Ατομική Διπλωματική Εργασία

ΠΡΟΤΙΜΗΣΕΙΣ ΣΤΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Ευαγγελία Βανέζη

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Προτιμήσεις στις βάσεις δεδομένων

Ευαγγελία Βανέζη

Επιβλέπων Καθηγητής
Δημόπουλος Γιάννης

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

Μάιος 2010

Περίληψη

Σε αυτή εδώ την εργασία ασχολήθηκα όπως υποδεικνύει και ο τίτλος, με τις προτιμήσεις στις βάσεις δεδομένων. Οι βάσεις δεδομένων χρησιμοποιούνται ευρέως σε πολλά συστήματα και εφαρμογές, αφού μπορούν να συνεισφέρουν στην προσφορά μιας πιο προσωποποιημένης μορφής της τεχνολογίας ως προς τον χρήστη, πέρα από τις υπόλοιπες χρησιμότητες τους. Διάφορα συστήματα αναπτύχθηκαν και αναπτύσσονται τα οποία πέρα από τους πίνακες στην σχεσιακή βάση με τα δεδομένα που έχουν για την εκάστοτε περίπτωση, έχουν αποθηκευμένες πληροφορίες για τον χρήστη, όπως κάποιο προφίλ ή κάποιες προτιμήσεις του πάνω στα δεδομένα. Η εφαρμογή αυτών των πληροφοριών που εκφράζει ο χρήστης, πάνω στα δεδομένα που είναι αποθηκευμένα στη βάση, οδηγεί σε αυτή την προσωποποιημένη υπηρεσία που αναφέρθηκε πιο πάνω. Τα αποτελέσματα δίνονται στον χρήστη με βάση τις δικές του προτιμήσεις και επιθυμίες.

Αυτή η εφαρμογή των προτιμήσεων πάνω στα δεδομένα, θα μπορούσε να γίνει πάνω στα συνολικά αποτελέσματα που επιστρέφει η βάση δεδομένων μέσω κάποιου λογισμικού, με κάποιου είδους φιλτράρισμα των πλειάδων. Σε αυτή την εργασία μελετήθηκε το πώς θα μπορούσε να γίνει μέσα από την βάση με την χρήση ερωτήσεων προτιμήσεων, την ώρα που θα αναζητείτε το αποτέλεσμα, μειώνοντας έτσι το εύρος της αναζήτησης, το μέγεθος των αποτελεσμάτων και συνεπώς το κόστος. Σε αυτή την μελέτη παρουσιάζονται σχετικά άρθρα, με προτάσεις όσο αφορά την αναπαράσταση των προτιμήσεων, των ερωτήσεων, ακόμη και των αποτελεσμάτων, και τον συνδυασμό των προτιμήσεων με τις ερωτήσεις, με αλγόριθμους για καλύτερες αποδόσεις και μικρότερο κόστος, καθώς και βοηθητικές δομές ή διαδικασίες. Κάθε πρόταση άγγιζε το θέμα με ένα εντελώς διαφορετικό τρόπο, παρατηρώντας έτσι τις διαφορετικές σκοπιές του. Μελέτησα επίσης την θεωρία των προτιμήσεων, καθώς και την δομή αναπαράστασης CP-Nets και την σημασιολογία ceteris paribus πάνω σε προτιμήσεις.

Τέλος εφάρμοσα μια από τις προτεινόμενες μεθοδολογίες -η οποία απευθυνόταν σε προτιμήσεις χωρίς συνθήκες και που δεν διέπονταν με ceteris paribus σημασιολογία-, τις αναπαραστάσεις της και τον αλγόριθμο της, σε προτιμήσεις με συνθήκες κάτω από ceteris paribus σημασιολογία, παίρνοντας μια τροποποιημένη μεθοδολογία για αυτό τον σκοπό.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Ανάγκη για προσωποποιημένα συστήματα βάσεων δεδομένων

Στις μέρες μας μέσω της όλο και αυξανόμενης χρήσης συστημάτων βάσεων δεδομένων και της εύκολης πρόσβασης προς αυτά, οι πληροφορίες γίνονται διαθέσιμες σε αυξανόμενα μεγάλες ποσότητες σε ένα ευρύ φάσμα χρηστών. Αυτό οδηγεί σε μια ταχέως αναπτυσσόμενη κλάση ανεκπαίδευτων στο θέμα, απλών χρηστών που πλοηγούνται μέσα σε τεράστιες βάσης δεδομένων διαθέσιμες προς αυτούς. Οι χρήστες αυτοί δεν έχουν ξεκάθαρη γνώση για το τι περιέχει μια βάση δεδομένων, ούτε έχουν συγκεκριμένα αποτελέσματα στο μυαλό τους. Απλώς προσπαθούν να αναγνωρίσουν αντικείμενα που είναι χρήσιμα προς αυτούς και που ταιριάζουν περισσότερο με τις προτιμήσεις τους. Θα ήταν καλύτερο για τους χρήστες αυτούς να περιγράφουν χαρακτηριστικά δεδομένων που θα ταίριαζαν στις δικές τους προτιμήσεις, και οι βάσεις δεδομένων να μπορούν να αντεπεξέλθουν σε αυτό.

Υπάρχει λοιπόν ανάγκη να γίνεται απόκτηση και διαχείριση προτιμήσεων χρηστών σε μια βάση δεδομένων ώστε τα αποτελέσματα που δίδονται μετά από την εκτέλεση του εκάστοτε ερωτήματος να ταιριάζουν στις προτιμήσεις του χρήστη που θα τα λάβει. Πρέπει να είναι δυνατό να γίνεται παροχή προς τους χρήστες μόνο σχετικών δεδομένων από την τεράστια ποσότητα των διαθέσιμων πληροφοριών. Οι βάσεις δεδομένων δηλαδή θα πρέπει να επεξεργάζονται τα ερωτήματα με ενσωματωμένες τις προτιμήσεις αυτές, και να δίνουν ως αποτέλεσμα μια ακολουθία δεδομένων με βάση της σειράς προτίμησης τους από τον εκάστοτε χρήστη.

Οι προτιμήσεις των χρηστών λοιπόν, εάν είναι γνωστές προς το σύστημα μπορούν να χρησιμοποιούνται για φιλτράρισμα πληροφοριών και μείωση του όγκου των δεδομένων που παρουσιάζονται στον χρήστη. Επίσης χρησιμοποιούνται για αποθήκευση κάποιων προφίλ για τους χρήστες έτσι ώστε να έχει σε γνώση του το σύστημα του τι αρέσει σε κάθε χρήστη , και σχηματισμό πολιτικών για βελτίωση και αυτοματοποίηση της λήψης αποφάσεων.

Έτσι η διαχείριση προτιμήσεων των χρηστών έχει γίνει ένα αυξανόμενα σημαντικό θέμα στις μέρες μας στα πληροφοριακά συστήματα. Το πρόβλημα της απόκτησης αυτών των προτιμήσεων από τα συστήματα και της διαχείρισης τους έχει προκαλέσει μεγάλο ενδιαφέρον στην κοινότητα που ασχολείται με συστήματα βάσεων δεδομένων τα τελευταία χρόνια.

Έχει διαπιστωθεί η ανάγκη για πρόσβαση πληροφοριών επικεντρωμένη στον χρήστη και πιο προσωποποιημένο πλαίσιο βάσεων δεδομένων. Προσωποποιημένες συμπεριφορές υπάρχουν ήδη σε ιστοσελίδες και άλλα συστήματα πληροφοριών όπου η απάντηση του συστήματος προς τον χρήστη είναι διαφορετική με βάση διάφορα χαρακτηριστικά αυτού που ζητά τις πληροφορίες. Τέτοιες συμπεριφορές όμως απουσιάζουν από συστήματα βάσεων δεδομένων που πάντα δίνουν την ίδια απάντηση προς όλους για ένα δεδομένο ερώτημα.

Μια βάση δεδομένων θα πρέπει να είναι σε θέση να επεξεργάζεται ερωτήματα προτιμήσεων δηλαδή ερωτήματα που περιγράφουν επιθυμητές ιδιότητες των τελικών αποτελεσμάτων της αναζήτησης με βάση τον κάθε χρήστη. Τέτοια ερωτήματα πρέπει να είναι διαισθητικά και ευέλικτα, σωστά μεταγλωττισμένα από το σύστημα, και υπολογιστικά αποδοτικά ως προς την επεξεργασία τους. Τα ερωτήματα προτιμήσεων είναι καίρια για διάφορες εφαρμογές αφού επιτρέπουν στους χρήστες να ανακαλύπτουν και να ταξινομούν τα δεδομένα που τους ενδιαφέρουν.

Σε μια τέτοια βάση υπάρχουν δυο κύριες ανησυχίες : η σημασιολογική σαφήνεια και επάρκεια και η υπολογιστική απόδοση. Το πρώτο βήμα στην υποστήριξη ερωτημάτων προτιμήσεων δηλαδή η σημασιολογική σαφήνεια και επάρκεια υποδεικνύει ότι πρέπει να μπορούν να μεταφραστούν ορθά οι προτιμήσεις των χρηστών από τον τρόπο που οι ίδιοι τις διατυπώνουν, ώστε να χρησιμοποιούνται στα ερωτήματα.

1.2 Σχετική βιβλιογραφία

Στην εργασία αυτή γίνεται μελέτη άρθρων με προτάσεις για προσωποποιημένα συστήματα, για εξαγωγή των κατάλληλων αποτελεσμάτων απευθείας από την βάση δεδομένων με χρήση ερωτημάτων προτιμήσεων όπως αναφέρθηκε και στην πιο πάνω ενότητα. Όλα τα άρθρα βασίζονται στο ίδιο σκεπτικό, αυτό που αναλύθηκε στην υποενότητα 1.1.

Όλα λαμβάνουν υπόψη την χρήση προτιμήσεων από τους χρήστες κάθε συστήματος, την ενσωμάτωση τους στα ερωτήματα, την εκτέλεση αυτών των ερωτημάτων και την εξαγωγή συμπερασμάτων.

Το κάθε άρθρο βλέπει όμως αυτό το θέμα από διαφορετική σκοπιά και φάσμα. Το κάθε ένα παρουσιάζει μια διαφορετική προσέγγιση ως προς την αναπαράσταση των προτιμήσεων που δίδονται από τους χρήστες για πιο αποδοτική εξαγωγή των αποτελεσμάτων και πιο αποτελεσματικό συνδυασμό τους με τα δεδομένα ερωτήματα.

Σε μερικά άρθρα δίνεται μεγαλύτερη έμφαση ως προς τις δομές αναπαράστασης που προτείνεται να χρησιμοποιηθούν, και σε άλλα μεγαλύτερη έμφαση ως προς τους αλγορίθμους εξαγωγής των αποτελεσμάτων.

Κάποια άρθρα ασχολούνται με τον τρόπο απόκτησης των σχετικών προτιμήσεων από τους χρήστες, και κάποια άλλα θεωρούν δεδομένη την ύπαρξη των προτιμήσεων στο σύστημα, και ασχολούνται με την περαιτέρω επεξεργασία τους και μόνο.

Άλλα άρθρα αναφέρουν τις δομές αποθήκευσης των χρηστών με τον γενικότερο όρο «προφίλ», και άλλα αναφέρουν λεπτομερώς τις κατάλληλες δομές που προτείνουν.

Μια άλλη σημαντική διαφορά είναι το είδος των προτιμήσεων με τις οποίες ασχολείται κάθε άρθρο. Σε όλα οι προτιμήσεις εκφράζονται πάνω σε χαρακτηριστικά σχέσεων σε μια σχεσιακή βάση δεδομένων. Σε κάποια οι προτιμήσεις είναι απλές, και σε κάποια άλλα εξαρτώνται από κάποιου είδους συνθήκες που ονομάζονται συμφραζόμενα.

Επίσης υπάρχει διαφορά ως προς τον τρόπο έκφρασης των προτιμήσεων. Οι δύο διαφορετικές προσεγγίσεις που ακολουθούνται από τα άρθρα είναι οι ακόλουθες:

Ποσοτική προσέγγιση: σε κάποια άρθρα γίνεται ανάθεση αριθμητικών σκορ στις προτιμήσεις και έτσι εκφράζουν οι χρήστες το ενδιαφέρον τους για τα εκάστοτε χαρακτηριστικά
Ποιοτική προσέγγιση: σε κάποια άρθρα οι προτιμήσεις ανάμεσα σε δύο κομμάτια δεδομένων καθορίζονται άμεσα, κυρίως με χρήση δυαδικών σχέσεων προτιμήσεων.

Στα επόμενα κεφάλαια δίνονται οι αναλύσεις των άρθρων και οι επεξηγήσεις του τι προτείνουν με τα κατάλληλα παραδείγματα.

1.3 Προτιμήσεις με συνθήκες, ceteris paribus και εφαρμογή τους

Έχει γίνει μελέτη επίσης πάνω στις προτιμήσεις σε βάσεις δεδομένων που εξαρτώνται από συνθήκες που δίνει και πάλι ο χρήστης. Η μελέτη αφορά την σημασιολογική ερμηνεία αυτών των προτιμήσεων με βάση τη σημασιολογία ceteris paribus.

Ακολούθως έχει γίνει εφαρμογή μεθοδολογιών από τα άρθρα πάνω σε προτιμήσεις αυτού του είδους, με τις κατάλληλες διαφοροποιήσεις στις μεθοδολογίες έτσι ώστε να μπορέσουν να τις υποστηρίξουν.

Κεφάλαιο 2

Προτιμήσεις (Με βάση το [6])

2.1 Εισαγωγή

Οι προτιμήσεις και η διαχείριση τους είναι ένα φλέγων θέμα όσο αφορά την σχεδίαση και ανάπτυξη συστημάτων που αυτοματοποιούν ή υποστηρίζουν λήψη αποφάσεων μέσω υπολογιστών, όπως θα το έπραττε και ένας άνθρωπος. Η κατανόηση αυτού του θέματος σε βάθος είναι κάτι τεράστιας σημασίας ως προς την επίτευξη των πιο πάνω στόχων. Κάθε τέτοιο σύστημα ή εφαρμογή προσπαθεί να εξάγει το καλύτερο αποτέλεσμα προς τον χρήστη με βάση τις προτιμήσεις και τα ζητούμενα του.

Σε αρκετά πεδία τιμών σε πραγματικά προβλήματα, ο χρήστης δεν μπορεί να γνωρίζει το σύνολο των πιθανών λύσεων που θα μπορούσε να πάρει και έτσι απλά ζητά την καλύτερη γι' αυτόν λύση χωρίς όμως να μπορεί να την περιγράψει ή να καθορίσει τα χαρακτηριστικά της. Εάν ο χρήστης έπρεπε να καθορίσει την απάντηση, χωρίς να έχει την γνώση, θα ζητούσε είτε κάποιο στόχο που δεν μπορεί να επιτευχθεί λόγω του ότι ίσως να μην προσφέρετε τέτοια απάντηση είτε θα ζητούσε πιο λίγα από ότι θα μπορούσε να πάρει διότι ίσως να υπήρχε καλύτερη (για αυτόν) απάντηση. Οπότε θέλουμε τα συστήματα αυτά να αντιλαμβάνονται και να κατανοούν τις προτιμήσεις των χρηστών πάνω σε διάφορες εναλλακτικές επιλογές και να εντοπίζουν την καταλληλότερη λύση από το σύνολο όλων των πιθανών λύσεων που μπορούν να υπάρξουν στο εκάστοτε πρόβλημα.

2.2 Γενική σημασιολογία των προτιμήσεων

Σημασιολογικά οι προτιμήσεις σε κάποιο πεδίο πιθανών επιλογών, ταξινομούν τις επιλογές αυτές, έτσι ώστε μια πιο επιθυμητή επιλογή να προηγείται μιας λιγότερο επιθυμητής. Κάθε στοιχεία αυτού του συνόλου επιλογών, δηλαδή κάθε πιθανή επιλογή/ανάθεση για το πρόβλημα θα αναφέρεται ως "έκβαση".
Ας υποθέσουμε ότι έχουμε ένα σύνολο V εκβάσεων. Μια σχέση προτιμήσεων σε αυτό το σύνολο είναι μια μεταβατική δυαδική σχέση $\succeq$.
Ιδιότητες ταξινομήσεων επιλογών:

1) Αν μια ταξινόμηση επιλογών είναι ολική (total), τότε κάθε ζεύγος επιλογών είναι συγκρίσιμο. Δηλαδή εάν για κάθε ζευγάρι α,α' ϵ V, ισχύει είτε ότι α $\succeq$ α' είτε α' $\succeq$ α.
2) Αν μια ταξινόμηση επιλογών είναι μερική (partial), τότε δεν είναι κάθε ζεύγος επιλογών συγκρίσιμο
3) Αν μια ταξινόμηση επιλογών είναι αυστηρή (strict), δεν υπάρχουν δύο εκβάσεις που να είναι το ίδιο επιθυμητές. Δηλαδή εάν η σχέση είναι αντισυμμετρική.
4) Αν μια ταξινόμηση επιλογών είναι αδύναμη (weak), τότε μπορεί να υπάρχουν και εκβάσεις που να είναι το ίδιο επιθυμητές.

Ας υποθέσουμε ότι έχουμε μια αδύναμη ταξινόμηση $\succeq$, τότε μπορεί να οριστεί η αυστηρή ταξινόμηση $\succ$ ως ακολούθως: α $\succ$ α' αν και μόνο εάν ισχύει α $\succeq$ α' , αλλά δεν ισχύει α' $\succeq$ α. Εάν ίσχυαν και τα δύο τότε θα ίσχυε και α $\sim$ α' , δηλαδή είναι και τα δύο το ίδιο επιθυμητά.

Ένα αρκετά σημαντικό ζήτημα είναι ο καθορισμός της σχέσης προτιμήσεων για ένα σύνολο επιλογών. Αυτό δεν είναι δύσκολο όταν αυτό που μας ενδιαφέρει είναι ένα και μόνο χαρακτηριστικό. Τυπικά όμως οι προτιμήσεις των χρηστών είναι πολύ πιο περίπλοκες και έτσι η ταξινόμηση-σύγκριση ακόμη και δύο εκβάσεων μεταξύ τους ίσως να γίνεται με σχετική δυσκολία. Καθώς το μέγεθος του διαστήματος των εκβάσεων μεγαλώνει, ταυτόχρονα μεγαλώνει και η δυσκολία, καθώς και επιπρόσθετα υπολογιστικά προβλήματα και προβλήματα αναπαράστασης εμφανίζονται.

Οι μέθοδοι απόκτησης προτιμήσεων στοχεύουν στο να κάνουν πιο εύκολη την διαδικασία ταξινόμησης ενός συνόλου εκβάσεων, ή εύρεσης του βέλτιστου αντικειμένου.

Περιλαμβάνουν γλώσσες κατάλληλες για έκφραση προτιμήσεων, στρατηγικές ερωτήσεων κ.ο.κ.

Οι μέθοδοι αναπαράστασης προτιμήσεων ψάχνουν τρόπους αποδοτικής αναπαράστασης πληροφοριών σχετικά με τις προτιμήσεις. Μια συμπαγής αναπαράσταση θα χρειάζεται λιγότερες καθορισμένες πληροφορίες και αυτό κάνει την απόκτηση των προτιμήσεων ευκολότερη.

Οι αλγόριθμοι διαχείρισης προτιμήσεων προσπαθούν να προσφέρουν αποδοτικά μέσα για να δίδονται απαντήσεις σε συνηθισμένα ερωτήματα σε μοντέλα προτιμήσεων.

Υπάρχουν διαφόρων τύπων περιοχές λήψης αποφάσεων όπου απαιτείται διαχείριση προτιμήσεων. Κάθε τέτοια περιοχή έχει διαφορετικές απαιτήσεις και μια λύση που είναι η ιδανική για την μια περιοχή, για την άλλη περιοχή μπορεί να μην θεωρείται και τόσο καλή. Χονδρικώς θα μπορούσαμε να κάνουμε διαχωρισμό σε τρεις κλάσεις: online εφαρμογές με οποιουσδήποτε χρήστες όπου οι χρήστες σε τέτοιες περιπτώσεις είναι απρόθυμοι να παρέχουν μεγάλη ποσότητα πληροφοριών, χρόνου και προσπάθειας, δεν τίθεται θέμα να εκπαιδευτούν, δεν περιμένουμε οι χρήστες α έχουν οποιεσδήποτε τεχνικές γνώσεις, και οι πληροφορίες είναι τυπικά " ποιοτικής" μορφής. Διαμόρφωση και σχεδιασμός συστημάτων, όπου οι προτιμήσεις καθοδηγούν την συμπεριφορά κάποιων συστημάτων για τα οποία οι χρήστες ενδιαφέρονται άρα και μπορούμε να περιμένουμε από αυτούς περισσότερη προσπάθεια και να μπορεί να υπάρχει μια λίγο πιο τεχνική γλώσσα. Τέλος πραγματικά προβλήματα αποφάσεων με αυξημένη πολυπλοκότητα τα οποία απαιτούν την βοήθεια κάποιου αναλυτή αποφάσεων, καθώς και εργαλεία όπως μοντέλα πιθανοτήτων και συναρτήσεις χρησιμότητας (utility functions)

2.3 Μοντελοποίηση προτιμήσεων

Ένα μοντέλο είναι μια μαθηματική δομή που επιχειρεί να συλλάβει όλες τις ιδιότητες της φυσικής, πνευματικής ή γνωστικής έννοιας που είναι υπό μελέτη. Σκοπός ενός μοντέλου είναι να παρέχει μια συγκεκριμένη μορφή στην αφηρημένη έννοια που αναπαριστά. Οπότε ένα μοντέλο προτιμήσεων πρέπει να έχει διαισθητική εφαρμογή, αφού ουσιαστικά δίνει ερμηνεία στις απαντήσεις που υπολογίζονται για τα ερωτήματα.

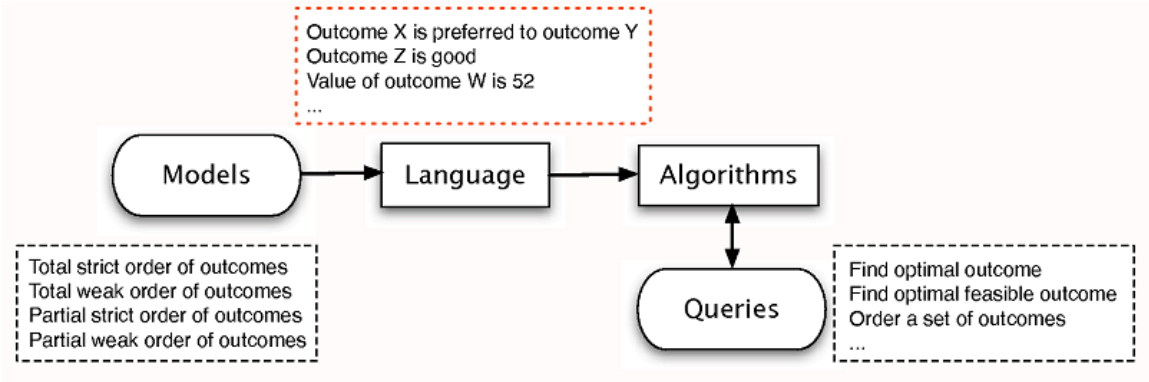
Ένα φυσικό μοντέλο προτιμήσεων είναι μια διάταξη $\succeq$ πάνω σε ένα σύνολο πιθανών εκβάσεων Ω.

Τυπικά ερωτήματα είναι το να βρεθεί η βέλτιστη έκβαση-λύση, να βρεθεί η βέλτιστη λύση που να ικανοποιεί ορισμένους περιορισμούς συγκρίνοντας δύο εκβάσεις, να ταξινομηθεί το σύνολο των εκβάσεων, να συναθροιστούν οι προτιμήσεις πολλαπλών χρηστών, να βρεθεί μια καλή δέσμευση υλικών κ.ο.κ.

Από την στιγμή που υπάρχει το μοντέλο και υπάρχει έστω ένα ερώτημα, τότε χρειαζόμαστε αλγορίθμους που υπολογίζουν μια απάντηση σε αυτό που ζητά το ερώτημα, με χρήση του μοντέλου.

Ένα πολύ σημαντικό θέμα είναι η απόκτηση των πληροφοριών που καθορίζουν το μοντέλο. Θα μπορούσαν να ερωτούνται οι χρήστες για αυτές τις πληροφορίες κάτι που είναι μεν απλό αλλά καθόλου αποδοτικό και εύκολο, εκτός αν το Ω είναι πολύ μικρό. Έτσι πρέπει να χρησιμοποιηθεί κάποιο εργαλείο για καθορισμό του μοντέλου με κάποιο συμπαγή τρόπο. Η γλώσσα είναι ένα τέτοιο εργαλείο. Αποτελείται από μια συλλογή πιθανών δηλώσεων που βοηθούν στην περιγραφή μοντέλων. Αυτές οι δηλώσεις για να αποκτήσουν νόημα χρειάζεται να χρησιμοποιηθεί πάνω τους μια φόρμουλα μεταγλώττισης που αντιστοιχεί δηλώσεις σε μοντέλα.

Σχήμα 2.3.1



Σχήμα από το άρθρο που απεικονίζει το μετα-μοντέλο

Άρα βλέπουμε ότι έχουμε πέντε βασικά στοιχεία που αποτελούν το μετά-μοντέλο:
1. Το μοντέλο που περιγράφει την δομή που μας ενδιαφέρει
2. Τα ερωτήματα, που είναι ερωτήσεις σχετικά με το μοντέλο, για τις οποίες ζητείται απάντηση

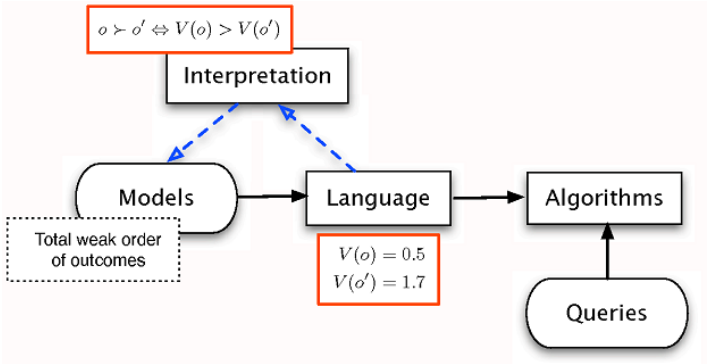
3. Τους αλγόριθμους για υπολογισμό απαντήσεων για τα ερωτήματα

4. Την γλώσσα που επιτρέπει τον έμμεσο καθορισμό του μοντέλου
5. Τις συναρτήσεις μεταγλώττισης που αντιστοιχούν τις εκφράσεις της γλώσσας σε μοντέλα.

2.3.1 Συναρτήσεις απόδοσης τιμής, Δομές και Ανεξαρτησία

Η πιο απλή περίπτωση είναι αυτή του μοντέλου που είναι μια ασθενές ολική διάταξη $\succeq$, και η γλώσσα είναι το ίδιο το μοντέλο. Τα θετικά σε αυτή την περίπτωση είναι ότι είναι πάντα πιθανό να απαντηθούν κοινά ερωτήματα και ταυτόχρονα κάνει την επιλογή της μεταγλώττισης τετριμμένη αφού απλά θα τεθεί ως η ταυτοτική συνάρτηση. Όμως παρά τα θετικά , έχουμε ένα μεγάλο αρνητικό που είναι ότι η επιλογή της γλώσσας εδώ είναι αρκετά φτωχή.

Σχήμα 2.3.1.1
Πιο κάτω βλέπουμε το μετά-μοντέλο για αυτή τη γλώσσα:



Άρα, στην προσπάθεια για βελτίωση του μοντέλου, αρχικά θα μπορούσε να ζητηθεί από τους χρήστες αντί να καθορίζουν άμεσα την ταξινόμηση των εκβάσεων/επιλογών, να αναθέτουν ένα πραγματικό αριθμό σε κάθε έκβαση. Έτσι τώρα η γλώσσα παίρνει την μορφή μιας συνάρτησης τιμών V: Ω → R , και η συνάρτηση μεταγλώττισης παίρνει την μορφή: o$\succeq$o' ⇔

V(o) ≥V(o').

Η ανάθεση πραγματικών τιμών σε κάθε έκβαση είναι ελάχιστα ευκολότερη από την απευθείας ταξινόμηση των εκβάσεων, αν και από κάποιους θεωρείται αρκετά δύσκολο. Αυτή η μέθοδος δηλαδή η ανάθεση ενός σκορ σε κάθε πιθανό αποτέλεσμα ονομάζεται ποσοτικοποίηση.

Έτσι η συνάρτηση τιμών ορίζεται άμεσα σε μορφή ενός πίνακα που συμπληρώνεται με τα στοιχεία(τιμές) που δίνει ο χρήστης για κάθε έκβαση. Θα μπορούσε να υπάρχει μια πιο

κατάλληλη και συμπαγής μορφή συνάρτησης, που να είναι διαισθητική προς τον χρήστη και να εξαφανίζει την πολυπλοκότητα του να αποκτούνται οι προτιμήσεις του χρήστη για μια-μια επιλογή ξεχωριστά.

Κάθε έκβαση έχει μια συγκεκριμένη δομή η οποία δίδεται από ένα σύνολο χαρακτηριστικών. Δηλαδή κάθε έκβαση περιέχει ένα σύνολο χαρακτηριστικών, και η επιλογή ανάμεσα στις εκβάσεις κατευθύνεται από τις τιμές που έχουν σε κάθε περίπτωση αυτά τα χαρακτηριστικά. Υποθέτουμε την ύπαρξη και την γνώση ενός συνόλου χαρακτηριστικών $X=X_1,...,X_n$ τέτοια ώστε $\Omega=X=x_{i=1\ to\ n}\ Dom(X_i)$, όπου $Dom(X_i)$ είναι το πεδίο τιμών του χαρακτηριστικού X_i.

2.3.1.1 Ανεξαρτησία προτιμήσεων

Αν τα Y και Z είναι διαμερίσεις του συνόλου χαρακτηριστικών X, δηλαδή είναι υποσύνολα των οποίων η τομή είναι το κενό, και η ένωση το X, λέμε ότι το Y είναι κατά προτίμηση ανεξάρτητο από το Z, εάν για όλα τα y1,y2 ϵ Dom(Y) και για κάθε z ϵ Dom(Z) :

y1z $\succeq$ y2z συνεπάγεται ότι για κάθε z' ϵ Dom(Z) : y1z' $\succeq$ y2z'

Αυτό σημαίνει ότι οι προτιμήσεις πάνω σε τιμές του Y συνόλου χαρακτηριστικών δεν εξαρτώνται από την τιμή του Z συνόλου χαρακτηριστικών όσο αυτή η τιμή είναι σταθερή. Δηλαδή αν το Z παίρνει την τιμή z τότε υπάρχει μεγαλύτερη προτίμηση για το y1z παρά για το y2z, και αν το Z πάρει την τιμή z' τότε και πάλι υπάρχει μεγαλύτερη προτίμηση για το y1z' παρά για το y2z' ανεξάρτητα από την σταθερή τιμή που παίρνει το Z.

Όταν το Y είναι κατά προτίμηση ανεξάρτητο από το Z, τότε μπορεί να οριστεί η προβολή του $\succeq$ στο Y με βάση κάποιο χαρακτηριστικό. Επίσης η κατά προτίμηση ανεξαρτησία του Y από το Z επιτρέπει στον χρήστη να εκφράσει τις προτιμήσεις του μεταξύ ζευγαριών εκβάσεων σε όρους του Y μόνο.

Η κατά προτίμηση ανεξαρτησία είναι μια κατευθυνόμενη έννοια, δηλαδή εάν το Y είναι κατά προτίμηση ανεξάρτητο από το Z, δεν σημαίνει ότι και το Z είναι κατά προτίμηση ανεξάρτητο από το Y.

Μια πιο αδύναμη έννοια ανεξαρτησίας είναι αυτή της υπό συνθήκες κατά προτίμηση ανεξαρτησίας. Εάν τα Y,Z,C είναι διαμερίσεις του συνόλου X, λέμε ότι το Y είναι υπό συνθήκες κατά προτίμηση ανεξάρτητο από το Z δοθέντος ότι C= c εάν όταν γίνεται προβολή

της ταξινόμησης των εκβάσεων όπου C=c, το Y θα είναι κατά προτίμηση ανεξάρτητο από το Z. Ουσιαστικά για κάθε y1,y2 ϵ Dom(Y) και για κάθε z ϵ Dom(Z) :

y1zc $\succeq$ y2zc συνεπάγεται ότι για κάθε z' ϵ Dom(Z) : y1z'c $\succeq$ y2z'c

Το Y είναι κατά προτίμηση ανεξάρτητο από το Z δοθέντος του C εάν για κάθε c ϵ Dom(C), ισχύει ότι το Y είναι κατά προτίμηση ανεξάρτητο από το Z δοθέντος ότι C=c. Σε αυτή την περίπτωση οι προτιμήσεις πάνω στις τιμές του Y δεν εξαρτώνται από τιμές του Z για οποιαδήποτε σταθερή τιμή του c. Η ταξινόμηση των προτιμήσεων πάνω στις τιμές του Y μπορεί να είναι διαφορετική για διαφορετικές τιμές του C, αλλά για κάθε τιμή του C, η ταξινόμηση είναι ανεξάρτητη από την τιμή του Z.

2.3.1.2 Συμπαγείς συναρτήσεις απόδοσης τιμών και αθροιστικότητα

Μια αναπαράσταση με βάση τα χαρακτηριστικά του χώρου των εκβάσεων είναι $\Omega = x_{i=1,\dots,n} Dom(Xi)$, και μια αναπαράσταση των συναρτήσεων απόδοσης τιμών είναι V: $x_{i=1,\dots,n} Dom(Xi) \rightarrow R$.

Με βάση την ταξινόμηση προτιμήσεων του χρήστη, είναι πιθανές πιο πυκνές/συμπαγείς μορφές του V. Πιθανών η πιο γνωστή είναι αυτή του αθροίσματος των συναρτήσεων πάνω σε ανεξάρτητα χαρακτηριστικά:

$$V(X1,\dots,Xn) = \sum_{i=1}^{n} Vi(Xi)$$

Τέτοιες συναρτήσεις είναι αθροιστικά ανεξάρτητες. Εάν μια συνάρτηση έχει αυτή την μορφή τότε για κάθε υποσύνολο χαρακτηριστικών Y του X, το Y είναι κατά προτίμηση ανεξάρτητο από το X\Y. Αυτή η ισχυρή μορφή ανεξαρτησίας ονομάζεται αμοιβαία ανεξαρτησία.

Τα μεγαλύτερο πλεονέκτημα σε μια τέτοια παραγοντοποιημένη μορφή συνάρτησης είναι ότι είναι πιο εύκολο να καθοριστεί, άρα και να αποκτηθεί από τους χρήστες. Εάν η V είναι αθροιστικά ανεξάρτητη πάνω σε n δυαδικά χαρακτηριστικά, ο καθορισμός της απαιτεί 2n τιμές σε αντίθεση με τις 2^n τιμές για μια tabular μη δομημένη αναπαράσταση. Επίσης το να βρεθεί η βέλτιστη έκβαση σε κάποια συναθροιστικά ανεξάρτητη V μπορεί να γίνει σε χρόνο O(n) , ενώ σε μια tabular μη δομημένη αναπαράσταση χρειάζεται $O(2^n)$ χρόνος.

Από την άλλη μια τέτοια μορφή είναι περιοριστική. Υπονοεί πως οι προτιμήσεις είναι χωρίς συνθήκες, δηλαδή η ταξινόμηση των προτιμήσεων πάνω σε τιμές χαρακτηριστικών είναι ίδιες χωρίς να έχει σημασία τι τιμές έχουν τα άλλα χαρακτηριστικά. Επίσης η δύναμη των διαφόρων προτιμήσεων μεταξύ τιμών χαρακτηριστικών είναι σταθερή.

2.3.1.3 Γενικευμένη συνάθροιση

Μια πιο γενική μορφή αθροιστικής ανεξαρτησίας που μας επιτρέπει να είμαστε όσο γενικοί επιθυμούμε και τίποτα περισσότερο, είναι αυτή της γενικευμένης αθροιστικής ανεξαρτησίας (GAI). Η μορφή αυτή είναι:

$$V(X1,....,Xn)= \sum_{i=1}^{k} Vi(Zi)$$

όπου Z1,...,Zk είναι υποσύνολα του X και αφορούν τα χαρακτηριστικά του X που επηρεάζονται από τις προτιμήσεις. Η V δηλαδή είναι ένα άθροισμα k παραγόντων, όπου κάθε παράγοντας εξαρτάται από κάποιο υποσύνολο χαρακτηριστικών X, και τα υποσύνολα χαρακτηριστικών που σχετίζονται με διαφορετικούς παράγοντες πρέπει να μην έχουν ως κενό την τομή τους.
Οι δύο ακραίες περιπτώσεις είναι να επιλεγεί για k=1 και Z1=X, όπου αναπαριστάται οποιαδήποτε συνάρτηση απόδοσης τιμών ή να επιλεγεί για k=n κα Zi={Xi} που είναι η γλώσσα των συναθροιστικά ανεξάρτητων συναρτήσεων απόδοσης τιμών.

Ένα ακόμη συστατικό του μετά-μοντέλου είναι η αναπαράσταση των πληροφοριών που παρέχονται από τον χρήστη. Είναι μια δομή που παίρνει όλες τις πληροφορίες, και έχει ως σκοπό να χρησιμοποιηθεί απευθείας από τον αλγόριθμο που βρίσκει απαντήσεις για τα ερωτήματα. Μια φυσική επιλογή για αναπαράσταση είναι η ίδια η γλώσσα. Κάποιες φορές όμως είναι χρήσιμο η έκφραση εισόδου να μεταλλάσετε ελαφρώς ή να αυξάνεται για να λαμβάνεται μια αναπαράσταση που να παρέχει ποιο διαισθητικά μέσα για ανάλυση της πολυπλοκότητας του να απαντιούνται τα ερωτήματα και για κατασκευή αποδοτικών αλγορίθμων για εύρεση απαντήσεων στα ερωτήματα.
Συνήθως τέτοιες αναπαραστάσεις έρχονται σε μορφή γράφου με σχόλια, που περιγράφει τις εξαρτήσεις μεταξύ διαφόρων συστατικών του προβλήματος. Οι αναπαραστάσεις εξάγονται από την γλώσσα και οι αλγόριθμοι κάνουν χρήση των ιδιοτήτων τέτοιων γραφικών δομών.
GAI-δίκτυο: είναι ένας γράφος με σχόλια, του οποίου οι κόμβοι αντιπροσωπεύουν τα χαρακτηριστικά στο X. Μια ακμή ενώνει τους κόμβους που αντιστοιχούν σε ζευγάρια χαρακτηριστικών Xi,Xj ϵ X, εάν Xi και Xj είναι μαζί σε ένα παράγοντα δηλαδή {Xi,Xj} υποσύνολα του Zm όπου m ϵ {1,...,k}. Κάθε κλίκα σε αυτό τον γράφο αντιστοιχεί σε ένα συνδυασμό χαρακτηρηστικών, και σχολιάζεται με ένα πίνακα που περιγράφει τις τιμές διαφορετικών στιγμιότυπων αυτού του συνδυασμού.

2.3.2 Μερική ταξινόμηση και ποιοτικοποιημένες γλώσσες

Μια γλώσσα πρέπει να έχει κάποιες ιδιότητες ώστε ο απλός χρήστης να μπορεί άνετα να εκφράσει τις προτιμήσεις του και να απαντήσει ερωτήσεις σχετικά με τις προτιμήσεις του. Πρώτον μια τέτοια γλώσσα πρέπει να βασίζεται σε πληροφορίες που οι χρήστες να μπορούν εύκολα να εκφράσουν. Δεύτερον οι πληροφορίες που υπάρχουν σε μια τέτοια γλώσσα θα πρέπει να είναι σχετικά εύκολο να μεταφραστούν. Τέλος θα πρέπει να είναι δυνατόν να απαντιούνται αποτελεσματικά τα ερωτήματα σχετικά με τις προτιμήσεις των χρηστών. Τέτοια γλώσσα άρα θα πρέπει να βασίζεται σε κομμάτια ποιοτικοποιημένων πληροφοριών σχετικά με το μοντέλο.

Ένα είδος ποιοτικοποιημένων δηλώσεων είναι μια άμεση σύγκριση μεταξύ εναλλακτικών εκβάσεων. Τέτοιες δηλώσεις είναι πολύ εύκολο να μεταφραστούν αφού υποδεικνύουν μια συσχέτιση ταξινόμησης. Αλλά από την άλλη τέτοιες δηλώσεις είναι πολύ συγκεκριμένες, και προσφέρουν μόνο μια μικρή ποσότητα πληροφοριών σχετικά με το μοντέλο προτιμήσεων του χρήστη.

Ένα άλλος είδος είναι οι γενικευμένες δηλώσεις όσο αφορά τιμές που μπορεί να πάρουν τα χαρακτηριστικά. Τέτοιες δηλώσεις είναι φυσικές στην καθημερινή ομιλία των χρηστών, αλλά ταυτόχρονα προσφέρουν πολύ περισσότερες πληροφορίες διότι μια τέτοια δήλωση έμμεσα κωδικοποιεί πολλές συγκρίσεις μεταξύ συγκεκριμένων εκβάσεων.

Τέτοιες γενικευμένες εκφράσεις προτιμήσεων, έχουν την μορφή:

$$S= \{s_1,..,s_m\}=\{<\varphi_1 \ominus_1 \psi_1>,\ldots,<\varphi_m \ominus_m \psi_m>\}$$

όπου κάθε $s_i= \varphi_i \ominus_i \psi_i$ είναι μια μονή δήλωση προτιμήσεων με:

- φ_i ,ψ_i να είναι κάποιες λογικές φόρμουλες πάνω στο X (για τις τιμές των χαρακτηριστικών)
- $\ominus_i \in \{\succ, \succeq, \sim\}$ και
- $\succ$, $\succeq$ και $\sim$ έχουν την σταθερή σημασιολογία των ισχυρών προτιμήσεων, αδύναμων προτιμήσεων και ισότητας προτιμήσεων αντίστοιχα.

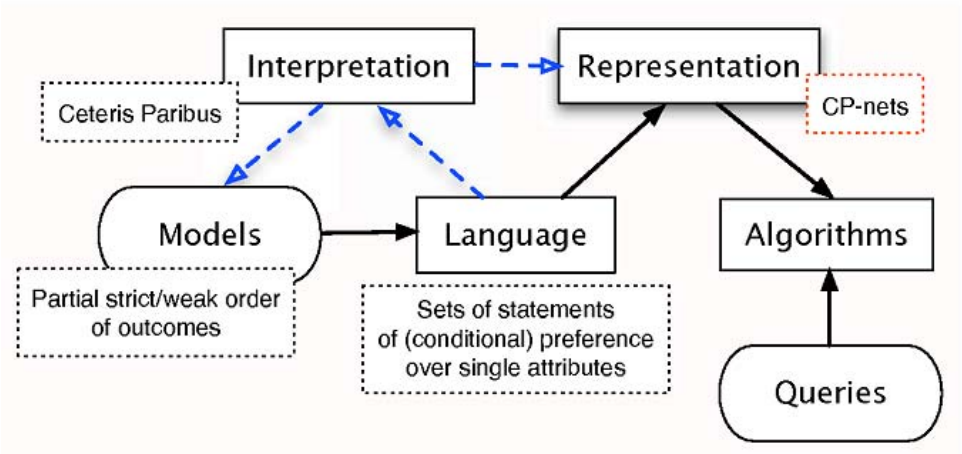
Με την προσθήκη τέτοιων γενικευμένων δηλώσεων στην γλώσσα, διευκολύνεται η πιο συμπαγής επικοινωνία του μοντέλου προτιμήσεων, αλλά έχει ένα τίμημα σε επίπεδο μεταγλώττισης της γλώσσας, διότι το ακριβές νόημα που βάζει ο χρήστης σε μια γενικευμένη δήλωση δεν είναι πάντα ξεκάθαρο.

Totalitarian σημασιολογία: σύμφωνα με αυτό τον τρόπο ερμηνείας, η προτίμηση του χρήστη για το φ1 σε σχέση με το ψ1, σημαίνει ότι ανεξάρτητα με τα υπόλοιπα χαρακτηριστικά που έχουν οι 2 εκβάσεις, είναι προτιμότερη αυτή που για το συγκεκριμένο χαρακτηριστικό ισχύει το φ1, σε σχέση με εκείνη στην οποία ισχύει το ψ1.

Ceteris Paribus σημασιολογία: σύμφωνα με αυτό τον τρόπο ερμηνείας, η προτίμηση του χρήστη για το φ1 σε σχέση με τοψ1, σημαίνει ότι ο χρήστης θα προτιμά την έκβαση στην οποία ισχύει το φ1 από αυτήν στην οποία ισχύει το ψ1, δεδομένου ότι όλα τα υπόλοιπα χαρακτηριστικά έχουν τις ίδιες ακριβώς τιμές στις 2 εκβάσεις. Αυτή είναι η πιο συντηρητική και ταυτόχρονα αδύναμη φυσική ερμηνεία τέτοιων γενικευμένων δηλώσεων. Από την μια παρέχει τους πιο αδύναμους περιορισμούς στο μοντέλο προτιμήσεων του χρήστη, και από την άλλη δεν είναι πιθανόν να φτάσει σε συμπεράσματα που δεν είναι σίγουρο αν ισχύουν.

Πιο κάτω γίνεται μια σύντομη παρουσιάση των CP-Networks.

Σχήμα 2.3.2.1
Στο πιο κάτω σχήμα είναι το μετά-μοντέλο για CP-Nets.



Το μοντέλο των προτιμήσεων του χρήστη στην προσέγγιση αυτή είναι αυτό της μερικής ταξινόμησης πάνω στις εκβάσεις.
Η γλώσσα που χρησιμοποιείται είναι αυτή των δηλώσεων προτίμησης πάνω στις τιμές μονών χαρακτηριστικών των εκβάσεων. Τυπικά η γλώσσα των CP-Nets αποτελείται από εκφράσεις προτιμήσεων της μορφής:

$$S \subseteq \{\mathbf{y} \wedge x_i \succ \mathbf{y} \wedge x_j \mid X \in \mathbf{X}, \mathbf{Y} \subseteq \mathbf{X} \setminus \{X\}, x_i, x_j \in Dom(X), \mathbf{y} \in Dom(\mathbf{Y})\}$$

Το Y μπορεί να είναι ένα οποιοδήποτε υποσύνολο χαρακτηριστικών το οποίο δεν περιέχει το αναφερόμενο χαρακτηριστικό X, και κάθε δήλωση $\mathbf{y} \wedge x_i \succ \mathbf{y} \wedge x_j$ στην έκφραση λέει ότι σε κάποιο πλαίσιο στο οποίο ανατίθεται το y στο Y, η τιμή xi είναι προτιμότερη από την xj για το X.

Η συνάρτηση απόδοσης ερμηνείας που χρησιμοποιείται στα CP-Nets αντιστοιχεί στην ceteris paribus ερμηνεία ξεχωριστών δηλώσεων, και ακολούθως ένας μεταβατικός γράφος προτιμήσεων προκαλείται από αυτές τις δηλώσεις.

Συγκεκριμένα μια δήλωση $\mathbf{y} \wedge x_i \succ \mathbf{y} \wedge x_j$ ερμηνεύεται ως μια ταξινόμηση προτιμήσεων όπου η έκβαση ο προτιμείται από την έκβαση ο' , οποτεδήποτε η έκβαση ο και η έκβαση ο' συμφωνούν σε όλα τα υπόλοιπα χαρακτηριστικά εκτός του X (X\{X}) και οι δύο εκβάσεις αναθέτουν την τιμή y στο Y, ενώ το ο αναθέτει την τιμή xi στο X και το ο' αναθέτει την τιμή xj στο X.

Στην προσέγγιση των CP-Nets η έκφραση προτιμήσεων αναπαριστάται με τη χρήση ενός σχολιασμένου γράφου. Οι κόμβοι αντιστοιχούν σε χαρακτηριστικά εκβάσεων, και οι ακμές περιέχουν την δομή των δηλώσεων προτίμησης. Συγκεκριμένα για κάθε δήλωση $\mathbf{y} \wedge x_i \succ \mathbf{y} \wedge x_j$ προστίθεται μια ακμή από κάθε κόμβο που αντιπροσωπεύει ένα χαρακτηριστικό Y'єY προς τον κόμβο που αντιπροσωπεύει το χαρακτηριστικό X, διότι οι προτιμήσεις για τις τιμές του X εξαρτώνται από την τιμή του Y'. Τέλος κάθε κόμβος σχολιάζεται με όλες τις δηλώσεις που περιγράφουν προτιμήσεις για την τιμή του.

Η σημασιολογία ceteris paribus και η ανεξαρτησία προτιμήσεων με συνθήκες είναι πολύ σχετικές έννοιες. Και στις δύο περιπτώσεις κάποιο κομμάτι είναι σταθερό, και ακολούθως η ταξινόμηση πάνω στις τιμές κάποιων χαρακτηριστικών είναι η ίδια ανεξάρτητα από τις σταθερές τιμές των εναπομεινάντων χαρακτηριστικών. Στα CP-Nets τα σύνολα χαρακτηριστικών είναι ένας κόμβος, οι γονείς του, και όλοι οι υπόλοιποι κόμβοι, και η ερμηνεία ceteris paribus αναγκάζει το χαρακτηριστικό που σχετίζεται με ένα κόμβο να είναι κατά προτίμηση ανεξάρτητο από όλα τα υπόλοιπα χαρακτηριστικά που παίρνουν την τιμή των χαρακτηριστικών των γονέων.

Η απεικόνιση των προτιμήσεων με χρήση γράφου βοηθά στην επιβεβαίωση ότι όλες οι δηλώσεις είναι συνεπής. Μια δήλωση είναι ασυνεπής εάν προκαλεί ένα κύκλο αυστηρών προτιμήσεων στην ταξινόμηση (ο $\succ$ ο για κάποιες περιπτώσεις). Στην περίπτωση των CP-Nets γνωρίζουμε ότι εάν το CP-Net είναι άκυκλο, τότε η δήλωση προτιμήσεων που αναπαριστά είναι συνεπής. Επίσης όταν το δίκτυο είναι άκυκλο, μπορούν να ταξινομηθούν

τοπολογικά οι κόμβοι του. Ακόμη βοηθά στην εύρεση μιας κατά προτίμηση συνεπούς ολικής ταξινόμησης ενός δοθέντος συνόλου εκβάσεων.

2.3.2 Σύνθεση προτιμήσεων

Μέχρι τώρα έχουμε δει δύο γενικές προσεγγίσεις για καθορισμό προτιμήσεων. Την πιο ποσοτικοποιημένη, βασισμένη σε συνάρτηση απόδοσης τιμών προσέγγιση, που είναι πολύ απλή η ερμηνεία της και υποστηρίζει αποδοτική σύγκριση και ταξινόμηση, αλλά βασίζεται σε μια γλώσσα εισόδου που λίγοι χρήστες θα βρουν φυσική. Την πιο ποιοτικοποιημένη γλώσσα που προσπαθεί να παρέχει στους χρήστες διαισθητικές εκφράσεις εισόδου, που όμως δεν είναι ξεκάθαρο πως αποδίδεται η ερμηνεία τους, οι συγκρίσεις μπορεί να είναι δύσκολες, και με βάση την κλάση των δηλώσεων μπορεί να είναι δύσκολη και η ταξινόμηση.
Η χρήση διαφορετικών τύπων δηλώσεων εισόδου είναι επιθυμητή γιατί παρέχει ευελιξία στους χρήστες αλλά η ερμηνεία και η αλληλεπίδραση μεταξύ των ετερογενών δηλώσεων δεν είναι εύκολο να καθοριστούν. Ακόμη και η ανάπτυξη αποδοτικών αλγορίθμων απάντησης σε ερωτήματα για τέτοιες εκφράσεις γίνεται αρκετά προβληματική.
Οι τεχνικές σύνθεσης προτιμήσεων προσπαθούν να αντιστοιχήσουν διαφορετικές δηλώσεις σε μια μονή, κατανοητή αναπαράσταση, αυτή των συμπαγών συναρτήσεων ανάθεσης τιμών. Αυτή η αντιστοιχία μεταφράζει ποιοτικοποιημένες δηλώσεις σε περιορισμούς στο χώρο των πιθανών συναρτήσεων απόδοσης τιμών. Τυπικά αυτοί οι περιορισμοί είναι συνεπείς με διάφορες συναρτήσεις απόδοσης τιμών, και με βάση την μέθοδο, ακολούθως απαντιούνται ερωτήματα με χρήση μιας από τις συνεπείς συναρτήσεις απόδοσης τιμών.

Κεφάλαιο 3

CP-Nets(Με βάση το [7])

3.1 Εισαγωγή

Σε πολλούς τομείς των αυτοματοποιημένων συστημάτων λήψης αποφάσεων, χρειαζόμαστε πρόσβαση στις πληροφορίες-προτιμήσεις των χρηστών με ένα ποιοτικοποιημένο (qualitative) τρόπο. Ένας τέτοιος γραφικός τρόπος αναπαράστασης προτιμήσεων είναι τα CP-Nets που αντικατοπτρίζουν εξαρτήσεις και ανεξαρτησία δηλώσεων προτιμήσεων με συνθήκες (conditional), με μετάφραση με βάση ceteris paribus (all else being equal) σημασιολογία.

Η δομή αυτών των δικτύων χρησιμεύει για τον καθορισμό του εάν και ποια έκβαση είναι προτιμότερη από την άλλη, για την ταξινόμηση ενός συνόλου εκβάσεων με βάση την σχέση

προτιμήσεων, και για την κατασκευή της καλύτερης έκβασης με βάση τα δεδομένα, όλα αυτά με ένα συμπαγές, διαισθητικό και δομημένο τρόπο.

Μέσα σε αυτή την ενότητα η περιγραφή των CP-Nets επικεντρώνεται γύρω από δύο καίρια ερωτήματα. Το πρώτο είναι : πως διεξάγουμε σύγκριση προτιμήσεων μεταξύ εκβάσεων, και το δεύτερο είναι πως εξάγουμε την βέλτιστη έκβαση δεδομένης μιας μερικής(partial) ανάθεσης στα χαρακτηριστικά του δεδομένου προβλήματος.

Ιδεατά ένας μηχανισμός αναπαράστασης προτιμήσεων των χρηστών, πρέπει να μπορεί να λάβει δηλώσεις που είναι φυσικές ως προς τον χρήστη, να είναι σχετικά συμπαγές, και να υποστηρίζει αποδοτική εξαγωγή συμπερασμάτων.

Η σημασιολογία ceteris paribus, απαιτεί από τον χρήστη να καθορίσει για κάθε συγκεκριμένο χαρακτηριστικό Α που τον αφορά, ποια άλλα χαρακτηριστικά μπορεί να έχουν επίδραση πάνω στις προτιμήσεις του όσον αφορά το χαρακτηριστικό Α. Αυτά τα χαρακτηριστικά ονομάζονται «γονείς του Α», και για κάθε στιγμιότυπό τους, ο χρήστης καλείται να καθορίσει την διάταξη των προτιμήσεων του πάνω στις τιμές του Α, με βάση κάποιες συνθήκες που θα επικρατούν στους γονείς του. Αυτό ουσιαστικά σημαίνει ότι, εάν και στις δύο εκβάσεις στους γονείς ισχύουν κάποιες τιμές β1 και γ1 ίδιες (όπως το όρισε ο χρήστης), τότε ο χρήστης προτιμά την έκβαση που έχει την τιμή α1 στο χαρακτηριστικό Α από την άλλη που έχει την τιμή α2 στο ίδιο χαρακτηριστικό (επίσης το ορίζει ο χρήστης) . Για κάποιες άλλες τιμές των γονέων τότε η προτίμηση ανάμεσα στις τιμές που παίρνει το Α, αλλάζει.

Ας υποθέσουμε ότι το σύνολο των εκβάσεων αποτελείται από ένα σύνολο πιθανών σπιτιών για αγορά. Κάθε σπίτι χαρακτηρίζεται από τα χαρακτηριστικά γνωρίσματα ΤΙΜΗ, ΜΕΓΕΘΟΣ, ΠΕΡΙΟΧΗ. Ο χρήστης ορίζεται ότι το χαρακτηριστικό ΤΙΜΗ επηρεάζεται από τα χαρακτηριστικά ΜΕΓΕΘΟΣ και ΠΕΡΙΟΧΗ, άρα αυτά τα δύο είναι οι γονείς του. Ακολούθως ορίζει ότι, σε περίπτωση που τα χαρακτηριστικά ΜΕΓΕΘΟΣ και ΠΕΡΙΟΧΗ παίρνουν ακριβώς τις ίδιες τιμές για δύο πιθανά σπίτια, τότε προτιμάει το σπίτι όπου το χαρακτηριστικό ΤΙΜΗ έχει την τιμή ΧΑΜΗΛΗ, από το σπίτι για το οποίο το χαρακτηριστικό ΤΙΜΗ έχει την τιμή ΥΨΗΛΗ. Αυτή η προτίμηση του όμως ισχύει σε περίπτωση που τα δύο σπίτια έχουν ακριβώς τις ίδιες ίσες τιμές στα άλλα δύο τους χαρακτηριστικά.

Η αναπαράσταση μέσω CP-Nets οδηγεί σε μια εξισορρόπηση της απόκτησης πληροφοριών σχετικά με τις προτιμήσεις, αφού ναι μεν επιτρέπει ελαστικές εκφράσεις προτιμήσεων, αλλά δε συσχετίζεται με μια συγκεκριμένη δομή προτιμήσεων και ζητούμενων στοιχείων. Ακόμη τα δίκτυα αυτά μπορούν να αναπαραστήσουν και προτιμήσεις που βασίζονται σε συνθήκες, σε αντίθεση με πολλούς άλλους μηχανισμούς αναπαράστασης.

3.2 Ορισμός των CP-Networks

Το «κτίσιμο» ενός CP-Net εν συντομία γίνεται ως ακολούθως:

Ζητείται από τον χρήστη να δώσει για κάθε μεταβλητή-χαρακτηριστικό Xi, το σύνολο των γονέων της, Pa(Xi), που όπως αναφέραμε και πιο πάνω αντιπροσωπεύει τα χαρακτηριστικά που μπορούν να επηρεάσουν τις προτιμήσεις του χρήστη πάνω σε διαφορετικές τιμές της Xi. Ακολούθως για κάθε δεδομένη ανάθεση τιμών στους γονείς της Xi, ο χρήστης θα πρέπει να καθορίσει μια ταξινόμηση προτιμήσεων για τις τιμές του χαρακτηριστικού Xi, δεδομένου ότι τα υπόλοιπα χαρακτηριστικά έχουν τις ίδιες τιμές για όλες τις τιμές του Xi για την συγκεκριμένη ταξινόμηση.

Με τις πιο πάνω πληροφορίες ως δεδομένα πλέον, κτίζεται ένας σχολιασμένος (annotated) κατευθυνόμενος γράφος. Οι κόμβοι αναπαριστούν τις μεταβλητές/χαρακτηριστικά του προβλήματος, και κάθε κόμβος Xi, έχει τους γονείς του ως άμεσους προγόνους. Ο κάθε κόμβος Xi σχολιάζεται (annotated) με ένα πίνακα προτιμήσεων με συνθήκες (conditional preference table-CPT), ο οποίος περιγράφει τις προτιμήσεις του χρήστη πάνω στις τιμές του εκάστοτε χαρακτηριστικού Xi, δεδομένου κάθε συνδυασμού των τιμών των γονέων.

Έτσι για κάθε δύο τιμές x και x’ που μπορεί να πάρει το χαρακτηριστικό Xi θα υπάρχει είτε μια σχέση όπου προτιμάται το x από το x’, είτε μια σχέση όπου προτιμάται το x’ από το x. Για σκοπούς απλοποίησης, η «αδιαφορία» (indifference) μεταξύ δύο τιμών δεν περιλαμβάνεται σε αυτό το κείμενο.

Ένας αυστηρός ορισμός των CP-Nets είναι ο πιο κάτω:

Ένα CP-Net πάνω σε ένα σύνολο μεταβλητών V={X1,...,Xn}, είναι ένας κατευθυνόμενος γράφος G πάνω στις X1,...,Xn, του οποίου οι κόμβοι σχολιάζονται με πίνακες προτιμήσεων

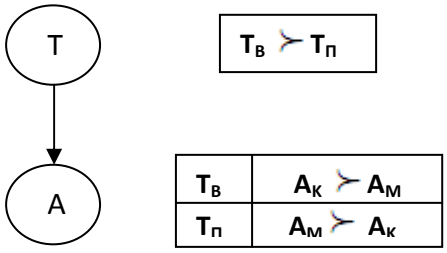
με συνθήκες CPT(Xi) για κάθε μεταβλητή-χαρακτηριστικό XiϵV. Κάθε πίνακας προτιμήσεων με συνθήκες CPT(Xi) αντιστοιχεί μια ολική διάταξη προτιμήσεων $\succ_u^i$, με κάθε στιγμιότυπο ανάθεσης τιμών u των γονέων της Xi, Pa(Xi)=U.

Ακολουθούν κάποια παραδείγματα που αποσκοπούν στην καλύτερη κατανόηση κάποιων της σημασιολογία και των συνεπειών των CP-Nets. Στα ακόλουθα παραδείγματα για λιγότερη πολυπλοκότητα στην παρουσίαση τους, τα πεδία τιμών κάθε μεταβλητής είναι δυαδικά.

Παράδειγμα 3.2.1:

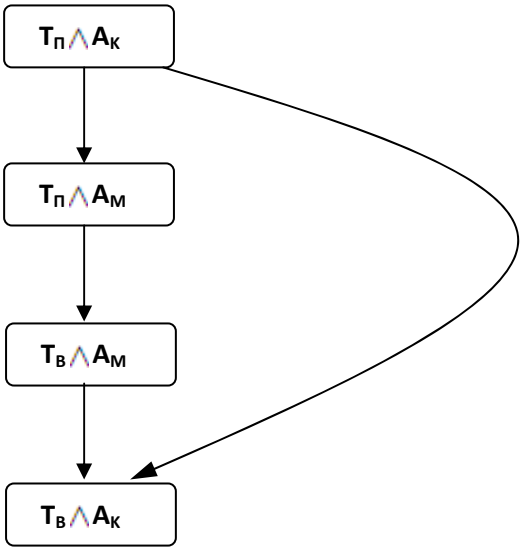
Ξαναγυρνάμε στο παράδειγμα της επιλογής ανάμεσα από ένα σύνολο υποψήφιων σπιτιών προς αγορά. Αυτή τη φορά τα χαρακτηριστικά που περιγράφουν ένα σπίτι είναι: ΑΡΧΙΤΕΚΤΟΝΙΚΗ και ΤΟΠΟΘΕΣΙΑ. Ας υποθέσουμε ότι ο χρήστης έχει αυστηρή προτίμηση στα σπίτια που βρίσκονται στο βουνό (ΤΟΠΟΘΕΣΙΑ=ΒΟΥΝΟ), από τα σπίτια που βρίσκονται στην πόλη (ΤΟΠΟΘΕΣΙΑ=ΠΟΛΗ). Οι προτιμήσεις του πάνω στην αρχιτεκτονική με πιθανές τιμές : ΜΟΝΤΕΡΝΑ, ΚΛΑΣΙΚΗ εξαρτάται από την τιμή που παίρνει το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ. Δηλαδή, εάν το σπίτι βρίσκεται στο βουνό τότε προτιμάει την κλασική αρχιτεκτονική. Αντιθέτως εάν το σπίτι βρίσκεται στην πόλη τότε προτιμάει την μοντέρνα αρχιτεκτονική.

Στο πιο κάτω σχεδιάγραμμα φαίνεται το CP-Net που αναπαριστά αυτό το απλό παράδειγμα. Για ευκολία αναπαράστασης, συμβολίζω το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ με Τ, και το χαρακτηριστικό ΑΡΧΙΤΕΚΤΟΝΙΚΗ με Α.



Ακολούθως βλέπουμε τον γράφο προτιμήσεων πάνω στις εκβάσεις που δημιουργείται από το πιο πάνω CP-Net. Κάθε ακμή σε αυτό τον γράφο η οποία κατευθύνεται από την έκβαση α

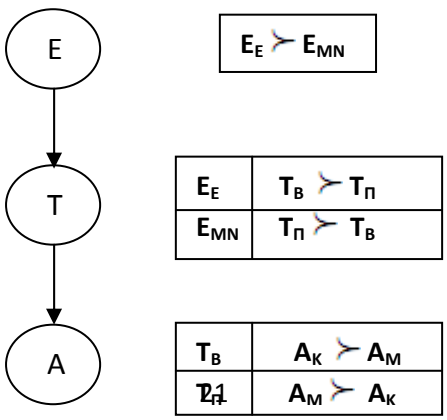
προς την έκβαση β, υποδεικνύει ότι μια προτίμηση για το β σε σύγκριση με το α, μπορεί να καθοριστεί άμεσα από ένα από τους πίνακες προτιμήσεων του CP-net.



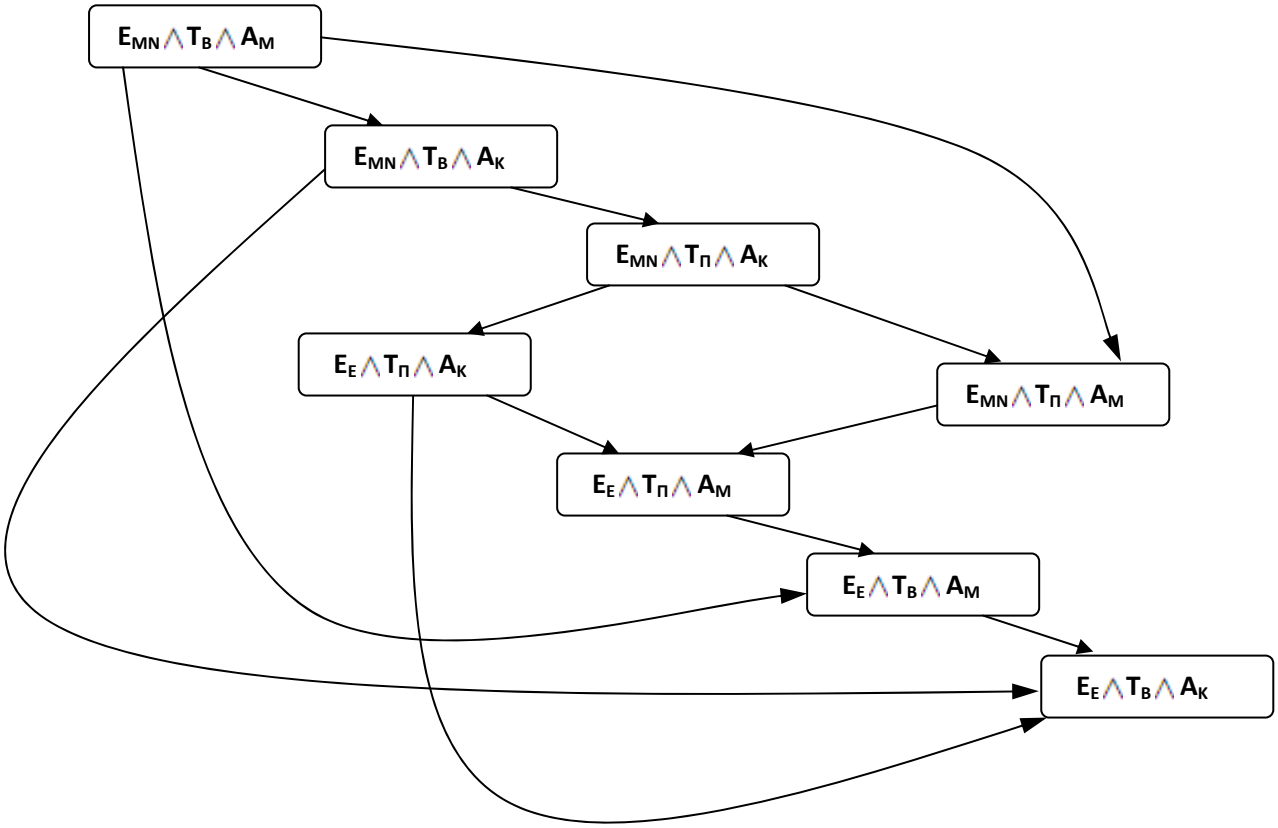
Παράδειγμα 3.2.2:

Συνεχίζοντας με το προηγούμενο παράδειγμα, θα προσθέσουμε ένα επιπλέον χαρακτηριστικό, το ΕΙΔΟΣ του σπιτιού. Υποθέτουμε ότι ο χρήστης προτιμάει να αγοράσει ένα σπίτι που θα το χρησιμοποιεί ως «εξοχικό» παρά ένα σπίτι που θα το χρησιμοποιεί ως μόνιμη κατοικία. Στην περίπτωση όμως που θα αγοράσει τελικά ένα σπίτι που θα το χρησιμοποιεί ως μόνιμη κατοικία, τότε προτιμάει αυστηρά το σπίτι αυτό να βρίσκεται στην πόλη. Από την άλλη, αν τελικά θα αγοράσει ένα σπίτι που θα το χρησιμοποιεί ως εξοχικό προτιμάει η τοποθεσία του να είναι στο βουνό.

Στο πιο κάτω σχεδιάγραμμα φαίνεται το CP-Net που αναπαριστά αυτό το παράδειγμα. Για ευκολία αναπαράστασης, συμβολίζω και πάλι το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ με Τ, το χαρακτηριστικό ΑΡΧΙΤΕΚΤΟΝΙΚΗ με Α και το χαρακτηριστικό ΕΙΔΟΣ με Ε.



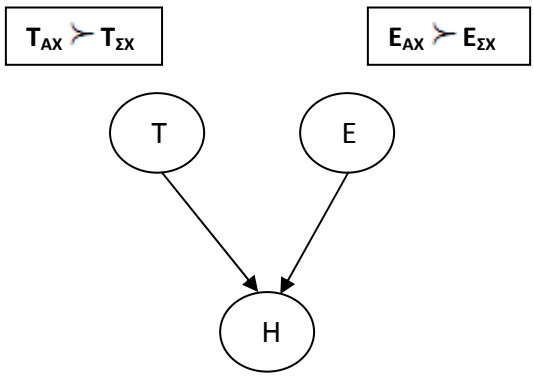
Ακολούθως βλέπουμε τον γράφο προτιμήσεων πάνω στις εκβάσεις που προκαλείται από το πιο πάνω CP-Net.



Παράδειγμα 3.2.3:

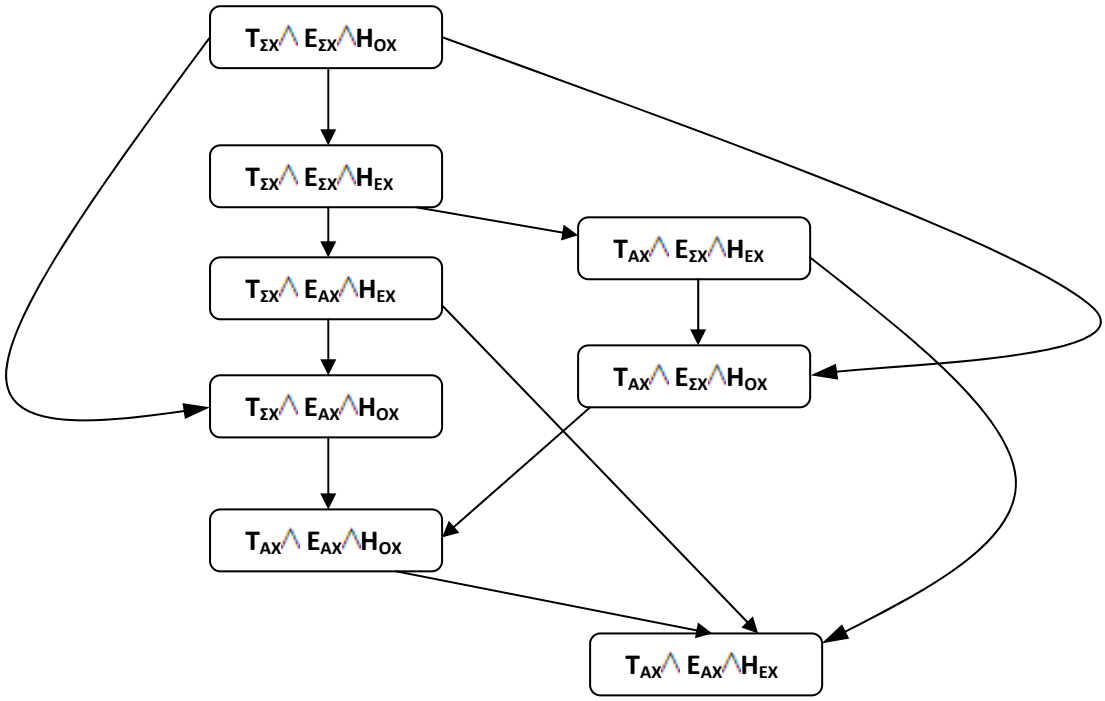
Ας υποθέσουμε ότι έχουμε ένα σύνολο εκβάσεων, που αποτελείται από την εσωτερική διακόσμηση ενός σπιτιού. Τα χαρακτηριστικά σε αυτή την περίπτωση είναι ΤΟΙΧΟΙ , ΕΠΙΠΛΑ και ΗΛΕΚΤΡΙΚΕΣ_ΣΥΣΚΕΥΕΣ. Ο χρήστης σε αυτό το παράδειγμα, προτιμά να έχει στο σπίτι του τοίχους και έπιπλα σε ανοικτό χρώμα παρά σε σκούρο χρώμα. Η επιλογή του για τις οικιακές συσκευές εξαρτάται από τον συνδυασμό των χρωμάτων στους τοίχους και τα έπιπλα. Εάν έχουν και τα δύο είτε σκούρο χρώμα είτε ανοικτό χρώμα, τότε στις ηλεκτρικές συσκευές προτιμάει το έντονο χρώμα ώστε να σπάζει η μονοτονία των υπόλοιπων αποχρώσεων. Εάν το ένα από τα δύο είναι σε ανοικτό χρώμα και το άλλο σε σκούρο χρώμα, τότε προτιμάει οι ηλεκτρικές του συσκευές να είναι σε ουδέτερο χρώμα ώστε να μην δημιουργείται δυσαρμονία.

Στο πιο κάτω σχεδιάγραμμα φαίνεται το CP-Net που αναπαριστά αυτό το παράδειγμα. Για ευκολία αναπαράστασης, συμβολίζω το χαρακτηριστικό ΤΟΙΧΟΙ με Τ, το χαρακτηριστικό ΕΠΙΠΛΑ με Ε και το χαρακτηριστικό ΗΛΕΚΤΡΙΚΕΣ_ΣΥΣΚΕΥΕΣ με Η.



$T_{AX} \wedge E_{AX}$	$H_{EX} \succ H_{OX}$
$T_{\Sigma X} \wedge E_{AX}$	$H_{OX} \succ H_{EX}$
$T_{AX} \wedge E_{\Sigma X}$	$H_{OX} \succ H_{EX}$
$T_{\Sigma X} \wedge E_{\Sigma X}$	$H_{\Sigma X} \succ H_{\Omega X}$

Ακολούθως βλέπουμε τον γράφο προτιμήσεων πάνω στις εκβάσεις που προκαλείται από το πιο πάνω CP-Net.



3.3 Σημασιολογία των CP-Networks

Έστω N είναι ένα CP-Net, πάνω στις μεταβλητές V, $X_i \epsilon V$ είναι κάποια μεταβλητή από το σύνολο V, και U c V είναι οι γονείς της X_i στο N. Επίσης Y=V-(U ∪ $\{X_i\}$). Το $\succ^i_u$ είναι η ταξινόμηση στις τιμές του $Dom(X_i)$, όπως υποδεικνύεται από τον πίνακα $CPT(X_i)$ για κάθε στιγμιότυπο u ϵ Asst(U) (ανάθεση των γονέων της). Τέλος έστω ότι $\succ$ είναι μια ταξινόμηση προτιμήσεων πάνω στο Asst(V) (τις αναθέσεις σε ολόκληρο το σύνολο μεταβλητών-χαρακτηριστικών άρα στις εκβάσεις).

1) Μια ταξινόμηση προτιμήσεων $\succ$ ικανοποιεί την ταξινόμηση $\succ^i_u$ αν και μόνο αν, για κάθε y ϵ Asst(Y) και για κάθε x,x' ϵ $Dom(X_i)$, ισχύει yxu $\succ$ yx'u όποτε ισχύει x $\succ^i_u$ x'.
2) Μια ταξινόμηση προτιμήσεων $\succ$ ικανοποιεί τον πίνακα $CPT(X_i)$ αν και μόνο αν ικανοποιεί τις ταξινομήσεις $\succ^i_u$ για κάθε u ϵ Asst(U).
3) Μια ταξινόμηση προτιμήσεων $\succ$ ικανοποιεί το CP-Net N αν και μόνο αν ικανοποιεί τους πίνακες $CPT(X_i)$ για κάθε μεταβλητή του X_i του προβλήματος.

Ένα CP-Net είναι ικανοποιήσιμο αν και μόνο αν υπάρχει μια ταξινόμηση προτιμήσεων $\succ$ η οποία το ικανοποιεί.

Άρα ουσιαστικά ένα CP-Net N, ικανοποιείται από την ταξινόμηση $\succ$ αν και μόνο αν η $\succ$ ικανοποιεί όλες τις προτιμήσεις με συνθήκες, που εκφράζονται στους πίνακες (CPTs), με βάση την σημασιολογία ceteris paribus.

Θεώρημα: Κάθε ακυκλικό (acyclic) CP-Net είναι ικανοποιήσιμο.

Παράδειγμα 3.3.1:

Στο παράδειγμα 3.2.1, οι πληροφορίες που αναπαριστά το CP-Net δίνουν μια (και μοναδική σε αυτή την περίπτωση) ολική ταξινόμηση (totally order) στις εκβάσεις, η οποία ικανοποιεί το CP-Net και είναι η εξής:

$$T_B \wedge A_K \succ T_B \wedge A_M \succ T_\Pi \wedge A_M \succ T_\Pi \wedge A_K$$

Παράδειγμα 3.3.2:

Από το CP-Net του παραδείγματος 3.2.2, μπορούμε να πάρουμε δύο ταξινομήσεις, οι οποίες ικανοποιούν το CP-Net αυτό, και είναι οι ακόλουθες:

1) $E_E \wedge T_B \wedge A_K \succ E_E \wedge T_B \wedge A_M \succ E_E \wedge T_\Pi \wedge A_M \succ E_{MN} \wedge T_\Pi \wedge A_M \succ E_E \wedge T_\Pi \wedge A_K \succ E_{MN} \wedge T_\Pi \wedge A_K \succ E_{MN} \wedge T_B \wedge A_K \succ E_{MN} \wedge T_B \wedge A_M$

2) $E_E \wedge T_B \wedge A_K \succ E_E \wedge T_B \wedge A_M \succ E_E \wedge T_\Pi \wedge A_M \succ E_E \wedge T_\Pi \wedge A_K \succ E_{MN} \wedge T_\Pi \wedge A_M \succ E_{MN} \wedge T_\Pi \wedge A_K \succ E_{MN} \wedge T_B \wedge A_K \succ E_{MN} \wedge T_B \wedge A_M$

Έστω τώρα ότι N είναι ένα CP-Net με μεταβλητές V, και o,o'ϵ Astt(V) είναι δύο οποιεσδήποτε εκβάσεις. Το N συνεπάγεται ότι o $\succ$ o', αν και μόνο αν το o $\succ$ o' ισχύει σε κάθε ταξινόμηση προτιμήσεων που ικανοποιεί το N. Αυτή η συνεπαγωγή γράφεται ως N $\models$ o $\succ$ o'.

Η συνεπαγωγή προτιμήσεων είναι *μεταβατική*. Δηλαδή εάν ισχύει N $\models$ o $\succ$ o' και ότι N $\models$ o' $\succ$ o'' , τότε θα ισχύει και ότι N = o $\succ$ o''.

Παράδειγμα 3.3.3:

Παρατηρώντας και πάλι τα διαγράμματα του παραδείγματος 3.2.2, ονομάζουμε : o την έκβαση $E_E \wedge T_B \wedge A_M$, o' την έκβαση $E_E \wedge T_\Pi \wedge A_M$ και o'' την έκβαση $E_E \wedge T_\Pi \wedge A_K$. Οι εκβάσεις o και o' διαφέρουν μεταξύ τους στο μόνο στην μεταβλητή T. Με βάση τον γονέα της μεταβλητής T που είναι η μεταβλητή E, βλέπουμε πως η ανάθεση T_B είναι προτιμότερη από την ανάθεση T_Π όταν όλες οι υπόλοιπες αναθέσεις είναι οι ίδιες. Άρα έχουμε ότι N $\models$ o $\succ$ o'.
Ακολούθως βλέπουμε ότι τα o' και o'' διαφέρουν μεταξύ τους στην μεταβλητή A. Με βάση τις μεταβλητές E και T που έχουν τις ίδιες αναθέσεις, συμπεραίνουμε ότι η ανάθεση A_M είναι προτιμότερη από την ανάθεση A_K και άρα έχουμε N $\models$ o' $\succ$ o''. Το o $\succ$ o'' δεν μπορεί να εξαχθεί απευθείας από τους πίνακες CPTs του N. Από την μεταβατική ιδιότητα όμως της συνεπαγωγής προτιμήσεων η σχέση αυτή μπορεί να προκύψει μέσω της μεταβατικής κλειστότητας (closure) των δύο απευθείας σχέσεων.

Σε ένα CP-Net μπορούμε να παρατηρήσουμε κάθε έκβαση ποιες προτιμήσεις με συνθήκες παραβιάζει.

Παράδειγμα 3.3.4:

Παρατηρώντας τώρα το CP-Net του παραδείγματος 2.2.1, μπορούμε να δούμε ότι η έκβαση $T_B \wedge A_K$ δεν παραβιάζει κανένα περιορισμό προτιμήσεων, η έκβαση $T_B \wedge A_M$ παραβιάζει την προτίμηση στο A, η έκβαση $T_\Pi \wedge A_M$ παραβιάζει την προτίμηση στο T, και η έκβαση $T_\Pi \wedge A_K$ παραβιάζει τις προτιμήσεις και στις δύο μεταβλητές. Η σημασιολογία ceteris paribus εξασφαλίζει ότι το να παραβιάζεται η προτίμηση στο T είναι πιο χειρότερο από το να παραβιάζεται η προτίμηση στο A. Δηλαδή οι προτιμήσεις-γονείς έχουν υψηλότερη

προτεραιότητα από τα παιδιά τους. Βλέπουμε και στο παράδειγμα ότι η έκβαση $T_B \wedge A_M$ είναι προτιμότερη από την έκβαση $T_\Pi \wedge A_M$.

3.4 Αδιαφορία (Indifference)

Πιο πάνω ήδη αναφέρεται ότι για κάθε μεταβλητή X_i με γονείς U, και για κάθε u ϵ Asst(U) ανάθεση, υποθέτουμε ότι η $\succ_u^i$ είναι μια ολική ταξινόμηση στο $Dom(X_i)$. Ο γενικός ορισμός των CP-Nets επιτρέπει μια αυθαίρετη ολική ταξινόμηση (arbitrary total preorder) , την $\succeq_u^i$ πάνω στο $Dom(X_i)$, η οποία επιτρέπει στον χρήστη να εκφράσει «αδιαφορία» μεταξύ δύο πιθανών τιμών του χαρακτηριστικού X_i, δεδομένου του u. Αν οι τιμές είναι οι x και x', τότε αυτό εκφράζεται ως x ~ x' στον πίνακα $CPT(X_i)$.

Παράδειγμα 3.4.1:

Ας υποθέσουμε ότι έχουμε τους 2 ακόλουθους πίνακες CPTs ενός CP-Net με μεταβλητές ΤΟΠΟΘΕΣΙΑ (Τ) με τιμές ΒΟΥΝΟ και ΠΟΛΗ και ΑΡΧΙΤΕΚΤΟΝΙΚΗ (Α) με τιμές ΚΛΑΣΙΚΗ και ΜΟΝΤΕΡΝΑ, και επίσης υποθέτουμε ότι το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ είναι γονέας του χαρακτηριστικού ΑΡΧΙΤΕΚΤΟΝΙΚΗ.

$T_B \sim T_\Pi$

T_B	$A_K \succ A_M$
T_Π	$A_M \succ A_K$

Οι πιο πάνω πίνακες δείχνουν ότι ενώ ο χρήστης δεν έχει κάποια προτίμηση ανάμεσα στα T_B και T_Π (αδιαφορία), όταν ισχύει το T_B προτιμά το A_K από το A_M, και όταν ισχύει το T_Π προτιμά το A_M από το A_K.
Από τους πίνακες εξάγουμε τις ακόλουθες προτιμήσεις πάνω στις εκβάσεις:

$$T_B \wedge A_K \succ T_B \wedge A_M \succeq T_\Pi \wedge A_M \succ T_\Pi \wedge A_K \succeq T_B \wedge A_K$$

Αυτές οι δηλώσεις δεν είναι συνεπής με καμία ταξινόμηση προτιμήσεων, και ως ακολούθως το CP-Net δεν είναι ικανοποιήσιμο.

Υπάρχει λοιπόν μια ιδιαιτερότητα με την χρήση της αδιαφορίας στα CP-Nets. Ο χρήστης δεν πρέπει να εκφράσει αδιαφορία μεταξύ δύο τιμών μιας μεταβλητής X, και ταυτόχρονα να εκφράσει προτίμηση με συνθήκες για κάποιο παιδί Y της X, που να εξαρτάται από εκείνες τις τιμές για τις οποίες ο χρήστης εξέφρασε την αδιαφορία. Με το να είναι η X γονέας της Y, σημαίνει ότι η ικανοποίηση της προτίμησης πάνω στην X είναι πιο σημαντική από την ικανοποίηση της προτίμησης πάνω στην Y, κάτι όμως που σε περίπτωση αδιαφορίας στις τιμές της X, δεν έχει νόημα.

Υπάρχει όμως τρόπος η αδιαφορία να εκφραστεί ορθά και με ασφάλεια. Ας υποθέσουμε ότι A είναι μια μεταβλητή του CP-Net N, με γονείς U, η B είναι μια μεταβλητή παιδί της A και C είναι οι υπόλοιποι γονείς της B εκτός από την A. Υποθέτουμε επίσης ότι για κάποια ανάθεση u ϵ Asst(U), και για α,α’ ϵ Dom(A), έχουμε αδιαφορία α^{uu} α’. Εάν οι τοπικές ταξινομήσεις στον πίνακα CPT(B) για κάθε στιγμιότυπο του C είναι ίδιες ανεξάρτητα από το αν ισχύει το α ή το α’, τότε το δίκτυο είναι ικανοποιήσιμο. Δηλαδή εάν ο χρήστης είναι αδιάφορος μεταξύ των τιμών της A, α και α’ τότε οι προτιμήσεις του πάνω στις τιμές των παιδιών της A πρέπει να είναι ανεξάρτητες από το αν ισχύει η τιμή α ή η τιμή α’.

Στο υπόλοιπο κείμενο υποθέτουμε ότι δεν υπάρχουν εκφράσεις αδιαφορίας στα CP-Nets που μελετούνται.

3.5 Βελτιστοποίηση εκβάσεων(outcome optimization)

Η εύρεση της βέλτιστης έκβασης μέσω ενός CP-Net είναι πολύ εύκολη υπόθεση. Απλά ξεκινώντας από την κορυφή του CP-Net (από μεταβλητή που δεν έχει γονείς) και κατεβαίνοντας προς τα κάτω, γίνεται ανάθεση σε κάθε μεταβλητή, της προτιμότερης τιμής, με βάση τις αναθέσεις στους γονείς της και τους πίνακες CPT της μεταβλητής. Το κάθε CP-Net μπορεί να καθορίσει μια μοναδική βέλτιστη έκβαση.

Ας υποθέσουμε ένα CP-Net N, πάνω στις μεταβλητές V και Z $\subseteq$ V. Υποθέτουμε επίσης μια μερική ανάθεση z ϵ Asst(Z), και στόχος είναι ο καθορισμός της μοναδικής ανάθεσης o ϵ Comp(z) (συμπλήρωμα του z), με βάση την ήδη δεδομένη ανάθεση z, τέτοια ώστε N $\models$ o $\succ$ o’ για κάθε άλλη ανάθεση o’ ϵ Comp(z) - {o}.

Τέτοια ερωτήματα βελτιστοποίησης εκβάσεων (εύρεση βέλτιστης έκβασης), μπορούν να απαντηθούν με την ακόλουθη διαδικασία που ονομάζεται “forward sweep”. Υποθέτουμε ότι $X_1,...X_n$ είναι κάθε μια τοπολογική(topological) ταξινόμηση των μεταβλητών του CP-Net. Θέτουμε Z=z, και αρχικοποιούμε κάθε $X_i \notin Z$ με την βέλτιστη τιμή που μπορεί να πάρει με βάση τις αναθέσεις που παίρνουν οι γονείς του.

Η πιο πάνω μέθοδος κατασκευάζει την πιο επιθυμητή έκβαση από όλα τα πιθανά συμπληρώματα του z (Comp(z)).

3.6 Σύγκριση εκβάσεων(comparing outcomes)

Ένα άλλο πολύ βασικό ερώτημα το οποίο πρέπει να μπορεί να απαντηθεί μέσω ενός τέτοιου μοντέλου αναπαράστασης είναι η σύγκριση προτιμήσεων μεταξύ εκβάσεων. Δυο εκβάσεις λοιπόν α και α’ μπορούν να συγκριθούν μέσω τριών πιθανών σχέσεων: $N \models \alpha \succ \alpha'$ ή $N \models \alpha' \succ \alpha$ ή $N \not\models \alpha \succ \alpha'$ και $N \not\models \alpha' \succ \alpha$. Η τρίτη σχέση σημαίνει ότι στο CP-Net δεν δίνονται αρκετές πληροφορίες για να αποδειχθεί ότι κάποια από τις δύο εκβάσεις είναι προτιμότερη από την άλλη.

Υπάρχουν δύο τρόποι με τους οποίους μπορούμε να συγκρίνουμε δύο εκβάσεις με την χρήση ενός CP-Net και είναι οι ακόλουθοι:

1. Ερωτήματα κυριαρχίας (Dominance queries) : Δεδομένου ενός CP-Net και ενός ζευγαριού εκβάσεων α και α’, το ερώτημα είναι εάν ισχύει $N \models \alpha \succ \alpha'$. Εάν αυτή η σχέση ισχύει τότε η έκβαση α είναι προτιμότερη από την έκβαση α’, και λέμε ότι η α κυριαρχεί της α’ σε σχέση με το CP-Net που μελετούμε.
2. Ερωτήματα ταξινόμησης (Ordering queries) : Δεδομένου ενός CP-Net και ενός ζευγαριού εκβάσεων α και α’, το ερώτημα είναι εάν ισχύει $N \not\models \alpha \succ \alpha'$. Εάν αυτή η σχέση ισχύει τότε υπάρχει μια ταξινόμηση προτιμήσεων στο CP-Net στην οποία το α είναι προτιμότερο από το α’. Σε αυτή την περίπτωση λέμε ότι το α είναι συνεπώς (consistenly) ταξινομημένο πριν από το α’ για το συγκεκριμένο CP-Net.

Τα ερωτήματα ταξινόμησης είναι πιο αδύναμα(weaker) από τα ερωτήματα κυριαρχίας. Εάν ισχύει $N \models \alpha \succ \alpha'$ τότε ισχύει και $N \not\models \alpha' \succ \alpha$. Αλλά μπορεί να υπάρξει και περίπτωση όπου ισχύει και $N \not\models \alpha' \succ \alpha$ και $N \not\models \alpha \succ \alpha'$.

3.6.1 Ερωτήματα ταξινόμησης

Τέτοια ερωτήματα μπορούν να απαντηθούν σε γραμμικό χρόνο με βάση τον αριθμό των μεταβλητών. Θα μπορούσε να κατασκευαστεί μια όχι αυστηρά αυξανόμενη ταξινόμηση των εκβάσεων, συνεπής με το CP-Net, με χρήση μόνο ερωτημάτων ταξινόμησης.

Ας υποθέσουμε ότι έχουμε ένα CP-Net N, και δύο εκβάσεις α και α'. Εάν υπάρχει μια μεταβλητή X στο N, τέτοια ώστε:

1. Η α και η α' αναθέτουν τις ίδιες τιμές σε όλους τους προγόνους της X στο N, και
2. δεδομένης της ανάθεσης αυτής των α και α' (που είναι η ίδια) στους γονείς της X (Pa(X)), η α αναθέτει μια πιο επιθυμητή τιμή στην X από ότι της αναθέτει η α',

τότε ισχύει ότι $N \not\models \alpha' \succ \alpha$.

Το πιο πάνω πόρισμα παρέχει μια συνθήκη η οποία είναι μεν επαρκής αλλά όχι αναγκαία για να αποδειχθεί ότι ισχύει το ερώτημα ταξινόμησης $N \not\models \alpha' \succ \alpha$.

Παράδειγμα 3.6.1.1:

Ας υποθέσουμε ότι έχουμε $\alpha=E_E \wedge A_K \wedge T_B$ και $\alpha'=E_{MN} \wedge A_K \wedge T_\Pi$. Με βάση το CP-Net κα την σημασιολογία ceteris paribus αυτές οι δύο εκβάσεις δεν είναι συγκρίσιμες διότι διαφέρουν σε δύο χαρακτηριστικά, άρα καμία δεν μπορεί να αποδειχθεί ότι είναι προτιμότερη από την άλλη. Όμως η σχέση $\alpha \not\succ \alpha'$ δεν μπορεί να παραχθεί μέσω του πιο πάνω πορίσματος, διότι αν θέσουμε X=E, και η E δεν έχει γονείς, τότε βλέπουμε ότι η έκβαση α αναθέτει στην μεταβλητή E μια πιο επιθυμητή τιμή από ότι της αναθέτει η έκβαση α'.

Το πόρισμα προσφέρει ένα μεν αποδοτικό αλγόριθμο για απάντηση ερωτημάτων ταξινόμησης, αλλά ενώ είναι έγκυρος , δηλαδή εάν αποφασίσει ότι το α είναι συνεπώς ταξινομημένο πριν από το α' τότε όντως ισχύει το $N \not\models \alpha' \succ \alpha$, είναι ταυτόχρονα και ελλιπής διότι εάν δώσει αρνητική απάντηση για το αν ισχύει το ερώτημα $N \not\models \alpha' \succ \alpha$ τότε υπάρχει περίπτωση να ισχύει τελικά, που θα μπορούσε να αποδειχθεί αν το ζευγάρι των εκβάσεων δινόταν αντίθετα στα α και α'.

Ένα σημαντικό θεώρημα είναι το ότι σε ένα CP-Net N με n μεταβλητές, αν έχουμε δύο εκβάσεις α και α', τότε η πολυπλοκότητα του να καθοριστεί αν ισχύει τουλάχιστον ένα από τα ερωτήματα ταξινόμησης δηλαδή είτε το $N \not\models \alpha \succ \alpha'$ είτε το $N \not\models \alpha' \succ \alpha$ είναι O(n).

Το θεώρημα αυτό παρέχει επίσης ένα αποδοτικό αλγόριθμο που και αυτός είναι έγκυρος και ταυτόχρονα είναι «μερικώς πλήρης». Αυτό διότι θα επιστρέψει μια θετική απάντηση για τουλάχιστον ένα από τα ερωτήματα $N \not\models \alpha \succ \alpha'$ και $N \not\models \alpha' \succ \alpha$, επιτρέποντας τον καθορισμό ότι μια από τις δύο εκβάσεις είναι συνεπώς ταξινομημένη πριν από την άλλη.

Η μερική πληρότητα αυτού του αλγορίθμου είναι επαρκής ώστε να βρεθεί μια συνεπής ταξινόμηση όλων των εκβάσεων σε πολυωνυμικό χρόνο.

Με τον συμβολισμό $N \mid\!\!-_{oq} \alpha >> \alpha'$ αναπαριστάται η περίπτωση όπου ο αλγόριθμος με τα διπλά ερωτήματα ταξινόμησης απαντάει ότι το $N \not\models \alpha' \succ \alpha$ ισχύει, και ότι το $N \not\models \alpha \succ \alpha'$ δεν ισχύει. Όταν ισχύει το $N \mid\!\!-_{oq} \alpha >> \alpha'$ τότε είναι σίγουρο ότι το α είναι ταξινομημένο πιο μπροστά από το α'. Λόγω όμως της μη-πληρότητας του αλγορίθμου δεν είναι σίγουρο ότι το α' δεν είναι και αυτό ταξινομημένο πιο μπροστά από το α. Με τον συμβολισμό $N \mid\!\!-_{oq} \alpha \cong \alpha'$ αναπαριστάται η περίπτωση όπου ο αλγόριθμος επιστρέφει μια θετική απάντηση και στα δύο ερωτήματα ταξινόμησης. Λόγω της μερικής πληρότητας του αλγορίθμου, θα ισχύει μόνο ένα από τα τρία : $N \mid\!\!-_{oq} \alpha >> \alpha'$, $N -_{oq} \alpha' >> \alpha$ είτε $N -_{oq} \alpha \cong \alpha'$

Δεδομένου ενός CP-Net N σε ένα σύνολο μεταβλητών V, και ενός συνόλου εκβάσεων $\alpha_1,\ldots\alpha_m$, η ταξινόμηση αυτών των εκβάσεων με συνέπεια ως προς το N, μπορεί να γίνει με χρήση μόνο ερωτημάτων ταξινόμησης. Με ένα τέτοιο σύνολο m εκβάσεων η πολυπλοκότητα της ταξινόμησης τους, είναι $O(nm^2)$ λόγω του κόστους της σύγκρισης κάθε ζευγαριού εκβάσεων και της κατάλληλης ταξινόμησης τους.

3.6.2 Ερωτήματα Κυριαρχίας και Flipping Sequences

Σε ένα CP-Net μπορεί να γίνει χρήση πληροφοριών από τον πίνακα CPT μιας μεταβλητής X, ώστε να αλλαχθεί η τιμή της X σε μια έκβαση έτσι ώστε να παραχθεί μια πιο επιθυμητή ή μια λιγότερο επιθυμητή έκβαση. Μια ακολουθία τέτοιων αλλαγών που οδηγούν από κάθε έκβαση σε μια πιο επιθυμητή, αποδεικνύουν ότι μια έκβαση είναι προτιμότερη ή όχι από μια άλλη σε κάθε ταξινόμηση που ικανοποιεί το δίκτυο.

Παράδειγμα 3.6.2.1:

Από το CP-Net του παραδείγματος 3.2.2, ας θυμηθούμε ότι μπορούμε να πάρουμε δύο ταξινομήσεις, οι οποίες ικανοποιούν το CP-Net αυτό, και είναι οι ακόλουθες:

1) $E_E \wedge T_B \wedge A_K \succ E_E \wedge T_B \wedge A_M \succ E_E \wedge T_\Pi \wedge A_M \succ \underline{E_{MN} \wedge T_\Pi \wedge A_M} \succ \underline{E_E \wedge T_\Pi \wedge A_K} \succ E_{MN} \wedge T_\Pi \wedge A_K \succ E_{MN} \wedge T_B \wedge A_K \succ E_{MN} \wedge T_B \wedge A_M$

2) $E_E \wedge T_B \wedge A_K \succ E_E \wedge T_B \wedge A_M \succ E_E \wedge T_\Pi \wedge A_M \succ \underline{E_E \wedge T_\Pi \wedge A_K} \succ \underline{E_{MN} \wedge T_\Pi \wedge A_M} \succ E_{MN} \wedge T_\Pi \wedge A_K \succ E_{MN} \wedge T_B \wedge A_K \succ E_{MN} \wedge T_B \wedge A_M$

Οι υπογραμμισμένες εκβάσεις είναι αυτές που δεν είναι ολικά ταξινομημένες ($E_{MN} \wedge T_\Pi \wedge A_M$ και $E_E \wedge T_\Pi \wedge A_K$). Εάν μια από τις δύο αφαιρεθεί από την ταξινόμηση, τότε η μετακίνηση από την κάθε έκβαση προς την επόμενη γίνεται με μια αλλαγή τιμής σε ακριβώς μια από τις μεταβλητές με βάση τον πίνακα της μεταβλητής CPT και τις αναθέσεις στους γονείς της. Για παράδειγμα από την πρώτη ανάθεση $E_E \wedge T_B \wedge A_K$ για να μεταβούμε στην δεύτερη $E_E \wedge T_B \wedge A_M$ γίνεται χρήση της προτίμησης $A_K \succ A_M$ όταν η ανάθεση στο Τ είναι T_B, και έτσι φαίνεται και ότι η δεύτερη έκβαση δεν προτιμείται τόσο όσο η πρώτη διότι η αλλαγή είναι στην τιμή του Α από μια πιο επιθυμητή τιμή σε μια λιγότερο επιθυμητή. Θα μπορούσε βέβαια να γινόταν και το ακριβώς αντίθετο, αλλάζοντας τιμή για να γίνει μετάβαση από την δεύτερη έκβαση στην πρώτη.

Για κάθε ζευγάρι εκβάσεων α και α', κάθε ακολουθία αλλαγών βελτίωσης από το α' προς το α, αντιστοιχεί μοναδικά σε κάποιο κατευθυνόμενο μονοπάτι από τον κόμβο α' προς τον κόμβο α, στον γράφο προτιμήσεων που εξάγεται από το CP-Net.

Παράδειγμα 3.6.2.2:

Ας θυμηθούμε το παράδειγμα 2.2.3. Υπάρχουν 4 πιθανές ακολουθίες αλλαγών από την έκβαση $T_{\Sigma X} \wedge E_{\Sigma X} \wedge H_{EX}$ προς την έκβαση $T_{AX} \wedge E_{AX} \wedge H_{EX}$ και είναι οι ακόλουθες:

$T_{\Sigma X} \wedge E_{\Sigma X} \wedge H_{EX} \rightarrow T_{AX} \wedge E_{\Sigma X} \wedge H_{EX} \rightarrow T_{AX} \wedge E_{AX} \wedge H_{EX}$

$T_{\Sigma X} \wedge E_{\Sigma X} \wedge H_{EX} \rightarrow T_{AX} \wedge E_{\Sigma X} \wedge H_{EX} \rightarrow T_{AX} \wedge E_{\Sigma X} \wedge H_{OX} \rightarrow T_{AX} \wedge E_{AX} \wedge H_{OX} \rightarrow T_{AX} \wedge E_{AX} \wedge H_{EX}$

$T_{\Sigma X} \wedge E_{\Sigma X} \wedge H_{EX} \rightarrow T_{\Sigma X} \wedge E_{AX} \wedge H_{EX} \rightarrow T_{AX} \wedge E_{AX} \wedge H_{EX}$

$T_{\Sigma X} \wedge E_{\Sigma X} \wedge H_{EX} \rightarrow T_{\Sigma X} \wedge E_{AX} \wedge H_{EX} \rightarrow T_{\Sigma X} \wedge E_{AX} \wedge H_{OX} \rightarrow T_{AX} \wedge E_{AX} \wedge H_{OX} \rightarrow T_{AX} \wedge E_{AX} \wedge H_{EX}$

Επομένως η έκβαση $T_{AX} \wedge E_{AX} \wedge H_{EX}$ είναι προτιμότερη από την έκβαση $T_{\Sigma X} \wedge E_{\Sigma X} \wedge H_{EX}$ αφού από την δεύτερη φτάσαμε στην πρώτη μέσω βελτιωτικών αλλαγών σε τιμές κάθε φορά μιας μεταβλητής. Παρατηρούμε ότι στον γράφο δεν υπάρχει μονοπάτι που να οδηγεί από την έκβαση $T_{\Sigma X} \wedge E_{AX} \wedge H_{OX}$ στην έκβαση $T_{AX} \wedge E_{\Sigma X} \wedge H_{EX}$, άρα ούτε και ακολουθία αλλαγών, και συνεπώς οι δύο αυτές εκβάσεις δεν είναι συγκρίσιμες.

Ας υποθέσουμε ότι N είναι ένα CP-Net με V μεταβλητές, $X_i \in V$, U είναι οι γονείς της X_i και $Y=V-(U \cup \{X_i\})$, $uxy \in Asst(V)$ είναι μια έκβαση, όπου $x \in Dom(X_i)$, $u \in Asst(U)$ και $y \in Asst(Y)$. Μια βελτιωτική αλλαγή τιμής στην έκβαση uxy στην μεταβλητή X_i είναι κάθε έκβαση ux'y τέτοια ώστε $x' \succ^{i}_{u} x$.

Μια ακολουθία βελτιωτικών αλλαγών στο N είναι κάθε ακολουθία εκβάσεων $\alpha_1,\ldots,\alpha_k$ τέτοια ώστε, για κάθε $i<k$, η έκβαση α_{i+1} να είναι μια βελτιωτική αλλαγή στην έκβαση α_i με βάση μια συγκεκριμένη μεταβλητή πάντα.

Μια ακολουθία βελτιωτικών αλλαγών από μια έκβαση α προς μια έκβαση α' είναι κάθε βελτιωτική ακολουθία $\alpha_1,\ldots,\alpha_k$ με $\alpha_1=\alpha$ και $\alpha_k=\alpha'$.

Συνεπώς μια ακολουθία αλλαγών που οδηγεί προς λιγότερο προτιμητέες εκβάσεις μπορεί να οριστεί αντιστρόφως από ότι ορίζεται μια βελτιωτική ακολουθία.

Μια ακολουθία αλλαγών λοιπόν μπορεί να χρησιμοποιηθεί για να δείξει ότι μια έκβαση είναι καλύτερη από μια άλλη, όπως είδαμε και στο προηγηθέν παράδειγμα, και ταυτόχρονα δείχνει ότι είναι χειρότερο να παραβιάζεται μια προτίμηση σε μεταβλητή που είναι πιο ψηλά στο δίκτυο από ότι σε κάποιες που βρίσκονται πιο χαμηλά.

Σε αυτό το σημείο ένα πολύ σημαντικό θεώρημα είναι το εξής: εάν υπάρχει μια βελτιωτική ακολουθία στο CP-Net από την έκβαση α προς την έκβαση α', τότε ισχύει $N \models \alpha' \succ \alpha$.

Ακόμη ένα σημαντικό θεώρημα είναι : εάν έχουμε το CP-Net N, και δεν υπάρχει καμία βελτιωτική ακολουθία αλλαγών από μια έκβαση α προς μια έκβαση α', τότε ισχύει ότι $N \not\models \alpha' \succ \alpha$.

Κεφάλαιο 4

Αποδοτικοί αλγόριθμοι αναδιατύπωσης ερωτημάτων προτιμήσεων

4.1 Εισαγωγή-Προσέγγιση άρθρου

Σε αυτό το κεφάλαιο παρουσιάζεται μια κριτική ανάλυση της εργασίας [5]. Η όλη ανάλυση που γίνεται στο άρθρο βασίζεται στο σκεπτικό ότι οι θα ήταν προτιμότερο για τους χρήστες να περιγράφουν χαρακτηριστικά δεδομένων που θα ταίριαζαν στις δικές τους προτιμήσεις, και οι βάσεις δεδομένων θα πρέπει να μπορούν να επεξεργάζονται τα ερωτήματα με ενσωματωμένες τις προτιμήσεις αυτές, και να δίνουν ως αποτέλεσμα μια ακολουθία μπλοκ δεδομένων, όπου το κάθε μπλοκ θα περιέχει δεδομένα που είναι πιο ενδιαφέροντα για τον χρήστη από τα δεδομένα του μπλοκ που ακολουθεί. Έτσι με αυτό τον τρόπο ο χρήστης θα

μπορεί να κοιτάζει τα δεδομένα από το κάθε μπλοκ με την σειρά, και να σταματήσει σε κάποιο μπλοκ όταν αισθάνεται ικανοποιημένος με όσα αποτελέσματα έλαβε μέχρι εκεί. Επίσης ο χρήστης θα μπορούσε αν το επιθυμούσε να δηλώσει από πριν τον αριθμό μπλοκ δεδομένων ή πλειάδων που θέλει να λάβει στο αποτέλεσμα, παίρνοντας έτσι μόνο τα πρώτα (πιο επιθυμητά) n μπλοκ ή πλειάδες.

Σε αυτό το άρθρο το ενδιαφέρον επικεντρώνεται στον αποδοτικό υπολογισμό τέτοιων ακολουθιών μπλοκ δεδομένων, όταν τα δεδομένα μοντελοποιούνται ως σχεσιακοί πίνακες και οι προτιμήσεις ως δυαδικές σχέσεις πάνω στα χαρακτηριστικά των πινάκων και τα αντίστοιχα πεδία ορισμού.

4.2 Βασικό παράδειγμα

Ακολουθεί ένα παράδειγμα, το οποίο θα χρησιμοποιηθεί σε όλη την ανάλυση του άρθρου για εφαρμογή των όσων περιγράφονται για καλύτερη κατανόηση και επεξήγηση.
Υποθέτουμε ότι έχουμε ένα πίνακα R(A,M,D) που αναπαριστά ένα κομμάτι του ηλεκτρονικού μενού μια ψησταριάς. Το χαρακτηριστικό A αντιστοιχεί στο ορεκτικό, το χαρακτηριστικό M στο κυρίως γεύμα και το D στο επιδόρπιο. Κάθε πλειάδα αναπαριστά ένα γεύμα και έχει ως μοναδικό χαρακτηριστικό τον μοναδικό αριθμό του γεύματος , tid.

Ο πίνακας φαίνεται πιο κάτω:

tid	A	M	D
t1	Λαχανικά	Σουβλάκια	Παγωτό
t2	Ψωμάκια	Σάντουιτς	Κέικ σοκολάτας
t3	Ψωμάκια	Σουβλάκια	Φρουτοσαλάτα
t4	Σαλάτα με ζυμαρικά	Σάντουιτς	Φρουτοσαλάτα
t5	Λαχανικά	Σουβλάκια	Κέικ σοκολάτας
t6	Αλλαντικά	Μακαρονάδα	Παγωτό
t7	Λαχανικά	Μακαρονάδα	Παγωτό
t8	Σαλάτα με ζυμαρικά	Σαλάτα με κοτόπουλο	Γλυκό κουταλιού

t9	Λαχανικά	Μακαρονάδα	Κέικ σοκολάτας
t10	Σαλάτα με ζυμαρικά	Σουβλάκια	Φρουτοσαλάτα

Ας υποθέσουμε τώρα ότι δεν φτιάχνουν καθημερινώς όλα αυτά τα γεύματα, και ότι οι πελάτες της ψησταριάς δηλώνουν τις προτιμήσεις τους, και κάθε μέρα του παραδίδεται στον χώρο εργασίας τους ένα γεύμα από αυτά που υπάρχουν την δεδομένη μέρα, και είναι πιο ψηλά στις προτιμήσεις τους από όλα τα υπόλοιπα της ημέρας εκείνης. Ένας πελάτης έχει δηλώσει τις ακόλουθες προτιμήσεις:

PA : "Τα λαχανικά τα προτιμώ από τα ψωμάκια και την σαλάτα με ζυμαρικά"

PM: "Τα σουβλάκια και την μακαρονάδα τα προτιμώ από την σαλάτα με κοτόπουλο "

PD: "Το παγωτό το προτιμώ από το κέικ σοκολάτας, και το κέικ σοκολάτας το προτιμώ από την φρουτοσαλάτα. "

Οι προτιμήσεις αυτές ορίζονται πάνω στα πεδία ορισμού των τριών χαρακτηριστικών του πίνακα.

Επίσης δηλώνει την προτίμηση:

"Το ορεκτικό (A) είναι εξίσου σημαντικό με το κυρίως πιάτο(M) , και ο συνδυασμός ορεκτικό-κυρίως πιάτο είναι πιο σημαντικός από το επιδόρπιο (D) "

η οποία ορίζεται πάνω στο σύνολο των τριών χαρακτηριστικών του πίνακα.

Η προτίμηση PA συνδέει 3 τιμές του χαρακτηριστικού Α, τις "Λαχανικά", "Ψωμάκια" και "Σαλάτα με ζυμαρικά". Άρα μέσα από αυτή την προτίμηση διαπιστώνουμε ότι οι μόνες πλειάδες που ενδιαφέρουν την έρευνα και θέλουμε να εμφανιστούν στα αποτελέσματα, είναι αυτές που περιέχουν μια από τις πιο πάνω τιμές.

Δηλαδή το σύνολο των πλειάδων που σχετίζονται με την προτίμηση P_A είναι η απάντηση στο ερώτημα:

QA: (A=Λαχανικά) ∨ (A=Κρύα Ζυμαρικά) ∨ (A=Ψωμάκια)

Η απάντηση με βάση τον πίνακα είναι :

Ans(QA)={t1,t2,t3,t4,t5,t7,t8,t9,t10}

Η προτίμηση P_A μοιράζει το πιο πάνω σύνολο σε δύο υποσύνολα, το πρώτο είναι αυτό που υπάρχει η τιμή Λαχανικά στο χαρακτηριστικό ορεκτικό δηλαδή {t1,t5,t7,t9} το οποίο είναι

προτιμότερο από το δεύτερο που είναι ένωση του υποσυνόλου με τις τιμές Σαλάτα με ζυμαρικά και ψωμάκια, δηλαδή {t4,t8,t10} ∪ {t2,t3}.

Η απάντηση λοιπόν στο ερώτημα PQA που συμβολίζει την προτίμηση PA μαζί με το ερώτημα QA, και:

Ans(PQA)={t1,t5,t7,t9}←{t4,t8,t10} ∪ {t2,t3}

Ας υποθέσουμε τώρα ότι έχουμε των συνδυασμό των προτιμήσεων PA και PM, με την ονομασία PAM. Μια πλειάδα θα ενδιαφέρει τον χρήστη εάν περιέχει μια τιμή του A που εμφανίζεται στο PA και μια τιμή του M που εμφανίζεται στο PM. Για να πάρουμε αυτές τις πλειάδες έχουμε το ερώτημα:

QAM: (A=Λαχανικά ∧ M=Σουβλάκια) ∨ (A =Λαχανικά∧ M=Μακαρονάδα)
∨ (A=Λαχανικά∧ M=Σάντουιτς)
∨ (A=Σαλάτα με Ζυμαρικά∧ M=Σουβλάκια)
∨ (A=Σαλάτα με Ζυμαρικά∧ M=Μακαρονάδα)∨ (A=Ψωμάκια∧ M=Σουβλάκια)
∨ (A=Ψωμάκια∧ M=Μακαρονάδα)∨ (A=Σαλάτα με Ζυμαρικά∧ M=Σάντουιτς)
∨ (A=Ψωμάκια∧ M=Σάντουιτς)

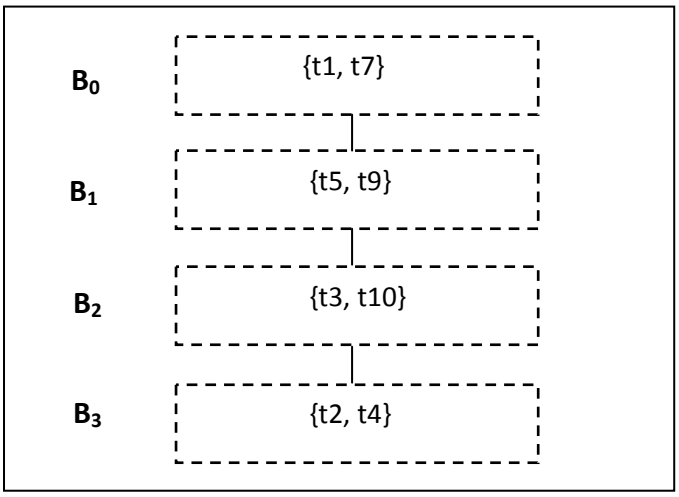
και την απάντηση του:

Ans(QAM)={t1,t5,t7,t9,t10,t3,t4,t2}

Η προτίμηση P_{AM} διαχωρίζει το πιο πάνω σύνολο σε διάφορα υποσύνολο, τα οποία αποτελούν και τα μπλοκ δεδομένων. Αφού το PA και το PM είναι εξίσου σημαντικά ως προς τον χρήστη, έχουμε ότι οι πλειάδες που περιέχουν τον συνδυασμό "Λαχανικά-Σουβλάκια"ή "Λαχανικά-Μακαρονάδα" είναι αυτές που προτιμά ο χρήστης πιο πολύ (άρα θα υπάγονται στο πρώτο μπλοκ) ενώ οι πλειάδες με τους συνδυασμούς "Σαλάτα με Ζυμαρικά-Σάντουιτς" και "Ψωμάκια-Σάντουιτς" είναι οι λιγότερο επιθυμητές (άρα θα υπάγονται στο τελευταίο μπλοκ), και βλέπουμε πιο κάτω την ακολουθία των μπλοκ-υποσυνόλων που αποτελεί την απάντηση του ερωτήματος PQ_{AM}:

Ans(PQAM)={t1,t5 } ∪ {t7,t9}←{t3 } ∪ {t10}← {t4} ∪ {t2}

Στο πιο κάτω σχήμα φαίνεται η ακολουθία των μπλοκ που θα είχε σαν αποτέλεσμα η εκτέλεση του ερωτήματος PQ_{AMD}:



4.3 Μοντελοποίηση προτιμήσεων χρήστη

Το άρθρο στηρίζεται σε μερικές ταξινομήσεις για μοντελοποίηση των προτιμήσεων. Πιο κάτω δίνονται οι συμβολισμοί και οι ορισμοί που χρησιμοποιούνται στο άρθρο.

Η σχέση $d \precsim d'$ υποδεικνύει ότι το d' είναι τουλάχιστον τόσο προτιμητέο όσο και το d σε ένα πεδίο ορισμού D, δηλαδή ότι το d' είναι προτιμότερο ή ίσης προτίμησης με το d για τον χρήστη.
Το συμμετρικό μέρος της σχέσης $\precsim$ είναι μια σχέση ισότητας , που ονομάζεται σχέση ίσης προτίμησης και συμβολίζεται με $\approx$, και το ασύμμετρο της μέρος είναι μια αυστηρή μερική σχέση (strict partial order) που δηλώνει την αυστηρή προτίμηση $d \prec d'$, όταν ισχύει η $d \precsim d'$ και δεν ισχύει η $d' \precsim d$.

Λόγω του ότι η $\precsim$ είναι μερική σχέση ταξινόμησης, δηλαδή δεν μπορούν να συγκριθούν μέσω αυτής όλα τα ζευγάρια στο πεδίο D, προκαλείται και μια σχέση μη-συγκρισιμότητας η οποία συμβολίζεται με $\parallel$.

Έτσι εάν για ένα πεδίο D ισχύει η σχέση $\precsim$ μπορεί να ισχύει ένα από τα ακόλουθα για κάθε ζευγάρι d και d' :

- είτε το d είναι προτιμότερο από το d' δηλαδή ισχύει $d \prec d'$
- είτε το d' είναι προτιμότερο από το d δηλαδή ισχύει $d' \prec d$

- είτε είναι και τα δύο ίσα στις προτιμήσεις του χρήστη δηλαδή $d \approx d'$
- είτε τα δύο δεν είναι συγκρίσιμα μεταξύ τους, δηλαδή $d \parallel d'$

Πολλές συνδυασμένες ανεξάρτητες δηλώσεις προτιμήσεων πάνω σε διάφορα χαρακτηριστικά και οι σημαντικότητες τους, θεωρούνται ως μια έκφραση προτιμήσεων.

Ας υποθέσουμε ότι έχουμε ένα σχεσιακό σχήμα $R(\mathcal{A})$, όπου το $\mathcal{A}=\{A1,\ldots,An\}$ είναι το σύνολο των χαρακτηριστικών, και A είναι ένα μη-κενό υποσύνολο του $\mathcal{A}$. Μια έκφραση προτιμήσεων πάνω στο A, συμβολίζεται ως PA, και ορίζεται ως εξής:

$$PA ::= PAi \mid (PX \approx PY) \mid (PX \prec PY) \qquad (\text{όπου } \mid \text{ σημαίνει είτε})$$

όπου $X \cup Y=A$ και $X \cap Y=\varnothing$. Με το PAi συμβολίζεται μια σχέση προτιμήσεων πάνω το Ai, καθώς το $\approx$ είναι μια σχέση ισότητας και το $\prec$ μια σχέση αυστηρής ταξινόμησης πάνω στο A.

Εάν έχουμε μια έκφραση προτιμήσεων $PX \approx PY$ (το X χαρακτηριστικό το ίδιο σημαντικό με το Y χαρακτηριστικό) , καθορίζουμε μια σχέση $\precsim_{PXY}$ στο πεδίο ορισμού dom(X) x dom(Y), ως εξής:

- $(x,y) \prec_{XY} (x',y')$ αν και μόνο αν $(x \prec_X x' \wedge y \precsim_Y y') \vee (x \precsim_X x' \wedge y \prec_Y y')$
- $(x,y) \approx_{XY} (x',y')$ αν και μόνο αν $x \approx_X x' \vee y \approx_Y y'$
- $(x,y) \parallel_{XY} (x',y')$ σε άλλη περίπτωση

Εάν έχουμε μια έκφραση προτιμήσεων $PX \prec PY$ (το X χαρακτηριστικό πιο σημαντικό από το Y χαρακτηριστικό) , καθορίζουμε μια σχέση $\precsim_{PXY}$ στο πεδίο ορισμού dom(X) x dom(Y), ως εξής:

- $(x,y) \prec_{XY} (x',y')$ αν και μόνο αν $x \prec_X x' \vee (x \approx_X x' \wedge y \prec_Y y')$
- $(x,y) \approx_{XY} (x',y')$ αν και μόνο αν $x \approx_X x' \vee y \approx_Y y'$
- $(x,y) \parallel_{XY} (x',y')$ σε άλλη περίπτωση

Παράδειγμα 4.3.1

Ας υποθέσουμε τις πλειάδες (α1,β1,γ1) και (α2,β2,γ2) και την μοναδική προτίμηση $\gamma 1 \prec_\Gamma \gamma 2$. Εφαρμόζοντας την σχέση στα A και B με βάση αυτά που ορίσαμε πιο πάνω, παίρνουμε $(\alpha 1,\beta 1) \parallel_{AB} (\alpha 2,\beta 2)$. Ακολούθως αν συνθέσουμε αυτό το αποτέλεσμα με το Γ, και πάλι με βάση αυτά που ορίσαμε πιο πάνω το τελικό αποτέλεσμα θα είναι $(\alpha 1,\beta 1,\gamma 1) \parallel_{AB\Gamma} (\alpha 2,\beta 2,\gamma 2)$ λόγω του ότι τα (α1,β1),(α2,β2) δεν είναι συγκρίσιμα.

Ένα ερώτημα προτιμήσεων PQ σε μια σχέση R ορίζεται από μια έκφραση προτίμησης PA, μαζί με ένα προαιρετικό ακέραιο αριθμό κ, που περιορίζει το μέγεθος των απαιτούμενων αποτελεσμάτων. Με βάση το PA, μπορεί να εξαχθεί μια προ-ταξινόμηση (preorder) $\precsim_{PA}$ στο αντίστοιχο καρτεσιανό γινόμενο των χαρακτηριστικών και μια τελική σχέση προτιμήσεων $\precsim_{T}$ στο R.

Σε αυτό το άρθρο αντί για μερικές προ-ταξινομήσεις (partial preorder), το αποτέλεσμα παρουσιάζεται σε ακολουθίες μπλοκ (block sequences) ως εξής : το πιο πάνω μπλοκ περιέχει το πιο επιθυμητό στοιχείο, και σε κάθε άλλο μπλοκ, για κάθε στοιχείο, υπάρχει ένα πιο επιθυμητό στοιχείο στο προηγούμενο μπλοκ. Αυτή η σχέση ονομάζεται σχέση κάλυψης (cover relation). Μια τέτοια ακολουθία μπλοκ κατασκευάζεται με επαναλαμβανόμενη εξαγωγή του επόμενου καλύτερου στοιχείου.

Σκοπός των αλγορίθμων σε αυτό το άρθρο είναι ο υπολογισμός αυτής της ακολουθίας μπλοκ που απαντούν σε ένα ερώτημα προτιμήσεων χωρίς να χρειαστεί να κατασκευαστεί η προκαλούμενη ταξινόμηση των πλειάδων από την σχέση. Αυτό επιτυγχάνεται με εκμετάλλευση της σημασιολογίας μιας έκφρασης προτιμήσεων, και συγκεκριμένα να γίνει με γραμμοποίηση (linearizing) του καρτεσιανού γινομένου όλων των όρων χαρακτηριστικών που εμφανίζονται στην έκφραση, ή ακόμα πιο απλά παράγοντας την ακολουθία μπλοκ του, από τις ακολουθίες μπλοκ των συστατικών σχέσεων προτιμήσεων του.

<u>Παράδειγμα 4.3.2</u>

Με βάση το βασικό παράδειγμα της υποενότητας 4.2, η ακολουθία μπλοκ για την P_A είναι {Λαχανικά}←{Ψωμάκια ,Σαλάτα με Ζυμαρικά} και για την P_M είναι {Σουβλάκια,Μακαρονάδα}←{Σάντουιτς}. Αυτά είναι τα μπλοκ των δύο σχέσεων προτιμήσεων, που θα αποτελούσαν συστατικές σχέσεις για την σχέση P_{AM}.

Θεώρημα 1: Εάν έχουμε τις ακολουθίες μπλοκ

$$X0 \leftarrow X1 \leftarrow \ldots \leftarrow Xn\text{-}2 \leftarrow Xn\text{-}1$$

και

$$Y0 \leftarrow Y1 \leftarrow \ldots \leftarrow Ym\text{-}2 \leftarrow Ym\text{-}1$$

δυο προτιμήσεων PX και PY, τότε η ακολουθία μπλοκ

$$\mathbf{Z0 \leftarrow Z1 \leftarrow \ldots \leftarrow Zn*m\text{-}2 \leftarrow Zn*m\text{-}2}$$

της προτίμησης που προκαλείται από την σχέση **PX≈PY** πάνω στο X x Y θα αποτελείται από n+m-1 μπλοκ, και κάθε μπλοκ Z_P θα περιέχει στοιχεία μόνο από τα μπλοκ X_q και Y_r έτσι ώστε q+r=p.

Θεώρημα 2: Εάν έχουμε τις ακολουθίες μπλοκ

X0←X1←... ← Xn-2←Xn-1 και

Y0←Y1←... ←Ym-2←Ym-1

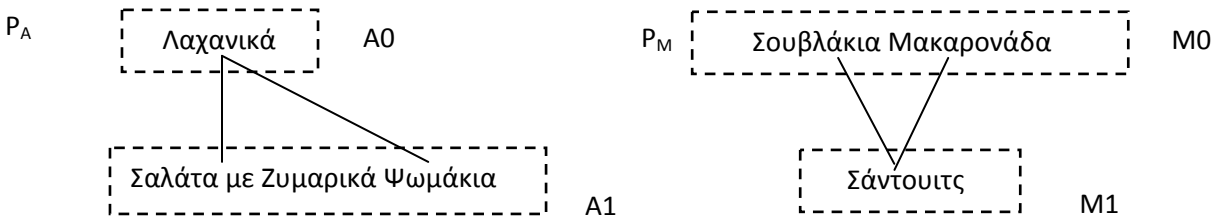
δυο προτιμήσεων PX και PY τότε η ακολουθία μπλοκ

Z0←Z1←... ←Zn*m-2←Zn*m-2

της προτίμησης που προκαλείται από την σχέση **PX≺ PY** πάνω στο X x Y θα αποτελείται από n*m μπλοκ, και κάθε μπλοκ Z_P θα περιέχει στοιχεία μόνο από τα μπλοκ X_q και Y_r έτσι ώστε p=q*m+r. Για κάθε τιμή του q από το 0 μέχρι το n-1, το r θα έχει πεδίο τιμών από το 0 εώς το m-1.

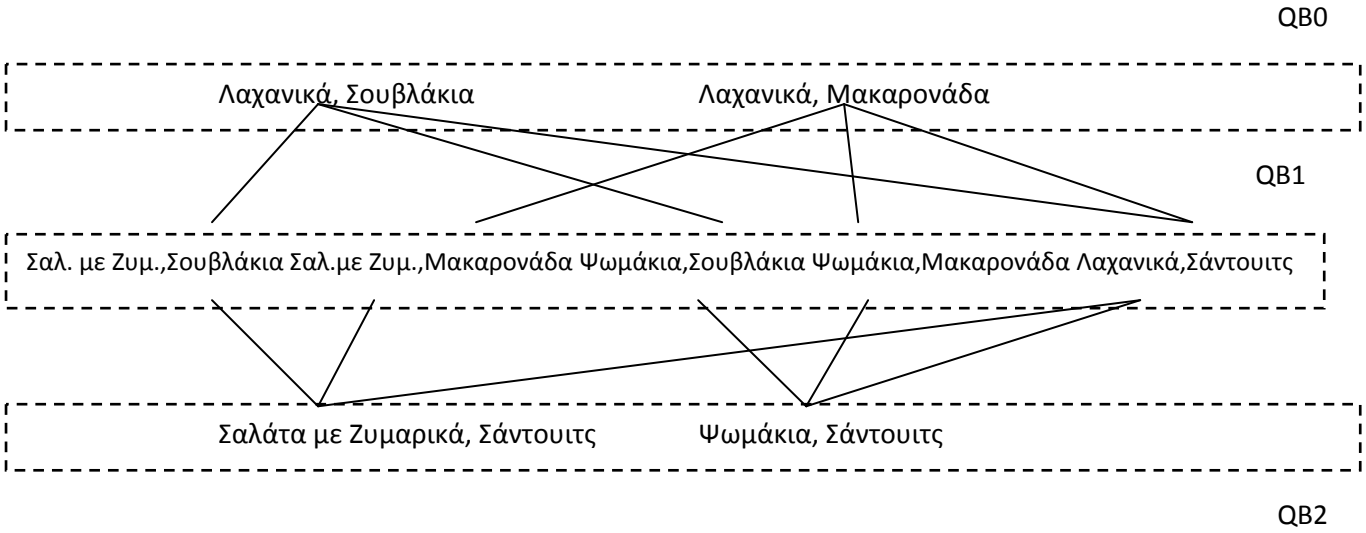
<u>Παράδειγμα 4.3.3</u>

Με βάση το βασικό παράδειγμα και το παράδειγμα 4.3.2, πιο κάτω βλέπουμε τις ακολουθίες των μπλοκ για τις P_A και P_M :



Δεδομένου αυτών των ακολουθιών μπλοκ, η ακολουθία μπλοκ για την σχέση προτιμήσεων που προκαλείται από την σχέση $P_A \approx P_M$ θα αποτελείται από 3 μπλοκ με βάση το θεώρημα 1: 2(αφού το τελευταίο μπλοκ για P_A είναι A1) +2 (αφού το τελευταίο μπλοκ για P_M είναι M1)-1.

Βλέπουμε πιο κάτω την ακολουθία μπλοκ που παράγεται για το ερώτημα QAM, από όπου και θα δημιουργηθεί η ακολουθία μπλοκ που θα δοθεί ως αποτέλεσμα:



Παρατηρούμε ότι το πιο πάνω μπλοκ (QB0) περιέχει στοιχεία από τα μπλοκ των συστατικών προτιμήσεων που το άθροισμα των δεικτών τους ισούται με μηδέν, δηλαδή A0 και M0. Το δεύτερο μπλοκ (QB1) περιέχει στοιχεία από μπλοκ των συστατικών προτιμήσεων που το άθροισμα των δεικτών τους είναι 1, δηλαδή A0 με M1, και A1 με M0, και το τρίτο μπλοκ (QB2) περιέχει στοιχεία από μπλοκ των συστατικών προτιμήσεων που το άθροισμα των δεικτών τους είναι 2, δηλαδή A1 και M1.

4.4 Αλγόριθμοι ταξινόμησης ερωτημάτων

Αρχικά παρουσιάζω τους συμβολισμούς που θα χρησιμοποιηθούν στο υπόλοιπο άρθρο:

- **V(P,Ai)** θα συμβολίζεται το σύνολο των ενεργών τιμών για μια προτίμηση PAi, πάνω στο χαρακτηριστικό Ai, δηλαδή το υποσύνολο των τιμών του χαρακτηριστικού A_i, οι οποίες αναφέρονται στις εκφράσεις προτιμήσεων του χρήστη.
- Εάν έχουμε μια προτίμηση PA πάνω σε ένα μη-κενό υποσύνολο A των χαρακτηριστικών του R:
    - **dom(A)** είναι το καρτεσιανό γινόμενο Xi(dom(Ai)), δηλαδή το καρτεσιανό γινόμενο των πεδίων των επί μέρους χαρακτηριστικών, και
    - **V(P,A)** είναι το ανάλογο πεδίο ορισμού των ενεργών προτιμήσεων.
- **T(P,A)** θα συμβολίζεται το σύνολο των ενεργών πλειάδων του R, δηλαδή εκείνων που έχουν ενεργές τιμές για κάθε χαρακτηριστικό που βρίσκεται στην P_A.
- **dp** θα συμβολίζεται η πυκνότητα προτιμήσεων μιας έκφρασης προτιμήσεων PA και ορίζεται ως |T(P,A)|/|V(P,A)|

- **ap** θα συμβολίζεται το ενεργό ποσοστό (active ratio) μια έκφρασης προτιμήσεων PA και ορίζεται ως |T(P,A)|/|R|.

4.4.1 Πλέγμα ερωτημάτων(Query Lattice)

Μια έκφραση προτιμήσεων P_A πάνω σε ένα σύνολο χαρακτηριστικών A={Ai1,Ai2,...,AiN} προκαλεί μια προτίμηση πάνω στα στοιχεία (ak1,ak2,...,akN) του πεδίου των ενεργών όρων V(P,A). Αυτά τα στοιχεία είναι συζευκτικά ερωτήματα της μορφής Ai1=ak1∧Ai2=ak2 ∧... ∧AiN=akN τα οποίο όταν εκτελεστούν θα δώσουν ως αποτέλεσμα τις ενεργές πλειάδες του συνόλου T(P,A).

Η ταξινόμηση των ερωτημάτων αυτών, θα αναφέρεται στο άρθρο ως «Πλέγμα Ερωτημάτων».

Παράδειγμα 4.4.1.1**:**
Υποθέτουμε ότι έχουμε την έκφραση προτίμησης PAM=(PA≈PM),όπου PA={Ψωμάκια≺Λαχανικά, Σαλάτα με Ζυμαρικά≺Λαχανικά} και P_M={Σάντουιτς≺Σουβλάκια, Σάντουιτς≺Μακαρονάδα}, που φαίνεται και στο πρώτο σχήμα του παραδείγματος 4.3.3. Στο δεύτερο σχήμα του ιδίου παραδείγματος βλέπουμε την προκαλούμενη προτίμηση P_{AM} πάνω στο καρτεσιανό γινόμενο των δύο ενεργών πεδίων τιμών, V(P,A) και V(P,M), καθώς και την δημιουργούμενη ακολουθία μπλοκ του $V(P_{AM},\{A,M\})$.

Έτσι για υπολογισμό του κορυφαίου μπλοκ B0, του ερωτήματος PQ_{AM}, πρέπει να εκτελεστούν τα ερωτήματα A=Λαχανικά∧M=Σουβλάκια και A=Λαχανικά∧M=Μακαρονάδα, αυτά δηλαδή που βρίσκονται στο πρώτο μπλοκ στην ακολουθία των ερωτημάτων, στο QB0. Τα αποτελέσματα των δύο εκτελέσεων δίνουν από μια πλειάδα ως αποτέλεσμα, οι οποίες θα αποτελέσουν το πρώτο μπλοκ.

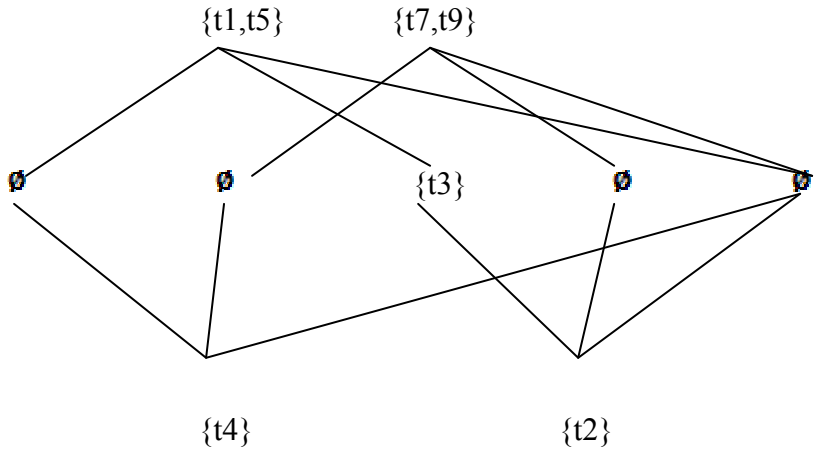
Όμως υπάρχει και το ενδεχόμενο, το αποτέλεσμα ενός από τα ερωτήματα να είναι κενό, δηλαδή καμία πλειάδα να μην αντιστοιχεί σε αυτό το ερώτημα. Βλέποντας το δεύτερο μπλοκ των ερωτημάτων, QB1, μόνο η ερωτήση A=Ψωμάκια∧M=Σουβλάκια έχει πλειάδα ως αποτέλεσμα, όλες οι άλλες ερωτήσεις επιστρέφουν το κενό σύνολο. Η πλειάδα αυτή που επιστρέφεται θα ανήκει στο δεύτερο μπλοκ στην ακολουθία. Μαζί με αυτή στο μπλοκ αυτό

μπαίνει όποια άλλη πλειάδα που είναι αποτέλεσμα κάποιου ερωτήματος που είναι διάδοχος των κενών ερωτημάτων του μπλοκ QB1, αλλά ταυτόχρονα να μην είναι και διάδοχος οποιουδήποτε άλλου ερωτήματος από το QB1, το οποίο να μην έχει κενό σύνολο ως αποτέλεσμα.

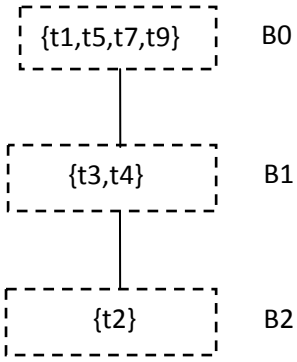
Στο δεύτερο σχήμα του παραδείγματος 3.3.3 μπορούμε να δούμε και τις σχέσεις διαδοχής (γονέων- παιδιών) μέσω των ευθειών που τα αντιστοιχούν.

Στα πιο κάτω σχήματα βλέπουμε,(1) τα αποτελέσματα των ερωτημάτων, και (2) την ταξινόμηση τους στην ακολουθία μπλοκ με βάση όσα αναφέραμε πιο πάνω.

(1)



(2)



4.4.2 Αλγόριθμος πλέγματος (Lattice Based Algorithm(LBA))

Ο αλγόριθμος που παρουσιάζεται σε αυτό το άρθρο παίρνει ως είσοδο μια σχέση R και μια έκφραση προτιμήσεων P_A η οποία εμπλέκει ένα υποσύνολο A των χαρακτηριστικών της R. Δίνει ως έξοδο ακολουθιακά μπλοκ του T(P,A). Κάθε φορά που υπολογίζεται ένα μπλοκ μπορεί να συνεχιστεί ο υπολογισμός του επόμενου με εντολή του χρήστη ή θα υπολογίζει μόνο τις πιο προτιμητέες κ πλειάδες με βάση το κ που όρισε ο χρήστης.

Algorithm LBA
input: a relation R, a preference expression P_A and a k>0
output: the block sequence of T(P, A)

```
1:  QB = ConstructQueryBlocks(P_A.root)
2:  totalsize = i = 0
3:  repeat
4:    Uq_i = GetBlockQueries(QB[i])
5:    totalsize += Evaluate(Uq_i)
6:    i += 1
7:  until ExitReq or totalsize >= k or i=|QB|
```

Εικόνα από το άρθρο , που παρουσιάζει σε γενικές γραμμές την λειτουργία του αλγορίθμου

Χρησιμοποιείται ένας πίνακας, ο QB για να κρατά στην μνήμη την δομή της ακολουθίας των μπλοκ του V(P,A) δηλαδή του πλέγματος ερωτημάτων. Ουσιαστικά κάθε κελί του πίνακα αντιστοιχεί σε ένα μπλοκ της ακολουθίας. Το πλέγμα ερωτημάτων όμως στην πραγματικότητα δεν υλοποιείται ως δομή, αλλά τα ερωτήματα που χρειάζονται για να παραχθούν τα μπλοκ του T(P,A) υπολογίζονται την στιγμή που χρειάζονται με βάση την δομή που κρατά ο QB και εκτελούνται εκείνη την ώρα, χωρίς να αποθηκεύονται κάπου.

Κάθε εγγραφή του πίνακα είναι μια λίστα της οποίας τα στοιχεία κρατούν τους δείκτες των στοιχείων του V(P,Ai) που σχηματίζουν ένα μπλοκ του V(P,A). Κάθε κόμβος της λίστας δηλαδή κρατά ένα δείκτη για κάθε χαρακτηριστικό που λαμβάνει μέρος, ουσιαστικά ένα συνδυασμό. Το πόσοι συνδυασμοί βρίσκονται στο ίδιο μπλοκ, υποδεικνύεται από τον αριθμό των κόμβων που αποτελούν την λίστα.

Παράδειγμα 4.3.1.1

Αν θυμηθούμε το παράδειγμα 4.3.3 και της ακολουθίες μπλοκ για τις προτιμήσεις PA και PM, τότε με βάση τα πιο πάνω, στο κελί QB0 θα έχουμε λίστα με ένα κόμβο που θα περιέχει τα στοιχεία <0,0> τα οποία αντιπροσωπεύουν τον συνδυασμό A0 με M0, στο κελί QB1 θα έχουμε μια λίστα με δύο κόμβους όπου ο ένας θα περιέχει τα στοιχεία <0,1> και ο άλλος τα

στοιχεία <1,0> που αντιπροσωπεύουν τους συνδυασμούς A0 με M1 και A1 με M0 αντίστοιχα, και στο κελί QB2 θα έχουμε τα στοιχεία <1,1> που αντιπροσωπεύουν τον συνδυασμό A1 με M1.

Αφού υπολογιστεί αρχικά ο QB, ο αλγόριθμος επαναληπτικά καλεί την συνάρτηση GetBlockQueries, για δημιουργία της σχετικής λίστας διαζευκτικών ερωτημάτων που χρειάζεται να εκτελεστούν για να ληφθούν τα αποτελέσματα, και την συνάρτηση Evaluate για την εξαγωγή των αποτελεσμάτων σε διαδοχικά μπλοκ του T(P,A). Η επανάληψη συνεχίζεται είτε μέχρι να δοθεί εντολή τερματισμού, είτε μέχρι να εκτελεστεί κ φορές (εάν έδωσε ο χρήστης όριο), είτε μέχρι να εκτελεστεί για όλα τα μπλοκ.

```
Function ConstructQueryBlocks
input: a preference expression P_A
output: a query block sequence QB
1:   if P is a leaf then          //a preference relation P on attribute A_i
2:     QB = PrefBlocks(V(P,A_i))
3:   else
4:     QB_left = ConstructQueryBlocks(P.left)
5:     QB_right = ConstructQueryBlocks(P.right)
6:     if P.type = '≈' then          // equally important preferences
7:                for w=0 to |QB_left| + |QB_right| - 1
                  //construct the block sequence of V(P.left ≈ P.right, A)
8:           QB[w] = ∪{QB_left[i] × QB_right[j] | i+j=w}
9:     else                    // strictly more important preferences
10:       w=0
11:       for j=0 to |QB_right|-1
12:           for i=0 to |QB_left|-1
                  // construct the block sequence of V(P.left≺P.right, A)
13:             QB[w] = QB  left[i] × QB  right[j]
14:             w+=1
15:  return QB
```

Εικόνα από το άρθρο , που παρουσιάζει την συνάρτηση ConstructQueryBlocks

Η συνάρτηση ConstructQueryBlocks διασχίζει το δέντρο μιας έκφρασης προτιμήσεων P_A την οποία παίρνει ως είσοδο, αναδρομικά αρχίζοντας από την ρίζα, και υπολογίζει όπως είπαμε και πριν την ακολουθία των μπλοκ που αναπαριστάτε με τον πίνακα QB. Για κάθε QB εισαγωγή παράγει την δομή της αντίστοιχης ακολουθίας μπλοκ με βάση της σχέσεις $\prec$ και $\approx$ανάμεσα στο αριστερό και δεξιό υποδέντρο, με τον κατάλληλο τρόπο όπως ορίζουν τα θεωρήματα 1 και 2. Για τα φύλλα, δηλαδή σχέσεις προτιμήσεων σε ξεχωριστά χαρακτηριστικά, υπολογίζονται οι αντίστοιχες εισαγωγές για το QB μέσω της συνάρτησης PrefBlocks

```
Function Evaluate
input: a list of queries Uq_i
output: the next block B_i
1:   for each q_i in Uq_i
2:    if q_i not in SQ then
3:          if ans(q_i) != ∅ then
4:                CurSQ ∪= {q_i}; B_i ∪= ans(q_i)
5:          else FQ ∪= {q_i}
6:    else FQ ∪= {q_i}
7:   while FQ != ∅
8:    for each q in FQ
9:          FQ \= {q}
10:         Q = {q | q=child(q)}
11:   for each q in Q
12:         if q not in SQ then
13:               if not q in succ(q') forall q' in CurSQ then
14:                     if ans(q) != ∅ then
15:                           CurSQ ∪= {q}; B_i ∪= ans(q)
16:                     else FQ ∪= {q}
17:         else FQ ∪= {q}
18:  SQ∪ = CurSQ; CurSQ = ∅
19:  print B_i, return |B_i|
```

Εικόνα από το άρθρο , που παρουσιάζει την συνάρτηση Evaluate

Η συνάρτηση Evaluate παίρνει ως είσοδο μια λίστα ερωτημάτων και εκτελεί κάθε ερώτημα από την λίστα. Κρατά εγγραφές για τα μη-κενά ερωτήματα (που έχουν έστω μια πλειάδα ως αποτέλεσμα) στην δομή SQ, έτσι ώστε να εκτελεστούν μια μόνο φορά. Επίσης κρατά εγγραφές για τα μη-κενά ερωτήματα του μπλοκ του T(P,A) που είναι υπό επεξεργασία την δεδομένη στιγμή στην δομή CursQ, καθώς και για τα κενά ερωτήματα στην δομή FQ.

Για κάθε μη-κενό ερώτημα τοποθετεί την απάντηση του στο τρέχων υπό επεξεργασία μπλοκ. Για κάθε κενό ερώτημα εφαρμόζει την ίδια μέθοδο στους διαδόχους του που δεν ανήκουν στο SQ, και ούτε στο CursSQ έτσι ώστε να επιβεβαιωθεί ότι δεν είναι ταυτόχρονα και διάδοχοι μη-κενών ερωτημάτων. Η διαδικασία αυτή τερματίζεται όταν δεν υπάρχουν άλλοι διάδοχοι, ή όταν δεν υπάρχουν άλλα κενά ερωτήματα προς διερεύνηση. Η συνάρτηση δίνει ως έξοδο το υπολογισμένο μπλοκ και το μέγεθός του.

4.4.3 Τιμές ορίων(The Threshold Values)

Στις περιπτώσεις που ο αριθμός των ενεργών συνδυασμών (V(P,A)) είναι αρκετά μικρότερος από τον αριθμό των ενεργών πλειάδων (T(P,A)) σημαίνει ότι θα εκτελεστούν στον

προηγούμενο αλγόριθμο πολλά ερωτήματα με κενά αποτελέσματα. Έτσι παρουσιάζεται στην συνέχεια ένας δεύτερος αλγόριθμος ο TBA, ο οποίος είναι υβριδικός συνδυάζοντας τον προηγούμενο αλγόριθμο με αλγορίθμους που κάνουν ελέγχους κυριαρχίας. Σκοπός του είναι να μειώσεις τις συγκρίσεις στο ελάχιστο δυνατό, και για να τον πετύχει βασίζεται σε κάποιες τιμές ορίων του V(P,A).

Θα αναφερθώ και πάλι στο παράδειγμα 3.3.3 για καλύτερη εξήγηση και κατανόηση της ενότητας. Το πιο πάνω μπλοκ QB_0 του πλέγματος ερωτημάτων περιέχει τις καλύτερες τιμές του V(P,A) αφού συνδυάζει στοιχεία από τα πιο πάνω μπλοκ A_0 και M_0 των σχετιζόμενων χαρακτηριστικών. Τα ζεύγη των τιμών που παράγονται με αυτό τον συνδυασμό λειτουργούν ως όρια, δηλαδή θεωρούμε ότι δεν μπορεί να υπάρξει κάποια πλειάδα που ακόμη δεν έχει διερευνηθεί στο αποτέλεσμα του PQ_{AM} η οποία να έχει καλύτερες τιμές από τις <Λαχανικά, Σουβλάκια> και <Λαχανικά, Μακαρονάδα>.

Ακολούθως θεωρούμε ένα διαζευκτικό ερώτημα q στο χαρακτηριστικό A, το οποίο σχηματίζεται από όλες τις τιμές του A_0. Στο παράδειγμα το q είναι W=Λαχανικά αφού δεν περιέχει άλλη τιμή το A_0. Προφανώς οποιαδήποτε πλειάδα της σχέσης R, που δεν ανήκει στο αποτέλεσμα του q, δεν μπορεί να είναι καλύτερη από πλειάδες οι οποίες ταιριάζουν σε ζευγάρια τιμών που λαμβάνονται από το επόμενο μπλοκ του V(P,A) για το χαρακτηριστικό A, δηλαδή από το A_1.Διευκρινίζουμε ότι όσο αφορά το χαρακτηριστικό M, έχουμε παραμείνει στο πρώτο μπλοκ του V(P,M) του. Στο παράδειγμα το βλέπουμε αυτό αφού καμία πλειάδα που δεν περιέχεται στο αποτέλεσμα του q, δεν είναι καλύτερη από τις <Σαλάτα με Ζυμαρικά, Σουβλάκια>, <Σαλάτα με Ζυμαρικά, Μακαρονάδα>, <Ψωμάκια, Σουβλάκια>, <Ψωμάκια, Μακαρονάδα>.

Δηλαδή, το όριο χαμηλώνει, πηγαίνοντας ένα μπλοκ κάτω στο V(P,A) δηλαδή στο χαρακτηριστικό από το οποίο επιλέχθηκαν οι τιμές για το q, και μένει το προηγούμενο όριο στο V(P,M).

Ακολούθως πρέπει να γίνει έλεγχος για κυριαρχία μεταξύ των πλειάδων που επιστρέφονται από την q. Στο παράδειγμα μας επιστρέφονται τέσσερις πλειάδες : t1,t5,t7,t9. Καμία από αυτές δεν κυριαρχεί πάνω σε άλλη. Εάν το νέο όριο, στην περίπτωση μας A_1 x M_0, καλύπτεται από το σύνολο των πλειάδων στην απάντηση της q που δεν κυριαρχεί καμία σε βάρος άλλης, τότε οι πλειάδες αυτές αποτελούν το μπλοκ, B_0.

Με επανάληψη αυτής της διαδικασίας, φτάνουμε στο τελευταίο μπλοκ, B2, και κατασκευάζουμε την ακολουθία μπλοκ των πλειάδων.

4.4.4 Αλγόριθμος με βάση τις τιμές ορίων (Threshold Based Algorithm)

Algorithm TBA
input: a relation R, a preference expression P_A and a k>0
output: the block sequence of T(P,A)

```
1:  for j=1 to m                    // m is the number of attributes in P_A
2:    PB[j] = PrefBlocks(V(P, Aj))
3:    Thres[j]=head(PB[j])
4:  U=D=∅; Totalsize=0
5:  repeat
6:    i = min_selectivity(Thres)
7:    Q = ∨(Ai=vj), ∀vj∈Thres[i]
8:    <D, U> = OrderTuples(ans(Q), D, U)
9:    if next(PB[i]) then
10:     Thres[i] = next(PB[i])
11:     <D, U, Totalsize >= CheckCover(U, D)
12:   else
13:     Thres={⊥}
14:     <D, U, Totalsize >= CheckCover(U, D)
15:     exit
16: until Totalsize >=k or ExitReq
```

Εικόνα από το άρθρο , που παρουσιάζει σε γενικές γραμμές την λειτουργία του αλγορίθμου

Ο αλγόριθμος αυτός καλεί την συνάρτηση PrefBlocks για υπολογισμό της ακολουθίας μπλοκ για τα $V(P,A_i)$. Διατηρεί τα αποτελέσματα όπως και ο πρώτος αλγόριθμος σε ένα πίνακα PB από λίστες , της οποίας τα στοιχεία κρατούν μόνο τους δείκτες των ενεργών τιμών.

Οι τιμές των ορίων φυλάγονται σε ένα άλλο πίνακα τον Thres μεγέθους m, όπου m είναι ο συνολικός αριθμός των χαρακτηριστικών A_i. Αρχικά έχει τις τιμές των κορυφαίων μπλοκ κάθε λίστας του PB, δηλαδή κάθε $V(P,A_i)$.

Κατά τη διάρκεια της εκτέλεσης ο αλγόριθμος κρατά στη μνήμη δυο σύνολα με τις πλειάδες που πάρθηκαν αλλά ακόμη δεν επιστράφηκαν: το σύνολο D που περιέχει πλειάδες για τις οποίες βρέθηκε κάποια καλύτερη και το σύνολο U που περιέχει τις ισοδύναμες κλάσεις πλειάδων για τις οποίες μια καλύτερη πλειάδα πρόκειται να βρεθεί.

Ακολούθως επαναλαμβάνονται τα ακόλουθα τέσσερα βήματα, μέχρι να δοθεί εντολή του χρήστη για τερματισμό, να υπολογιστούν οι πιο προτιμητέες κ πλειάδες με βάση το κ που όρισε ο χρήστης, ή να μην μπορεί να υπάρξει περαιτέρω διερεύνηση:

(1) Ο αλγόριθμος αναγνωρίζει το χαμηλότερης επιλεκτικότητας μπλοκ $V(P,A_i)$ ανάμεσα σε εκείνα που αναφέρονται στο Thres, και εκτελεί το αντίστοιχο q.

(2) Καλείται η συνάρτηση OrderTuples για να συγκρίνει ανά ζεύγη τις πλειάδες που επιστραφήκαν ως αποτέλεσμα της εκτέλεσης της q και να ανανεώσει αναλόγως τα σύνολα D και U.

(3) Το επόμενο καλύτερο μπλοκ $V(P,A_i)$ ανανεώνει το Thres

(4) Καλείται η συνάρτηση CheckCover για να δώσει ως αποτέλεσμα ένα ή περισσότερα από τα μπλοκ της απάντησης, και να ανανεώσει τα σύνολα D και U.

```
Function OrderTuples
input : sets of tuples A, Dom, set of tuple classes Und
output: a pair of sets <UptDom, UptUnd>,
    UptUnd: set of tuple classes, UptDom: set of tuples
1:   UptDom=Dom                          // [] denotes a class of tuples
2:   if Und=∅ then UptUnd = {[t1]} else UptUnd=Und
                                         // t1 is the first active tuple of A
3:   for each active tuple t in A
4:     IsDominated=false
5:     for each t′ in UptUnd
6:       if t<t′ then
7:          IsDominated =true
8:          UptDom ∪={t}; break   //inner for
9:       elseif t~t′ then
10:         [t′] ∪= {t}; break   //inner for
11:      elseif t′<t then
12:         UptUnd \= {[t′]}
13:         UptDom ∪= {t′}
14:    if not IsDomilnated then
15:      UptUnd ∪= [t]
16:  return <UptDom, UptUnd>
```

Εικόνα από το άρθρο , που παρουσιάζει την συνάρτηση OrderTuples

Η τρίτη περίπτωση τερματισμού, δηλαδή να μην μπορεί να υπάρξει περαιτέρω διερεύνηση ενεργοποιείται όταν μια από τις λίστες-στοιχεία του Thres εξαντληθεί. Όταν ενεργοποιηθεί αυτή η κατάσταση τερματισμού, γίνεται το εξής: χρησιμοποιείται ένα ειδικό σύμβολο ως κατώτατο όριο το $\perp$, έτσι η συνάρτηση CheckCover θα βρει οποιαδήποτο σύνολο μη-

κυρίαρχων πλειάδων καλύτερο από το $\perp$ και θα δώσει ως αποτέλεσμα τα επόμενα μπλοκ όπως ζητήθηκε.

Η συνάρτηση OrderTuples παίρνει ως είσοδο το σύνολο πλειάδων A (απάντηση στο q), το D και το U. Εάν το U είναι άδειο τότε αρχικοποιείται με την κλάση της πρώτης πλειάδας στο A. Η συνάρτηση ενημερώνει τα D και U αφού συγκρίνει κάθε πλειάδα t που υπάρχει στο A, με μια αντιπρόσωπο t' όλων των κλάσεων πλειάδων του U. Κατά τη σύγκριση μπορεί να υπάρξουν τέσσερις περιπτώσεις:
(1) Εάν η t είναι χειρότερη από την t', τοποθετείτε στο D και δεν χρειάζεται να συγκριθεί με τις υπόλοιπες του U.
(2) Εάν η t είναι εξίσου προτιμητέα με την t', τοποθετείτε στην κλάση της t' στο U και πάλι δεν χρειάζονται περαιτέρω συγκρίσεις με το U.
(3) Εάν η t αποδειχθεί καλύτερη από την t' η κλάση της t' μετακινείτε από το U στο D, και η συνάρτηση συνεχίζει να συγκρίνει την t με το υπόλοιπο U.
(4) Εάν η t δεν είναι συγκρίσιμη με την t', οι συγκρίσεις συνεχίζονται με το υπόλοιπο U χωρίς άλλες ενέργειες. Στο τέλος των συγκρίσεων εάν η t αποδειχθεί ότι κυριαρχείτε από κάποιο στοιχείο του U, τοποθετείτε στο U μια νέα κλάση που περιέχει την t.

Η συνάρτηση CheckCover παίρνει ως είσοδο το D, το U και την τρέχουσα τιμή του μεγέθους του αποτελέσματος. Χρησιμοποιώντας το τρέχων όριο, το ζητούμενο k, και τις παραμέτρους που έλαβε, αναδρομικά δίνει ως αποτέλεσμα όσα περισσότερα μπλοκ του αποτελέσματος είναι δυνατόν να δώσει. Επιστρέφει ανανεωμένες τις παραμέτρους εισόδου της. Ελέγχει κατά πόσο το U καλύπτει το όριο. Εάν το καλύπτει τότε το U αποτελεί το επόμενο μπλοκ, και ανανεώνεται το μέγεθος του αποτελέσματος, και το U μηδενίζεται.

4.5 Αξιολόγηση

Στο άρθρο περιλαμβάνεται μια ανάλυση της πολυπλοκότητας των δύο πιο πάνω αλγορίθμων. Όσο αφορά τον LBA αλγόριθμο το κόστος στην καλύτερη περίπτωση είναι O(log|R|) και στην χειρότερη περίπτωση O(|V(P,A)|). Από την άλλη ο αλγόριθμος TBA έχει στην καλύτερη περίπτωση κόστος O(log|R|) και στην καλύτερη κόστος $O(|T(P,A)|^2)$.

Κεφάλαιο 5

Σύγκριση και τροποποίηση μεθοδολογίας κεφαλαίου 4

Σκοπός είναι η εφαρμογή της μεθόδου που περιγράφεται στο άρθρο «Efficient Rewriting Algorithms for Preference Queries» , σε δίκτυα προτιμήσεων με συνθήκες και σημασιολογία ceteris paribus.

Ανάμεσα στις προτιμήσεις χωρίς συνθήκες και χωρίς σημασιολογία ceteris paribus που εφαρμόζει το άρθρο και στις προτιμήσεις με συνθήκες, ceteris paribus σημασιολογίας υπάρχουν βασικές διαφορές οι οποίες εγείρουν κάποια βασικά θέματα προς επίλυση και διαφοροποίηση.

Αρχικά στην μεθοδολογία του άρθρου υπήρχε η πιθανότητα εφαρμογής της μεθοδολογίας σε δύο προτιμήσεις P_A και P_B όπου $P_A \approx P_B$ για εύρεση αποτελεσμάτων για την σχέση προτιμήσεων P_{AB} (παράδειγμα 4.3.3).

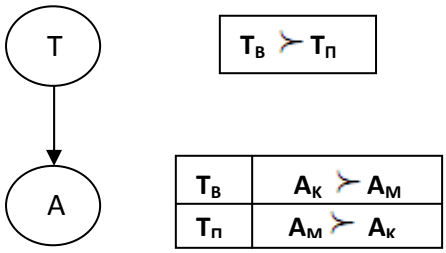
Αυτό δεν είναι δυνατό να υπάρξει σε προτιμήσεις με συνθήκες διότι με προτιμήσεις ίσης σημασίας δεν μπορούν να υφίστανται συνθήκες. Από την στιγμή που κάποια προτίμηση πρέπει να εξαρτάται από άλλες, σημαίνει ότι έστω μια από είναι πιο σημαντική από τις υπόλοιπες.

Έτσι έχουμε δύο περιπτώσεις :

(1) Σχέση προτιμήσεων που συνδυάζει προτιμήσεις πάνω σε χαρακτηριστικά, που είναι η μια πιο μεγάλης σημαντικότητας από τη άλλη, δηλαδή η κάθε μια εξαρτάται από την προηγούμενη της, εκτός η πρώτη.
Ας χρησιμοποιήσουμε ένα παράδειγμα από το κεφάλαιο 3 για καλύτερη επεξήγηση και κατανόηση :

Ας υποθέσουμε ότι το σύνολο των εκβάσεων αποτελείται από ένα σύνολο πιθανών σπιτιών για αγορά. Τα χαρακτηριστικά που περιγράφουν ένα σπίτι είναι: ΑΡΧΙΤΕΚΤΟΝΙΚΗ και ΤΟΠΟΘΕΣΙΑ. Ας υποθέσουμε ότι ο χρήστης έχει αυστηρή προτίμηση (P_T) στα σπίτια που βρίσκονται στο βουνό (ΤΟΠΟΘΕΣΙΑ=ΒΟΥΝΟ), από τα σπίτια που βρίσκονται στην πόλη (ΤΟΠΟΘΕΣΙΑ=ΠΟΛΗ). Οι προτιμήσεις του πάνω στην αρχιτεκτονική (P_A) με πιθανές τιμές : ΜΟΝΤΕΡΝΑ, ΚΛΑΣΙΚΗ εξαρτάται από την τιμή που παίρνει το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ. Δηλαδή, εάν το σπίτι βρίσκεται στο βουνό τότε προτιμάει την κλασική αρχιτεκτονική. Αντιθέτως εάν το σπίτι βρίσκεται στην πόλη τότε προτιμάει την μοντέρνα αρχιτεκτονική.

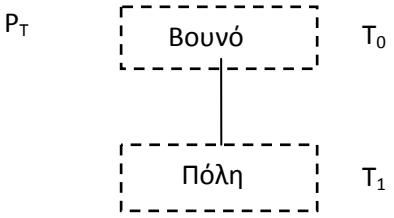
Στο πιο κάτω σχεδιάγραμμα φαίνεται το CP-Net που αναπαριστά αυτό το απλό παράδειγμα. Για ευκολία αναπαράστασης, συμβολίζω το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ με Τ, και το χαρακτηριστικό ΑΡΧΙΤΕΚΤΟΝΙΚΗ με Α.



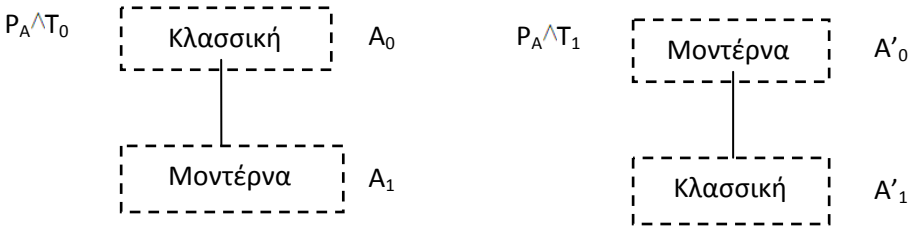
Η ακολουθία των μπλοκ που χαρακτηρίζει κάθε V(P,Ai) μπορεί να εξαχθεί απλά από την αναπαράσταση στο CP-Net, παίρνοντας τα δεδομένα από τους πίνακες κάθε χαρακτηριστικού.

Σε αυτό το σημείο εντοπίζουμε μια άλλη βασική διαφορά ανάμεσα στις προτιμήσεις του άρθρου και στις προτιμήσεις ceteris paribus με συνθήκες: την ύπαρξη πολλαπλών πιθανών ακολουθιών μπλοκ για κάθε V(P,Ai).

Αρχικά έχουμε την ακολουθία μπλοκ για το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ και την προτίμηση P_T που δεν διαφέρει σε κάτι από τις ακολουθίες μπλοκ που μελετήσαμε στο άρθρο:



Ακολούθως χρειάζεται να κατασκευαστεί η ακολουθία μπλοκ για το χαρακτηριστικό ΑΡΧΙΤΕΚΤΟΝΙΚΗ και την προτίμηση P_A. Εδώ διαπιστώνουμε ότι τα δεδομένα αλλάζουν. Η προτίμηση του χρήστη δεν είναι πλέον «μοναδική» αλλά εξαρτάται από την επιλογή της τιμής για το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ. Έτσι με βάση τον πίνακα στο CP-Net του χαρακτηριστικού ΑΡΧΙΤΕΚΤΟΝΙΚΗ, μπορούμε να φτιάξουμε δύο πιθανές ακολουθίες μπλοκ:

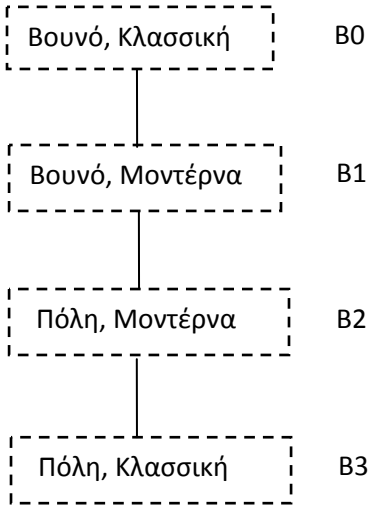


Εδώ λοιπόν ζητούνται αποτελέσματα για την σχέση προτιμήσεων P_{TA}, η οποία παράγεται από τις σχέσεις P_A και P_T. Να θυμηθούμε ότι θεωρούμε ότι τα αποτελέσματα θα είναι σε μορφή ακολουθίας μπλοκ όπως ακριβώς και στο άρθρο. Μέχρι στιγμής έχουμε εξάγει τις ακολουθίες μπλοκ για τα V(P,Ai) κάθε εμπλεκόμενου χαρακτηριστικού, παίρνοντας τα δεδομένα για την δημιουργία τους από τους αντίστοιχους πίνακες του CP-Net.

Το επόμενο βήμα είναι να δημιουργηθεί το query lattice για την σχέση προτιμήσεων μας, συνδυάζοντας τις τιμές των δύο χαρακτηριστικών. Δεδομένου ότι $P_A \prec P_T$, θα

χρησιμοποιηθεί το θεώρημα 2 του άρθρου. Παρόλο που για το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ έχουμε μια ακολουθία μπλοκ την Τ0←Τ1 με n=2 όπως ορίζεται στο θεώρημα, για το χαρακτηριστικό ΑΡΧΙΤΕΚΤΟΝΙΚΗ έχουμε δύο πιθανές ακολουθίες. Την Α0←Α1 και την Α'0←Α'1 ίσου μεγέθους, m=2.

Στο πρόβλημα αυτό η λύση είναι ο διαχωρισμός περιπτώσεων. Η πρώτη ακολουθία είναι για την περίπτωση που θα εφαρμόζεται το T_0 στο χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ $P_A \wedge T_0$ και η δεύτερη στην περίπτωση που θα χρησιμοποιείται το T_1. Εφαρμόζοντας αυτή την μέθοδο, όντως θα πάρουμε n*m μπλοκ στην ακολουθία, και κάθε μπλοκ Z_P θα έχει στοιχεία μόνο από τα μπλοκ X_q και Y_r τέτοια ώστε p=q*m+r. Δηλαδή θα έχουμε την πιο κάτω ακολουθία:

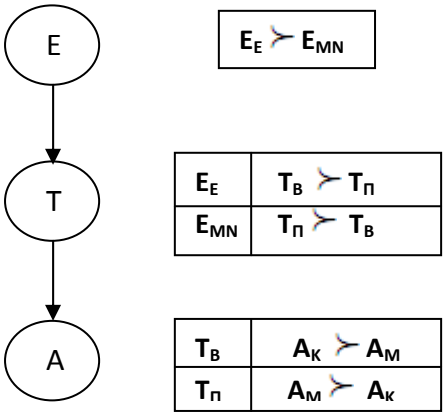


Έτσι βλέπουμε ότι με μια μικρή αλλαγή του τρόπου δημιουργίας του query lattice δημιουργώντας πολλαπλές ακολουθίες μπλοκ για χαρακτηριστικά που εξαρτούνται από άλλα χαρακτηριστικά, και με χρήση της κατάλληλης ακολουθίας κάθε φορά, έχουμε φτάσει σε επιθυμητό αποτέλεσμα.

Ας προσθέσουμε όμως λίγη πολυπλοκότητα στο παράδειγμα, χρησιμοποιώντας ακόμη ένα παράδειγμα από το κεφάλαιο 2:

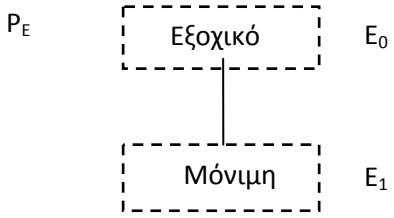
Στο προηγούμενο παράδειγμα προσθέτουμε ένα επιπλέον χαρακτηριστικό, το ΕΙΔΟΣ του σπιτιού. Υποθέτουμε ότι ο χρήστης προτιμάει (P_E) να αγοράσει ένα σπίτι που θα το χρησιμοποιεί ως «εξοχικό» παρά ένα σπίτι που θα το χρησιμοποιεί ως μόνιμη κατοικία. Στην περίπτωση όμως που θα αγοράσει τελικά ένα σπίτι που θα το χρησιμοποιεί ως μόνιμη κατοικία, τότε προτιμάει αυστηρά το σπίτι αυτό να βρίσκεται στην πόλη. Από την άλλη, αν τελικά θα αγοράσει ένα σπίτι που θα το χρησιμοποιεί ως εξοχικό προτιμάει η τοποθεσία του να είναι στο βουνό(P_T).

Στο πιο κάτω σχεδιάγραμμα φαίνεται το CP-Net που αναπαριστά αυτό το παράδειγμα. Για ευκολία αναπαράστασης, συμβολίζω και πάλι το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ με Τ, το χαρακτηριστικό ΑΡΧΙΤΕΚΤΟΝΙΚΗ με Α και το χαρακτηριστικό ΕΙΔΟΣ με Ε.



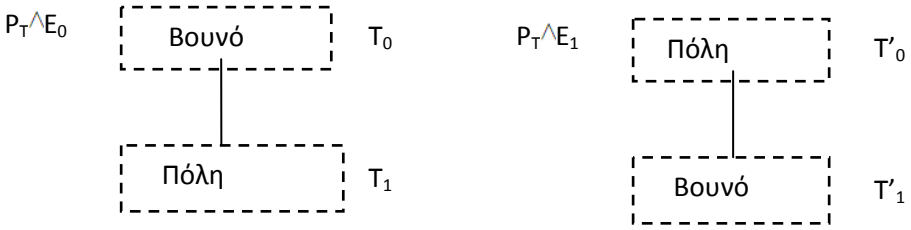
Θα κτίσουμε και πάλι την ακολουθία των μπλοκ που χαρακτηρίζει κάθε V(P,Ai), παίρνοντας τα δεδομένα από τους πίνακες κάθε χαρακτηριστικού.

Αρχικά έχουμε την ακολουθία μπλοκ για το χαρακτηριστικό ΕΙΔΟΣ και την προτίμηση P_E :

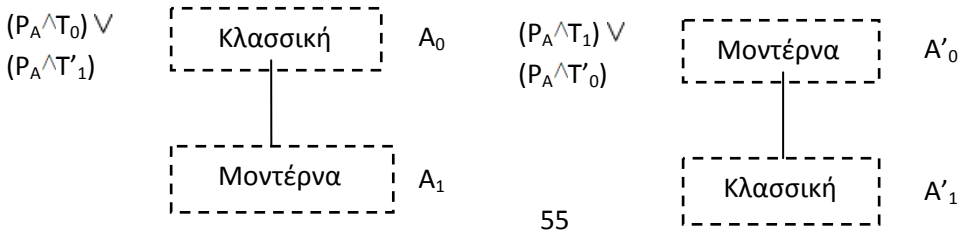


Ακολούθως κατασκευάζονται οι ακολουθίες μπλοκ για τα χαρακτηριστικά ΑΡΧΙΤΕΚΤΟΝΙΚΗ και την προτίμηση P_A και τοποθεσία και την προτίμηση P_T με τον ίδιο τρόπο όπως και στο προηγούμενο παράδειγμα.

ΤΟΠΟΘΕΣΙΑ

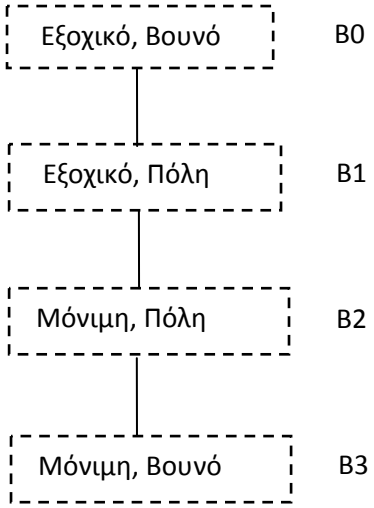


ΑΡΧΙΤΕΚΤΟΝΙΚΗ

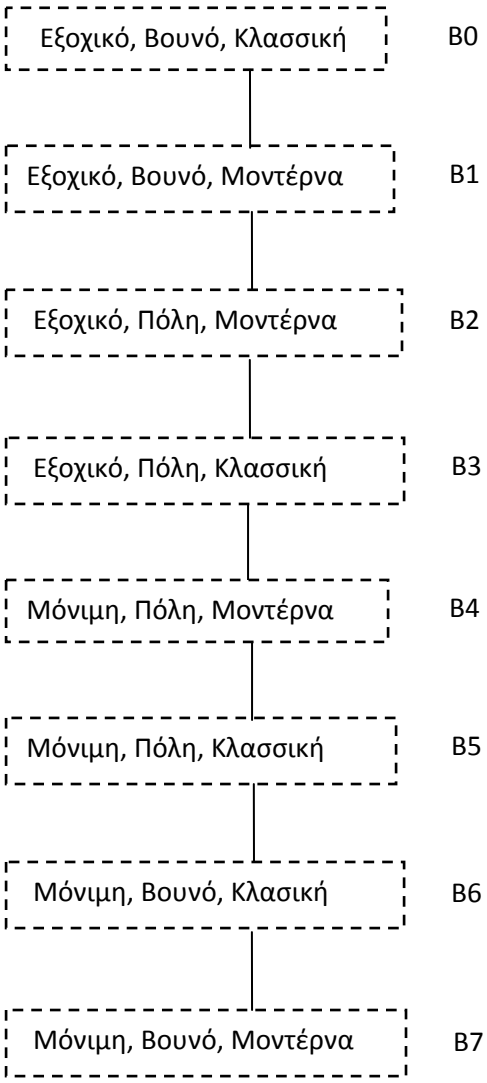


Εδώ παρατηρούμε ακόμη μια ενίσχυση στην αλλαγή που θέσαμε για διαχωρισμό των πολλαπλών ακολουθιών μπλοκ. Η προτίμηση για το χαρακτηριστικό ΑΡΧΙΤΕΚΤΟΝΙΚΗ εξαρτάται από την προτίμηση για το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ, στο οποίο ήδη υπήρξε διαχωρισμός ανάμεσα στις δύο ακολουθίες μπλοκ. Παρόλο που το χαρακτηριστικό ΤΟΠΟΘΕΣΙΑ μπορεί να πάρει μόνο δύο πιθανές τιμές, λόγω των δύο ακολουθιών μπλοκ που έχει, κάθε χαρακτηριστικό εμφανίζεται σε δύο διαφορετικές θέσεις. Έτσι τώρα ο διαχωρισμός των ακολουθιών για την ΑΡΧΙΤΕΚΤΟΝΙΚΗ, πρέπει να βασιστεί και στις τέσσερις διαφορετικές «θέσεις» του χαρακτηριστικού ΤΟΠΟΘΕΣΙΑ.

Τώρα πρέπει να υπολογιστεί το query lattice για την σχέση προτιμήσεων P_{ETA}, η οποία παράγεται από τις σχέσεις P_E, P_A και P_T. Δεδομένου και πάλι ότι $P_A \prec P_T \prec P_E$, θα χρησιμοποιηθεί το θεώρημα 2 του άρθρου δύο συνεχόμενες φορές. Εάν εφαρμοστεί, με βάση τον τρόπο που ορίζεται, θα πρέπει αρχικά να παράξει το query lattice για την σχέση P_{ET}, με τις ακολουθίες να έχουν μήκος m=2 και n=2 και άρα η ακολουθία θα έχει 4 μπλοκ. και θα είναι η ακόλουθη:



Ακολούθως πρέπει να δημιουργηθεί το query lattice για την σχέση P_{ETA}, συνδυάζοντας το query lattice που εξάχθηκε από την P_{ET} μαζί με την ακολουθία μπλοκ της P_A. Δεδομένου ότι n=4 και m=2 πρέπει να παραχθεί μια ακολουθία μήκους 8, η οποία θα είναι όπως φαίνεται πιο κάτω:



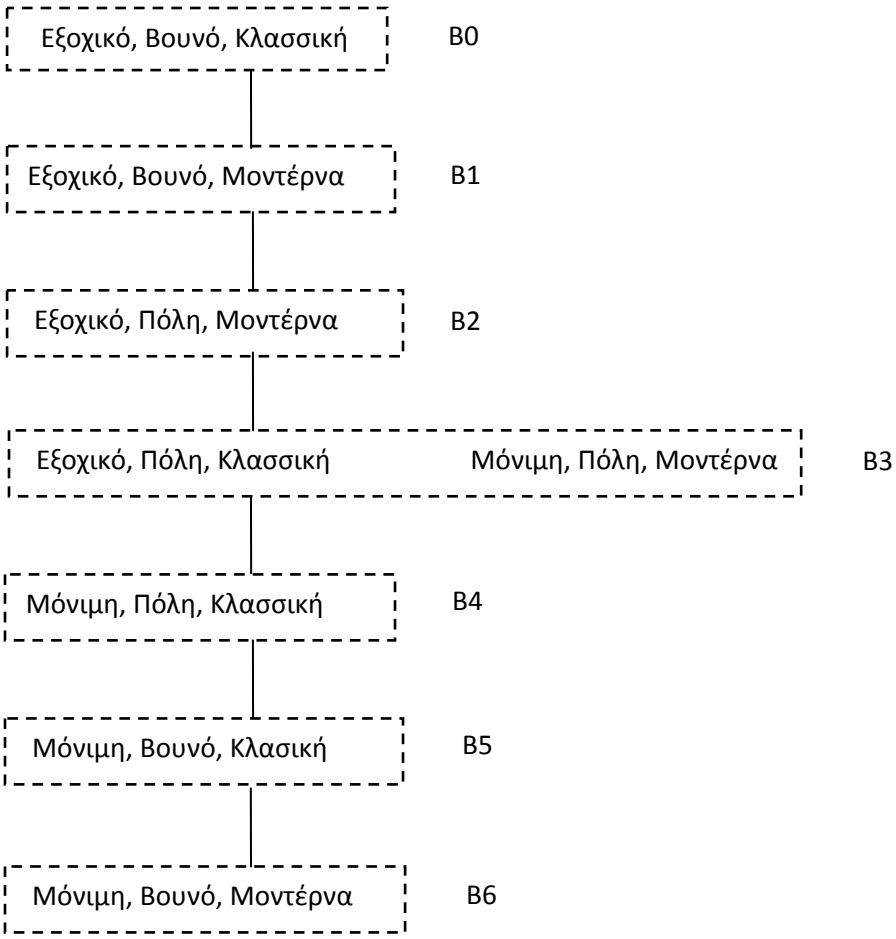
Σε αυτό το σημείο παρατηρούμε ένα άλλο σημαντικό πρόβλημα που υπάρχει. Τα B3 και B4 μπλοκ, δεν επιτρέπεται να είναι ακολουθιακά και κατά συνέπεια συγκρίσιμα μεταξύ τους, στην σημασιολογία ceteris paribus, διότι έχουν πάνω από μια διαφορετικές τιμές σε χαρακτηριστικά. Και τα δύο όμως είναι συγκρίσιμα με το B2 αλλά και το B5. Έτσι αυτά τα δύο θα πρέπει να βρίσκονται σε παράλληλη θέση μεταξύ τους, δηλαδή στο ίδιο μπλοκ.

Για να λυθεί αυτό το ζήτημα προσθέτουμε άλλη μια αλλαγή στην μεθοδολογία κατασκευής του query lattice. Κάθε φορά που υπολογίζεται το περιεχόμενο για το επόμενο μπλοκ, πρέπει να γίνεται ο ακόλουθος έλεγχος:

> Σύγκρινε το περιεχόμενο του υπό-επεξεργασία μπλοκ, με το περιεχόμενο του ακριβώς προηγούμενου μπλοκ:
>
> - Εάν έχουν διαφορετικές τιμές σε περισσότερο από ένα χαρακτηριστικά, τότε βάλε το κομμάτι αυτό, στο ήδη υπάρχων μπλοκ με το οποίο έγινε η σύγκριση

- Διαφορετικά, δημιούργησε ένα νέο μπλοκ και βάλε το μέσα

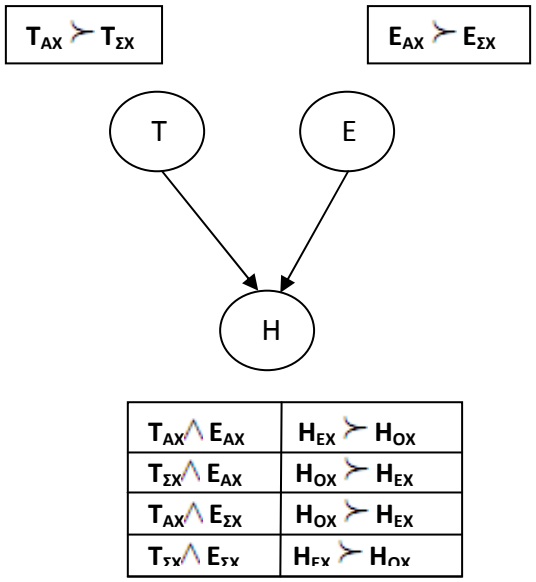
Έτσι τώρα η ακολουθία διαφοροποιείται ως εξής (επιθυμητή):



(2) Σχέση προτιμήσεων που συνδυάζει προτιμήσεις πάνω σε χαρακτηριστικά, που κάποιες είναι η μια πιο μεγάλης σημαντικότητας από τη άλλη, και κάποιες άλλες είναι ίσης σημαντικότητας. Ας χρησιμοποιήσουμε ξανά ένα παράδειγμα από το κεφάλαιο 2:

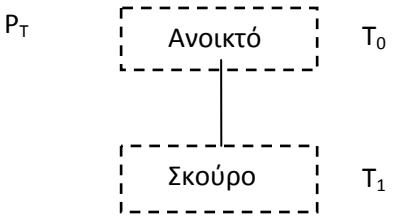
Ας υποθέσουμε ότι έχουμε ένα σύνολο εκβάσεων, που αποτελείται από την εσωτερική διακόσμηση ενός σπιτιού. Τα χαρακτηριστικά σε αυτή την περίπτωση είναι ΤΟΙΧΟΙ , ΕΠΙΠΛΑ και ΗΛΕΚΤΡΙΚΕΣ_ΣΥΣΚΕΥΕΣ. Ο χρήστης σε αυτό το παράδειγμα, προτιμά να έχει στο σπίτι του τοίχους και έπιπλα σε ανοικτό χρώμα παρά σε σκούρο χρώμα. Η επιλογή του για τις οικιακές συσκευές εξαρτάται από τον συνδυασμό των χρωμάτων στους τοίχους και τα έπιπλα. Εάν έχουν και τα δύο είτε σκούρο χρώμα είτε ανοικτό χρώμα, τότε στις ηλεκτρικές συσκευές προτιμάει το έντονο χρώμα ώστε να σπάζει η μονοτονία των υπόλοιπων αποχρώσεων. Εάν το ένα από τα δύο είναι σε ανοικτό χρώμα και το άλλο σε σκούρο χρώμα, τότε προτιμάει οι ηλεκτρικές του συσκευές να είναι σε ουδέτερο χρώμα ώστε να μην δημιουργείται δυσαρμονία.

Στο πιο κάτω σχεδιάγραμμα φαίνεται το CP-Net που αναπαριστά αυτό το παράδειγμα. Για ευκολία αναπαράστασης, συμβολίζω το χαρακτηριστικό ΤΟΙΧΟΙ με Τ, το χαρακτηριστικό ΕΠΙΠΛΑ με Ε και το χαρακτηριστικό ΗΛΕΚΤΡΙΚΕΣ_ΣΥΣΚΕΥΕΣ με Η.

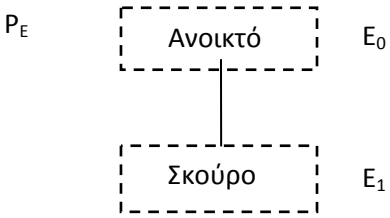


Ακολουθώντας και πάλι την διαδικασία θα κτίσουμε αρχικά την ακολουθία των μπλοκ που χαρακτηρίζει κάθε V(P,Ai), παίρνοντας τα δεδομένα από τους πίνακες κάθε χαρακτηριστικού από το CP-Net.

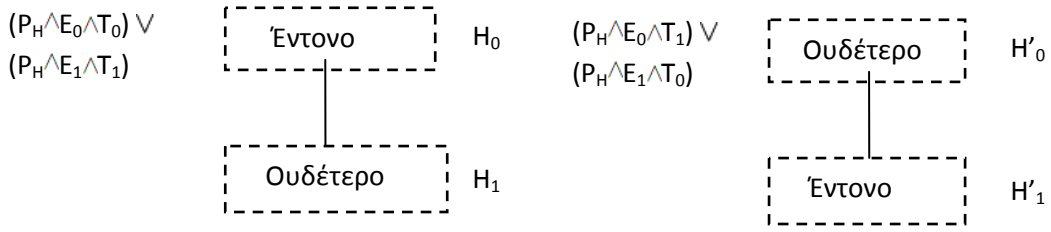
Αρχικά έχουμε την ακολουθία μπλοκ για το χαρακτηριστικό ΤΟΙΧΟΙ και την προτίμηση P_T :



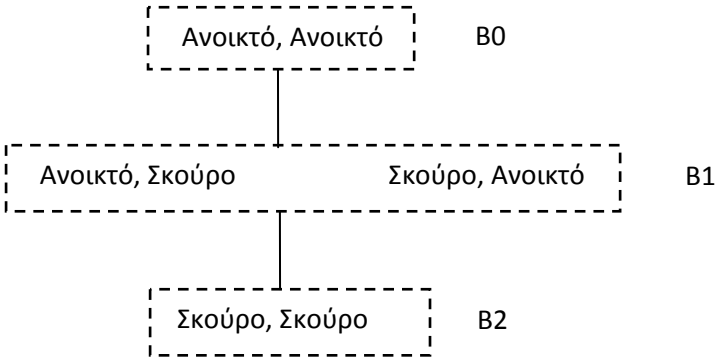
και την ακολουθία μπλοκ για το χαρακτηριστικό ΕΠΙΠΛΑ και την προτίμηση P_E:



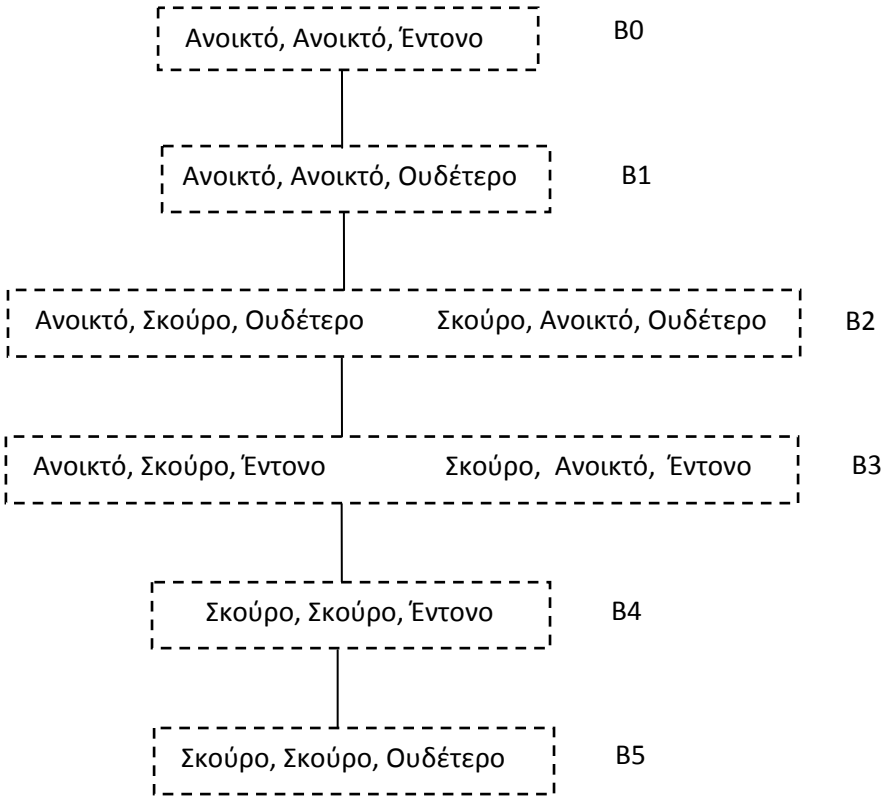
Ακολούθως κατασκευάζεται η ακολουθία μπλοκ για το χαρακτηριστικό ΗΛΕΚΤΡΙΚΕΣ_ΣΥΣΚΕΥΕΣ και την προτίμηση P_H με τον ίδιο τρόπο όπως και στα προηγούμενο παραδείγματα, μόνο που τώρα θα πρέπει κάθε ακολουθία μπλοκ να συσχετίζεται με ένα συνδυασμό των τιμών για το χαρακτηριστικό ΤΟΙΧΟΣ και το χαρακτηριστικό ΗΛΕΚΤΡΙΚΕΣ_ΣΥΣΚΕΥΕΣ.



Τώρα πρέπει να υπολογιστεί το query lattice για την σχέση προτιμήσεων P_{TEH}, η οποία παράγεται από τις σχέσεις P_T, P_E και P_H. Δεδομένου τώρα ότι $P_T \prec (P_T \approx P_E)$, θα χρησιμοποιηθούν και τα δύο θεωρήματα του άρθρου διαδοχικά. Εάν εφαρμοστούν, με βάση τον τρόπο που ορίζεται στο άρθρο, θα πρέπει αρχικά να παραχθεί το query lattice για την σχέση P_{TE}, με τις ακολουθίες να έχουν μήκος m=2 και n=2 και με χρήση του θεωρήματος 1 αφού $P_T \approx P_E$. Άρα η ακολουθία θα έχει 3 μπλοκ και θα είναι η ακόλουθη:



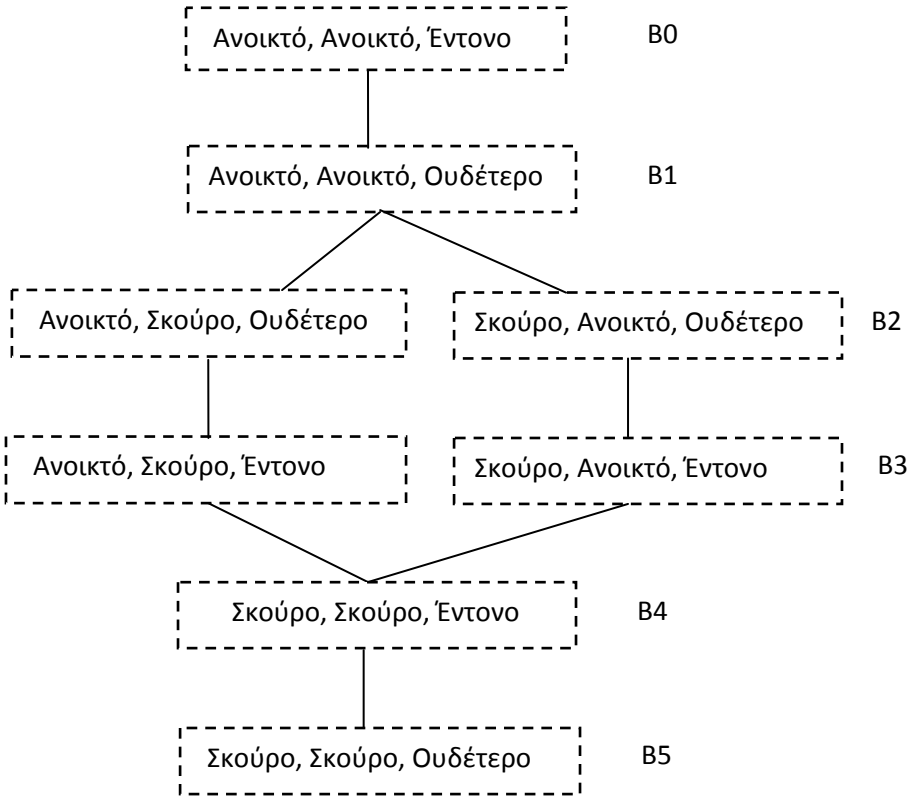
Ακολούθως πρέπει να δημιουργηθεί το query lattice για την σχέση P_{TEH}, συνδυάζοντας το query lattice που εξάχθηκε από την P_{TE} μαζί με την ακολουθία μπλοκ της P_H. Τώρα πρέπει να χρησιμοποιηθεί το θεώρημα 2 δεδομένου ότι $P_T \prec (P_T \approx P_E)$. Τα μεγέθη των ακολουθιών είναι n=3 και m=2 οπότε πρέπει να παραχθεί μια ακολουθία μήκους 6, η οποία θα είναι όπως φαίνεται πιο κάτω:



Παρατηρούμε στην πιο πάνω ακολουθία ακόμη ένα σημαντικό πρόβλημα, που αφορά την σημασιολογία ceteris paribus. Στα μπλοκ B2 και B3, υπάρχουν τέσσερις τιμές, οι οποίες αν και σε συγκεκριμένα ζεύγη είναι συγκρίσιμες με βάση την σημασιολογία, αν διασταυρώσουμε τα ζευγάρια βλέπουμε ότι υπάρχουν διαφορετικές τιμές σε περισσότερο από ένα χαρακτηριστικά. Για παράδειγμα η τιμή «Σκούρο, Ανοικτό, Ουδέτερο», δεν μπορεί να συγκριθεί με την τιμή «Ανοικτό, Σκούρο, Έντονο», αλλά με την εφαρμογή του θεωρήματος έχουν τοποθετηθεί σε διαδοχικά μπλοκ. Επίσης παρατηρούμε ότι από το μπλοκ B1 μπορούμε να πάμε και στις δύο τιμές του B2, και αντίστοιχα για το B3 και B4.

Η λύση είναι ο διαχωρισμός των ακολουθιών μπλοκ σε εκείνο το συγκεκριμένο κομμάτι που εμφανίζεται το εμπόδιο. Πρέπει δηλαδή τα μπλοκ B2 και B3 να σπάσουν σε δύο παράλληλες ακολουθίες μπλοκ, που ξεκινούν από το B1 και καταλήγουν στο B4, και ενώνονται μεταξύ τους ανά συγκρίσιμα ζευγάρια.

Αυτό μπορεί να γίνει ως εξής: όταν βρισκόμαστε σε ενδιάμεσες τιμές, δηλαδή όχι στο καλύτερο και στο χειρότερο σενάριο, θα πρέπει να δημιουργείται μια ακολουθία για κάθε συνδυασμό αυτών των ενδιάμεσων τιμών. Δηλαδή στη περίπτωση μας, θα πρέπει να δημιουργηθεί μια ακολουθία για τις περιπτώσεις του συνδυασμού «Ανοικτό, Σκούρο» και μια ακολουθία για τις περιπτώσεις του συνδυασμού «Σκούρο, Ανοικτό». Το αποτέλεσμα της μεθοδολογίας αυτής θα είναι το ακόλουθο:


Ανοικτό, Ανοικτό, Έντονο
B0
Ανοικτό, Ανοικτό, Ουδέτερο
B1
Ανοικτό, Σκούρο, Ουδέτερο
Σκούρο, Ανοικτό, Ουδέτερο
B2
Ανοικτό, Σκούρο, Έντονο
Σκούρο, Ανοικτό, Έντονο
B3
Σκούρο, Σκούρο, Έντονο
B4
Σκούρο, Σκούρο, Ουδέτερο
B5

Κεφάλαιο 6

Ταχεία ανάθεση σκορ σε πλειάδες βάσεων δεδομένων με βάση ερωτήσεις προτιμήσεων και συμφραζόμενα

6.1 Εισαγωγή-Προσέγγιση άρθρου

Σε αυτό το κεφάλαιο παρουσιάζεται μια κριτική ανάλυση της εργασίας [3]. Στόχος είναι η παροχή στους χρήστες μόνο τα σχετικά με τις προτιμήσεις τους δεδομένα από τον συνολικό όγκο διαθέσιμων πληροφοριών, και να τους επιτραπεί να εκφράσουν το ενδιαφέρον τους, το οποίο μπορεί να ποικίλει ανάλογα με την κατάσταση του χρήστη τη συγκεκριμένη στιγμή.

Οι προτιμήσεις των χρηστών μπορεί να διαφέρουν ανάλογα με το περιβάλλον (context). Με τον όρο "συμφραζόμενα" εκφράζουμε την κατάσταση στην οποία βρίσκεται ο χρήστης την στιγμή που υποβάλει το ερώτημα (τοποθεσία, ώρα, άτομα που τον περιβάλλουν). Εκφράζουμε ουσιαστικά οποιοδήποτε χαρακτηριστικό που δεν είναι μέρος του σχήματος της βάσης δεδομένων μας.

Δοθέντος ενός συνόλου "συμφραζομένωνual" προτιμήσεων P, ενός στιγμιότυπου βάσης δεδομένων r, και ενός ερωτήματος q, επιδιώκουν στο να δώσουν στον χρήστη τις προτιμότερες πλειάδες από το r με βάση το τρέχων "συμφραζόμενο", χωρίς να χρειάζεται για κάθε ερώτημα και κάθε "συμφραζόμενο" να παίρνουν σκορ και να ταξινομούνται όλες οι πλειάδες της βάσης δεδομένων (αχρείαστη σπατάλη πόρων και καθυστέρηση).

Σε αυτό το άρθρο:

1) Προτείνεται μια ακολουθία τεχνικών για γρήγορη παροχή των χρηστών με δεδομένα που τους ενδιαφέρουν στην περίπτωση "συμφραζομένωνual" quantitative προτιμήσεων.
2) Εξετάζεται μια "συμφραζομένηual" μέθοδος συγκέντρωσης που εκμεταλλεύεται την ιεραρχική φύση των "συμφραζομένων" χαρακτηριστικών.
3) Γίνεται εισαγωγή σε μια μέθοδο ομαδοποίησης των προτιμήσεων που παράγουν παρόμοια σκορ για όλες τις πλειάδες της βάσης δεδομένων. Η μέθοδος βασίζεται σε bitmap αναπαράσταση.

6.2 Μοντέλο προτιμήσεων με βάση τα συμφραζόμενα:

Για την μοντελοποίηση του "συμφραζομένου" γίνεται χρήση ενός πεπερασμένου συνόλου χαρακτηριστικών που ονομάζεται *"παράμετροι συμφραζομένων",* τα οποία διαχωρίζονται σε δύο τύπους: απλές που περιλαμβάνουν ένα μονό "συμφραζομένο" χαρακτηριστικό C_i με πεδίο τιμών $dom(C_i)$, και σύνθετες που αποτελούνται από ένα σύνολο μονών "συμφραζομένων" χαρακτηριστικών $C_{j1},C_{j2},...,C_{ji}$ με πεδία ορισμού $dom(C_{j1}),dom(C_{j2}),...,dom(C_{ji})$ αντίστοιχα, και το πεδίο ορισμού της σύνθετης παραμέτρου C_j ορίζεται ως $dom(C_j)=dom(C_{j1}) \times dom(C_{j2}) \times ... \times dom(C_{ji})$.

Για μια εφαρμογή καθορίζουμε το "συμφραζομένο" περιβάλλον της CE_x ως ένα σύνολο n "συμφραζομένων" παραμέτρων $\{C_1,C_2,..,C_n\}$, είτε απλών είτε σύνθετων. Μέσα στο "συμφραζομένο" ίσως να συμπεριλαμβάνεται και ο χρήστης.

Υποθέτουμε ότι κάθε "συμφραζομένο" χαρακτηριστικό συμμετέχει σε μια ιεραρχία χαρακτηριστικών. Μια ιεραρχία χαρακτηριστικών είναι της μορφής $(L, \prec)$: $L=(L_1,...,L_{m-1},ALL)$ με m επίπεδα και το $\prec$ είναι μια μερική διάταξη ανάμεσα στα επίπεδα της L τέτοια ώστε $L_1 \prec L_i \prec ALL$ για κάθε $1<i<m$. Το πιο πάνω επίπεδο της L πρέπει να είναι πάντα το επίπεδο ALL, έτσι ώστε να μπορούν να ομαδοποιηθούν όλες οι τιμές σε μια και μόνο τιμή, την ALL.

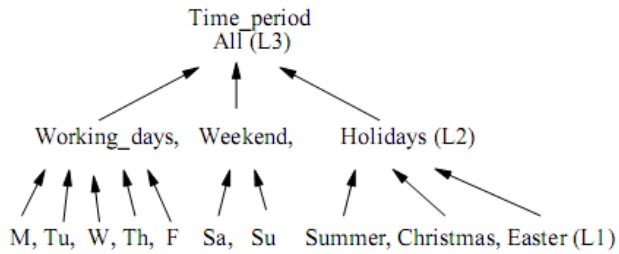
Μια "συμφραζομένη" κατάσταση (cs) είναι μια n-πλειάδα της μορφής $(c_1,c_2,...,c_n)$, όπου $c_i \in dom(C_i)$, δηλαδή μια ανάθεση τιμών στα "συμφραζομένα" χαρακτηριστικά.

Ορισμός: Δεδομένου ενός σχήματος βάσης δεδομένων $R(A_1,A_2,...,A_d)$, μια "συμφραζομένη" προτίμηση p στο R είναι μια τριάδα (cs,Pred,score), όπου το cs είναι μια "συμφραζομένη" κατάσταση, Pred είναι ένα κατηγόρημα της μορφής $A_{i1}\Theta_1 a_{i1} \wedge A_{i2}\Theta_2 a_{i1} \wedge ... \wedge A_{ik}\Theta_\kappa a_{ik}$ που καθορίζει συνθήκες Θ_i (πχ =, <, > , ≤ , ≥, ≠, κλπ) πάνω στις τιμές $a_{ij} \in dom(A_{ij})$ των χαρακτηριστικών A_{ij}, $1 \leq i_j \leq d$, του σχήματος της βάσης δεδομένων, και το score είναι ένας πραγματικός αριθμός μεταξύ 0 και 1.

Μια "συμφραζόμενη" προτίμηση δεν πρέπει οπωσδήποτε να εξαρτάται από όλα τα "συμφραζόμενα" χαρακτηριστικά.

Το σύνολο όλων των "συμφραζομένων" προτιμήσεων για μια εφαρμογή ονομάζεται προφίλ P. Με τον όρο CS(P) εννοείται το σύνολο των "συμφραζομένων" καταστάσεων που εμφανίζονται σε τουλάχιστον μια προτίμηση στο P.

Παράδειγμα Ιεραρχίας χαρακτηριστικών:



6.3 Υπολογισμός Σκορ Ενδιαφέροντος

Δοθέντος ενός συνόλου προτιμήσεων που καθορίζονται σε ένα προφίλ P, γίνεται συσχέτιση ενός σκορ ενδιαφέροντος για κάθε πλειάδα t κάθε στιγμιότυπου r της βάσης δεδομένων R, για κάθε "συμφραζόμενη" κατάσταση cs ϵ CS(P).

Εάν καμία από τις προτιμήσεις που καθορίζονται για την "συμφραζόμενη" κατάσταση cs δεν εφαρμόζεται στην πλειάδα t, δηλαδή δεν υπάρχει πλειάδα που να αντιστοιχεί με το κατηγόρημα, τότε στην t ανατίθεται το σκορ 0.

Μπορεί να υπάρχουν περισσότερες από μια προτιμήσεις που να εφαρμόζονται σε μια συγκεκριμένη πλειάδα της βάσης δεδομένων. Σε τέτοια περίπτωση επιλέγουμε εκείνη με το υψηλότερο σκορ.

Ορισμός: Εάν το P είναι ένα προφίλ, cs μια "συμφραζόμενη" κατάσταση και t ϵ r μια πλειάδα. Εάν το P' $\subseteq$ P είναι το σύνολο των προτιμήσεων p_i=(cs,$Pred_i$,score), έτσι ώστε το $Pred_i$[t] να ισχύει και $\neg \exists$ p_j=(cs,$Pred_j$,score) ϵ P' έτσι ώστε $Pred_j$ να περικλείει οποιοδήποτε από τα $Pred_i$, τότε το σκορ της t στην cs (score(t,cs)) καθορίζεται ως ακολούθως:

$$score(t,cs) = \begin{cases} \max_{p_i \epsilon P'} score_i, & \text{αν } P' \neq 0 \\ 0, & \text{διαφορετικά} \end{cases}$$

6.4 Σχηματισμός προβλημάτων:

Κάθε ερώτημα που δίδεται από κάποιο χρήστη συνδέεται με μια ή περισσότερες "συμφραζόμενες" καταστάσεις. Το "συμφραζόμενο" που συνδέεται με το εκάστοτε ερώτημα συνήθως αντιστοιχεί με το τρέχων "συμφραζόμενο" του χρήστη. Τα ερωτήματα μπορεί να δίδονται «εμποτισμένα» με τις "συμφραζόμενες" καταστάσεις του χρήστη.

Δοθέντος των "συμφραζόμενων" καταστάσεων CS_q και ενός ερωτήματος q, στόχος είναι:
1) Η αναγνώριση του συνόλου P_q $\subseteq$ P των προτιμήσεων (cs,Pred,score) για τις οποίες

$cs=cs_q$, για κάποιο $cs_q \epsilon CS_q$ και έπειτα,

2) Η χρήση τους για υπολογισμό του σκορ για κάθε πλειάδα t στο αποτέλεσμα της q.

Ένα πρόβλημα προέρχεται από το γεγονός ότι για κάποιες καταστάσεις cs_q από το CS_q, μπορεί να μην υπάρχει προτίμηση (cs,Pred,score) στο προφίλ P, για την οποία να ισχύει $cs=cs_q$. Σε τέτοιες περιπτώσεις γίνεται χρήση των προτιμήσεων του P που έχουν τις πιο όμοιες "συμφραζόμενες" καταστάσεις. Για κάθε cs_q, γίνεται χρήση της προτίμησης (cs,Pred,score) του P με $\min_{cs\epsilon CS(P)} dist_S(cs,cs_q)$.

Ορισμός: Δεδομένου ενός προφίλ P, ενός συνόλου "συμφραζόμενων" καταστάσεων $CS \subseteq CS(P)$ και μιας πλειάδας $t \epsilon r$, το αθροιστικό σκορ της t στο CS είναι : $score(t,CS)=\max_{cs\epsilon CS} score(t,cs)$.

Σε αυτό το σημείο μπορούμε να ορίσουμε το πρόβλημα της απόδοσης σκορ. Δεδομένου ενός στιγμιοτύπου βάσης δεδομένων r, ενός προφίλ P, ενός ερωτήματος q με ένα σύνολο "συμφραζόμενων" καταστάσεων CSq, και $CS \subseteq CS(P)$ το σύνολο των "συμφραζόμενων" καταστάσεων cs με το $\min_{cs\epsilon CS(P)} dist_S(cs,cs_q)$, ζητούμενο είναι η ταξινόμηση όλων των πλειάδων t στα αποτελέσματα της q βασισμένη στα σκορ score(t,CS).

Μια πιθανή λύση είναι να εντοπιστεί πρώτα το σύνολο CSq, να υπολογιστούν τα σκορ για όλες τις πλειάδες t στο αποτέλεσμα και να ταξινομηθούν με βάση τα σκορ. Μια άλλη λύση είναι να υπολογιστούν τα σκορ για κάθε πλειάδα της βάσης , για κάθε πιθανή "συμφραζόμενη" κατάσταση, κάτι που οδηγεί σε σπατάλη πόρων και αργές καθυστερήσεις στην έξοδο αποτελεσμάτων.

Μια τρίτη λύση είναι ο υπολογισμός σκορ για όλες τις καταστάσεις που εμφανίζονται στο προφίλ, και μετά να συνδυάζονται τα σκορ των πιο όμοιων την στιγμή που ζητείται. Λόγω του ότι ο αριθμός των καταστάσεων μπορεί να είναι τεράστιος, προτείνονται η πιο κάτω προσέγγιση:

1. Δεδομένων των προτιμήσεων που ισχύουν κάτω από διαφορετικές συνθήκες, γίνεται μια ομαδοποίηση με βάση:
   α. είτε το μέρος που σχετίζεται με το "συμφραζόμενο", δηλαδή δημιουργία ομάδων προτιμήσεων που εφαρμόζονται σε παρόμοιες "συμφραζόμενες" καταστάσεις,
   β. είτε το μέρος που δεν σχετίζεται με το "συμφραζόμενο", δηλαδή δημιουργία ομάδων προτιμήσεων που παράγουν παρόμοια σκορ.

2. Με χρήση των προτιμήσεων κάθε ομάδας, υπολογίζουμε ένα σκορ ενδιαφέροντος για κάθε πλειάδα για την ομάδα
3. Για ένα ερώτημα που δόθηκε από τον χρήστη, εντοπίζονται οι πιο όμοιες στο "συμφραζόμενο" του ερωτήματος ομάδες. Γίνεται χρήση των σκορ των πλειάδων των ομάδων που εντοπίζονται, και γρήγορη ταξινόμηση των αποτελεσμάτων με βάση τα υπολογισμένα σκορ.

6.5 Εντοπισμός αντιπροσωπευτικών "συμφραζόμενων" καταστάσεων:

Αντί να υπολογίζονται τα σκορ για κάθε πλειάδα για κάθε πιθανή "συμφραζόμενη" κατάσταση, μπορεί να γίνει αναγνώριση των αντιπροσωπευτικών "συμφραζόμενων" καταστάσεων και να προ-υπολογιστούν τα σκορ με βάση αυτές. Ο υπολογισμός των σκορ ενδιαφέροντος με χρήση μόνο των αντιπροσωπευτικών "συμφραζόμενων" καταστάσεων βασίζεται στην υπόθεση ότι οι προτιμήσεις που καθορίζονται για παρόμοιες "συμφραζόμενες" καταστάσεις θα έδιναν ως αποτέλεσμα την παραγωγή παρόμοιων σκορ για τις πιο πολλές πλειάδες.

6.5.1 Ομοιότητα μεταξύ καταστάσεων συμφραζομένων:

Μια απευθείας μέθοδος για υπολογισμό της απόστασης μεταξύ δύο τιμών μιας "συμφραζόμενης" παραμέτρου, είναι να συσχετιστεί η απόσταση με το μήκος του ελάχιστου μονοπατιού που τις συνδέει στην ιεραρχία τους. Αυτή η μέθοδος όμως μπορεί να μην είναι ακριβής λόγω του ότι οι τιμές στα ψηλά επίπεδα της ιεραρχίας είναι διαισθητικώς λιγότερο όμοια από ότι τιμές στα χαμηλά επίπεδα που συνδέονται με μονοπάτια του ίδιου μήκους.

Με βάση έρευνες που έγιναν για τον καθορισμό της σημασιολογικής ομοιότητας μεταξύ όρων προτείνεται η ακόλουθη μέθοδος:

Λαμβάνονται υπόψη τόσο το μήκος του μονοπατιού, όσο και το βάθος των ιεραρχικών επιπέδων στα οποία ανήκουν οι δύο τιμές.

Θεωρούμε ότι $lca(c_1,c_2)$ είναι ο χαμηλότερος πρόγονος των "συμφραζόμενων" τιμών c_1 και c_2.

Ορισμός: Η απόσταση του μονοπατιού, $dist_P(c_1,c_2)$ μεταξύ δύο "συμφραζόμενων" τιμών c_1 ϵ $dom_{Lj}(C_i)$ και c_2 ϵ $dom_{Lk}(C_i)$:

- ισούται με 0, εάν $c_1=c_2$
- ισούται με 1, εάν τα c_1 και c_2 είναι τιμές του πιο χαμηλού ιεραρχικού επιπέδου και ο $lca(c_1,c_2)$ είναι η τιμή στην ρίζα της αντίστοιχης ιεραρχίας
- ή σε άλλη περίπτωση υπολογίζεται μέσω της συνάρτησης $f_P=(1-e^{-\alpha x \rho})$, όπου το α είναι μια σταθερά (βάρος),α>0, και το ρ είναι το μήκος του ελάχιστου μονοπατιού που συνδέει τις δύο τιμές στη ιεραρχία τους

Ορισμός: Η απόσταση βάθους, $dist_D(c_1,c_2)$ μεταξύ δύο "συμφραζόμενων" τιμών c_1 ϵ $dom_{Lj}(C_i)$ και c_2 ϵ $dom_{Lk}(C_i)$:

- ισούται με 0, εάν $c_1=c_2$
- ισούται με 1, εάν ο $lca(c_1,c_2)$ είναι η τιμή στην ρίζα της αντίστοιχης ιεραρχίας
- ή σε άλλη περίπτωση υπολογίζεται μέσω της συνάρτησης $f_D=(1-e^{-\beta x \gamma})$, όπου το β είναι μια σταθερά (βάρος), β>0, και το γ είναι το μήκος του ελάχιστου μονοπατιού που συνδέει την τιμή του $lca(c_1,c_2)$ και την τιμή στην ρίζα της ιεραρχίας τους.

Ορισμός: Η τιμή της ολικής απόστασης μεταξύ δύο "συμφραζόμενων" τιμών c_1 και c_2 υπολογίζεται ως εξής

$$dist_V(c_1,c_2)= dist_P(c_1,c_2) \text{ x } dist_D(c_1,c_2)$$

Με αναθέσεις διαφόρων τιμών στα α και β, μπορούν να συνδυαστούν μεταξύ τους με διαφορετικούς τρόπους με διάφορα βάρη ενδιαφέροντος.

Ορισμός: Δοθέντων δύο "συμφραζόμενων" καταστάσεων $cs^1=(c^1_1,c^1_2,...,c^1_n)$ και $cs^2=(c^2_1,c^2_2,...,c^2_n)$ η απόσταση των καταστάσεων ορίζεται ως:

$dist_S(cs^1,cs^2)= \Sigma^n_{i=1} w_i x dist_V(c^1_i,c^{2i})$, όπου κάθε w_i είναι ένα βάρος για την συγκεκριμένη "συμφραζόμενη" παράμετρο.

6.5.2 Ομαδοποίηση με βάση τα συμφραζόμενα:

Γίνεται χρήση του d-max αλγόριθμου ο οποίος λειτουργεί ως εξής:
Αρχικά τοποθετεί κάθε "συμφραζόμενη" κατάσταση σε μια ομάδα μόνη της. Ακολούθως σε κάθε βήμα, συγχωνεύει τις δύο ομάδες με την μικρότερη απόσταση. Η απόσταση μεταξύ δύο ομάδων καθορίζεται ως η μέγιστη απόσταση μεταξύ δύο οποιονδήποτε "συμφραζόμενων" καταστάσεων που ανήκουν σε αυτές τις ομάδες. Ο αλγόριθμος τερματίζεται όταν οι δύο πιο κοντινές ομάδες έχουν απόσταση μεγαλύτερη από d_{cl}, όπου d_{cl} είναι μια παράμετρος εισόδου. Τέλος για κάθε ομάδα, επιλέγεται μια "συμφραζόμενη" κατάσταση ως αντιπρόσωπος, η οποία είναι αυτή με την μικρότερη συνολική απόσταση από όλες τις καταστάσεις μέσα στην ομάδα της.

Ορισμός: Υποθέτουμε ότι cl_i είναι μια ομάδα που παράχθηκε από τον πιο πάνω αλγόριθμο η οποία αποτελείται από ένα σύνολο CS_{cli} από m "συμφραζόμενες" καταστάσεις, cs_{ij}. Η αντιπρόσωπος της cl_i είναι η "συμφραζόμενη" κατάσταση cs ϵ CS_{cli} , με ελάχιστη τιμή $\Sigma^{m}_{j=1}$ $dist_S(cs,cs_{ij})$.

Algorithm 1 *d-max* Algorithm

Input: A set of preferences with context states cs_i, a distance value d_{cl}.
Output: A set of clusters.

Begin
1. Create a cluster for each context state cs_i.
2. Repeat.
   2.1 If the minimum distance among any pair of clusters is smaller than d_{cl}.
      2.1.1 Merge these two clusters.
   2.2 Else, end loop.
3. Compute the representative context state of each produced cluster.

End

Μετά την δημιουργία των ομάδων, γίνεται υπολογισμός για κάθε μια ομάδα ένα σκορ για κάθε πλειάδα που καθορίζεται σε κάποια από τις προτιμήσεις της. Για κάθε ομάδα cl_i, διατηρείται ένας πίνακας με σκορ, cl_iScores(tuple_id,score), στον οποίο αποθηκεύονται σε φθίνουσα σειρά μόνο τα σκορ των πλειάδων που ικανοποιούν τουλάχιστον ένα από τα κατηγορήματα των προτιμήσεων στην ομάδα.

Κάθε φορά που δίδεται ένα ερώτημα, γίνεται έρευνα για την εύρεση της πιο όμοιας ομάδας (ή πιο όμοιων ομάδων), δηλαδή της ομάδας της οποίας η αντιπρόσωπος "συμφραζόμενη" κατάσταση είναι η πιο όμοια με το "συμφραζόμενο" του ερωτήματος.

Έπειτα με χρήση του πίνακα με τα σκορ των αντίστοιχων ομάδων, μπορούν να ανακτηθούν οι πλειάδες με τα μεγαλύτερα σκορ γρήγορα.

Ιδιότητα: Έστω cs είναι μια "συμφραζόμενη" κατάσταση και CS ένα σύνολο "συμφραζόμενων" καταστάσεων. Εάν cs ϵ CS, τότε για κάθε t ϵ r, το score(t,CS)≥score(t,cs)

Αυτό σημαίνει πως το σκορ μιας πλειάδας που υπολογίζεται με βάση την αντιπρόσωπο "συμφραζόμενη" κατάσταση, δεν είναι μικρότερο από το σκορ της πλειάδας που υπολογίζεται με βάση οποιαδήποτε "συμφραζόμενη" κατάσταση που ανήκει στην ομάδα. Δηλαδή εάν η "συμφραζόμενη" κατάσταση που είναι η πιο όμοια στο "συμφραζόμενο" του ερωτήματος, ανήκει στην ομάδα της οποίας η αντιπρόσωπος "συμφραζόμενη" κατάσταση είναι η πιο όμοια με το ερώτημα, τότε το σκορ που υπολογίζεται από την προσεγγιστική προσέγγιση για μια πλειάδα δεν μπορεί να είναι χαμηλότερο από το ακριβές σκορ.

Ομαδοποίηση με βάση τα κατηγορήματα:

Μια εναλλακτική μέθοδος είναι η ομαδοποίηση μαζί "συμφραζόμενων" καταστάσεων που παράγουν παρόμοια σκορ για τις περισσότερες πλειάδες της βάσης δεδομένων.

Για να γίνει αυτό χρησιμοποιείται μια "bitmap" αναπαράσταση για τις προτιμήσεις που εφαρμόζονται σε μια "συμφραζόμενη" κατάσταση cs.

Ορισμός: Ένας πίνακας προτιμήσεων B(cs) για μια "συμφραζόμενη" κατάσταση είναι ένας "bitmap" πίνακας l x $|P|$, όπου $|P|$ είναι ο αριθμός όλων των διαφορετικών κατηγορημάτων που εμφανίζονται στο P, και l ο αριθμός όλων των διαφορετικών σκορ στο P, έτσι ώστε, B(cs)[i,j]=1, αν και μόνο αν, υπάρχει μια προτίμηση (cs,j,i) ϵ P.

Εάν οι πίνακες $B(cs_1)$ και $B(cs_2)$ δύο "συμφραζόμενων" καταστάσεων cs_1 και cs_2 είναι οι ίδιοι, τότε όλες οι πλειάδες της βάσης δεδομένων έχουν τα ίδια σκορ για τις "συμφραζόμενες" καταστάσεις cs_1 και cs_2.

Λόγω του ότι αυτοί οι πίνακες προτιμήσεων μπορεί να είναι τεράστιοι, καθορίζονται προσεγγίσεις τους ως ακολούθως.

Ορισμός: Μια αναπαράσταση κατηγορημάτων για μια "συμφραζόμενη" κατάσταση cs και σκορ s, BV(cs,s), $0 \leq s \leq 1$, είναι ένα δυαδικό διάνυσμα μεγέθους |P | τέτοιο ώστε, $BV(cs,s)[j]=BOR_{i \geq s}B(cs)[i,j]$, όπου BOR είναι ο δυαδικός OR τελεστής.

Ιδιότητα: Έστω BV(cs,s) είναι η αναπαράσταση κατηγορημάτων για την "συμφραζόμενη" κατάσταση cs και το σκορ s. Εάν BV(cs,s)[j]=1 ➔ $\forall$ t ϵ r, για το οποίο το κατηγόρημα j ισχύει , score(y,cs)≥s (είναι τουλάχιστον ίσο, αν θυμηθούμε η κάθε πλειάδα παίρνει το μέγιστο σκορ από τις προτιμήσεις που τις ταιριάζουν).

Ιδιότητα: Έστω $BV(cs_1,s)$ και $BV(cs_2,s)$ είναι οι αναπαραστάσεις κατηγορημάτων δύο "συμφραζόμενων" καταστάσεων cs_1 και cs_2 για το σκορ s:

- Εάν $\forall$ j, $BV(cs_1,s)[j]=1$ ➔ $BV(cs_2,s)[j]=1$, τότε το σύνολο των πλειάδων που έχουν σκορ μεγαλύτερο από s στην cs_2 είναι ένα υπερσύνολο του συνόλου των πλειάδων που έχουν σκορ μεγαλύτερο από s στην cs_1.
- Εάν $\forall$j, $BV(cs_1,s)[j]=1 \leftrightarrow BV(cs_2,s)[j]=1$, τότε το σύνολο των πλειάδων με σκορ μεγαλύτερα ή ίσα με s είναι τα ίδια και στις δύο "συμφραζόμενες" καταστάσεις.

Η απόσταση μεταξύ δύο δυαδικών διανυσμάτων εξαρτάται από τον αριθμό των bits στα οποία διαφέρουν.
Έστω V_1 και V_2 δύο δυαδικά διανύσματα μεγέθους m, τότε $diff= \Sigma^{m}_{i=1} |V_1(i)-V_2(i)|$.
Έστω pos ο αριθμός των bits που ισούνται με 1 και για τα δύο V_1 και V_2.

Ορισμός: Η απόσταση των δύο διανυσμάτων V_1 και V_2 μεγέθους m ισούται με

$$dist_V(V_1,V_2)= \frac{diff}{diff+pos}$$, εάν diff+pos≠0 αλλιώς 1

Εάν για δύο "συμφραζόμενες" καταστάσεις cs_1 και cs_2, $dist_V(BV(cs_1,s),BV(cs_2,s))= 0$, τότε το σύνολο των πλειάδων με σκορ μεγαλύτερο ή ίσο με το s, που συσχετίζεται με κάθε BV είναι το ίδιο.

Αντί να αποθηκεύονται "bitmap" διανυσματικές αναπαραστάσεις BV για όλα τα διαφορετικά σκορ ενδιαφέροντος s_i, δημιουργείται ένας συνολικός "bitmap" πίνακας αναπαράστασης BM με μόνο b γραμμές.

Ορισμός: Ένας συνολικός "bitmap" πίνακας αναπαράστασης κατηγορημάτων BM για μια "συμφραζόμενη" κατάσταση cs και b σκορ $s_1,s_2,...,s_b$, με $0 \leq s_1 < s_2 < ... < s_b \leq 1$, είναι ένας "bitmap" b x |P | πίνακας, όπου |P | ο αριθμός των κατηγορημάτων στο P, έτσι ώστε, $BM(cs)[i,j]=BV(cs,s_i)[j]$, $1 \leq i \leq b$, $1 \leq j \leq |P|$.

Ορισμός: Η απόσταση μεταξύ δύο συνολικών πινάκων b x |P | αναπαράστασης κατηγορημάτων, $BM(cs_1)$ και $BM(cs_2)$ δύο "συμφραζόμενων" καταστάσεων cs_1 και cs_2, καθορίζεται ως:

$$dist_{BM}(BM(cs_1), BM(cs_2)) = \frac{\Sigma^{b}_{i=1} dist_V(BV(cs_1,s_i),BV(cs_2,s_i))}{b}$$

Με χρήση των αποστάσεων μεταξύ συνολικών πινάκων κατηγορημάτων, δημιουργούνται ομάδες προτιμήσεων που έχουν παρόμοια απόδοση σκορ σε πλειάδες της βάσης δεδομένων. Για να γίνει αυτό χρησιμοποιείται και εδώ ο d-max αλγόριθμος, ο οποίος λειτουργεί ως εξής: Αρχικά τοποθετεί κάθε προτίμηση με μια συγκεκριμένη "συμφραζόμενη" κατάσταση σε μια ομάδα μόνη της. Ακολούθως σε κάθε βήμα, συγχωνεύει τις δύο ομάδες με την μικρότερη απόσταση. Η απόσταση μεταξύ δύο ομάδων καθορίζεται ως η μέγιστη απόσταση μεταξύ δύο οποιονδήποτε συνολικών πινάκων αναπαράστασης κατηγορημάτων για "συμφραζόμενες" καταστάσεις που ανήκουν σε αυτές τις ομάδες. Ο αλγόριθμος τερματίζεται όταν οι δύο πιο κοντινές ομάδες έχουν απόσταση μεγαλύτερη ή ίση με s_{cl}, όπου s_{cl} είναι μια παράμετρος εισόδου.

Μετά την δημιουργία των ομάδων, γίνεται υπολογισμός για κάθε μια ομάδα ενός σκορ για τις πλειάδες. Για κάθε ομάδα cl_i, διατηρείται ένας πίνακας με σκορ, cl_iScores(tuple_id,score), στον οποίο αποθηκεύονται σε φθίνουσα σειρά μόνο τα σκορ των πλειάδων που ικανοποιούν τουλάχιστον ένα από τα κατηγορήματα των προτιμήσεων στην ομάδα.

Κάθε φορά που δίδεται ένα ερώτημα, γίνεται έρευνα για την εύρεση της πιο όμοιας ομάδας (ή πιο όμοιων ομάδων), δηλαδή της ομάδας η οποία περιέχει την προτίμηση(προτιμήσεις) με την ίδια ή την πιο όμοια "συμφραζόμενη" κατάσταση με την "συμφραζόμενη" κατάσταση του ερωτήματος.

Διαχείριση των ταξινομήσεων ενδιαφέροντος:

Μετά την δημιουργία των ομάδων και την δημιουργία των σκορ για κάθε ομάδα, θεωρείται ότι για κάθε ομάδα τα σκορ αυτά αποθηκεύονται σε ένα «πίνακα σκορ» με δύο χαρακτηριστικά το tuple_id και το score. Για κάθε ομάδα υπάρχει ένας πίνακας σκορ, και η ταξινόμηση γίνεται με βάση το χαρακτηριστικό score.

Όταν ένα “συμφραζόμενοual ερώτημα” δίδεται από τον χρήστη, ο «πίνακας σκορ» που σχετίζεται με το “συμφραζόμενο” χρησιμοποιείται. Σε περίπτωση ομαδοποίησης με βάση το “συμφραζόμενο” ο πίνακας αυτός είναι εκείνος που αντιστοιχεί στην ομάδα της οποίας η αντιπρόσωπος “συμφραζόμενη” κατάσταση cs είναι η πιο όμοια στην cs_q. Στην περίπτωση ομαδοποίησης με βάση τα κατηγορήματα, χρησιμοποιείται ο πίνακας που αντιστοιχεί στην ομάδα που περιέχει είτε την cs_q, είτε αν δεν εμφανίζεται πουθενά στο προφίλ η cs_q, την “συμφραζόμενη” κατάσταση που είναι πιο όμοια με την cs_q.

Ο εντοπισμός του κατάλληλου «πίνακα σκορ» μπορεί να επιτευχθεί με την διατήρηση ενός επιπρόσθετου πίνακα (C_1,C_2,...,C_n,table_id), όπου Ci, $1 \leq i \leq n$, είναι ένα “συμφραζόμενο” χαρακτηριστικό και table_id είναι ο «πίνακας σκορ» που συσχετίζεται με την “συμφραζόμενη” κατάσταση.

Η επιλογή του κατάλληλου πίνακα μπορεί να γίνει πιο αποτελεσματικά με την ανάπτυξη δεικτών στα “συμφραζόμενα” χαρακτηριστικά που εμφανίζονται στο προφίλ P. Μια τέτοια δομή ονομάζεται δέντρο προφίλ.

Επίσης η φυσική αποθήκευση των προ-υπολογισμένων αποτελεσμάτων θα μπορούσε να βελτιωθεί. Αντί να υπολογίζονται τα σκορ για κάθε πλειάδα και να αποθηκεύονται σε πίνακες σκορ, θα ήταν πιο αποτελεσματικό να ομαδοποιούνται οι προτιμήσεις στο προφίλ P και να δημιουργούνται κατάλληλοι δείκτες στις πλειάδες με βάση τα κατηγορήματα που εμφανίζονται στις προτιμήσεις κάθε ομάδας. Δηλαδή θα μπορούσε για κάθε ομάδα να πάνε δείκτες σε πλειάδες που ικανοποιούν κατηγορήματα που συσχετίζονται με υψηλά σκορ. Σε αυτή την περίπτωση και πάλι πρώτα πρέπει να εντοπιστεί η κατάλληλη ομάδα με βάση το “συμφραζόμενο” του ερωτήματος, cs_q. Έπειτα θα χρησιμοποιούνται οι ανάλογοι δείκτες κατηγορημάτων για να βρεθούν οι πλειάδες με τα υψηλότερα σκορ.

Όταν πάνω από μια ομάδα χρησιμοποιείται για τον υπολογισμό των αποτελεσμάτων ενός ερωτήματος μπορεί να χρησιμοποιηθεί ένας top-k αλγόριθμος, έτσι ώστε να συνδυάσει τις ταξινομημένες λίστες που διατηρούνται στον πίνακα σκορ των σχετικών ομάδων.

Τέλος μια σκέψη είναι να εφαρμόζονται και οι δύο μέθοδοι ομαδοποίησης μαζί. Για παράδειγμα θα μπορούσε πρώτα να εφαρμοστεί η ομαδοποίηση ανά κατηγορήματα πρώτα, και ακολούθως η ομαδοποίηση κατά “συμφραζόμενα” για ομαδοποίηση των ήδη παραγόμενων ομάδων.

Κεφάλαιο 7

Εξατομίκευση ερωτήσεων σε συστήματα βάσεων δεδομένων

7.1 Εισαγωγή-Προσέγγιση άρθρου

Σε αυτό το κεφάλαιο παρουσιάζεται μια κριτική ανάλυση του άρθρου [1]. Η όλη μελέτη που παρουσιάζεται στο άρθρο βασίζεται πάνω στο σκεπτικό ότι υπάρχει ανάγκη για πρόσβαση πληροφοριών επικεντρωμένη στον χρήστη.

Προτείνεται ένα πλαίσιο εξατομίκευσης για συστήματα βάσεων δεδομένων, βασισμένο σε προφίλ χρηστών και παρουσιάζεται η αρχιτεκτονική που χρειάζεται για να υποστηριχθεί κάτι τέτοιο.

Καθορίζεται ένα μοντέλο προτιμήσεων το οποίο αναθέτει σε κάθε ατομικό στοιχείο ερωτήματος ένα βαθμό ενδιαφέροντος, και το οποίο παρέχει μηχανισμό για υπολογισμό του βαθμού ενδιαφέροντος σε οποιαδήποτε σύνθετη συνθήκη, με βάση τους βαθμούς των ατομικών που την αποτελούν. Τα αποτελέσματα των ερωτημάτων ταξινομούνται με βάση τους βαθμούς ενδιαφέροντος του συνδυασμού προτιμήσεων που ικανοποιούν.

Στην πρόταση αυτή οι προτιμήσεις των χρηστών αποθηκεύονται σε προφίλ, και όταν πρόκειται να εκτελεστεί ένα ερώτημα τότε αρχικά επιλέγονται οι σχετικές προτιμήσεις του χρήστη και στη συνέχεια ενσωματώνονται αυτές οι προτιμήσεις στα ερωτήματα των οποίων τα αποτελέσματα ζητά ο χρήστης.

Παράδειγμα 7.1.1

Παρουσιάζω ένα παράδειγμα στο οποίο θα στηρίζεται κάθε παράδειγμα στις υπόλοιπες ενότητες του άρθρου για καλύτερη κατανόηση και επεξήγηση των όσων θα παρουσιάζονται. Ας υποθέσουμε το εξής σχήμα βάσης δεδομένων για ποδοσφαιρικούς αγώνες:

TV_CHANNEL(tid, name, phone, transmission)
MATCH(mid, country, field)
PLAY(tid, mid, date)
TEAM(aid, name)
TEAM_IN_MATCH(mid, aid, home)
SPORTSCASTER(rid, name)
MATCH_SPORTSCASTER(mid, rid)
KIND(mid, kind)

Θεωρούμε ότι έχουμε δυο χρήστες, τον Γιώργο και την Μαρία που θέλουν και οι δυο να ενημερωθούν για τους ποδοσφαιρικούς αγώνες που προβάλλονται απόψε “21/4/2010”. Μια απλή διαπροσωπεία θα έδινε το πιο κάτω ερώτημα:

```
select MT.title
from MATCH MT, PLAY PL
where MT.mid=PL.mid and PL.date='21/4/2010'
```

που θα έδινε την ίδια λίστα αποτελεσμάτων και για τους δυο χρήστες.

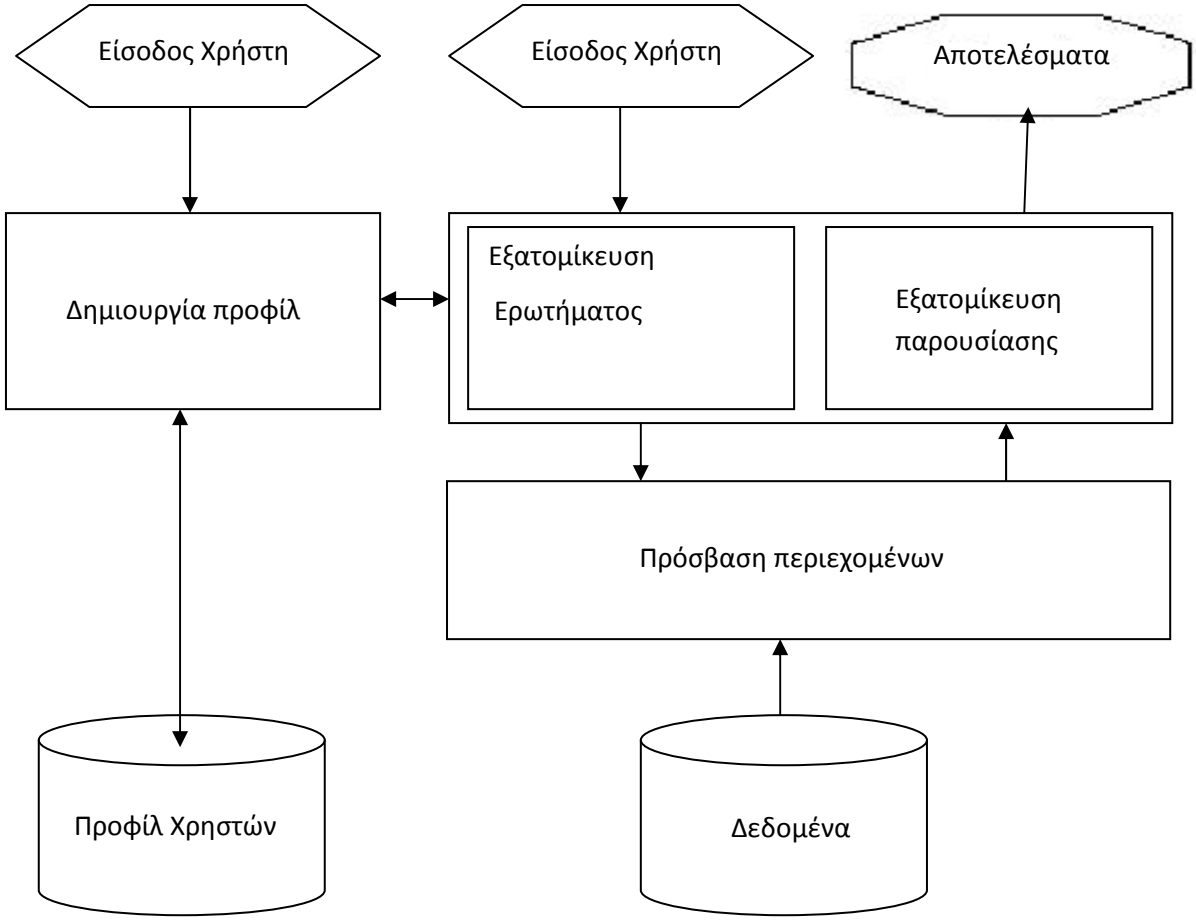
Οι δυο χρήστες όμως , ως άτομα έχουν δικές τους προτιμήσεις. Στην Μαρία αρέσουν οι αγώνες του Super League Ελλάδας και οι αγώνες της Primera Division στην Ισπανία, ενώ στον Γιώργο αρέσουν οι αγώνες του Champions League και η ομάδα Chelsea. Με μια σωστή διαχείριση θα αποθηκεύονταν οι προτιμήσεις τους αυτές σε κάποια προφίλ και θα εφαρμόζονταν στα ερωτήματα ώστε να γίνουν πιο εξατομικευμένα όπως τα ακόλουθα:

1. *select MT.title*
   from MATCH MT, PLAY PL, KIND KN
   where MT.mid=PL.mid and PL.date='21/04/2010'
   and MT.mid=KN.mid
   and (KN.kind='Super League' or KN.kind='Primera Division')

2. *select MT.title*
   from MATCH MT, PLAY PL, KIND KN, TEAM_IN_MATCH TIM, TEAM TM
   where MT.mid=PL.mid and PL.date='21/04/2010'
   and MT.mid=KN.mid
   and MT.mid=TIM.mid
   and TIM.aid=TM.aid
   and (KN.kind='Champions League' or TM.name='Chelsea')

7.2 Γενική αρχιτεκτονική ενός εξατομικευμένου συστήματος βάσης δεδομένων

Σχήμα 7.2.1



Αρχιτεκτονική προσωποποιημένου συστήματος βάσης δεδομένων

Πιο πάνω φαίνεται η γενική αρχιτεκτονική ενός προσωποποιημένου συστήματος βάσης δεδομένων. Στο κέντρο έχουμε μια παραδοσιακή ενότητα «Πρόσβασης περιεχομένων». Το σύστημα αποθηκεύει πληροφορίες για τους χρήστες («Προφίλ Χρηστών»).
Αυτές οι πληροφορίες από τα προφίλ ενσωματώνονται σε εισερχόμενα ερωτήματα τόσο κατά τη διάρκεια της «Εξατομίκευσης Ερωτήματος» όσο και κατά τη διάρκεια της «Προσωποποίηση παρουσίασης» και έτσι η ολική διαδικασία προσωποποιείτε.

Το άρθρο αυτό επικεντρώνεται στην χρήση των προφίλ χρηστών για ενσωμάτωση των προτιμήσεων στα ερωτήματα.

7.3 Μοντέλο Προτιμήσεων Χρήστη

Η μεθοδολογία του άρθρου αναλύεται με επικέντρωση σε εντολές "selection", "project" και "join" σε σχεσιακές βάσεις δεδομένων. Τα ερωτήματα που χρησιμοποιούνται είναι συνδυασμός συζεύξεων και διαζεύξεων ατομικών συνθηκών "select" και "join" , και δίνουν τα αποτελέσματα μέσω ατομικών εντολών "project" πάνω σε χαρακτηριστικά.
Το μοντέλο προτιμήσεων αναθέτει προτιμήσεις σε κομμάτια κατασκευής ερωτήματος που μπορούν να χρησιμοποιηθούν για τροποποίηση κάποιου ερωτήματος.

Παράδειγμα 7.3.1:

Η προτίμηση της Μαρίας για τον σχολιαστή "Α.Κωνσταντίνου" εκφράζεται ως προτίμηση για την συνθήκη:

SPORTSCASTER.name='Α.Κωνσταντίνου'

Οι οντότητες όμως μεταξύ τους είναι άμεσα συσχετισμένες. Αυτό σημαίνει ότι προτιμήσεις στην μια οντότητα υπονοούν και προτιμήσεις στην άλλη οντότητα όπως φαίνεται στο πιο κάτω παράδειγμα.

Παράδειγμα 7.3.2:

Αφού η Μαρία προτιμά τον σχολιαστή "Α.Κωνσταντίνου", τότε λογικό είναι να ενδιαφέρεται και για αγώνες τους οποίους σχολιάζει ο ίδιος. Αυτό εκφράζεται ως προτίμηση για την συνθήκη:

MATCH.mid=MATCH_SPORTSCASTER.mid and
MATCH_SPORTSCASTER.rid=SPORTSCASTER.rid and
SPORTSCASTER.name= 'Α.Κωνσταντίνου'

7.4 Αποθηκευμένες ατομικές προτιμήσεις χρηστών

Πρέπει λοιπόν να διατηρείται ένα προφίλ για κάθε χρήστη που αποθηκεύει τις προτιμήσεις αυτές των χρηστών σε επίπεδο ατομικών στοιχείων ενός ερωτήματος, όπως ατομικές συνθήκες εντολών "select" και "join", όπως φαίνεται στο παράδειγμα 7.3.1.

Το ενδιαφέρον ενός χρήστη προς ένα ατομικό στοιχείο εκφράζεται μέσω ενός βαθμού ενδιαφέροντος που είναι ένας πραγματικός αριθμός στο πεδίο [0,1], με την τιμή 1 να δείχνει πλήρες ενδιαφέρον και το 0 κανένα ενδιαφέρων και έτσι δεν αποθηκεύεται τέτοια πληροφορία. Ο βαθμός ενδιαφέροντος για κάθε συνθήκη υποδεικνύει κατά πόσο θα έπρεπε να ενταχθεί η συνθήκη στο τελικό ερώτημα ώστε να διαφοροποιηθούν τα αποτελέσματα.

Η αναπαράσταση των προτιμήσεων ενός συγκεκριμένου χρήστη, πάνω στα περιεχόμενα μιας βάσης δεδομένων γίνεται με ένα γράφο προσωποποίησης.

Ένας γράφος εξατομίκευσης είναι ένας κατευθυνόμενος γράφος G(V,E), όπου:

V: το σύνολο των κόμβων
E: το σύνολο των ακμών

Υπάρχουν 3 τύποι κόμβων σε ένα τέτοιο γράφο:

α) Σχεσιακοί κόμβοι: ένας για κάθε σχέση (οντότητα) στο σχήμα
β) Κόμβοι τιμών: ένας για κάθε τιμή κάθε χαρακτηριστικού κάθε σχέσης (οντότητας), όταν η τιμή ενδιαφέρει τον χρήστη
γ) Κόμβοι χαρακτηριστικών: ένας για κάθε χαρακτηριστικό σε κάθε σχέση

Υπάρχουν 2 τύποι ακμών:

α) Ακμές εντολής "selection": από ένα κόμβο χαρακτηριστικού, προς ένα κόμβο τιμής, και αντιπροσωπεύει την πιθανή συνθήκη "selection" που συνδέει το χαρακτηριστικό με την αντίστοιχη τιμή
β) Ακμές εντολής "join": από ένα κόμβο χαρακτηριστικού, προς ένα άλλο κόμβο χαρακτηριστικού, και αντιπροσωπεύει την πιθανή συνθήκη "join" μεταξύ των αντίστοιχων χαρακτηριστικών.

Δύο κόμβοι χαρακτηριστικών θα μπορούσαν να ενωθούν μεταξύ τους με δυο διαφορετικές ακμές εντολής "join", προς δυο πιθανές κατευθύνσεις. Ο βαθμός ενδιαφέροντος φαίνεται σε «ετικέτες» στις ακμές του γράφου.

Παράδειγμα 7.4.1:

Έχουμε μια πιθανή περιγραφή προτιμήσεων για τον χρήστη Μαρία:

> Προτιμά τα τηλεοπτικά κανάλια με μετάδοση "ανοικτής ζώνης". Είναι λάτρης των αγώνων της "Superleague Ελλάδος", " Primera Division Ισπανίας" και λιγότερο και των ποδοσφαιρικών αγώνων της " Premier League Αγγλίας".
> Οι αγαπημένοι της sportscaster είναι οι "Α.Κωνσταντίνου" και λιγότερο ο "Μ.Ανδρέου".
> Οι αγαπημένες τις ομάδες είναι οι "Real Madrid", λιγότερο "Παναθηναϊκός" και ακόμη λιγότερο "Barcelona". Τέλος θεωρεί τις ομάδες που λαμβάνουν μέρος στον αγώνα πιο σημαντικό στοιχείο από ότι τον sportscaster.

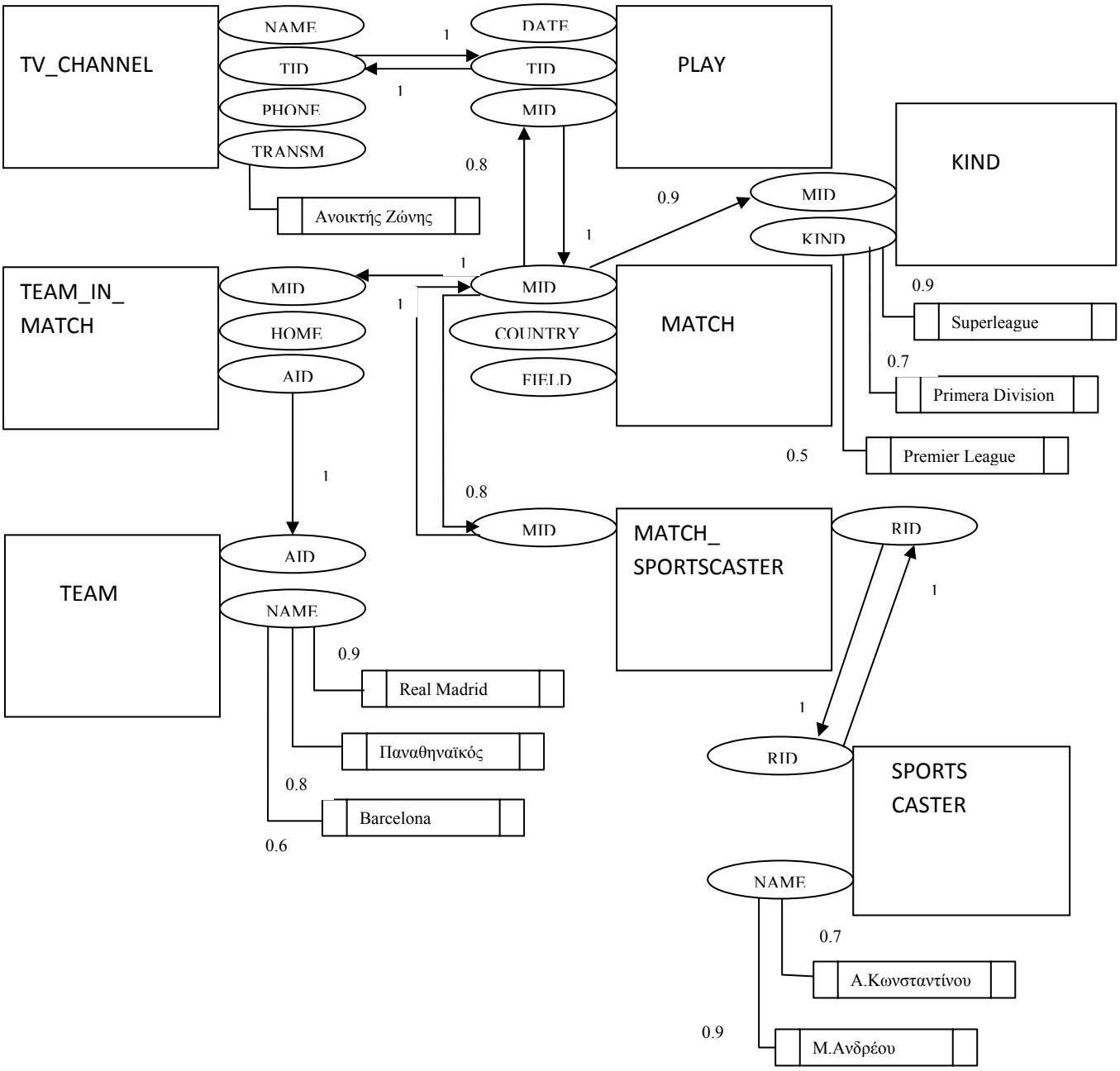
Πιο κάτω βλέπουμε ένα μέρος του προφίλ της με βάση τις προτιμήσεις της, και τους βαθμούς ενδιαφέροντος που υποθέτουμε ότι γνωρίζουμε για την Μαρία:

> [TV_CHANNEL.tid=PLAY.tid, 1]
> [PLAY.tid=TV_CHANNEL.tid, 1]
> [PLAY.mid=MATCH.mid, 1]
> [MATCH.mid=PLAY.mid, 0.8]
> [MATCH.mid=KIND.mid, 0.9]
> [TEAM.name='Παναθηναϊκός', 0.8]
> [KIND.kind='Primera Division Ισπανίας', 0.9]
> [KIND.kind='Premier League Αγγλίας', 0.7]

Το πιο πάνω αποτελεί στιγμιαία απεικόνιση του προφίλ, που μπορεί ανά πάσα στιγμή να αλλάξει.

Σχήμα 7.4.1

Στο ακόλουθο σχήμα φαίνεται ένας γράφος προσωποποίησης του προφίλ της Μαρίας με βάση τις προτιμήσεις της. Με τα μεγάλα ορθογώνια απεικονίζονται οι σχεσιακοί κόμβοι , με τις ελλείψεις οι κόμβοι χαρακτηριστικών και με τα μικρά ορθογώνια οι κόμβοι τιμών. Παρατηρούμε επίσης ότι με ετικέτες πάνω στις ακμές υποδεικνύονται οι βαθμοί ενδιαφέροντος.



7.5 Υπονοούμενες και μεταβατικές προτιμήσεις χρηστών

Εάν γίνει σύνθεση των ατομικών προτιμήσεων των χρηστών, τότε μπορούν να δημιουργηθούν μεταβατικές προτιμήσεις. Τέτοιες προτιμήσεις χαρτογραφούνται σε κατευθυνόμενα μονοπάτια ενός γράφου προσωποποίησης.

Έχουμε δυο τύπους μεταβατικών προτιμήσεων:

1) Μεταβατικό "join": χαρτογραφείται σε μονοπάτι του γράφου μεταξύ 2 κόμβων χαρακτηριστικών, το οποίο αποτελείται από ατομικές ακμές εντολής "join" και αντιπροσωπεύει την πιθανή υπονοούμενη συνθήκη εντολής "join" μεταξύ αντίστοιχων χαρακτηριστικών.

2) Μεταβατικό "select": χαρτογραφείται σε μονοπάτι του γράφου που ξεκινά από ένα κόμβο χαρακτηριστικών και καταλήγει σε ένα κόμβο τιμών και αποτελείται από n-1 ατομικές ακμές εντολής "join" και μια ακμή εντολής "select". Αντιπροσωπεύει την υπονοούμενη συνθήκη εντολής "select" μεταξύ χαρακτηριστικού και τιμής.

Αν συμβολίσουμε ως

PN: το σύνολο των N ατομικών προτιμήσεων που συμμετέχουν
DN: το σύνολο των αντίστοιχων βαθμών ενδιαφέροντος

τότε το DN ορίζεται ως:

DN: { di | di: βαθμός ενδιαφέροντος στο Pi ϵ PN , i=1..N }

Ο βαθμός ενδιαφέροντος μεταβατικών προτιμήσεων πρέπει να είναι μια συνάρτηση από τους βαθμούς ενδιαφέροντος των ατομικών προτιμήσεων που συμμετέχουν στην μεταβατική.

Για κάθε τέτοια συνάρτηση f πρέπει να ισχύει:

$f((DN)) \leq \min(DN)$, δηλαδή ο βαθμός ενδιαφέροντος να μειώνεται όσο αυξάνεται το μήκος του αντίστοιχου κατευθυνόμενου μονοπατιού.

Στο άρθρο αυτό, έχει επιλεχθεί ως συνάρτηση υπολογισμού βαθμού ενδιαφέροντος μεταβατικών προτιμήσεων ο πολλαπλασιασμός και άρα έχουμε:

$f((DN)) = d_1 d_2 ... d_N$

Παράδειγμα 7.5.1 :

Έχουμε την ακόλουθη περιγραφή προτιμήσεων για τον χρήστη Μαρία:

Της αρέσει η ομάδα "Real Madrid" που εκφράζεται ως προτίμηση για το "selection":

TEAM.name='Real Madrid' , με βαθμό ενδιαφέροντος 0.9

Της αρέσουν επίσης αγώνες που συμμετέχει η συγκεκριμένη ομάδα, κάτι που εκφράζεται ως υπονοούμενη προτίμηση ως εξής:

MATCH.mid=TEAM_IN_MATCH.mid and
TEAM_IN_MATCH.aid=TEAM.aid and
TEAM.name= 'Real Madrid',
με βαθμό ενδιαφέροντος 0.8*1*0.9=0.72 θεωρώντας ότι
MATCH.mid=TEAM_IN_MATCH.mid έχει βαθμό 0.8 και
TEAM_IN_MATCH.aid=TEAM.aid έχει βαθμό 1.

7.6 Λογικός συνδυασμός προτιμήσεων χρήστη

Έχοντας ως δεδομένα, ένα σύνολο προτιμήσεων (ατομικών και μεταβατικών), μπορούν να σχηματιστούν λογικοί συνδυασμοί, με χρήση των Boolean τελεστών 'and' και 'or', δίνοντας ως αποτέλεσμα προτιμήσεις σε μορφή συζεύξεων και σε μορφή διαζεύξεων.

Αν συμβολίσουμε ως

PN: το σύνολο των N ατομικών και μεταβατικών προτιμήσεων που συμμετέχουν
DN: το σύνολο των αντίστοιχων βαθμών ενδιαφέροντος

τότε το DN ορίζεται ως:

DN: { di | di: βαθμός ενδιαφέροντος στο Pi ϵ PN , i=1..N }

Όσο αφορά τον βαθμό ενδιαφέροντος των νέων σύνθετων συζευκτικών και διαζευκτικών προτιμήσεων που δημιουργούνται, ισχύει το ίδιο με τις μεταβατικές προτιμήσεις, δηλαδή υπολογίζεται μέσω μιας συνάρτησης.

Στο άρθρο αυτό, έχουν επιλεχθεί ως συναρτήσεις υπολογισμού βαθμού ενδιαφέροντος οι ακόλουθες:

α) $f_{\wedge}$ (DN) = 1 – (1-d1)(1-d2)...(1-dN)

β) $f_{\vee}$ (DN) = (d1+d2+...+ dN)/N

Παράδειγμα 7.6.1 :

Ας υποθέσουμε ότι έχουμε τα δυο ακόλουθα μεταβατικά “selections”:

1) *MATCH.mid=MATCH_SPORTSCASTER.mid and MATCH_SPORTSCASTER.rid=SPORTSCASTER.rid and SPORTSCASTER.name= ’Α.Κωνσταντίνου’*

2) *MATCH.mid=KIND.mid and KIND.kind= ‘Superleague Ελλάδος’*

Με βάση πληροφορίες που υποθέτουμε ότι έχουμε από το προφίλ της Μαρίας υπολογίζουμε τους βαθμούς ενδιαφέροντος:

1) Στην σύζευξη τους = 1-(1-1*1*0.7)(1-0.9*0.9)=0.943
2) Στην διάζευξη τους = (0.7+0.81)/2=0.755

7.7 Πλαίσιο προσωποποίησης των ερωτημάτων

Προσωποποίηση ενός ερωτήματος ονομάζεται η διαδικασία ενίσχυσης του, με προτιμήσεις βασισμένες στον χρήστη που είναι αποθηκευμένες σε ένα προφίλ.

Θεωρούμε ότι:

Q : το ερώτημα που χρησιμοποιείται
U: το προφίλ του χρήστη

Για την προσωποποίηση ενός ερωτήματος χρειάζεται να ορισθούν οι ακόλουθες παράμετροι:

α) Ο αριθμός K, των προτιμήσεων που εξάγονται από το προφίλ του χρήστη που πρέπει να επηρεάζουν το ερώτημα.

β) Ο αριθμός M ($0 \leq M \leq K$), των προτιμήσεων από το σύνολο των επιλεγμένων K, που πρέπει να θεωρούνται υποχρεωτικές.

γ) Ο αριθμός L ($L \leq K\text{-}M$), των K-M προτιμήσεων που απομένουν που πρέπει τουλάχιστον να συσχετίζονται με τα αποτελέσματα.

Παράδειγμα 7.7.1:

Κριτήριο για το K : οι πρώτες 5 προτιμήσεις να επηρεάζουν το ερώτημα

1) Κριτήριο για το M: προτιμήσεις με βαθμό ενδιαφέροντος=1, να θεωρούνται υποχρεωτικές
2) Κριτήριο για το L: τουλάχιστον 2 από τις K-M προτιμήσεις πρέπει επίσης να ικανοποιούνται

Η διαδικασία προσωποποίησης ενός ερωτήματος γίνεται σε 2 φάσεις:

α) Επιλογή προτιμήσεων : σε αυτή την φάση γίνεται η αναγνώριση και η εξαγωγή του συνόλου των K προτιμήσεων από το προφίλ του χρήστη που σχετίζονται με το ερώτημα.

β) Ένωση προτιμήσεων-ερωτήματος: σε αυτή την φάση γίνεται η ενσωμάτωση των προτιμήσεων στο ερώτημα, έτσι ώστε να παραχθεί ένα προσωποποιημένο ερώτημα που να δίνει αποτελέσματα που ικανοποιούν M από τις προτιμήσεις και κάποιες από τις L που απομένουν.

7.8 Επιλογή προτιμήσεων

Η πρώτη φάση της προσωποποίησης ενός ερωτήματος είναι η επιλογή των K προτιμήσεων από το προφίλ του εκάστοτε χρήστη. Μια προτίμηση εξάγεται από το προφίλ δεδομένου ότι έχει τις ακόλουθες ιδιότητες:

1) Συσχετίζεται με το ερώτημα
2) Δεν έρχεται σε σύγκρουση με το ερώτημα

Η συσχέτιση και η σύγκρουση μιας προτίμησης με ένα ερώτημα, μπορεί να είναι σε 2 διαφορετικά επίπεδα:

1) Σημασιολογικό επίπεδο:

   για να διαπιστωθεί ότι μια προτίμηση συσχετίζεται ή συγκρούεται με ένα ερώτημα σε αυτό το επίπεδο, είναι απαραίτητο να υπάρχει μια επιπλέον γνώση για τα δεδομένα, εκτός από τις πληροφορίες που εξάγονται από το σχήμα.

   Παράδειγμα 7.8.1 :

   Μια προτίμηση για τον sportscaster “Α.Κωνσταντίνου” είναι σημασιολογικά συσχετισμένη με ένα ερώτημα για αγώνες τύπου “Superleague Ελλάδος”.
   Από την άλλη μια προτίμηση για τον sportscaster “L.Carlos” είναι σημασιολογικά

συγκρουόμενη με το πιο πάνω ερώτημα, και αν συνδυάζονταν δεν θα επιστρέφονταν αποτελέσματα.

2) Συντακτικό επίπεδο:

για να διαπιστωθεί ότι μια προτίμηση σχετίζεται/συγκρούεται με ένα ερώτημα σε αυτό το επίπεδο, είναι αρκετές οι πληροφορίες από το σχήμα. Η διαπίστωση γίνεται ως εξής:

α. για συντακτικά συσχετιζόμενες προτιμήσεις:
ένα ερώτημα μπορεί να αναπαρασταθεί πάνω στον γράφο προσωποποίησης ως ένας υπό-γράφος, ο οποίος θα περιέχει όλους τους κόμβους που αντιστοιχούν σε σχέσεις (οντότητες) που συμμετέχουν στο ερώτημα και όλες τις ακμές τύπου "selection" και "join" που αντιστοιχούν στις ατομικές συνθήκες του ερωτήματος. Επίσης οι προτιμήσεις σχηματίζουν κατευθυνόμενα μονοπάτια στον γράφο προσωποποίησης.
Μια προτίμηση θεωρείται συντακτικά συσχετιζόμενη με το ερώτημα, εάν σχηματίζει ένα μονοπάτι που είναι ενωμένο με τον υπό-γράφο του ερωτήματος.

β. για συντακτικά συγκρουόμενες προτιμήσεις:
μια προτίμηση συγκρούεται με ένα ερώτημα σε αυτό το επίπεδο εάν συγκρούεται με μια συνθήκη που βρίσκεται ήδη εκεί. Δυο συνθήκες συγκρούονται σε αυτό το επίπεδο εάν αποτελούνται από μια κοινή μεταβατική συνθήκη "join" και ατομικές συνθήκες "selection" στο ίδιο χαρακτηριστικό μιας οντότητας και όλες οι συστατικές ατομικές συνθήκες "join", στην κατεύθυνση του "selection" είναι προς-ένα τύπος σχέσης.

Παράδειγμα 7.8.2:

Υποθέτουμε ότι έχουμε ένα ερώτημα με την συνθήκη

TV_CHANNEL.transmission='free zone'

η προτίμηση

TV_CHANNEL.transmission='pay zone'

δεν μπορεί να συμπεριληφθεί στο ερώτημα αφού ένα τηλεοπτικό κανάλι μπορεί να μεταδίδει μόνο σε ένα είδος ζώνης.

Παράδειγμα 7.8.3:

Έχουμε τα ακόλουθα δυο ερωτήματα:

1) *TV_CHANNEL.tid=PLAY.tid and PLAY.mid=MATCH.mid and MATCH.country='Spain'*

2) *TV_CHANNEL.tid=PLAY.tid and PLAY.mid=MATCH.mid and MATCH.country='International'*

Επιπλέον έχουμε και την συνθήκη:

TV_CHANNEL.tid=PLAY.tid and PLAY.date='2/7/2003 21:30'

η οποία δεν συγκρούεται με κάθε συνθήκη ξεχωριστά, αλλά εάν ενωθούν με σύζευξη τα δυο ερωτήματα με την πιο πάνω συνθήκη δεν θα επιστρέφονται καθόλου αποτελέσματα, γνωρίζοντας από το σχήμα ότι ένα τηλεοπτικό κανάλι προβάλει ένα μόνο αγώνα κάποια συγκεκριμένη μέρα και ώρα.

7.9 Διαμόρφωση του προβλήματος επιλογής προτιμήσεων

Θεωρούμε:

α) Ένα γράφο προσωποποίησης (Gp) για το προφίλ του χρήστη και την βάση δεδομένων που υπάρχει

β) Ένα ερώτημα (Q) που αντιπροσωπεύεται από έναν υπό-γράφο πάνω στον Gp

γ) Ένα κριτήριο ενδιαφέροντος (CI) που χρησιμοποιείται για καθορισμό των K προτιμήσεων που επιλέγονται από το προφίλ

δ) Ένα σύνολο (PN) όλων τον μονοπατιών Pi στον Gp, που σχετίζονται αλλά δεν συγκρούονται με το Q, σε φθίνουσα σειρά με βάση τον βαθμό ενδιαφέροντος τους. Αν απεικονίσουμε το σύνολο ως:

$PN=\{ Pi \mid i \epsilon [1,N], d_{i-1} \geq di\}$

τότε το σύνολο K των προτιμήσεων που πρέπει να επιλεγούν με βάση το κριτήριο ενδιαφέροντος CI είναι το ταξινομημένο υποσύνολο:

$PK=\{ Pi \mid i \epsilon [1,K], d_{i-1} \geq di \}$ του PN, ώστε:

$K= \max(\{ t \mid t \epsilon [1,N]: CI(Pt) \text{ holds } \})$

Το κριτήριο ενδιαφέροντος (CI) μπορεί να αντιστοιχεί σε διαφόρων ειδών εκφράσεις, μερικές από αυτές φαίνονται πιο κάτω:

α) $t \leq r$, επιλέγει το πολύ r προτιμήσεις

β) $dt > d$, επιλέγει προτιμήσεις με βαθμό ενδιαφέροντος μεγαλύτερο από το d

Αλγόριθμος επιλογής προτιμήσεων

Είσοδος αλγορίθμου: ένα ερώτημα Q, το προφίλ του χρήστη U, και το κριτήριο ενδιαφέροντος CI

Έξοδος αλγορίθμου: το σύνολο προτιμήσεων PK που εξάγονται από το U, οι οποίες συσχετίζονται και δεν συγκρούονται με το Q, και που ο αριθμός τους K, καθορίζεται με την βοήθεια του CI.

Γενική περιγραφή αλγορίθμου: γίνεται σταδιακή ανάπτυξη κατευθυνόμενων μονοπατιών, σε φθίνουσα σειρά με βάση τον βαθμό ενδιαφέροντος, τα οποία ξεκινούν από τον γράφο του ερωτήματος και επεκτείνονται προς τα έξω. Έτσι γίνεται εξαγωγή των προτιμήσεων που είναι συντακτικά συσχετιζόμενες με το Q σε φθίνουσα σειρά του βαθμού ενδιαφέροντος τους. Οι προτιμήσεις που συγκρούονται απορρίπτονται αμέσως. Ένα μονοπάτι που κατασκευάζεται και αντιστοιχεί σε μια επιλογή, δίδεται ως έξοδος εάν ικανοποιεί το

CI. Ο αλγόριθμος σταματάει όταν δεν υπάρχουν άλλες προτιμήσεις που να ικανοποιούν το CI και να μπορούν να εξαχθούν από το προφίλ.

7.10 Συνένωση προτιμήσεων

Ονομάζεται η διαδικασία ενσωμάτωσης των K επιλεγμένων προτιμήσεων στο αρχικό ερώτημα και η παραγωγή του νέου προσωποποιημένου ερωτήματος που θα δίνει τα επιθυμητά αποτελέσματα.

Θεωρούμε ότι έχουμε:

α) το σύνολο των K προτιμήσεων που πάρθηκαν από το προφίλ , σε φθίνουσα σειρά με βάση τον βαθμό ενδιαφέροντος.

β)τις πρώτες M συνθήκες που θεωρούνται ως υποχρεωτικές. Το M είναι ακέραιος στο πεδίο ορισμού [0,K] και καθορίζεται με τη βοήθεια ενός κριτηρίου

γ)αριθμό L, που είναι ακέραιος στο πεδίο ορισμού [0,K-M], ο οποίος μπορεί να εκφραστεί ρητά ή υπονοούμενα, καθορίζοντας τον ελάχιστο βαθμό ενδιαφέροντος d για κάθε σειρά στα αποτελέσματα.

Έχοντας τα πιο πάνω στοιχεία, το προσωποποιημένο ερώτημα μπορεί να κατασκευαστεί ακολουθώντας δυο διαφορετικές προσεγγίσεις:

A) <u>SQ (Single Query)</u>:

Μια μονή σύνθετη συνθήκη που είναι μια σύζευξη από:

i. την σύζευξη των υποχρεωτικών συνθηκών

ii.την διάζευξη όλων των πιθανών συζεύξεων των L συνθηκών από τις υπόλοιπες K-M

Η συνθήκη ενσωματώνεται στο αρχικό ερώτημα και αφαιρούνται τυχών επαναλαμβανόμενες συνθήκες από αυτό. Επιπλέον επιπρόσθετες μεταβλητές πλειάδων που απαιτούνται για τις συνθήκες προτίμησης, ενσωματώνονται στο ερώτημα.

Παράδειγμα 7.10.1 :

Το αρχικό ερώτημα της Μαρίας μεταφράζεται στο ακόλουθο προσωποποιημένο ερώτημα

```
select distinct MT.country
from MATCH MT, PLAY PL, TEAM_IN_MATCH TM, TEAM TE,
KIND KN, MATCH_SPORTSCASTER MS, SPORTSCASTER SP
where MT.mid=PL.mid and PL.date='21/04/2010 21:30' and  (
(  (MT.mid=KN.mid and KN.kind='Primera Division Ισπανίας'
and MT.mid=TM.mid and TM.aid=TE.aid
and TE.name='Real Madrid')  ) or
(  (MT.mid=TM.mid and TM.aid=TE.aid
and TE.name='Real Madrid'
and MT.mid=MS.mid and MS.rid=SP.rid
and SP.name='R.Carlos')  ) or
(  (MT.mid=KN.mid and KN.kind='Superleague Ελλάδος'
and MT.mid=MS.mid and MS.rid=MS.rid
and SP.name='Α.Κωνσταντίνου')  )  )
```

B) MQ (Multiple Queries):

Σχηματισμός ενός συνόλου K-M ερωτημάτων, που το κάθε ένα περιέχει μια απλούστερη συνθήκη, που είναι μια σύζευξη των υποχρεωτικών συνθηκών και μιας συνθήκης από τις K-M που απομένουν.
Κάθε ένα από αυτά τα ερωτήματα κτίζεται γύρω από το αρχικό ερώτημα, επεκτείνοντας το με την αντίστοιχη συνθήκη και το σύνολο των νέων μεταβλητών πλειάδων.
Τα επιθυμητά αποτελέσματα επιστρέφονται εάν παρθεί η ένωση των επί μέρους αποτελεσμάτων, ομαδοποιημένα βάση των χαρακτηριστικών που γίνονται "project" στο αρχικό ερώτημα και βγάζοντας εκτός όλες τις ομάδες που περιέχουν λιγότερο από L σειρές.

Παράδειγμα 7.10.2 :

Το αρχικό ερώτημα της Μαρίας μεταφράζεται στο ακόλουθο προσωποποιημένο ερώτημα

```
select MT.country
from ( (select distinct MT.country
        from    MATCH MT, PLAY PL, KIND KN
        where  MT.mid=PL.mid and
                        PL.date='21/04/2010 21:30' and
               MT.mid=KN.mid and
               KN.kind='Primera Division')
Union All
        ( select distinct MT.country
        from     MATCH MT, PLAY PL,
               TEAM_IN_MATCH MT, TEAM TE
        where   MT.mid=PL.mid and
               PL.date='21/04/2010 21:30' and
               MT.mid=TM.mid and
               TM.aid=TE.aid and
               TE.name='Real Madrid')
Union All
        ( select distinct MT.country
        from    MATCH MT, PLAY PL,
               MATCH_SPORTSCASTER MS, SPORTSCASTER SP
        where   MT.mid=PL.mid and
               PL.date='21/04/2010 21:30' and
               MT.mid=MS.mid and
               MS.rid=SP.rid and
               SP.name='R.Carlos') ) TEMP
group by   MT.country
having     count(*) >= 2
```

Πλεονεκτήματα προσέγγισης MQ:

α) το L μπορεί να καθοριστεί «έμμεσα», δίνοντας ένα επιθυμητό ελάχιστο βαθμό ενδιαφέροντος d, σε κάθε σειρά των αποτελεσμάτων. Είναι εύκολο να κτιστεί ένα ερώτημα με αποτελέσματα που να έχουν βαθμό ενδιαφέροντος >d, απλά γίνεται αλλαγή στο τμήμα "having" ως εξής:

having DEGREE_OF_CONJUCTION(*) > d

β) Είναι εύκολο να ταξινομηθούν τα αποτελέσματα με βάση τον βαθμό ενδιαφέροντος τοποθετώντας την πιο πάνω συνάρτηση στο μέρος του "select" και τα αποτελέσματα ταξινομούνται με "order by" εντολή.

Μειονεκτήματα προσέγγισης MQ:

α) Γίνεται επαναλαμβανομένη εκτέλεση του υποχρεωτικού μέρους του ερώτημα.

Για την κατασκευή συζεύξεων όμως, είτε γίνεται χρήση της προσέγγισης SQ, είτε της MQ πρέπει να μελετηθούν τα ακόλουθα δυο ζητήματα:

α) Συνθήκες που συγκρούονται δεν μπορούν ταυτόχρονα να ικανοποιηθούν άρα μπορούν να συνδυαστούν μεταξύ τους μόνο με διάζευξη, και τους φερόμαστε σαν να είναι 1 συνθήκη.

β) Οι συνθήκες μπορεί να μοιράζονται κοινές σχέσεις , σε 2 περιπτώσεις:

i. οι συνθήκες φτιάχνουν μονοπάτια που διασχίζουν διαφορετικά σύνολα κόμβων στον γράφο και συναντιούνται σε ένα κόμβο. Μπορούν να χρησιμοποιηθούν μεταβλητές της ίδιας ή διαφορετικής πλειάδας για αυτή την σχέση.
ii. οι συνθήκες μοιράζονται ένα κοινό αρχικό μεταβατικό "join" που ακολουθείται από ατομικά "selections" ή διαφορετικά μεταβατικά "join". Αν τα κοινά ατομικά "joins" είναι σχέση προς-ένα στην κατεύθυνση του "selection" τότε το να χρησιμοποιηθούν μεταβλητές κοινής πλειάδας για όλες τις κοινές σχέσεις , είναι η μόνη επιλογή. Αν

αυτές οι συνθήκες συγκρούονται τότε η χρήση διάζευξης είναι υποχρεωτική. Αν υπάρχει ένα κοινό ατομικό "join" στην κατεύθυνση του "selection" που να είναι σχέση προς-πολλά, είναι πιθανή η χρήση μεταβλητών διαφορετικής πλειάδας σε αυτό το σημείο.

Και στις δύο περιπτώσεις η χρήση μεταβλητών κοινής πλειάδας, υπονοεί ένα επιπρόσθετο περιορισμό που όλες οι συνθήκες πρέπει να ικανοποιούνται από το ίδιο αντικείμενο.

Παράδειγμα 7.10.3 :

Η σύζευξη των πιο κάτω συνθηκών, θα δίνει ως αποτελέσματα αγώνες που η ομάδα "Real Madrid" παίζει ως γηπεδούχος στο στάδιο "Santiago Bernabeu"

MATCH.mid=TEAM_IN_MATCH.mid and
TEAM_IN_MATCH.home='YES'
MATCH.mid=TEAM_IN_MATCH.mid and
TEAM_IN_MATCH.aid=TEAM.aid and
TEAM.name='Real Madrid'

Παράδειγμα 7.10.4 :

Μια σύζευξη των πιο κάτω συνθηκών, θα δίνει ως αποτελέσματα αγώνες στους οποίους συμμετέχει και η ομάδα "Real Madrid" και η ομάδα "Barcelona"

MATCH.mid=TEAM_IN_MATCH.mid and
TEAM_IN_MATCH.aid=TEAM.aid and
TEAM.name='Real Madrid'

MATCH.mid=TEAM_IN_MATCH.mid and
TEAM_IN_MATCH.aid=TEAM.aid and
TEAM.name='Barcelona'

7.11 Πειραματικά αποτελέσματα

Στο τέλος της μελέτης υλοποιήθηκε ένα σύστημα στο Oracle 9i, το οποίο ήταν περιείχε μια βάση δεδομένων με πληροφορίες για περίπου 340000 ταινίες.

Σε κάποιο πίνακα υπήρχαν αποθηκευμένα προφίλ χρηστών τόσο με αληθινά στοιχεία όσο και προφίλ που παρήχθησαν αυτόματα από το σύστημα.

Με ένα σύνολο 100 τυχαία δημιουργημένων ερωτημάτων και αριθμό υποχρεωτικών προτιμήσεων M=0, έγιναν 4 σύνολα πειραμάτων, τα οποία αναφέρονται πιο κάτω.

Α) Επιρροή του μεγέθους του προφίλ στον χρόνο επιλογής των προτιμήσεων

Δηλαδή μελετήθηκε ο χρόνος εκτέλεσης του αλγορίθμου επιλογής προτιμήσεων για διάφορα μεγέθη προφίλ (το μέγεθος του προφίλ ορίζεται ως ο αριθμός των ατομικών "selection" που υπάρχουν). Για κάθε μέγεθος, χρησιμοποιήθηκαν 100 προφίλ.
Μετρήθηκε ο χρόνος για κάθε ερώτημα και συνδυασμό για L=1.
Εκτελέστηκε η διαδικασία 3 φορές με μεγέθη K=5, K=10 και K=15.
Το συμπέρασμα ήταν ότι όσο μικρότερο είναι το μέγεθος του προφίλ τόσο μεγαλύτερος είναι ο χρόνος, διότι υπάρχει μεγαλύτερη πιθανότητα ο προτιμήσεις να μπουν αραιά στον γράφο, και άρα να υπάρχει ανάγκη για περισσότερες προσβάσεις στην βάση.

Β) Μέγεθος των αποτελεσμάτων από τα προσωποποιημένα ερωτήματα

Δηλαδή μετρήθηκε η αλλαγή του μεγέθους των αποτελεσμάτων από το προσωποποιημένο ερώτημα σε σύγκριση με το αρχικό ερώτημα. Αυτό έγινε με τυχαία επιλογή 200 προφίλ χρηστών και εκτέλεση του αλγορίθμου για διάφορες τιμές των K και L.

Γ) Σύγκριση προσέγγισης SQ και προσέγγισης MQ

Δηλαδή έγινε σύγκριση της απόδοσης των δυο μεθόδων μεταξύ τους. Έγινε επιλογή συνόλου δυο τυχαίων προφίλ χρηστών και εκτέλεση της κάθε μεθόδου για διάφορες τιμές του K και του L.

Δ) Απόδοση της προσωποποίησης

Έγινε επιλογή 200 τυχαίων προφίλ και χρήση της προσέγγισης MQ. Το συμπέρασμα ήταν ότι ο συνολικός χρόνος για προσωποποίηση ενός ερωτήματος και χρήση αυτού του ερωτήματος είναι λιγότερος από τον χρόνο που απαιτεί η χρήση του αρχικού ερωτήματος. Η προσωποποίηση αποδίδει καλά όταν το K είναι ανεξάρτητο από το L.

Κεφάλαιο 8

Προτιμήσεις στις βάσεις δεδομένων και σημασιολογία ceteris paribus

---

8.1 Εισαγωγή-Προσέγγιση άρθρου

Σε αυτό το κεφάλαιο παρουσιάζεται μια κριτική ανάλυση του άρθρου [4]. Όλη η μελέτη που αναλύεται στο άρθρο βασίζεται στην ανάγκη να γίνεται απόκτηση και διαχείριση προτιμήσεων χρηστών σε μια βάση δεδομένων ώστε τα αποτελέσματα που δίδονται μετά από την εκτέλεση του εκάστοτε ερωτήματος να ταιριάζουν στις προτιμήσεις του χρήστη που θα τα λάβει.

Για να δουλέψει μια τέτοια βάση δεδομένων, το σύστημα πρέπει να είναι ικανό να επεξεργάζεται ερωτήματα προτιμήσεων, τα οποία πρέπει να είναι ευέλικτα για να

σχηματίζονται από τον χρήστη, να μεταφράζονται ορθά από το σύστημα, να είναι υπολογιστικά αποδοτικά και να δίδουν γρήγορη πρόσβαση και αποτελέσματα. Σε μια τέτοια βάση υπάρχουν και δυο κύριες ανησυχίες : η σημασιολογική σαφήνεια και επάρκεια, και η υπολογιστική απόδοση.

Το πρώτο βήμα στην υποστήριξη ερωτημάτων προτιμήσεων είναι να μεταφραστούν ορθά οι προτιμήσεις των χρηστών από τον τρόπο που οι ίδιοι τις διατυπώνουν σε φυσική γλώσσα, ώστε να χρησιμοποιούνται στα ερωτήματα.

Στόχοι του άρθρου:

(1) Να δείξει ότι η προσέγγιση "totalitarian intersection" είναι ακατάλληλη για μετάφραση των προτιμήσεων από την φυσική γλώσσα καθώς αποτυγχάνει στο να πετύχει την βασική απαίτηση για σημασιολογική επάρκεια
(2) Να δώσει μια διαφορετική προσέγγιση για μετάφραση, την "ceteris paribus union", η οποία σημασιολογικά υπερέχει αλλά υπολογιστικά έχει προβλήματα.
(3) Εισαγωγή μιας παραλλαγή του αλγορίθμου BEST (παίρνει από την βάση το σύνολο των πιο επιθυμητών αντικειμένων και έχει χειρότερο χρόνο $O(\exp(n).D^2)$), το τελεστή ORD που με συγκεκριμένα ερωτήματα υπολογίζεται σε χρόνο O(nDlogD).

8.2 Κύριες έννοιες, θέματα και ορισμοί

Το άρθρο επικεντρώνεται στην ποιοτική προσέγγιση ως προς τα ερωτήματα προτιμήσεων, όπου οι προτιμήσεις των χρηστών αναπαριστούνται από μια γενική δυαδική σχέση προτιμήσεων πάνω σε ένα σχεσιακό σχήμα. Δηλαδή εάν έχουμε το σχεσιακό σχήμα $\mathbb{R}$, τότε ένα ερώτημα προτιμήσεων Q πάνω στο $\mathbb{R}$ αποτελείται από ένα σύνολο Q = {s1, ... ,sm} , από δηλώσεις προτιμήσεων. Αυτές οι δηλώσεις καθορίζουν σχέσεις προτιμήσεων $\{\succ 1, \ldots, \succ m\}$ αντίστοιχα, από όπου μπορεί να εξαχθεί η καθολική σχέση Q. Επομένων δοθέντος ενός στιγμιότυπου R στο $\mathbb{R}$, το σύστημα της βάσης μπορεί να επιστρέψει ένα υποσύνολο του R με τις πλειάδες που είναι βέλτιστες σύμφωνα με την καθολική σχέση.

Παράδειγμα 8.2.1:

Θεωρούμε ότι έχουμε το ακόλουθο σχεσιακό σχήμα "TEAMS", με τα στιγμιότυπα περασμένα μέσα.

	Άθλημα	Ομάδα	Ηλικίες
t1	Ποδόσφαιρο	Γυναικεία	Κάτω των 17
t2	Ποδόσφαιρο	Γυναικεία	Άνω των 17
t3	Ποδόσφαιρο	Αντρική	Κάτω των 17
t4	Ποδόσφαιρο	Αντρική	Άνω των 17
t5	Καλαθόσφαιρα	Γυναικεία	Κάτω των 17
t6	Καλαθόσφαιρα	Γυναικεία	Άνω των 17
t7	Καλαθόσφαιρα	Αντρική	Κάτω των 17
t8	Καλαθόσφαιρα	Αντρική	Άνω των 17

Θεωρούμε επίσης ότι ο χρήστης εκφράζει την προτίμηση

s: «Προτιμώ τις γυναικείες ομάδες ποδοσφαίρου κάτω των 17 από τις αντρικές ομάδες ποδοσφαίρου άνω των 17 ετών»

Η σχέση προτιμήσεων που προκαλείται από αυτή την δήλωση πάνω στις πλειάδες του R είναι

$\succ_s = \{t1 \succ_s t4\}$

Επίσης υπάρχει και ο πιο κάτω πίνακας με δηλώσεις προτιμήσεων πάνω στην πιο πάνω οντότητα, ο οποίος θα χρησιμοποιηθεί σε παραδείγματα σε επόμενες σελίδες:

s1	Προτιμώ τις γυναικείες ποδοσφαιρικές ομάδες από τις αντρικές
s2	Προτιμώ τις αντρικές καλαθοσφαιρικές ομάδες από τις γυναικείες
s3	Στις αντρικές ομάδες προτιμώ να είναι άνω των 17 ετών
s4	Στις γυναικείες ομάδες προτιμώ τις κατηγορίες κάτω των 17 ετών
s5	Προτιμώ τις ποδοσφαιρικές ομάδες από τις καλαθοσφαιρικές

8.3 Μετάφραση και αξιολόγηση των ερωτημάτων προτιμήσεων

Συνήθως οι δηλώσεις των προτιμήσεων (εκφρασμένες σε φυσική γλώσσα), αναφέρονται μόνο σε ένα υποσύνολο των χαρακτηριστικών μιας οντότητας.

Παράδειγμα 8.3.1:

Έχουμε την ακόλουθη δήλωση προτίμησης που δεν αναφέρεται σε όλα τα χαρακτηριστικά της οντότητας :

s': «Προτιμώ τις γυναικείες ποδοσφαιρικές ομάδες από τις αντρικές καλαθοσφαιρικές»

Βασικά σημασιολογικά ερωτήματα:

1) «Ποια εντολή προτίμησης πάνω στις πλειάδες της βάσης προκαλείται από μια δήλωση προτιμήσεων πάνω στις τιμές ενός αυστηρού υποσυνόλου των χαρακτηριστικών;»

   Λύση σημασιολογίας "totalitarian": να αγνοηθούν τα υπόλοιπα χαρακτηριστικά

   Παράδειγμα 8.3.2:

   Το s' από το παράδειγμα 8.3.1, θεωρείται ότι υπονοεί ότι οποιαδήποτε πλειάδα όπου η κατηγορία είναι το άθλημα=ποδόσφαιρο και το φύλο=γυναίκες προτιμείτε από πλειάδες όπου το άθλημα=καλαθόσφαιρα και το φύλο=άντρες. Δηλαδή θα έχουμε

   $\succ s' = \{t1 \succ s' t7, t1 \succ s' t8, t2 \succ s' t7, t2 \succ s' t8\}$

   Λύση σημασιολογίας "ceteris paribus": διόρθωση των τιμών

   Παράδειγμα 8.3.3**:**

   Το s' από το παράδειγμα 8.3.1 υπονοεί ότι μια πλειάδα στη οποία η κατηγορία είναι τύπου "minivan" και το εξωτερικό χρώμα έχει τιμή "red" προτιμείτε από πλειάδες όπου η κατηγορία είναι τύπου "SUV" και το εξωτερικό χρώμα έχει τιμή "white", εάν και οι δυο πλειάδες έχουν τις ίδιες ακριβώς τιμές στα υπόλοιπα

χαρακτηριστικά. Δηλαδή θα έχουμε

$\succ_{s'}=\{t1 \succ_{s'} t7,\ t2 \succ_{s'} t8\}$

2) «Πως πρέπει ένα σύνολο εντολών προτιμήσεων να αθροιστούν σε μια μόνο εντολή προτιμήσεων»

Λύση με αθροιστικούς τελεστές τύπου "Boolean": θεωρεί όλες τις δηλώσεις προτιμήσεων του Q εξίσου σημαντικές και συνδυάζει τις ανταποκρινόμενες σχέσεις προτιμήσεων χρησιμοποιώντας ένα σύνολο τελεστών, όπως η ένωση ή η τομή.

Λύση με αθροιστικούς τελεστές τύπου "prioritized compositions": αξιολογεί τις δηλώσεις προτιμήσεων του Q σε κάποια ιεραρχία σημασίας $\triangleright \subset \mathbf{Q X Q}$, όπου si $\triangleright$ sj σημαίνει ότι η δήλωση si είναι σημαντικότερη από την sj

Τα πιο πάνω αφορούν την μετάφραση μιας δήλωσης προτιμήσεων. Όσο αφορά την αξιολόγηση δηλώσεων προτιμήσεων και την εξαγωγή των πλειάδων που ταιριάζουν στα κριτήρια, έχει προταθεί ένας τελεστής που εξάγει όλες τις κατάλληλες πλειάδες, ο BEST που εκφράζεται ως εξής:

$$\mathsf{BEST}(R, \succ_Q) \;=\; \{t \in R \mid \forall t' \in R \;:\; t' \not\succ_Q t\}.$$

8.4 Η εμβέλεια των ερωτημάτων προτιμήσεων

Θεωρούμε ότι:

$A(\mathbb{R})$ = {a1,…,an} , το σύνολο των χαρακτηριστικών που καθορίζουν το σχεσιακό σχήμα $\mathbb{R}$

D(a1),...,D(an) , τα πεδία ορισμού των χαρακτηριστικών

$D(\mathbb{R})$ = XD(ai) , το διάστημα πλειάδων του σχεσιακού σχήματος $\mathbb{R}$

Περιορισμοί για την ανάλυση στο άρθρο:

Επικέντρωση σε ερωτήματα εξαρτώμενα από το σχήμα που βασίζονται σε ένα μόνο χαρακτηριστικό. Κύρια απαίτηση είναι οι προτιμήσεις να εκφράζονται σε όρους τιμών με βάση τα χαρακτηριστικά του σχήματος και μόνο.

Δήλωση προτίμησης εξαρτώμενη από το σχεσιακό σχήμα: αν και μόνο αν υπάρχουν δυο ασύνδετα υποσύνολα χαρακτηριστικών $A^c(s), A^r(s) \subseteq A(\mathbb{R})$, όπου ώστε $A^r(s) \neq 0$ και $\alpha \Rightarrow \langle\{\beta_1 \succ \beta_1'\}, \cdots, \{\beta_l \succ \beta_l'\}\rangle$, όπου $\alpha \in \mathcal{D}(\mathcal{A}^c(s))$ και για $1 \leq j \leq l$, έχουμε $\beta_j, \beta_j' \in \mathcal{D}(\mathcal{A}^r(s))$ and $\beta_j \neq \beta_j'$

Ερώτημα προτιμήσεων εξαρτώμενο από το σχήμα: αν και μόνο αν κάθε δήλωση του Q είναι εξαρτώμενη από το σχήμα.

Παράδειγμα 8.4.1:

Αν υποθέσουμε ότι έχουμε ένα σχεσιακό σχήμα που περιέχει τα χαρακτηριστικά:

άθλημα,φύλο ομάδας,κατηγορία,χώρα

και την δήλωση:

S1: «Στις γυναικείες ποδοσφαιρικές ομάδες, προτιμώ να είναι α' κατηγορίας»

Η δήλωση μπορεί να εκφραστεί ως:

[sport=Football] ∧ [gender=Women] ==> {[category=A] ≻ [category=B]}

και $A^c(s1)$={ sport,gender} , $A^r(s1)$={category} άρα είναι εξαρτώμενη από το σχήμα

Αν είχαμε την δήλωση:

s2: «Προτιμώ να είναι κυπριακή ομάδα»

τότε έχουμε $A^r(s2)$={country} αλλά έχουμε $A^c(s2)$=0 διότι δεν υπάρχει χαρακτηριστικό-συνθήκη

άρα δεν είναι εξαρτώμενη από το σχήμα

Παράδειγμα 8.4.2:

Ας υποθέσουμε ότι έχουμε τις ακόλουθες 3 δηλώσεις:

s1: «Προτιμώ να έχω sunroof»

s2: «Στα σπορ αυτοκίνητα, προτιμώ να έχω sunroof»

s3: «Προτιμώ τα σπορ αυτοκίνητα με sunroof,από τα οικογενειακά χωρίς sunroof»

Οι δυο πρώτες δηλώσεις, έχουν ως χαρακτηριστικό αναφοράς το sunroof και μόνο, άρα

$|A^r|=1$

Στην Τρίτη δήλωση έχουμε δυο αναφερόμενα συγκρίσιμα χαρακτηριστικά, άρα $|A^r|=2$

8.5 Η σημασιολογία "totalitarian"

Με παραδείγματα επιδεικνύεται πιο κάτω ότι αυτή η σημασιολογία είναι ανεπαρκής

Παράδειγμα 8.5.1:

Ας υποθέσουμε πως έχουμε την δήλωση προτίμησης:

«Προτιμώ το παγωτό σοκολάτα, από το παγωτό βανίλια»

εάν χρησιμοποιηθεί για επιλογή παγωτών τότε είναι εντάξει, αν όμως χρησιμοποιηθεί για επιλογή γευμάτων με βάση αυτή την σημασιολογία θα εννοηθεί ότι οποιαδήποτε επιλογή γεύματος προτιμάται από μια άλλη εάν η πρώτη περιέχει παγωτό σοκολάτα και η δεύτερη παγωτό βανίλια, κάτι που μάλλον δεν ανταποκρίνεται στις προθέσεις που είχε ο χρήστης όταν έκανε την δήλωση

Παράδειγμα 8.5.2:

Ας υποθέσουμε πως έχουμε τις δηλώσεις προτιμήσεων:

s1: «Προτιμώ το κόκκινο από το άσπρο για χρώμα του αυτοκινήτου μου»

s2: «Προτιμώ τα minivans από τα οικογενειακά αυτοκίνητα»

εάν τα αθροίσουμε με χρήση ένωσης βάση της σημασιολογίας αυτής παίρνουμε μια ασυμβίβαστη σχέση προτιμήσεων η οποία λέει ότι με βάση το s1 ένα κόκκινο οικογενειακό αυτοκίνητο θα ήταν προτιμότερο από ένα άσπρο minivan, και με βάση το s2 ένα άσπρο minivan θα ήταν προτιμότερο από ένα κόκκινο οικογενειακό αυτοκίνητο.

Παράδειγμα 8.5.3:

Ας υποθέσουμε πως έχουμε τις δηλώσεις προτιμήσεων:

s1: «Όσο αφορά τα minivans προτιμώ τα Ford από τα Chrysler»

s2: «Όσο αφορά τα SUVs, προτιμώ τα Toyota από τα Isuzu»

Η τομή των δύο σχέσεων προτιμήσεων είναι κενή, δείχνοντας πως με προσθήκη νέων

δηλώσεων προτιμήσεων μπορεί να προκαλέσει την αγνόηση πληροφοριών από προηγούμενες δηλώσεις.

Θα μπορούσε να χρησιμοποιηθούν και πιο σύνθετοι αθροιστικοί τελεστές, ή να χρησιμοποιηθεί ένωση σε ασύνδετα περιεχόμενα και τομή σε παρόμοια περιεχόμενα. Όλα αυτά όμως είναι πολύπλοκα αν έχουμε πολλές δηλώσεις. Επίσης θα μπορούσε να χρησιμοποιηθεί ένωση με ιδιότητες, οι οποίες όμως πρέπει να ζητούνται από τον χρήστη και αυτό κάνει πιο περίπλοκη την διαδικασία εισαγωγής των προτιμήσεων στο σύστημα, με αυξανόμενο υπολογιστικό κόστος.

8.6 Η σημασιολογία "Ceteris Paribus"

Η σημασιολογία αυτή συγκρίνει τις πλειάδες με βάση το χαρακτηριστικό που αναφέρθηκε στην δήλωση, αλλά μόνο αυτές που έχουν όλα τα υπόλοιπα χαρακτηριστικά ίσα.

Παράδειγμα 8.6.1:

Ας υποθέσουμε ότι έχουμε την δήλωση:

«Ένα στρογγυλό τραπέζι είναι καλύτερο από ένα τετράγωνο»

Ο χρήστης με αυτή την δήλωση δεν εννοεί ότι ανεξαρτήτως από τις οποιεσδήποτε άλλες ιδιότητες, οποιοδήποτε τραπέζι είναι στρογγυλό θα το προτιμούσε από ένα τετράγωνο. Εννοούσε ότι κάθε στρογγυλό τραπέζι είναι καλύτερο από ένα τετράγωνο εάν τα δυο τραπέζια δεν διαφέρουν ιδιαίτερα στα υπόλοιπα χαρακτηριστικά τους.

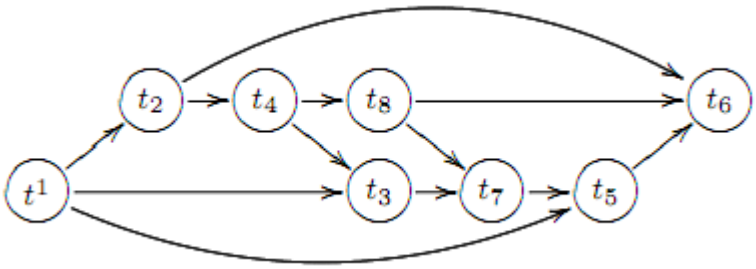
Με αυτή την μέθοδο δεν έχουμε αντικρουόμενα συμπεράσματα, και έχουμε λιγότερες πλειάδες να συγκρίνουμε. Επίσης ο αθροιστικός τελεστής της ένωσης που με την σημασιολογία "totalitarian" ήταν προβληματικός, με αυτή την σημασιολογία δουλεύει καλά.

Όσο πιο πολλές δηλώσεις προτιμήσεων υπάρχουν, τότε τόσο πιο πολλές πλειάδες είναι συγκρίσιμες, και δημιουργούνται πιο πολύπλοκες εντολές προτιμήσεων.

Παράδειγμα 8.6.2:

Με χρήση της οντότητας και του πίνακας δηλώσεων προτιμήσεων από το παράδειγμα 8.2.1. Στο πιο κάτω σχήμα φαίνεται η ένωση των σχέσεων προτιμήσεων $\succ_1, \cdots, \succ_5$ με βάση την

σημασιολογία “ceteris paribus”



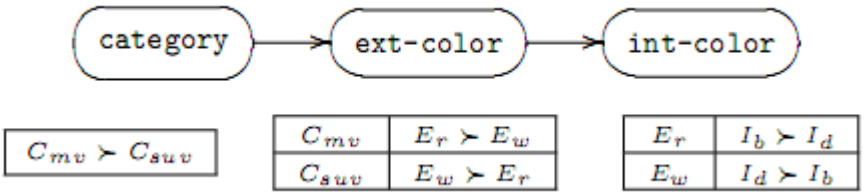
Οι κόμβοι συμβολίζουν τις πλειάδες και υπάρχει απευθείας ακμή από ένα t προς ένα t’ αν για κάποια δήλωση si, που ισχύει 1≤ i ≤5, έχουμε $t \succ_i t'$. Η τελική σχέση προτιμήσεων καθορίζεται από το μεταβατικό “closure” του γράφου.

8.7 CP-nets

Ορισμός: Ένα CP-net N για τις μεταβλητές V={X1,...Xn} είναι ένας κατευθυνόμενος γράφος G με κόμβους X1,...Xn και υπάρχει μια κατευθυνόμενη ακμή από το Xi στο Xj αν η προτίμηση πάνω στην τιμή του Xj είναι με συνθήκη, πάνω στην τιμή Xi. Κάθε κόμβος Xi $\in$ V συμβολίζεται με ένα πίνακα προτιμήσεων με συνθήκες (CPT(Xi)) που αντιστοιχεί μια αυστηρή μερική εντολή $\succ_{u_i}$ με κάθε στιγμιότυπο ui .

Ουσιαστικά ένα CP-net αντιστοιχεί με τα χαρακτηριστικά του σχεσιακού σχήματος, και μια κατευθυνόμενη ακμή υπάρχει από το u στο u’ εάν υπάρχει μια δήλωση προτιμήσεων για την οποία ισχύει ότι $v \in \mathcal{A}^c(s) \text{ and } v' = \mathcal{A}^r(s)$, δηλαδή περιγράφει την προτίμηση πάνω στο u’, που ελέγχεται με συνθήκη στην τιμή του u και ίσως και άλλων χαρακτηριστικών.

Παράδειγμα 8.7.1:
Με χρήση της οντότητας και του πίνακας δηλώσεων προτιμήσεων από το παράδειγμα 1.
Το CP-net N^Q για το συγκεκριμένο σχήμα και προτιμήσεις φαίνεται πιο κάτω:



Οι πίνακες είναι τα CPTs, και οι τιμές {Cmu,Csuv},{Er,Ew} και {Ib,Id} αντιπροσωπεύουν τα {minivan,SUV}, {red,white} και {bright,dark} αντίστοιχα.

Ένα CP-net θεωρείται εντελώς καθορισμένο εάν για κάθε μεταβλητή Xi, κάθε ανάθεση στο Ui ισχύει ότι το είναι μια ολική εντολή πάνω στο πεδίο ορισμού του X.

8.8 Τελεστές BEST και ORD

Τελεστής BEST:
Θεωρούμε ότι έχουμε μια σχέση προτιμήσεων $\succ_Q$ που καθορίζεται από ένα εξαρτημένο από το σχήμα ερώτημα Q={s1,…,sm}. Η βασικότερη υπορουτίνα του αλγορίθμου αυτού κάνει ένα έλεγχο επικράτησης μεταξύ των πλειάδων. Εδώ υπάρχει ένα πρόβλημα, η πολυπλοκότητα του ελέγχου επικράτησης. Το πρόβλημα είναι NP-δύσκολο και για μη-δυαδικά χαρακτηριστικά δεν είναι καν NP.

Τελεστής ORD:
Βασίζεται στην ταξινόμηση του συνόλου R με βάση την σχέση προτιμήσεων του ερωτήματος και να εμφανίζει στον χρήστη τις επικρατέστερες K πλειάδες σε μια μη-αυξανόμενη σειρά προτιμήσεων.

Ορισμός: Έστω έχουμε ένα σχεσιακό σχήμα $\mathbb{R}$, και Q το ερώτημα προτιμήσεων που είναι εξαρτώμενο από το σχήμα το οποίο προκαλεί μια αυστηρή μερική εντολή προτιμήσεων πάνω στο D($\mathbb{R}$). Εάν έχουμε ένα στιγμιότυπο του $\mathbb{R}$ το R τότε η ORD(R, $\succ_Q$) περιέχει όλες τις πλειάδες του R, τελείως ταξινομημένες, έτσι ώστε για κάθε t,t' $\in$ R, εάν το t εμφανίζεται στο

ORD(R, $\succ_{\mathcal{Q}}$) πριν το t', τότε έχουμε $t' \not\succ_{\mathcal{Q}} t$.

Θεώρημα: Έχουμε το Q , ένα ερώτημα εξαρτημένο από το σχεσιακό σχήμα $\mathbb{R}$ με n χαρακτηριστικά, και το R που είναι ένα στιγμιότυπο του $\mathbb{R}$, και |R|=D. Εάν το Q προκαλέσει ένα μη-κυκλικό εντελώς καθορισμένο CP-net, τότε το ORD(R, $\succ_{\mathcal{Q}}$) μπορεί να υπολογιστεί σε χρόνο $O(nD^2)$

Για να ισχύσει όμως το πιο πάνω θεώρημα και να υπολογίζεται το ORD σε αυτό τον χρόνο, απαιτείται να έχουμε ένα εντελώς καθορισμένο CP-net όπως αναφέρεται πιο πάνω. Αυτό σημαίνει ότι πρέπει για κάθε χαρακτηριστικό να δωθεί μια ολική εντολή πάνω στις τιμές για κάθε πιθανή τιμή που μπορεί να ανατεθεί στους «γονείς» του στο CP-net.
Πιο κάτω φαίνεται πως μπορεί να υπάρξει ένα παρόμοιο αποτέλεσμα αλλά με χρήση όχι τελείως καθορισμένου μη-κυκλικού CP-net.

Λήμμα: Ας υποθέσουμε ότι έχουμε το N που είναι ένα μη κυκλικό CP-net, και t≠t' είναι ένα ζευγάρι πλήρους ανάθεσης στις μεταβλητές του N. Έχουμε και την Xi μια μεταβλητή του N τέτοια ώστε και το t και το t' αναθέτουν τις ίδιες τιμές σε όλους του «προγόνους» του Xi στο N και διαφορετικές τιμές στο Xi. Εάν με δοθέν την ανάθεση u που δίδεται από το t στο Ui, έχουμε $t[X_i] \succ_{\mathbf{u}} t'[X_i]$τότε έχουμε $N \not\models t' \succ t$. Αλλιώς, εάν t[Xi] και t'[Xi] είναι μη-συγκρίσιμα ισχύει και το $N \not\models t' \succ t$ και το $N \not\models t \succ t'$.

Η πιο πάνω διαδικασία ονομάζεται «Τελεστής ταξινόμησης», αλλά ο τελεστής αυτός μέχρι εδώ είναι ανολοκλήρωτος, δηλαδή ορισμένα ερωτήματα απαντούνται αποτελεσματικά.

Παράδειγμα 8.8.1:
Με χρήση της οντότητας και του πίνακας δηλώσεων προτιμήσεων από το παράδειγμα 8.2.1 ,τον γράφο από το παράδειγμα 8.6.2 και το CP-net από το παράδειγμα 8.7.1
Έχουμε τις πλειάδες $t_3 = C_{mv} \wedge E_w \wedge I_h$και $t_8 = C_{suv} \wedge E_w \wedge I_d$. Με βάση το CP-net του ερωτήματος αυτές οι δυο αναθέσεις είναι μη-συγκρίσιμες. Αλλά το $N \not\models t_3 \succ t_8$ δεν μπορεί να συρρικνωθεί με χρήση της συνθήκης στο πιο πάνω λήμμα, διότι το χαρακτηριστικό "category" είναι το μόνο χαρακτηριστικό ρίζας αυτού του CP-net και το t3 αναθέτει σε αυτό μια τιμή που προτιμάται από την τιμή που του αναθέτει το t8.

Το πιο κάτω λήμμα δείχνει ότι ο τελεστής ταξινόμησης είναι ολοκληρωμένος σε μια μεν αποδυναμωμένη δε επαρκή και δυνατή μορφή.

Λήμμα: Εάν έχουμε ένα μη-κυκλικό CP-net το N, και δυο ολοκληρωμένες αναθέσεις t και t' στις μεταβλητές του N, η αλήθεια τουλάχιστον ενός από τα ερωτήματα $N \not\models t' \succ t$ ή $N \not\models t \succ t'$ μπορεί να καθοριστεί με χρήση ενός ζευγαριού κατάλληλων τελεστών ταξινόμησης.

Έτσι δίδεται ένας ενισχυμένος τελεστής ταξινόμησης που καθορίζει μια πλήρες επέκταση των εντολών προτιμήσεων. Ο τελεστής ονομάζεται order-pair και πιο κάτω φαίνεται ο ορισμός του:

order–pair (N, t, t')

1. Let $\mathbf{V}_{t,t'}$ be the set of all variables X_i, such that t and t' assign different values to X_i but the same values to all ancestors of X_i in N (in particular, $\mathbf{u}_t = \mathbf{u}_{t'}$ for all such X_i). Identify $\mathbf{V}_{t,t'}$ by a top-down traversal of N.
2. Fix a total topological ordering $\triangleright$ of the variables of N. For each variable $X_i \in N$, and each assignment $\mathbf{u}$ to $\mathbf{U}_i$, fix a total order $>_{\mathbf{u}}$, consistent with the strict partial order $\succ_{\mathbf{u}}$ specified by $CPT(X_i)$.
3. For each $X_i \in \mathbf{V}_{t,t'}$, let $\mathbf{u}^*$ be the assignment to $\mathbf{U}_i$ made by t (and t'). If, for each $X_i \in \mathbf{V}_{t,t'}$, we have $t[X_i] >_{\mathbf{u}^*} t'[X_i]$, then return $t \gg t'$. Otherwise, if for each $X_i \in \mathbf{V}_{t,t'}$, we have $t'[X_i] >_{\mathbf{u}^*} t[X_i]$, then return $t' \gg t$.
4. Otherwise, order $\mathbf{V}_{t,t'}$ with respect to $\triangleright$, and pick the first variable X_i in the sorted $\mathbf{V}_{t,t'}$. If $t[X_i] >_{\mathbf{u}^*} t'[X_i]$, then return $t \gg t'$, otherwise return $t' \gg t$.

Στο επόμενο θεώρημα φαίνεται η ορθότητά του:

Θεώρημα: Εάν έχουμε ένα μη-κυκλικό CP-net το N, πάνω στο σύνολο μεταβλητών V, η σχέση προτιμήσεων >> που προκαλείται από τον τελεστή order-pair είναι μια ολική εντολή πάνω στο D(V), συνεπής με την σχέση προτιμήσεων $\succ$ που προκαλείται από το N.

Πιο κάτω βλέπουμε το βασικό αποτέλεσμα για την σημασιολογία "Ceteris Paribus"

Θεώρημα: Εάν Q είναι ένα ερώτημα εξαρτώμενο από το σχεσιακό σχήμα $\mathbb{R}$ με n χαρακτηριστικά, το R είναι ένα στιγμιότυπο του $\mathbb{R}$ και |R|=D, εάν το Q προκαλεί ένα μη-κυκλικό CP-net τότε η εντολή ORD(R, $\succ_Q$) μπορεί να υπολογιστεί σε χρόνο O(nDlogD).

Κεφάλαιο 9

Συμπεράσματα

Ο τομέας αυτός είναι συνεχώς κάτω από επεξεργασία, συνεχώς προτείνονται νέες μεθοδολογίες με όλο και καλύτερες επιδόσεις ως προς τον υπολογισμό των αποτελεσμάτων, την αποθήκευση όλων των απαραίτητων δεδομένων, και την επιτυχία των αποτελεσμάτων.

Είναι ένας τομέας που απαιτείται λόγω της ανάπτυξης και της εξάπλωσης της χρήσης της τεχνολογίας, να ερευνηθεί σε όλες τις πτυχές τους, ώστε να βρεθούν οι καλύτερες τεχνικές για την αξιοποίηση του.

Όλες αυτές οι τεχνικές που προτάθηκαν στα άρθρα που ανάλυσα στα προηγούμενα κεφάλαια, μπορούν να εφαρμοστούν και σε άλλα είδη προτιμήσεων εκτός από αυτά που προτείνονται με επιτυχία, και μπορούν να αναπτυχθούν περαιτέρω με βελτιώσεις.

Βιβλιογραφία

[1] Yannis Ioannidis and Georgia Koutrika, “Personalization of Queries in Database Systems”, ICDE 2004: 597-608

[3] Kostas Stefanidis and Evaggelia Pitoura, “Fast Contextual Preference Scoring of Database tuples”, EDBT 2008: 344-355

[4] Ronen I. Brafman and Carnel Domshlak, “Database Preference Queries Revisited“

[5] P.Georgiadis, I.Kapantaidakis, V.Christophides, E.M.Nguer, N.Spyratos, “Efficient Rewriting Algorithms for Preference Queries”, ICDE 2008: 1101-1110

[6] Ronen I. Brafman and Carmel Domshlak, ”Preference Handling – An Introductory Tutorial”

[7] C. Boutilier, R.I.Brafman, C.Domshlak, H.H.Hoos, D.Poole, “CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements”