς επέκτασης θα μπορούσε να γίνει χρήση άλλων επεκτάσεων, όπως για παράδειγμα της στηριγμένης επέκτασης (Grounded Semantics), της προτιμώμενης επέκτασης (Preferred Semantics) και άλλων. Με βάση αυτές τις επεκτάσεις θα μπορούσε να δημιουργείται η καινούργια θεωρία. Ακόμη η γνώση με βάση την οποία θα χτίζεται η καινούργια θεωρία μπορεί να είναι συνδυασμοί κάποιων επεκτάσεων. Στο τέλος μπορεί να γίνει σύγκριση μεταθύ των αποτελεσμάτων της κάθε περίπτωσης.

Ακόμη θα μπορούσε να γίνει μελέτη για εντοπισμό μεθόδων που υπολογίζουν τη διαφορά που υπάρχει μεταξύ κάποιων δοθέντων θεωριών επιχειρηματολογίας. Με άλλα λόγια μπορεί να υλοποιηθούν αλγόριθμοι οι οποίοι δοθέντος κάποιων θεωριών να εντοπίζουν κατά πόσο οι θεωρίες αυτές είναι όμοιες μεταξύ τους. Για σκοπούς αυτής τη ενέργειας μπορεί να γίνει μελέτη για συνδυασμό μεθόδων οι οποίοι στηρίζονται στην επιστήμη των μαθηματικών.

Τέλος, θα μπορούσε να γίνει μια επέκταση της υλοποίησης σε κάποια αντικειμενοστρεφή γλώσσα προγραμματισμού. Σε μια τέτοια γλώσσα προγραμματισμού μας παρέχονται περισσότερες διευκολύνσεις, αφού αρκετά πράγματα είναι ήδη υλοποιημένα σε βιβλιοθήκες. Ένα παράδειγμα μιας γλώσσα αντικειμενοστρεφούς προγραμματισμού είναι η Java.

Βιβλιογραφία

- [1] Phan Minh Dung: “On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games”. Artificial Intelligence 77(1995) 321-357.
- [2] Iyad Rahwan and Guillermo R. Simari, Argumentation in Artificial Intelligence, Chapter 2, Pietro Baroni and Massimiliano Giacomin, “Semantics of Abstract Argument Systems”.
- [3] Martin Caminada: “A gentle Introduction to argumentation semantic”, University of Luxembourg, summer 2008.
- [4] Sanjay Modgil: Reasoning about preferences in argumentation frameworks. Artif. Intell. 173(9-10): 901-934 (2009).
- [5] Μιχαήλ Δημήτρης, Αλγόριθμοι και Πολυπλοκότητα, NP και Υπολογιστική Δυσεπιλυσιμότητα, Αθήνα
- [6] Toby Walsh , « SAT v CSP» University of York, York England
<http://www.cs.toronto.edu/~jdavies/daviesCP11.pdf>
- [7] <http://maxsat.ia.udl.cat/requirements/>

Παράρτημα Α

Σε αυτό το παράρτημα παραθέτεται η υλοποίηση του αλγορίθμου εντοπισμού όλων των ευσταθών επεκτάσεων μιας θεωρίας αφαιρώντας της υποσύνολα επιχειρημάτων.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    int nodes, akmes, count, flag1=1, flag=0, i, j, k;
    float piknotita, pin;
    int** table;
    //Pairnw ta dedomena apo to xristi(arithmo komvwn kai piknotita)
    /*printf("Dose arithmo komvon gia ton grafo: \n");
    scanf("%d", &nodes);
    printf("Dose tin piknotita tou grafou: \n");
    scanf("%f", &piknotita);*/
    nodes= 0;
    piknotita= 0;
    //ipologizw poses akmes tha exei o grafos
    akmes=piknotita*nodes*(nodes-1);
    if (akmes>nodes*nodes-nodes){
        printf("Poli megali Piknotita! To programa tha termatisi!\n");
        exit(-1);
    }
    //Desmeusi mninis gia pinaka-grafo
    table = (int**) malloc(nodes*sizeof(int*));
    for (i = 0; i < nodes; i++)
        table[i] = (int*) malloc(nodes*sizeof(int));
    //Arxikopoihsh pinaka me miden
    for (i=0; i<nodes;i++)
        for (j=0;j<nodes;j++)
            table[i][j]=0;
    srand(time(NULL));
    //Topothesisi akmwon sto grafo tixaia
    for (k=0;k<akmes;k++){
        i=0;j=0;
        while (i==j||flag==1||i>nodes||j>nodes||i<0||j<0) {
            i = rand() % nodes;
            j = rand() % nodes;
            if (table[i][j]==0)
                flag=0;
            else
                flag=1;
        }
        table[i][j]=1;
    }
    //Typwma pinaka-grafou
    for (i=0; i<nodes;i++){
        if (i==nodes) printf("\n");
        for (j=0;j<nodes;j++){
            printf("%d ", table[i][j]);
        }
        printf("\n");
    }
}
```

```

FILE *f5 = fopen("random-newTheories.txt", "w");
fprintf(f5,"%d\n",nodes);
for (i=0; i<nodes;i++){
    if (i==nodes) fprintf(f5,"\n");
    for (j=0;j<nodes;j++){
        fprintf(f5,"%d ", table[i][j]);
    }
    fprintf(f5,"\n");
}
fprintf(f5,"\n");
fclose(f5);
}
int createExtesnsionsCNF_file(char **table2,int nodes,int count,FILE
*f){
    int i=0,j=0,flag1=1;
    //metra posa clausures tha exei t arxeio pou dimiourgw
    count=nodes;
    for (i=0;i<nodes;i++){
        for (j=0;j<nodes;j++){
            if (table2[i][j]=='1')
                count++;
        }
    }
    fprintf(f, "p cnf %d %d \n", nodes, count);
    //a. kanonas.Diatrexb ton pinaka kai opou entopizw 1 "typwnw" to
seimio
    for (i=1;i<=nodes;i++){
        for (j=1;j<=nodes;j++){
            if (table2[i-1][j-1]=='1') {
                fprintf(f, "-%d -%d 0\n", i, j);
            }
        }
    }
    //b. kanonas.Elegxb gia kathe grami tin antistoixi stili kai
opou exei 1 "typwnw" to seimio
    j=1;
    for (i=1;i<=nodes;i++) {
        do {
            if (table2[j-1][i-1]=='1'&&flag1==0) {
                fprintf(f, "%d ", j);
            }
            if (table2[j-1][i-1]=='1'&&flag1==1) {
                fprintf(f, "%d %d ", i, j);
                flag1=0;
            }
            j++;
        } while(j<=nodes);
        j=1;
        if (flag1==0) fprintf(f, "0\n");
        else if (flag1==1) fprintf(f, "%d 0\n", i);
        flag1=1;
    }
    return count;
}

int createExtesnsions2CNF_file(char **table2,int *random,int nodes,int
count,FILE *f,int i,int *ptr1){
    int a,j,e,b=0,k=0;
    //metra posa clausures tha exei t arxeio pou dimiourgw
    count=nodes-ptr1[i];

```



```

for(a=1;a<=nodes;a++){
    if( b==ptr1[i] || a!=random[b]){
        k=0;
        for(e=1;e<=nodes;e++){
            if(k==ptr1[i] || e!=random[k]){
                if (table2[a-1][e-1]=='1') {
                    count++;}}
            else{k++;}}
        else{ b++;}}
    fprintf(f, "p cnf %d %d \n", nodes, count);
    b=0;k=0;
    //a. kanonas.Diatrexw ton pinaka kai opou entopizw 1 "typwnw" to
seimio
    for(a=1;a<=nodes;a++){
        if( b==ptr1[i] || a!=random[b]){
            k=0;
            for(e=1;e<=nodes;e++){
                if(k==ptr1[i] || e!=random[k]){
                    if (table2[a-1][e-1]=='1') {
                        fprintf(f, "-%d -%d 0\n", a, e);}}
                    else{k++;}}
                else{b++;}}
            k=0;b=0;
            //b. kanonas.Elegxw gia kathe stili tin antistoixi grami kai
opou exei 1 "typwnw" to seimio
            for (e = 1; e <= nodes; e++){
                if(k==ptr1[i] || e!=random[k]){
                    fprintf(f, "%d " , e);
                    b=0;
                    for (a = 1; a <= nodes; a++) {
                        if(b==ptr1[i] || a!=random[b]){
                            if (table2[a-1][e-1] == '1'){
                                fprintf(f, "%d " , a);}}
                            else{b++;}}
                        fprintf(f, "%d\n" , 0);}
                    else{k++;}}
                return count;}
void quickSort(int x[],int first,int last){
    int pivot,j,temp,i;
    if(first<last){
        pivot=first;
        i=first;
        j=last;
        while(i<j){
            while(x[i]<=x[pivot]&&i<last)
                i++;
            while(x[j]>x[pivot])
                j--;
            if(i<j){
                temp=x[i];
                x[i]=x[j];
                x[j]=temp;}}
        temp=x[pivot];
        x[pivot]=x[j];
        x[j]=temp;
        quickSort(x,first,j-1);
        quickSort(x,j+1,last);}}

int go(int count,int plithos_extensions){
    int flag01=1;

```

```

int extension=0;
int plithos=0;
FILE *f2=NULL;
while(flag01=1){
    FILE *f = fopen("resultLingeling.txt", "r");
    FILE *f3=fopen("extensions.txt","r");//extensions
    char c1;
    char c;
    int num;
    fscanf(f,"%c",&c);
    while(c=='c' || c=='s'){
        if(c=='s'){
            fscanf(f,"%c",&c);
            fscanf(f,"%c",&c);
            if(c=='U'){
                flag01=0;break;}}
        while(c!='\n'){
            fscanf(f,"%c",&c);}
        fscanf(f,"%c",&c);}
    f2=f;
    if(flag01==0) break;
    plithos_extensions=plithos_extensions+1;
    extension=extension+1;
    FILE *f4=fopen("extensionsCNF.txt","r+");
    int num1;
    fscanf(f4,"%c",&c1);
    fscanf(f4,"%c",&c1);
    fscanf(f4,"%c",&c1);
    fscanf(f4,"%c",&c1);
    fscanf(f4,"%c",&c1);
    fscanf(f4,"%c",&c1);
    fscanf(f4,"%c",&c1);
    fscanf(f4,"%d",&num1);
    fscanf(f4,"%c",&c1);
    fprintf(f4,"%d\n",count+1);
    fseek(f4,0,SEEK_END);
    fseek(f3,0,SEEK_END);
    FILE *f6=fopen("plithoiExtensions.txt","r+");
    fseek(f6,0,SEEK_END);
    while(c=='v'){
        while(c!='\n'){
            fscanf(f2,"%d",&num);
            if(num>0){
                fprintf(f3,"%d ",num);
                fprintf(f4,"-%d ",num);
                plithos=plithos+1;}
            if(num==0){
                fprintf(f6,"%d\n",plithos);
                plithos=0;
                fprintf(f4,"%d\n",num);
                fprintf(f3,"%d\n",num);
                count=count+1;}
            fscanf(f,"%c",&c);}
        fscanf(f,"%c",&c);}
    fclose(f);
    fclose(f3);
    fclose(f4);
    fclose(f6);
    int status = system("./lingeling < extensionsCNF.txt >
resultLingeling.txt");}
return plithos_extensions;}

```

```

int main(int argc, char *argv[]){
    int plithos, plithos_extensions=0;
    int i, a, j, nodes, num, count=0;
    char c;
    int plithos_ipo=atoi(argv[1]);
    //pairnw tin random thewria=table2
    FILE *f=fopen("random-newTheories.txt", "r");
    fscanf(f, "%d", &num);
    printf("nodes:%d\n", num);
    nodes=num;

    char **table2 = (char**) malloc(nodes*sizeof(char*));
    for (i = 0; i < nodes; i++)
        table2[i] = (char*) malloc(nodes*sizeof(char));
    for (i = 0; i < nodes; i++)
        for(j=0; j<nodes; j++)
            table2[i][j] = 0;

    fscanf(f, "%c", &c);
    for(i=0; i<nodes; i++){
        a=0;
        for(j=0; j<2*nodes+1; j++){
            fscanf(f, "%c", &c);
            if(c!=' '){
                table2[i][a]=c; a++;
            }
        }
    }
    //Typwma pinaka-grafou
    for (i=0; i<nodes; i++){
        if (i==nodes) printf("\n");
        for (j=0; j<nodes; j++){
            printf("%c ", table2[i][j]);
        }
        printf("\n");
    }
    /*int plithos_ipo=0;
    printf("Dwse to plithos tw'n iposinolwn pou tha dokimastoun:
\n");
    scanf("%d", &plithos_ipo);
    printf("Yposimola: %d \n", plithos_ipo);*/
    int *ptr1 = malloc(sizeof(int) * plithos_ipo);
    if (ptr1 == NULL){
        printf(" Out of memory!\n");
        exit(1);
    }
    //pairnw to plithiko arithmo tv'n iposinolwn
    if(plithos_ipo>0){
        for(i=0; i<plithos_ipo; i++){
            int max=nodes-1;
            plithos=rand() % max +1;
            ptr1[i]=plithos;}}
    //vriskw tous pirines xwris na afairesw kapoio node
    //anoigma input arxeiou pou dimiourgw
    f = fopen("extensionsCNF.txt", "w");
    if (f == NULL){printf("Error with file!\n");
        exit(1);}
    count=createExtesnsionsCNF_file(table2, nodes, count, f);
    printf("count %d\n", count);
    fclose(f);

```

```

        int status = system("./lingeling < extensionsCNF.txt >
resultLingeling.txt");
        FILE *f3=fopen("extensions.txt", "w");//lineling result
        fclose(f3);
        f3=fopen("plithoiExtensions.txt", "w");//lineling result
        fclose(f3);

        plithos_extensions=go(count,plithos_extensions);
        f=fopen("out5.txt", "w");
        fprintf(f,"%d ",nodes);
        fprintf(f,"%d\n",plithos_ipo);
        fprintf(f,"%d\n",plithos_extensions);
        fclose(f);
        srand(time(NULL));
        for(i=0;i<plithos_ipo;i++){
            int ran=0;
            int flagR=0;
            int *random = (int*) malloc(ptr1[i]*sizeof(int));
            for (j = 0; j < ptr1[i]; j++)
                random[j] = 0;
            if(ptr1[i]!=0){
                int r;
                a=0;
                while(a!=ptr1[i]){
                    ran=rand() % nodes +1;
                    for(j=0;j<a;j++){
                        if(ran==random[j]){
                            flagR=1;break;}}
                    if(flagR!=1){random[a]=ran;a++;}
                    flagR=0;}}
                quickSort( random, 0, ptr1[i]-1);
                printf("plithos: %d \n",ptr1[i]);
                //if(ptr1[i]!=nodes){
                //anoigma input arxeiou pou dimiourgw
                FILE *f = fopen("extensionsCNF.txt", "w");
                if (f == NULL){printf("Error with file!\n");
                    exit(1);}
                FILE *f4 = fopen("extensions.txt", "r+");
                fseek(f4,0,SEEK_END);
                int e;
                for(e=0;e<ptr1[i];e++){
                    fprintf(f4,"%d ",random[e]);}
                fprintf(f4,"\n");
                fclose(f4);
                count=createExtesnsions2CNF_file(table2,random,nodes
,count,f,i,ptr1);
                fclose(f);
                int status = system("./lingeling <
extensionsCNF.txt > resultLingeling.txt");
                FILE *f6=fopen("plithoiExtensions.txt", "r+");
                fseek(f6,0,SEEK_END);
                fprintf(f6,"%d\n",ptr1[i]);
                fclose(f6);
                plithos_extensions=go(count,plithos_extensions);
                free(random);}
        f=fopen("out5.txt", "r+");
        fseek(f,0,SEEK_END);
        fprintf(f,"%d\n",plithos_extensions);
        fclose(f);

```

Παράρτημα Β

Σε αυτό το παράρτημα παραθέτεται η υλοποίηση του αλγορίθμου εντοπισμού της ελάχιστης θεωρίας.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int plinNode;
    int num,arguments,plithos_ipo=0,extension_arxi,extension;
    int count=0;
    int plithos=0;
    int i=0,j=0,k,flag=0,c=0;
    FILE *f3=fopen("out5.txt","r");
    fscanf(f3,"%d",&num);
    arguments=num;
    fscanf(f3,"%d",&num);
    plithos_ipo=num;
    fscanf(f3,"%d",&num);
    extension_arxi=num;
    fscanf(f3,"%d",&num);
    extension=num;
    fclose(f3);
    int *ptr1 = malloc(sizeof(int) * extension);
    if (ptr1 == NULL){
        printf(" Out of memory!\n");
        exit(1);
    }
    int *ptr2 = malloc(sizeof(int) * plithos_ipo);
    if (ptr2 == NULL){
        printf(" Out of memory!\n");
        exit(1);
    }
    f3=fopen("plithoiExtensions.txt","r");
    for(i=0;i<extension_arxi;i++){
        fscanf(f3,"%d",&num);
        ptr1[i]=num;
    }
    for(i=0;i<extension_arxi;i++){
        count=count+(arguments-
ptr1[i])+ptr1[i]*ptr1[i]+((arguments-ptr1[i])*ptr1[i]); //vaze tis
pithanes akmes san unit clause me mikro varos gi na anagkastei na
valei mono mia true
    }
    int fl=0;j=0;
    while(fscanf(f3,"%d",&num)!=EOF){
        while(!feof(f3) && num>=0){
            ptr1[i]=num;
            count=count+(arguments-ptr2[j-1]-
ptr1[i])+ptr1[i]*ptr1[i]+((arguments-ptr2[j-1]-
ptr1[i])*ptr1[i]);
            i++;
            fscanf(f3,"%d",&num);
            if(feof(f3)) {fl=1;break;}
        }
    }
}
```

```

    }
    if(fl!=1){
        ptr2[j]=num * (-1);
        j++;}
    else break;}
fclose(f3);
//anoigma input arxeiou pou dimiourgw
FILE *f = fopen("output2.cnf", "w");
if (f == NULL){
    printf("Error with file!\n");
    exit(1);
}
fprintf(f, "p wcnf %d %d %d\n", arguments*arguments,
count,count);
FILE *f2 = fopen("extensions.txt", "r");
for(i=0;i<extension_arxi;i++){
    int *ptr = malloc(sizeof(int) * ptr1[i]);
    if (ptr == NULL){
        printf(" Out of memory!\n");
        exit(1);
    }
    for(j=0;j<ptr1[i];j++){
        fscanf(f2,"%d",&num);
        ptr[j]=num;

    }
    fscanf(f2,"%d",&num);//prospernw to miden
    for(j=0;j<ptr1[i];j++)
        for(k=0;k<ptr1[i];k++){
            fprintf(f,"1 -%d 0\n",arguments*(ptr[j]-1)+ptr[k]);}
    for(k=1;k<arguments+1;k++){
        for(j=0;j<ptr1[i];j++)
            if(ptr[j]==k) {
                flag=1;break;}
        if (flag!=1){
            for(j=0;j<ptr1[i];j++)
                if(j==0)
                    fprintf(f,"%d %d ",count,arguments
                    *(ptr[j]-1)+k);
                else
                    fprintf(f,"%d ",arguments*(ptr[j]-
                    1)+k);
            fprintf(f,"0\n");
            for(j=0;j<ptr1[i];j++)
                fprintf(f,"1 -%d 0\n",arguments*(ptr[j]-1)+k);
        } flag=0;
    }
    free(ptr);
}
int flaGG=0,tmp=0,d;
fscanf(f2,"%d",&num);
tmp=num*(-1);
for(j=0;j<plithos_ipo;j++){
    int *iposinolo=malloc(sizeof(int)*ptr2[j]);
    if (iposinolo == NULL){
        printf(" Out of memory!\n");
        exit(1);
    }
    for(d=0;d<ptr2[j];d++){
        if(d==0)

```

```

        iposinolo[d]=tmp;
    else{
        fscanf(f2,"%d",&num);
        iposinolo[d]=num*(-1);}}
fscanf(f2,"%d",&num);
while(num>=0){
    int *ptr = malloc(sizeof(int) * ptr1[i]);
    if (ptr == NULL){
        printf(" Out of memory!\n");
        exit(1);
    }
    ptr[0]=num;
    c=0;
    for(c=c+1;c<ptr1[i];c++){
        fscanf(f2,"%d",&num);
        ptr[c]=num;
    }
    fscanf(f2,"%d",&num);//prospernw to miden
    for(c=0;c<ptr1[i];c++){
        for(k=0;k<ptr1[i];k++){
            fprintf(f,"1 -%d 0\n",arguments*(ptr[c]-
            1)+ptr[k]);
        }
    }
    int s=0;
    for(k=1;k<arguments+1;k++){
        if (s<ptr2[j] && k==iposinolo[s]){
            flag=1;
            s++;
        }else{
            for(c=0;c<ptr1[i];c++)
                if(ptr[c]==k) {
                    flag=1;
                    break;
                }
        }
        if (flag!=1){
            for(c=0;c<ptr1[i];c++)
                if(c==0)
                    fprintf(f,"%d %d",count,arguments*ptr[c]-1)+k);
                else
                    fprintf(f,"%d ",arguments*(ptr[c]-
                    1)+k);
                    fprintf(f,"0\n");
            for(c=0;c<ptr1[i];c++)
                fprintf(f,"1 -%d 0\n",arguments*(ptr[c]-
                1)+k);
        }
        flag=0;
    }
    free(ptr);
    i++;
    fscanf(f2,"%d",&num);
    if(feof(f2)) {flaGG=1;break;}
}
//an den exw feof
if(flaGG==1) break;
tmp=num*(-1);
free(iposinolo);

```

```

    }
    free(ptr1);
    free(ptr2);
    fclose(f);
    int status = system("./maxino-2015-kdyn-static < output2.cnf >
output3.txt");
    f = fopen("output3.txt", "r");
    if (f == NULL){
        printf("Error with file!\n");
        exit(1);
    }
    f2=fopen("output3.txt", "r");
    int grami;
    int stili;
    char c1;
    fscanf(f,"%c",&c);
    while(c=='c' || c=='o' || c=='s'){
        while(c!='\n'){
            fscanf(f,"%c",&c);}
        fscanf(f,"%c",&c);}
    if(c=='v'){
        //Desmeusi mninis gia pinaka-grafo
        int **table = (int**) malloc(arguments*sizeof(int*));
        for (i = 0; i < arguments; i++)
            table[i] = (int*) malloc(arguments*sizeof(int));
        for(i=0;i<arguments;i++)
            for(j=0;j<arguments;j++)
                table[i][j]=0;

    f2=f;
    fscanf(f,"%c",&c);
    fscanf(f2,"%d",&num);
    while( num!= arguments*arguments){
        while(c!='\n'){
            if(num>0){
                grami=(num/arguments)+1;
                stili=num % arguments;
                if(stili==0){
                    stili=arguments;
                    grami=grami-1;
                }
                //printf("o komvos %d kanei epithesi ston
                komvo %d. \n", grami,stili);}
                fscanf(f,"%c",&c);
                fscanf(f2,"%d",&num);}
            fscanf(f2,"%d",&num);
            if(num==-1*arguments*arguments) break;}
        fclose(f);
        FILE *f5 = fopen("newTheoria.txt", "w");
        fprintf(f5,"%d\n",arguments);
        for (i=0; i<arguments;i++){
            if (i==arguments) fprintf(f5,"\n");
            for (j=0;j<arguments;j++){
                fprintf(f5,"%d ", table[i][j]);}
            fprintf(f5,"\n");}
        fclose(f5);
        free(table);
        printf("1\n");
    }
    else{printf("0\n");fclose(f);}//den vrethike i thewria}

```


Παράρτημα Γ

Σε αυτό το παράρτημα παραθέτεται η υλοποίηση του αλγορίθμου που υπολογίζει την διαφορά με βάση τις ευσταθείς επεκτάσεις.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int numdigits(int num) {
    int count1=0;
    if(num!=0) {
        while(num) {
            num=num/10;
            count1++;
        }
    }
    else count1=1;
    return count1;
}
int main() {
    int count01=0, f1=0, num2, num, i, plithos2, diafora=0, j=0;
    int
    plithos_extensions1, plithos_extensions1b, xwrisplin, xwrisplin_new, flag1
    0=0, plinNodes, plinNodes2, ab=0;
    int arnitikos1=0, arnitikos2=0, flag001, flag002, a;
    int *ptr;
    FILE *f2=fopen("extensions.txt", "r");
    FILE *f3=fopen("extensions2.txt", "r");
    FILE *f4=fopen("plithoiExtensions.txt", "r");
    FILE *f5=fopen("out5.txt", "r");
    fscanf(f5, "%d", &num2);
    fscanf(f5, "%d", &num2);
    plinNodes=num2;
    fscanf(f5, "%d", &num2);
    xwrisplin=num2;
    fscanf(f5, "%d", &num2); //pairnw to sinoliko arirhmo tw n extensions tis
    random
    fclose(f5);
    plithos_extensions1=num2;
    for(i=0; i<xwrisplin; i++) {
        flag10=0;
        fscanf(f4, "%d", &plithos2); //to plithos kathe fora tou extensions
    tis random
        int *ptr = malloc(sizeof(int) * plithos2+1);
        for(j=0; j<plithos2+1; j++) {
            fscanf(f2, "%d", &num2);
            ptr[j]=num2;
        }
        fscanf(f3, "%d", &num2);
        j=0;
        while(num2>=0) {
            if(ptr[j]!=num2) {
                while(num2!=0) {
                    fscanf(f3, "%d", &num2);
                }
                j=0;
            }
            else{
```

```

        j++;
        if(j==plithos2+1){
            flag10=1;
            break;}}
        fscanf(f3,"%d",&num2);
        if(feof(f3)) break;}
    if (flag10=0) diafora++;
    free(ptr);
    rewind(f3);}
f2=fopen("extensions.txt","r");
f3=fopen("extensions2.txt","r");
FILE *f6=fopen("plithoiextensions2.txt","r");
f5=fopen("out5b.txt","r");
fscanf(f5,"%d",&num2);
fscanf(f5,"%d",&num2);
plinNodes2=num2;
fscanf(f5,"%d",&num2);
xwrisplin_new=num2;
fscanf(f5,"%d",&num2);
fclose(f5);
plithos_extensions1b=num2;
for(i=1;i<=xwrisplin_new;i++){
    rewind(f2);
    flag10=0;
    fscanf(f6,"%d",&plithos2);
    int *ptr = malloc(sizeof(int) * plithos2+1);
    for(j=0;j<plithos2+1;j++){
        fscanf(f3,"%d",&num2);
        ptr[j]=num2;}
    fscanf(f2,"%d",&num2);
    j=0;
    while(num2>=0){
        if(ptr[j]!=num2){
            while(num2!=0){
                fscanf(f2,"%d",&num2);}
            j=0;}
        else{ j++;
            if(j==plithos2+1){
                flag10=1;
                break;}}
        fscanf(f2,"%d",&num2);
        if(feof(f2)) break;
    }
    if (flag10==0) diafora++;
    free(ptr);
}
fclose(f2);
fclose(f3);
if(plinNodes!=0){
    f2=fopen("extensions.txt","r");
    f3=fopen("extensions2.txt","r");
    fscanf(f2,"%d",&num);
    while(num>=0){
        fscanf(f2,"%d",&num);}
    fscanf(f3,"%d",&num2);
    if(num2<0){count01=count01+numdigits(num2)+2;f1=1;}
    else {count01=count01+numdigits(num2)+1;}
    while(num2>=0){
        fscanf(f3,"%d",&num2);
        count01+numdigits(num2)+1;}
}

```

```

if(fl!=1){
count01=count01+1;
arnitikos1=num;arnitikos2=num2;flag001=0;flag002=0;a=0;
flag10=0; count01=0;
ab=abs(plithos_extensions1-(plithos_extensions1b-xwrisplin_new));
for(i=0;i<plithos_extensions1-xwrisplin;i++){
flag10=0;
if(flag001==0){
fscanf(f4,"%d",&plithos2);
ptr = malloc(sizeof(int) * plithos2+1);
fscanf(f2,"%d",&num);
while(num<0 && !feof(f2)){
arnitikos1=num;
fscanf(f2,"%d",&num);}
j=0;
ptr[j]=num;
for(j=j+1;j<plithos2+1;j++){
fscanf(f2,"%d",&num);
ptr[j]=num;}}
if(flag002==0){
fscanf(f3,"%d",&num2);
if(num2<0){count01=0;}
while(num2<0 && !feof(f3)){
arnitikos2=num2;
fscanf(f3,"%d",&num2);}
if(arnitikos1==arnitikos2){
a=0;
while(num2>=0){
if(ptr[a]!=num2){
while(num2!=0){
fscanf(f3,"%d",&num2);
count01=count01+numdigits(num2)+1;}
a=0;}
else{ a++;
if(a==plithos2+1){
flag10=1;
break;}}
fscanf(f3,"%d",&num2);
if(feof(f3)) break;
count01=count01+numdigits(num2)+1;}
flag001=0;flag002=0;
if (flag10==0) {diafora++;}
fscanf(f2,"%d",&num);
fseek(f2, (numdigits(num)+1)*(-1),SEEK_CUR);
if(num>=0 && num2<0){count01=count01+1;}
if(num>=0){
fseek(f3, (count01)*(-1),SEEK_CUR);
count01=0;}
if(num2<0 && num<0){arnitikos2=num2;count01=0;}
if(num2==0 && num<0){count01=0;}
free(ptr);}
else if(arnitikos1<arnitikos2){
flag001=1;flag002=0;
while(num2>=0){
fscanf(f3,"%d",&num2);
}arnitikos2=num2;
count01=0;
i--;}
else {flag002=1;flag001=0;diafora++;
while(num>=0){

```

```

        fscanf(f2,"%d",&num);
        if(num==0){fscanf(f4,"%d",&plithos2);diafora++;i++;}}
        }arnitikos1=num;
        free(ptr);}}
fclose(f2);fclose(f3);fclose(f4);}
if(plinNodes2!=0){
    f2=fopen("extensions.txt","r");
    f3=fopen("extensions2.txt","r");
    fl=0;
    fscanf(f2,"%d",&num2);
    count01=0;
    if(num2<0){count01=count01+numdigits(num2)+2;fl=1;}
    else {count01=count01+numdigits(num2)+1;}
    while(num2>=0){
        fscanf(f2,"%d",&num2);
        count01=count01+numdigits(num2)+1;}
    if(fl!=1)count01=count01+1;
    fscanf(f3,"%d",&num);
    while(num>=0){
        fscanf(f3,"%d",&num);}
    arnitikos1=num;arnitikos2=num2;flag001=0;flag002=0;a=0;
    flag10=0;count01=0;
    ab=abs(plithos_extensions1b-(plithos_extensions1-xwrisplin));
    for(i=0;i<plithos_extensions1b-xwrisplin_new;i++){
        flag10=0;
        if(flag001==0){
            fscanf(f6,"%d",&plithos2);
            ptr = malloc(sizeof(int) * plithos2+1);
            fscanf(f3,"%d",&num);
            while(num<0 && !feof(f3)){
                arnitikos1=num;
                fscanf(f3,"%d",&num);}
            j=0;ptr[j]=num;
            for(j=j+1;j<plithos2+1;j++){
                fscanf(f3,"%d",&num);
                ptr[j]=num;}}
        if(flag002==0){
            fscanf(f2,"%d",&num2);
            if(num2<0){count01=0;}
            while(num2<0 && !feof(f2)){
                arnitikos2=num2;
                fscanf(f2,"%d",&num2);}
            count01=count01+numdigits(num2)+1;
            if(arnitikos1==arnitikos2){
                a=0;
                while(num2>=0){
                    if(ptr[a]!=num2){
                        while(num2!=0){
                            fscanf(f2,"%d",&num2);
                            count01=count01+numdigits(num2)+1;
                        }a=0;}
                    else{ a++;
                        if(a==plithos2+1){
                            flag10=1;break;}}
                    fscanf(f2,"%d",&num2);
                    if(feof(f2)) break;
                    count01=count01+numdigits(num2)+1;}
                flag001=0;flag002=0;
                if (flag10==0) { diafora++;}
                fscanf(f3,"%d",&num);

```

```

fseek(f3, (numdigits(num)+1)*(-1), SEEK_CUR);
if(num>=0 && num2<0){count01=count01+1;}
if(num>=0){
    fseek(f2, (count01)*(-1), SEEK_CUR);count01=0;}
if(num2<0 && num<0){arnitikos2=num2;count01=0;}
if(num2==0 && num<0){count01=0;}
free(ptr);
}
else if(arnitikos1<arnitikos2){
    flag001=1;
    flag002=0;
    while(num2>=0){
        fscanf(f2,"%d",&num2);
    }arnitikos2=num2;
    count01=0;
    i--;}
else {flag002=1;flag001=0;diafora++;
    while(num>=0){
        fscanf(f3,"%d",&num);
        if(num==0){fscanf(f6,"%d",&plithos2);diafora++;i++;}
    }arnitikos1=num;
    free(ptr);}}
printf("%d  \n",diafora);
fclose(f2); fclose(f3);fclose(f6);}}}}

```

Παράρτημα Δ

Σε αυτό το παράρτημα παραθέτεται η υλοποίηση του αλγορίθμου που υπολογίζει την διαφορά με βάση τις σχέσεις επίθεσης.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(){
    int num,nodes,a=0,i,j;
    char c;
    //pairnw tin random thewria=table2
    FILE *f=fopen("random-newTheories.txt","r");
    fscanf(f,"%d",&num);
    nodes=num;

    char **table2 = (char**) malloc(nodes*sizeof(char*));
    for (i = 0; i < nodes; i++)
        table2[i] = (char*) malloc(nodes*sizeof(char));

    fscanf(f,"%c",&c);
    for(i=0;i<nodes;i++){
        a=0;
        for(j=0;j<2*nodes+1;j++){
            fscanf(f,"%c",&c);
            if(c!=' '){
                table2[i][a]=c;a++;}}}

    //pairnw tin kainourgia thewria
    f=fopen("newTheoria.txt","r");
    fscanf(f,"%d",&nodes);
    printf("nodes:%d\n",nodes);
    char **table = (char**) malloc(nodes*sizeof(char*));
    for (i = 0; i < nodes; i++)
        table[i] = (char*) malloc(nodes*sizeof(char));

    fscanf(f,"%c",&c);
    for(i=0;i<nodes;i++){
        a=0;
        for(j=0;j<2*nodes+1;j++){
            fscanf(f,"%c",&c);
            if(c!=' '){table[i][a]=c;a++;}}}

    int countRandom=0,countNew=0;
    int countR=0,countN=0;

    //metrw poses diaforetikas epithesei exei h random thewria apo tin new
    thewria
    for(i=0;i<nodes;i++){
        for(j=0;j<nodes;j++){
            if(table2[i][j]=='1'){
                countR=countR+1;
                if(table2[i][j]!=table[i][j]){
                    countRandom=countRandom+1;}}}}}
```

```

//metrw poses diaforetikes epithesei exei h NEW thewria apo tin RANDOM
thewria
    for(i=0;i<nodes;i++){
        for(j=0;j<nodes;j++){
            if(table[i][j]=='1'){
                countN=countN+1;
                if(table[i][j]!=table2[i][j]){
                    countNew=countNew+1;}}}}

    int sum,ginomenoNodes;
    double diafora;

    sum=countRandom+countNew;
    ginomenoNodes=nodes*(nodes-1);

    diafora=(double)sum/ginomenoNodes;
    printf("%f  ",diafora);
}

```

Παράρτημα Ε

Σε αυτό το παράρτημα παραθέεται το shell script αρχείο μέσω του οποίου αλληλεπιδρούν οι αλγόριθμοι που έχουμε υλοποιήσει.

```
#!/bin/bash
rm data_output

echo"#Nodes Piknotita Yposinola Vrethike Diafora-diadikes
sxeseis Diafora-Eystathis epektaseis" >>data_output

echo"Dwse ton arithmo tw n thewriwn poy epithimeis na dimiourgithoun: "
read epanaliyeis

for c in `seq 1 $epanaliyeis`;
do
    echo "Dwse tw n arithmo tw n epixeirimatwn pou tha apoteloun
    ti thewria: "
    read nodes

    echo "Dwse tin piknotita tis thewrias: "
    read piknotita

    perl -i -pe "s/nodes= 0/nodes= $nodes/g" $1 #replace the
    old nodes with new value
    perl -i -pe "s/piknotita= 0/piknotita= $piknotita/g" $1
    #replace the old piknotita with new value

    echo "Random thewria: "
    gcc createRandom.c -o createRandom && ./createRandom
    cycles=./createRandom
    echo $cycles

    echo "Dwse to plithos tw n forwn pou epithimeis gia na
    ginei afairesi tw n epixeirimatwn apo ti thewria:"
    read fores

    for p in `seq 1 $fores`;
    do
        echo "Dwse to plithos tw n iposinolwn pou epithimeis
        na afairethoun:"
        read iposin
        gcc iposinola.c -o iposinola && ./iposinola $iposin
        cycle=./iposinola
        echo $cycle

        cyc=$(gcc newMAxIpo4.c -o newMaxIpo4 && ./newMaxIpo4)

        if [ $cyc -eq 1 ]
        then
            cycl=$(gcc diafora.c -o diafora && ./diafora)

        fi
        if [ $cyc -eq 1 ]
        then
            echo "kaninourgia thewria: "
```



```

gcc findExtR_N_IPO.c -o findExtR_N_IPO &&
./findExtR_N_IPO $p
cycl2=./findExt_R_N_IPO
echo $cycl2

cycl3=$(gcc new3.c -o new3 && ./new3)
else
    echo "Den vrethike thewria"
fi
echo "$nodes      $piknotita  $iposin      $cyc
$cycl  $cycl3">>data_output

echo "H diafora tw n dio thewriwn me vasi tis
eustathis epektaseis einai: $cycl3"

echo "H diafora tw n dio thewriwn me vasi tis
diadikes sxeseis epitheseis einai: $cycl"
done
perl -i -pe "s/nodes= $nodes/nodes= 0/g" $1 #replace the
old nodes with new value

perl -i -pe "s/piknotita= $piknotita/piknotita= 0/g" $1
#replace the old piknotita with new value
done

```