ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Αλγόριθμοι παραγωγής συνδυαστικών κανόνων (ARM)»

Του φοιτητή
Σιμεόνοβ Στανισλάβ
Αρ. Μητρώου: 164740

Επιβλέπων
Ονοματεπώνυμο Δέρβος Δημήτριος
Βαθμίδα Καθηγητής

26 Μαΐου 2023

Τίτλος Π.Ε. Αλγόριθμοι παραγωγής συνδυαστικών κανόνων (ARM)
Κωδικός Π.Ε. 22121
Ονοματεπώνυμο φοιτητή/τών Σιμεόνοβ Στανισλάβ
Ονοματεπώνυμο εισηγητή Δέρβος Δημήτριος
Ημερομηνία ανάληψης Π.Ε. 28-02-2022
Ημερομηνία περάτωσης Π.Ε. 26-05-2023

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιερώνω αυτή την εργασία στην οικογένεια και τους φίλους μου για την πολύτιμη στήριξή τους.»

Πρόλογος

Ο κύριος λόγος που με οδήγησε στην επιλογή της συγκεκριμένης πτυχιακής εργασίας είναι η ίδια η φύση της εργασίας. Η έρευνα γενικότερα πάντα αποτελούσε και αποτελεί σημαντικό κομμάτι της ζωής μου. Πέρα όμως από το κομμάτι της έρευνας, ιδιαίτερο ενδιαφέρον μου προξένησε και η υλοποίηση του θέματος της εργασίας, των αλγορίθμων δηλαδή, μέσα από το περιβάλλον R/RStudio. Έτσι με την επιλογή που έκανα θέλησα να εξερευνήσω όλο το φάσμα που προσφέρεται μέσα από την παρούσα εργασία. Τέλος πιστεύω πως η εργασία αυτή αποτελεί μία πρωτης τάξεως ευκαιρία να εμβαθύνω τις γνώσεις μου στον τομέα του Data Mining.

Περίληψη

Η συγκεκριμένη πτυχιακή εργασία έχει ως σκοπό την παρουσίαση και ανάλυση σε βάθος των τριών αλγορίθμων εύρεσης συχνών στοιχειοσυνόλων, τα οποία στη συνέχεια οδηγούν στην παραγωγή των συνδυαστικών κανόνων. Οι τρεις αλγόριθμοι εύρεσης συχνών στοιχειοσυνόλων είναι: ο αλγόριθμος Apriori, ο αλγόριθμος Eclat και ο αλγόριθμος FP-Growth. Προτού ωστόσο πραγματοποιηθεί η ανάλυση του κάθε αλγορίθμου γίνεται αναφορά στο περιβάλλον R/RStudio μέσα από το οποίο συγκρίνονται οι τρεις αλγόριθμοι στην πράξη. Έπειτα κάθε ένας από τους αλγορίθμους αναλύεται και παρουσιάζονται τα χαρακτηριστικά τα οποία φέρει. Η αποτύπωση του κάθε αλγορίθμου αφορά αρχικά το θεωρητικό του κομμάτι μέσα από την ανάλυση του ψευδοκώδικα και ενός παραδείγματος, από τα οποία μπορεί να διαπιστωθεί πλήρως η πορεία εκτέλεσης του αλγορίθμου. Επίσης το θεωρητικό κομμάτι ακολουθεί και η καταγραφή των αλγορίθμων στην πράξη χρησιμοποιώντας το περιβάλλον R/RStudio μέσα από ορισμένα παραδείγματα σύγκρισης των αλγορίθμων. Τα παραπάνω παραδείγματα αφορούν το καθένα, ξεχωριστά σύνολα δεδομένων, πάνω στα οποία εκτελούνται και συγκρίνονται οι αλγόριθμοι καταγράφοντας μια σειρά από μετρήσιμα αποτελέσματα, όπως για παράδειγμα οι χρόνοι εκτέλεσης ή η κατανάλωση της μνήμης. Έπειτα, από την ανάλυση των αλγορίθμων, ακολουθεί η ανάδειξη των παραπάνω αποτελεσμάτων για κάθε αλγόριθμο και η σύγκρισή τους μέσα από πίνακες και σχήματα που αποτυπώνουν τα παραγόμενα αποτελέσματα. Αμέσως μετά, συνοψίζεται ο κάθε αλγόριθμος για τις επιδόσεις του έναντι των υπολοίπων δύο ώστε να διαπιστωθεί εάν προκύπτουν χρήσιμα συμπεράσματα και τι είδους συμπεράσματα είναι αυτά. Τέλος, τονίζονται οι προτεινόμενες επεκτάσεις οι οποίες αφορούν τους παραπάνω αλγορίθμους και ολοκληρώνεται η παρουσα εργασία με τον Επίλογο, στον οποίο συνοψίζεται τι ακριβώς παρουσιάστηκε στην εργασία αυτή.

« Αλγόριθμοι παραγωγής συνδυαστικών κανόνων (ARM)»

(Association Rule Mining Algorithms)

«Στανισλάβ Σιμεόνοβ»

(Stanislav Simeonov)

Abstract

The aim of the present BSc thesis is the presentation and the in-depth study of three association rule mining (ARM) algorithms: Apriori, Eclat, and FP-Growth. The frequent itemsets are first found, followed by the calculation of the association rules. Prior to considering each algorithm, the R IDE used (R/RStudio) is presented. Next, each one ARM algorithm is presented at the pseudocode level, as well as by means of example applications on a set of datasets that are publicly available. The latter comprise the testbed used for all three algorithms. This way the performance of each algorithm is compared/contrasted to that of the other two. More specifically, performance comparison consists of execution times and memory utilisation measurements. The results obtained are organised in tables and in explanatory figures. In Chapter 6 (Performance Measurements and Concluding Remarks), FP-Growth is seen to outperform Apriori and Eclat both in execution time as well as in memory utilisation. In addition, FP-Growth is the only one of the three algorithms that can be applied as is to dynamic datasets that grow in size in an incremental fashion. Possible extensions to the present study are outlined in Chapter 7 (Proposed Extensions).

Ευχαριστίες

Θα ήθελα σε αυτό το σημείο να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Δημήτριο Δέρβο καθώς στάθηκε αρωγός σε όλη την πορεία υλοποίησης της παρούσας εργασίας. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου που μου στάθηκε όταν το είχα ανάγκη. Τέλος, θα ήθελα να ευχαριστήσω και τον κ. Κελεσίδη Κωνσταντίνο για την συνεργασία που είχαμε.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Εισαγωγή

1.1 Σκοπός της εργασίας

Η συγκεκριμένη εργασία έχει σαν σκοπό την μελέτη, την εφαρμογή και την συγκριτική αξιολόγηση των αλγορίθμων παραγωγής συνδυαστικών κανόνων (association rules). Αυτό επιτυγάνεται μέσω της ανάλυσης των αλγορίθμων Apriori, Eclat και FP-Growth αρχικά σε θεωρητικό επίπεδο για το τι ισχύει σε κάθε έναν από τους αλγορίθμους αυτούς μέσω ψευδοκώδικα και παραδειγμάτων, κι έπειτα με την καταγραφή του τρόπου υλοποίησης και υποστήριξής τους στο περιβάλλον R/RStudio. Για τη σύγκριση των αλγορίθμων δεν αρκεί μόνο η σύγκριση τους σε θεωρητικό επίπεδο, καταθέτοντας τι ισχύει για κάθε έναν αλγόριθμο και τα χαρακτηριστικά τα οποία συνοδεύουν τον αλγόριθμο αυτόν, αλλά και η σύγκρισή τους στην πράξη μέσα από τις επιδόσεις τους πάνω σε δύο επιλεγμένα σύνολα δεδομένων. Τέλος, είναι επιθυμητή και η σύγκριση των αποτελεσμάτων των αλγορίθμων προκειμένου να διαπιστωθεί εάν προκύπτουν χρήσιμα και απτά συμπεράσματα για τον εκάστοτε αλγόριθμο.

1.2 Δομή της εργασίας

Στο κεφάλαιο 2 παρουσιάζεται η μεθοδολογία και το περιβάλλον μέσα από το οποίο παρατηρούνται η υλοποίηση και η υποστήριξη του κάθε αλγορίθμου για τον οποίο εκτελούνται τα αντίστοιχα πειράματα.

Στο κεφάλαιο 3 το οποίο και αποτελεί το βασικότερο κεφάλαιο της παρούσας εργασίας αναλύονται πλήρως οι τρεις αλγόριθμοι Apriori, Eclat και FP-Growth. Για κάθε έναν αλγόριθμο τονίζονται τι ακριβώς ισχύει για τον αλγόριθμο αυτό, τα βήματα εκτέλεσής του, παρατείθεται ο ψευδοκώδικας για τον αλγόριθμο αυτόν καθώς και ένα αναλυτικό παράδειγμα εκτέλεσης του αλγορίθμου και αναφέρονται τα συμπεράσματα που προκύπτουν από τον συγκεκριμένο αλγόριθμο. Επίσης ιδιαίτερη βαρύτητα δίνεται στο κύριο πακέτο που χρησιμοποιείται στο περιβάλλον της R, το οποίο μας δίνει τη δυνατότητα υλοποίησης και σύγκρισης των αλγορίμων που προαναφέρθηκαν.

Στο κεφάλαιο 4 αναλύονται τα δύο σύνολα δεδομένων που χρησιμοποιούνται στα πλαίσια της εργασίας για την σύγκρισή των αλγορίθμων μέσα από το περιβάλλον της R. Πιο συγκεκριμένα παρουσιάζονται αναλυτικά η δομή και τα χαρακτηριστικά του κάθε συνόλου δεδομένων τόσο πριν την προεπεξεργασία τους, για να έρθουν στην απαιτούμενη μορφή ώστε να εκτελεστούν οι αλγόριθμοι, όσο και έπειτα από αυτή.

Στο κεφάλαιο 5 περιγράφεται η ανάλυση των δεδομένων και τα αποτελέσματα που προκύπτουν για τον κάθε αλγόριθμο ανά περίπτωση. Επίσης τονίζεται ο τρόπος με τον οποίο μπορούν και συγκρίνονται τα παραγόμενα αποτελέσματα ανά περίπτωση κάνοντας χρήση των δυνατοτήτων του πακέτου arules. Ακόμη περιγράφεται αναλυτικά ο τρόπος με τον οποίο υλοποιούνται τα πειράματα για την σύγκριση των αλγορίθμων, ποια μεθοδολογία χρησιμοποιείται και ποια η σειρά των βημάτων που ακολουθούνται.

Στο κεφάλαιο 6 έπονται το ιδιαίιτέρως σημαντικά σχόλια και συμπεράσματα που προκύτπουν από τη σύγκριση των αλγορίθμων μέσα από την αποτύπωση των πινάκων με τα μετρήσιμα αποτελέσματα για την κάθε περίπτωση σύγκρισης των αλγορίθμων και των γραφημάτων που αποτυπώνουν τις αντίστοιχες εκτελέσεις των αλγορίθμων.

Στο κεφάλαιο 7 ακολουθούν οι προτεινόμενες επεκτάσεις για τους αλγορίθμους μέσα από τις οποίες μπορεί να εξεταστεί αν και κατα πόσο βοηθούν στην καλύτερη απόδοση των αλγορίθμων. Τέλος ακολουθεί το 8^{ο} κεφάλαιο με τον επίλογο και μία σύνοψη του περιεχομένου της εργασίας αυτής.

Κεφάλαιο 2ο: Περιβάλλον επεξεργασίας δεδομένων

2.1 Η γλώσσα προγραμματισμού R

Για την πραγματοποίηση των σεναρίων σύγκρισης των τριών αλγορίθμων χρησιμοποιείται η γλώσσα προγραμματισμού R. Η συγκεκριμένη γλώσσα προγραμματισμού, είναι διαθέσιμη ως ελεύθερο λογισμικό ανοιχτού κώδικα, σύμφωνα με τους όρους άδειας GNU. Χρησιμοποιείται κυρίως για ανάλυση δεδομένων, στατιστική ανάλυση και παραγωγή γραφημάτων. Ειδικά για τις δύο προαναφερεθίσες έννοιες η R παρέχει μία πληθώρα από τεχνικές. Την συγκεκριμένη γλώσσα προγραμματισμού μπορεί κανείς να την εγκαταστήσει στις πλατφόρμες Windows, Linux και Mac μιας και υπάρχουν οι αντίστοιχες εκδόσεις. Επίσης αυτό που χαρακτηρίζει ιδιαίτερα την R είναι η επεκτασιμότητά της, όπως και το γεγονός πως αποτελείται από μία μεγάλη κοινότητα. Για τους παραπάνω λόγους, καταλήγει να προτιμάται από αρκετούς επιστήμονες των δεδομένων [1].

2.2 Ανάλυση των δεδομένων με τη χρήση της γλώσσας R

Η ανάλυση των δεδομένων με τη χρήση της γλώσσας R καθίσταται ιδιαιτέρως εύκολη. Αυτό προκύπτει από το γεγονός πως η R υποστηρίζει μια ποικιλία από τεχνικές αναλυτικής μοντελοποίησης. Για παράδειγμα ορισμένες από αυτές είναι: κλασικές στατιστικές δοκιμές, ομαδοποίηση, ανάλυση χρονοσειρών, γραμμική και μη γραμμική μοντελοποίηση καθώς και αρκετές άλλες [2],[5].

Κεφάλαιο 3ο: Παραγωγή συνδυαστικών κανόνων

3.1 Βασικές έννοιες

Έστω *I* ένα σύνολο στοιχείων και D μία βάση συναλλαγών, όπου κάθε συναλλαγή έχει μοναδικό αναγνωριστικό (tid) και περιέχει ένα σύνολο στοιχείων. Ένα σύνολο στοιχείων ονομάζεται επίσης στοιχειοσύνολο. Ένα σύνολο στοιχείων με k στοιχεία λέγεται k-στοιχειοσύνολο [4]. Η τιμή υποστήριξης ενός στοιχειοσυνόλου X, που συμβολίζεται με σ(X), είναι ο αριθμός των συναλλαγών στις οποίες λαμβάνει χώρα ως υποσύνολο. Ένα στοιχειοσύνολο είναι συχνό εάν η τιμή υποστήριξής του είναι μεγαλύτερη από μια τιμή ελάχιστης υποστήριξης που ορίζεται από το χρήστη (min_sup). Το σύνολο των συχνών k-στοιχειοσυνόλων συμβολίζεται με Fk, εννίοτε και ως Lk [8]. Ένας συνδυαστικός κανόνας είναι μια έκφραση A => B, όπου τα A και B είναι στοιχειοσύνολα. Το στοιχειοσύνολο A ή αριστερό μέλος (Left-Hand-Side-LHS) [3] του κανόνα αναφέρεται και ως το σώμα (Antecedent) του κανόνα ενώ το στοιχειοσύνολο B ή δεξιό μέλος (Right-Hand-Side-RHS) [3] του κανόνα αναφέρεται και η κεφαλή του κανόνα (Consequent). Η τιμή υποστήριξης (Support) του κανόνα ορίζεται ως σ(AUB) και ορίζεται ως ο λόγος των συναλλαγών που περιέχουν το στοιχειοσύνολο {A,B} επί του συνόλου των συναλλαγών [3] και λαμβάνει τιμές στο διάστημα [0,1] ενω η εμπιστοσύνη (Confidence) ορίζεται ως σ(AUB) / σ(A) (δηλαδή πόσο συχνά το στοιχειοσύνολο B εμφανίζεται σε συναλλαγές οι οποίες περιέχουν το A ή η πιθανότητα ότι μία συναλλαγή περιέχει το B, δεδομένου ότι περιέχει το A) [3] και αντίστοιχα λαμβάνει τιμές στο διάστημα [0,1]. Η τιμή ανύψωσης(lift) [4] αναδεικνύει τη σημασία ενός κανόνα και ορίζεται ως lift(A=>B) = σ(AUB) / σ(A) * σ(B) ή την αναλογία της παρατηρούμενης υποστήριξης προς την αναμενόμενη αν τα A και B ήταν ανεξάρτητα. Ένας κανόνας είναι ισχυρός εάν τόσο η τιμή της εμπιστοσύνης του, όσο και η τιμή υποστήριξης του, είναι μεγαλύτερες από τις ελάχιστες τιμές εμπιστοσύνης (min_conf) και υποστήριξης (min_supp) τις οποίες έχει ορίσει ο χρήστης. Γενικότερα, η παραγωγή συνδυαστικών κανόνων μπορεί να θεωρηθεί ως μια διαδικασία δύο βημάτων [36]:

1. Η εύρεση όλων των συχνών στοιχειοσυνόλων (Frequent Itemset Mining): Τα στοιχειοσύνολα τα οποία εμφανίζονται στη βάση των συναλλαγών τουλάχιστον τόσες φορές όσες και η ελάχιστη τιμή υποστήριξης (min_supp).

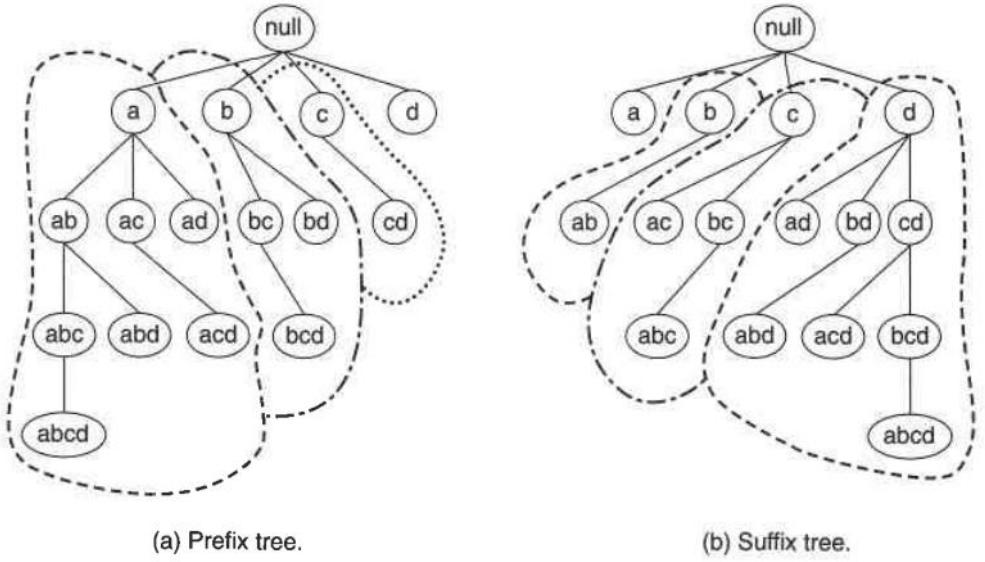
2. Η παραγωγή ισχυρών συνδυαστικών κανόνων (Association Rule Mining) από τα συχνά στοιχειοσύνολα: Σύμφωνα με τον ορισμό των ισχυρών συνδυαστικών κανόνων οι κανόνες αυτοί πρέπει να ικανοποιούν την ελάχιστη τιμή υποστήριξης και την ελάχιστη τιμή εμπιστοσύνης τις οποίες ορίζει ο χρήστης.

3.1.1 Αναζήτηση συχνών στοιχειοσυνόλων

Η αναζήτηση των συχνών στοιχειοσυνόλων μπορεί να θεωρηθεί ουσιαστικά ως ο τρόπος διέλευσης στο πλέγμα (lattice) [11] των στοιχειοσυνόλων, όπου ως πλέγμα εννοείται ο χώρος αναζήτησης (search space) των στοιχειοσυνόλων. Η στρατηγική που χρησιμοποιείται από έναν αλγόριθμο υπαγορεύει τον τρόπο διέλευσης της δομής του πλέγματος κατά τη διάρκεια της διαδικασίας αναζήτησης συχνών στοιχειοσυνόλων [42]. Ορισμένες από τις στρατηγικές που παρουσιάζονται είναι καλύτερες από άλλες, ανάλογα με τη διαμόρφωση των συχνών στοιχειοσυνόλων στο πλέγμα. Παρακάτω παρουσιάζεται μια σύνοψη των στρατηγικών αυτών:

Equivalence Classes (Κλάσεις Ισοδυναμίας):

Ένας διαφορετικός τρόπος προσπέλασης του πλέγματος είναι αρχικά ο διαχωρισμός του πλέγματος σε ασύνδετες ομάδες κόμβων (ή κλάσεις ισοδυναμίας). Ένας αλγόριθμος εύρεσης συχνών στοιχειοσυνόλων αναζητά αρχικά τα συχνά στοιχειοσύνολα εντός μίας συγκεκριμένης κλάσης ισοδυναμίας προτού μεταβεί σε μία άλλη κλάση ισοδυναμίας. Παράδειγμα αποτελεί ο αλγόριθμος Apriori και η στρατηγική ανα επίπεδο (level-wise) [15] την οποία ακολουθεί καθώς μπορεί να θεωρηθεί πως διαχωρίζει το πλέγμα με βάση το μέγεθος των στοιχειοσυνόλων και πιο συγκεκριμένα ο αλγόριθμος εντοπίζει αρχικά τα συχνά στοιχειοσύνολα που έχουν μήκος ένα και στη συνέχεια εκείνα τα οποία έχουν μεγαλύτερο μέγεθος [24]. Οι κλάσεις ισοδυναμίας μπορούν επίσης να ορίστούν σύμφωνα με τις ετικέτες προθέματος (prefix) ή κατάληξης (suffix) ενός στοιχειοσυνόλου [41]. Στην παραπάνω περίπτωση δύο στοιχειοσύνολα θεωρείται πως ανήκουν στην ίδια κλάση ισοδυναμίας, εάν μοιράζονται ένα κοινό πρόθεμα ή επίθημα μήκους k. Σύμφωνα με τα παραπάνω στην πρώτη προσέγγιση, αυτή δηλαδή η οποία βασίζεται στο πρόθεμα (prefix), ο αλγόριθμος δύναται να αναζητήσει συχνά στοιχειοσύνολα τα οποία ξεκινούν με το πρόθεμα a, προτού αναζητήσει εκείνα τα οποία ξεκινούν με προθέματα b,c. Μία τέτοια περίπτωση αποτελεί ο αλγόριθμος Eclat [27], ο οποίος χρησιμοποιεί τις κλάσεις ισοδυναμίας (Equivalence Classes), καθώς όπως θα διαπιστωθεί και παρακάτω, με βάση το πρόθεμα του εκάστοτε συχνού στοιχειοσυνόλου μήκους 1, δημιουργεί τις κλάσεις ισοδυναμίας για το αντίστοιχο στοιχειοσύνολο και κατόπιν προχωρά στην εύρεση συχνών στοιχειοσυνόλων μεγαλύτερου μήκους τα οποία ωστόσο έχουν το ίδιο πρόθεμα [42],[43].

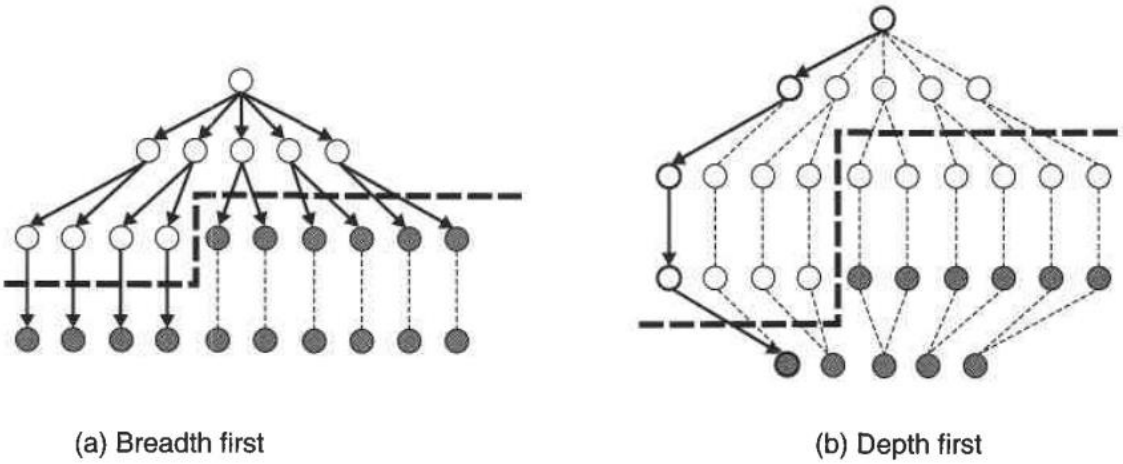


Σχήμα 3.1: Κλάσεις ισοδυναμίας [4]

Αναζήτηση με βάση το πλάτος εναντίον αναζήτησης με βάση το βάθος (Breadth vs Depth Search)

Αρχικά ο αλγόριθμος Apriori, όπως θα διαπιστωθεί και παρακάτω, προσπελαύνει το πλέγμα με τέτοιον τρόπο ο οποίος δίνει βάση πρώτα στο πλάτος (Breadth First) [25]. Πρώτα δηλαδή παράγει όλα τα συχνά στοιχειοσύνολα μήκους 1, και κατόπιν ακολουθούν τα συχνά στοιχειοσύνολα μήκους 2, έως ότου δεν βρεθούν νέα συχνά στοιχειοσύνολα. Το πλέγμα των στοιχειοσυνόλων μπορεί επίσης να προσπελαστεί με τέτοιον τρόπο ο οποίος να δίνει βάση πρώτα στο βάθος (Depth First) [31]. Ο αλγόριθμος μπορεί να εκκινήσει από το στοιχειοσύνολο a , να υπολογίσει την τιμή υποστήριξης του προκειμένου να καθορίσει εάν το συγκεκριμένο στοιχειοσύνολο είναι συχνό. Στη συνεχεια και εφόσον το παραπάνω

στοιχειοσύνολο είναι συχνό, ο αλγόριθμος επεκτείνεται προοδευτικά στο επόμενο επίπεδο στοιχειοσυνόλων, δηλαδή ab, abc έως ότου βρεθεί κάποιο στοιχειοσύνολο το οποίο δεν υποστηρίζει την ελάχιστη τιμή υποστήριξης. Στη συνέχεια η πορεία του καταλήγει σε κάποιο άλλο συχνό στοιχειοσύνολο προηγούμενου επιπέδου και συνεχίζει την αναζήτηση από εκεί. Μία τέτοια περίπτωση, αναζήτησης, δηλαδή δίνοντας βάση πρώτα στο βάθος, αποτελεί ο αλγόριθμος Eclat όπως θα εξεταστεί παρακάτω, μιας και η αναδρομική κλίση του αλγορίθμου επιτρέπει την εύρεση των συχνών στοιχειοσυνόλων που έχουν το ίδιο πρόθεμα στο μεγαλύτερο επίπεδο, προτού ο αλγόριθμος περάσει στα στοιχειοσύνολα με διαφορετικό πρόθεμα [4].



Σχήμα 3.2: Μέθοδοι αναζήτησης συχνών στοιχειοσυνόλων [4]

3.1.2 Αναπαράσταση του συνόλου δεδομένων

Υπάρχουν πολλοί τρόποι για την αναπαράσταση ενός συνόλου δεδομένων μιας βάσης η οποία περιέχει συναλλαγές. Η επιλογή της αναπαράστασης μπορεί να επηρεάσει το I/O κόστος που προκύπτει κατά τον υπολογισμό της τιμής υποστήριξης των υποψήφιων στοιχειοσυνόλων. Έναν τρόπο αναπαράστασης αποτελεί η οριζόντια διάταξη των δεδομένων (Horizontal Data Layout), η οποία και υιοθετείται από τον αλγόριθμο Apriori καθώς και από αρκετούς άλλους αλγορίθμους. Επίσης υπάρχει και η δυνατότητα αποθήκευσης της λίστας των αναγνωριστικών των συναλλαγών (Transaction Identifiers) (TID-list) που σχετίζονται με κάθε ένα στοιχειοσύνολο [31-32]. Μια τέτοιου είδους αναπαράσταση ονομάζεται κατακόρυφη διάταξη δεδομένων (Vertical Data Layout). Η τιμή υποστήριξης για κάθε υποψήφιο στοιχειοσύνολο λαμβάνεται από την τομή της λίστας των αναγνωριστικών των συναλλαγών (TID-list) των στοιχειοσυνόλων που αποτελούν υποσύνολα του υποψήφιου στοιχειοσυνόλου. Παρατηρείται πως το μήκος των λιστών των αναγνωριστικών των συναλλαγών (TlD-lists) τείνει να μικραίνει με τη μετάβαση σε στοιχειοσύνολα όλο και αυξανόμενου μεγέθους [34]. Παρακάτω στα παραδείγματα συναντάται καθένας από τους παραπάνω τρόπους αναπαράστασης των δεδομένων.

3.1.3 Παραγωγή συνδυαστικών κανόνων

Στη συγκεκριμένη υποενότητα περιγράφεται η αποδοτική παραγωγή συνδυαστικών κανόνων από ένα συχνό στοιχειοσύνολο. Κάθε ένα συχνό k-στοιχειοσύνολο (μήκους k) Y, δύναται να παράξει έως 2^k - 2 συνδυαστικούς κανόνες μη λαμβάνοντας υπόψιν του εκείνους οι οποίοι έχουν κενό σώμα ή κενή κεφαλή ($\{\} \rightarrow Y$ ή $Y \rightarrow \{\}$) . Η παραγωγή του εκάστοτε συνδυαστικού κανόνα επιτυγχάνεται από το διαχωρισμό του στοιχειοσυνόλου Y σε δύο μη κενά υποσύνολα X και Y - X, τέτοια ώστε ο κανόνας

X → Y - X να ικανοποιεί την ελάχιστη τιμή εμπιστοσύνης (minconf), η οποία έχει οριστεί από τον χρήστη.[16-17] Οι παραπάνω κανόνες ικανοποιούν την ελάχιστη τιμή υποστήριξης, καθώς η παραγωγή τους προέρχεται από συχνά στοιχειοσύνολα, τα οποία ικανοποιούν την ελάχιστη τιμή υποστήριξης, γι'αυτό και τους έχει αποδοθεί ο χαρακτηρισμός "συχνά" [4],[23],[24]. Παρακάτω παρατίθεται ένα σύντομο παράδειγμα:

Έστω το συχνό στοιχειοσύνολο X = {1,2,5}. Το σύνολο των έξι υποψήφιων συνδυαστικών κανόνων οι οποίοι μπορούν να παραχθούν από το παραπάνω συχνό στοιχειοσύνολο είναι: ({1,2} → {5]), ({1,5} → {2}), ({2,5} → {1}), ({1} → {2,5}), ({2} → {1,5}), ({5} → {1,2}). Η τιμή υποστήριξης του κάθε κανόνα είναι η ίδια με αυτή του συχνού στοιχειοσυνόλου X και με αυτόν τον τρόπο ικανοποιείται η ελάχιστη τιμή υποστήριξης για τους παραπάνω κανόνες.

Αλγόριθμος παραγωγής συνδυαστικών κανόνων [5]:

Input: Το σύνολο των συχνών στοιχειοσυνόλων (Frequent Itemsets)

Output: Το σύνολο των συνδυαστικών κανόνων (Association Rules).

```
function rules (F);
1)  R := ∅;
2)  forall f ∈ F do begin
3)      m   := 1;
4)      H_m := ⋃_{i∈f} {{i}};
5)      repeat
6)          forall h ∈ H_m do
7)              if s_T(f) / s_T(f−h) ≥ c_min
8)              then R   := R ∪ {[(f − h) → h]};
9)              else  H_m := H_m − {h};
10)         H_{m+1} := candidates(H_m);
11)         m       := m + 1;
12)     until H_m = ∅ or m ≥ |f|;
13) end;
14) return R;
end; (* rules *)
```

Σχήμα 3.3: Αλγόριθμος παραγωγής συνδυαστικών κανόνων [5]

Βήματα εκτέλεσης του αλγορίθμου παραγωγής συνδυαστικών κανόνων [5]:

1) Αρχικοποίηση του συνόλου των συνδυαστικών κανόνων (γραμμή 1).

2) Για κάθε ένα συχνό στοιχειοσύνολο, το οποίο ανήκει στο σύνολο των συχνών στοιχειοσυνόλων (γραμμή 2).

3) Αρχικά για τις κεφαλές των κανόνων μήκους οι οποίες περιέχουν ένα στοιχειοσύνολο μήκους 1 (γραμμές 3-4).

4) Επανάληψη για κάθε κεφαλή του κανόνα με αυξανόμενο μέγεθος, για κάθε πιθανή κεφαλή του κανόνα (γραμμές 5-6).

5) Αν η τιμή υποστήριξης είναι μεγαλύτερη ή ίση από την ελάχιστη, έπεται προσθήκη του κανόνα στο σύνολο των συνδυαστικών κανόνων (γραμμές 7-8).

6) Ειδάλλως η κεφαλή του κανόνα απορρίπτεται, δημιουργούνται κεφαλές με ένα στοιχειοσύνολο περισσότερο και αυξάνεται η τιμή του μετρητή μήκους της κεφαλής (γραμμές 9-11)

7) Έως ότου δεν υπάρχει καμία άλλη κεφαλή κανόνων ή το σώμα του κανόνα καταστεί κενό (γραμμές 12-13)

8)Επιστροφή του συνόλου των συνδυαστικών κανόνων που βρέθηκαν (γραμμή 14).

Input: Το σύνολο των συχνών στοιχειοσυνόλων (Frequent Itemsets)

Output: Το σύνολο των υποψηφίων συνδυαστικών κανόνων (Association Rules) μεγέθους (k+1).

function candidates (F_k)
begin
1) $E := \emptyset$;
2) **forall** $f_1, f_2 \in F_k$
3) **with** $f_1 = \{a_1, \ldots, a_{k-1}, a_k\}$
4) **and** $f_2 = \{a_1, \ldots, a_{k-1}, a'_k\}$
5) **and** $a_k < a'_k$ **do begin**
6) $f := f_1 \cup f_2 = \{a_1, \ldots, a_{k-1}, a_k, a'_k\}$;
7) **if** $\forall a \in f : f - \{a\} \in F_k$
8) **then** $E := E \cup \{f\}$;
9) **end**;
10) **return** E;
end (* candidates *)

Σχήμα 3.4: Αλγόριθμος παραγωγής υποψήφιων συνδυαστικών κανόνων [5]

Βήματα εκτέλεσης της συνάρτησης candidates, για την παραγωγή (k+1) υποψήφιων συνδυαστικών κανόνων [5]:

1)Αρχικοποίηση του συνόλου των υποψηφίων συνδυαστικών κανόνων (γραμμή 1).

2)Για όλα τα ζεύγη των συχνών στοιχειοσυνόλων, τα οποία διαφέρουν κατά ένα στοιχείο και τα οποία είναι ταξινομημένα κατά λεξικογραφική τάξη (γραμμές 2-5).

3)Η ένωση περιέχει k+1 στοιχεία, αν και μόνο αν όλα τα υποσύνολα είναι συχνά (γραμμές 6-7).

4)Προσθήκη του νέου στοιχειοσυνόλου στα υποψήφια, ειδάλλως δεν μπορεί να είναι συχνό (γραμμές 8-9).

5)Επιστροφή του συνόλου των υποψηφίων συνδυαστικών κανόνων (γραμμή 10).

3.1.4 Μέθοδοι για τη δημιουργία υποψηφίων στοιχειοσυνόλων

- Η μέθοδος Brute Force:

Η συγκεκριμένη μέθοδος θεωρεί κάθε ένα στοιχειοσύνολο μεγέθους k ως ένα δυνητικό υποψήφιο στοιχειοσύνολο και στη συνέχεια απορρίπτει εκείνα για τα οποία δεν ικανοποιείται η ελάχιστη τιμή υποστήριξης (minimum support), υλοποιώντας με τον τρόπο αυτόν το Prune Step [4].

- Η μέθοδος $F_{κ-1}$ x F1:

Η συγκεκριμένη μέθοδος δημιουργίας υποψηφίων στοιχειοσυνόλων επεκτείνει κάθε ένα συχνό (k-1)-στοιχειοσύνολο με άλλα συχνά στοιχειοσύνολα. Ιδιαίτερη προσοχή θα πρέπει να δοθεί ωστόσο στην τήρηση της αλφαβητικής σειράς των στοιχειοσυνόλων, καθώς σε διαφορετική περίπτωση ορισμένα υποψήφια στοιχειοσύνολα δημιουργούνται περισσότερες από μία φορές.

- Η μέθοδος $F_{κ-1}$ x $F_{κ-1:}$

Η παραπάνω μέθοδος είναι αυτή την οποία υλοποιεί ο αλγόριθμος Apriori. Πιο συγκεκριμένα ένα ζεύγος συχνών (k - l) στοιχειοσυνόλων [26] ενώνεται μόνο εάν τα πρώτα (k - 2) στοιχεία τους είναι πανομοιότυπα. Έστω A= {a1,a2,…,$a_{κ-1}$} και B={b1,b2,...,$b_{κ-1}$}, ένα ζεύγος συχνών στοιχειοσυνόλων. Το ζεύγος αυτό ενώνεται αν και μόνο αν ισχύει:

$$ai=bi \text{ (για } i = 1,2,...k-2) \text{ και } a_{κ-1} \neq b_{κ-1.}$$

Παρόλο που, κάθε υποψήφιο στοιχειοσύνολο λαμβάνεται με την ένωση ενός ζεύγους συχνών (k-1)-στοιχειοσυνόλων, απαιτείται επίσης κι ένα βήμα κλαδέματος (Pruning Step) για να διασφαλιστεί ότι τα υπόλοιπα (k - 2) - υποσύνολα του υποψηφίου στοιχειοσυνόλου είναι συχνά [4].

3.1.5 Αλγόριθμος Brute Force για την εύρεση των συχνών στοιχειοσυνόλων

```
BruteForce (D, I, minsup):
1  F ← ∅ // set of frequent itemsets
2  foreach X ⊆ I do
3      sup(X) ← ComputeSupport (X, D)
4      if sup(X) ≥ minsup then
5          F ← F ∪ {(X, sup(X))}
6  return F

ComputeSupport (X, D):
7  sup(X) ← 0
8  foreach ⟨t, i(t)⟩ ∈ D do
9      if X ⊆ i(t) then
10         sup(X) ← sup(X) + 1
11 return sup(X)
```

Σχήμα 3.5: Ο αλγόριθμος BruteForce [6]

Αλγόριθμος BruteForce για την εύρεση συχνών στοιχειοσυνόλων [6]:

Input: Η βάση των συναλλαγών D, το σύνολο *I* των στοιχείων και η ελάχιστη τιμή υποστήριξης σ.

Output: Το σύνολο των συχνών στοιχειοσυνόλων (Frequent Itemsets).

Ο υπολογισμός της τιμής υποστήριξης (support) παίρνει στην χειρότερη περίπτωση χρόνο ίσο με O(|*I*| * |D|) και εξαιτίας της ύπαρξης $O(2^{|I|})$ πιθανών υποψηφίων στοιχειοσυνόλων η υπολογιστική πολυπλοκότητα του αλγορίθμου BruteForce καταλήγει να είναι ίση με O(|*I*| * |D| $*2^{|I|}$). Ένας ακόμη σημαντικός παράγοντας ο οποίος θα πρέπει να ληφθεί υπόψη είναι και το συνολικό μέγεθος της βάσης D, καθώς εάν η αντίστοιχη βάση D είναι αρκετά μεγάλη, είναι επίσης σημαντικό να υπολογιστεί η πολυπλοκότητα εισόδου/εξόδου (input/output complexity). Προκειμένου να υπολογιστεί η τιμή υποστήριξης για κάθε υποψήφιο στοιχειοσύνολο, χρειάζεται να πραγματοποιηθεί ένα συνολικό πέρασμα της βάσης. Με αυτόν τον τρόπο η πολυπλοκότητα Input/Output του αλγορίθμου Bruteforce είναι $O(2^{|I|})$ σαρώσεις της βάσης D. Σύμφωνα με τα παραπάνω ο αλγόριθμος BruteForce καταλήγει να μην συμφέρει υπολογιστικά, συμπεριλαμβανομένου και του υπολογιστικού κόστους για το Input/Output [13], ακόμα και για σύνολα που περιέχουν μικρό πλήθος στοιχείων, την ώρα μάλιστα που στην πράξη ο αριθμός των στοιχείων δύναται να καταστεί ιδιαίτερα υψηλός όπως ο αριθμός των βιβλίων σε μία βιβλιοθήκη ή ο αριθμός των προϊόντων που πωλούνται σε μία υπεραγορά. Παρακάτω παρουσιάζονται οι τρεις αλγόριθμοι εύρεσης των συχνών στοιχειοσυνόλων και αποτυπώνονται οι διαφορές με τον παραπάνω αλγόριθμο καθώς και οι βελτιώσεις που παρατηρούνται τόσο ως προς τον τρόπο υπολογισμού της τιμής υποστήριξης των τριών αλγορίθμων, όσο και από τον τρόπο με τον οποίο δημιουργούνται τα υποψήφια στοιχειοσύνολα.

Παράδειγμα εύρεσης συχνών στοιχειοσυνόλων:

TID	Items
1	I1.I2,I4,I5
2	I2,I3,I5
3	I4,I5
4	I1,I2,I4,I5

Σχήμα 3.6: Η βάση με τις συναλλαγές του παραδείγματος

Έστω η ελάχιστη τιμή υποστήριξης είναι το 0.5 ή 50%. Η τιμή υποστήριξης αρχικά για το{I1} είναι Supp(I1) = 2 / 4 = 50%. Για το {I2} η τιμή υποστήριξης είναι Supp(I2) = 3 / 4 = 75%. Για το {I3} η τιμή υποστήριξης είναι Supp(I3) = 1 / 4 = 25%. Για το {I4} η τιμή υποστήριξης είναι Supp(I4) = 3 / 4 = 75%. Τέλος για το {I5} η τιμή υποστήριξης είναι Supp(I5) = 4 / 4 = 100%. Τα συχνά στοιχειοσύνολα με βάση την ελάχιστη τιμή υποστήριξης που είναι 0.5, είναι όσα έχουν τιμή υποστήριξης ίση ή μεγαλύτερη από 0.5 και πιο συγκεκριμένα τα {I1}, {I2}, {I4}, {I5}. Κατόπιν, εξετάζονται οι συνδυασμοι των συχνών στοιχειοσυνόλων: Supp(I1,I2) = 2 / 4 = 50%, Supp(I1,I4) = 2 / 4 = 50%, Supp(I1,I5) = 2 / 4 = 50%, Supp(I2,I4) = 2 / 4 = 50%, Supp(I2,I5) = 3 / 4 = 75%, Supp(I4,I5) = 3 / 4 = 75%. Παρατηρείται ότι όλα τα στοιχειοσύνολα ικανοποιούν την ελάχιστη τιμή υποστήριξης οπότε όλα χαρακτηρίζονται ως συχνά. Επίσης ελέγχονται και τα στοιχειοσύνολα: Supp(I1,I2,I4) = 2 / 4 = 50%, Supp(I1,I2,I5) = 2 / 4 = 50%, Supp(I1,I4,I5) = 2 / 4 = 50%, Supp(I2,I4,I5) = 2 / 4 = 50%. Αντίστοιχα και σε αυτήν την περίπτωση παρατηρείται ότι όλα τα στοιχειοσύνολα ικανοποιούν την ελάχιστη τιμή

υποστήριξης οπότε όλα χαρακτηρίζονται ως συχνά. Τέλος ελέγχεται το στοιχειοσύνολο Supp(I1,I2,I4,I5) = 2 / 4 = 50% όπου και σε αυτήν την περίπτωση ικανοποιείται η ελάχιστη τιμή υποστήριξης. Το σύνολο των συχνών στοιχειοσυνόλων είναι: {(I1), (I2), (I4), (I5), (I1,I2), (I1,I4), (I1,I5), (I2,I4), (I2,I5), (I4,I5), (I1,I2,I4), (I1,I2,I5), (I1,I4,I5), (I2,I4,I5), (I1,I2,I4,I5)}.

3.1.6 Incremental mining

Το μεγαλύτερο μέρος των αλγορίθμων εύρεσης συχνών στοιχειοσυνόλων, μέσα από μία μεγάλου μεγέθους βάση δεδομένων, λειτουργούν με αποτελεσματικό τρόπο σε μια "στατική" βάση η οποία δεν έχει περιθώρια τακτικής ενημέρωσης [7]. Παρόλα αυτά, όλες οι βάσεις που αποτελούνται από συναλλαγές είναι δυναμικές, ενημερώνονται και αυξάνονται τακτικά από άποψη χρόνου. Χαρακτηριστικό παράδειγμα αποτελεί μία βάση, η οποία περιέχει συναλλαγές, μιας βιβλιοθήκης καθώς σε καθημερινή βάση βιβλία δανείζονται ή επιστρέφονται. Η παραπάνω βάση αναδεικνύει την δυναμική της φύση και επ' ουδενί δεν μπορεί να χαρακτηριστεί ως στατική. Παρακάτω θα διαπιστωθεί στην αντίστοιχη υποενότητα για τον κάθε αλγόριθμο ποιοί από αυτούς υποστηρίζουν το incremental mining, τι ισχύει γενικότερα και αν ο καθένας από αυτούς μπορεί να χαρακτηριστεί ως incremental mining αλγόριθμο, να υποστηρίζει δηλαδή το incremental mining.

3.2 Ο αλγόριθμος Apriori

Ο αλγόριθμος Apriori ο οποίος ορίστηκε από τους Agrawal and Srikant 1994 [8], αποτελεί έναν από τους πιο δημοφιλείς αλγορίθμους στο πεδίο της Εξόρυξης των Δεδομένων. Χρησιμοποιείται για την εύρεση συχνών στοιχειοσυνόλων (Frequent Itemsets), καθώς και συνδυαστικών κανόνων (Association Rules) μέσα από μία βάση συναλλαγών (Transactional Database). Ο συγκεκριμένος αλγόριθμος εκτελείται σε επίπεδα (Level-Wise), όπου το κάθε επίπεδο αντιστοιχίζεται στο πλήθος των στοιχείων ενός στοιχειοσυνόλου που εξετάζεται στο εκάστοτε επίπεδο. Κατά την παραπάνω διαδικασία k-συχνά στοιχειοσύνολα χρησιμοποιούνται προκειμένου να βρεθούν (k+1) υποψήφια στοιχειοσύνολα. [14] Έπειτα, μέσα από επαναλήψεις ο αλγόριθμος περνάει από επίπεδο σε επίπεδο έως ότου φτάσει στο επίπεδο για το οποίο δεν μπορούν να προκύψουν νέα συχνά στοιχειοσύνολα. Προκειμένου να επιτευχθεί βελτίωση της αποτελεσματικότητας της δημιουργίας υποψήφιων συχνών στοιχειοσυνόλων, γίνεται χρήση της ιδιότητας Apriori (Apriori Property) [18], η οποία συμβάλλει στην μείωση του χώρου στον οποίο γίνεται η αναζήτηση των συχνών στοιχειοσυνόλων.

Ιδιότητα Apriori: Όλα τα μη κενά υποσύνολα ενός συχνού στοιχειοσυνόλου είναι επίσης συχνά. Εάν ένα στοιχειοσύνολο είναι μη συχνό, τότε όλα τα υπερσύνολα του θα είναι μη συχνά.

Σε κάθε επανάληψη υφίστανται δύο σημαντικά βήματα [19]. Κατά τη διάρκεια του πρώτου βήματος (Join Step), λαμβάνει χώρα η δημιουργία των υποψήφιων στοιχειοσυνόλων, ενώ κατά το δεύτερο σημαντικό βήμα (Prune Step) πραγματοποιείται η καταμέτρηση της υποστήριξης του κάθε υποψήφιου στοιχειοσυνόλου, και όσα υποψήφια στοιχειοσύνολα δεν ικανοποιούν τη συνθήκη για την ελάχιστη υποστήριξη (minimum support), δεν λαμβάνονται υπόψη σε αντίθεση με όσα ικανοποιούν την παραπάνω συνθήκη και τα οποία πλέον από υποψήφια κατατάσσονται στα συχνά στοιχειοσύνολα [19-20].

Δημιουργία του συνόλου των υποψηφίων στοιχειοσυνόλων Ck για τον αλγόριθμο Apriori:

Για να δημιουργηθούν τα υποψήφια στοιχειοσύνολα μεγέθους k και να σχηματιστεί το σύνολο Ck, από τα συχνά στοιχειοσύνολα μεγέθους k-1, λαμβάνει χώρα ένα Self Join για το σύνολο F_{K-1}, δηλαδή το

σύνολο των συχνών k-1 στοιχειοσυνόλων, ώστε να προκύψει το σύνολο Ck [21-22]. Ένα μικρό παράδειγμα αποτελεί το παρακάτω:

Έστω το σύνολο των συχνών στοιχειοσυνόλων μεγέθους k = 2:

F2 = {ΑΒ,ΑΓ,ΑΔ,ΑΕ,ΒΓ,ΒΔ,ΒΕ}

Το σύνολο των υποψηφίων στοιχειοσυνόλων μεγέθους k = 3 που προκύπτει είναι:

C3 = {ΑΒΓ,ΑΒΔ,ΑΒΕ,ΑΓΔ,ΑΓΕ,ΑΔΕ,ΒΓΔ,ΒΓΕ,ΒΔΕ}

Παρατηρείται ότι κάθε ένα υποψήφιο στοιχειοσύνολο μεγέθους 3, προκύπτει από την ένωση δύο συχνών στοιχειοσυνόλων μεγέθους 2, με την προϋπόθεση ύπαρξης ενός κοινού στοιχείου στη συγκεκριμένη περίπτωση.

Αλγόριθμος Apriori για την εύρεση των συχνών στοιχειοσυνόλων [6]:

Input: Η βάση των συναλλαγών, η οποία συμβολίζεται με T και η ελάχιστη τιμή υποστήριξης minsup σ.

Output: Το σύνολο των συχνών στοιχειοσυνόλων.

3.2.1 Βήματα εκτέλεσης του αλγορίθμου Apriori

Έχοντας μία δοσμένη βάση συναλλαγών (Transactional Database D) και μία ελάχιστη τιμή υποστήριξης (Minimum Support minsup).

1. Εύρεση των συχνών στοιχειοσυνόλων μήκους 1 καθώς αρχικά το μέγεθος k ισούται με 1. (γραμμή 1)
2. Επανάληψη της διαδικασίας μέχρις ότου δεν εντοπίζονται νέα συχνά στοιχειοσύνολα (το μήκος του συνόλου L_{k-1} ισούται με μηδέν). (γραμμή 2)
3. Δημιουργία μήκους (k) υποψήφιων στοιχειοσυνόλων από τα μήκους (k-1) συχνά στοιχειοσύνολα μέσω της apriori-gen, η οποία σε πρώτη φάση υλοποιεί το Join Step (εικόνα 3.8). (γραμμή 3)
4. Αφαίρεση των υποψήφιων στοιχειοσυνόλων μήκους (k), τα οποία περιέχουν υποσύνολα μήκους (k-1) τα οποία δεν είναι συχνά υλοποιώντας το Prune Step (εικόνα 3.9). (γραμμή 3)
5. Σε κάθε μία συναλλαγή t, υπολογισμός της τιμής υποστήριξης για κάθε υποψήφιο στοιχειοσύνολο c. (γραμμές 4-8)
6. Σχηματισμός και επιστροφή του συνόλου των συχνών στοιχειοσυνόλων. (γραμμές 9-11)

```
1)  L1 = {large 1-itemsets};
2)  for ( k = 2; Lk-1 ≠ ∅; k++ ) do begin
3)     Ck = apriori-gen(Lk-1); // New candidates
4)     forall transactions t ∈ D do begin
5)        Ct = subset(Ck, t); // Candidates contained in t
6)        forall candidates c ∈ Ct do
7)           c.count++;
8)     end
9)     Lk = {c ∈ Ck | c.count ≥ minsup}
10) end
11) Answer = ⋃k Lk;
```

Σχήμα 3.7: Ο αλγόριθμος Apriori [6]

Η συνάρτηση apriori-gen:

```
insert into Ck
select p.item1, p.item2, ..., p.itemk-1, q.itemk-1
from Lk-1 p, Lk-1 q
where p.item1 = q.item1, ..., p.itemk-2 = q.itemk-2,
      p.itemk-1 < q.itemk-1;
```

Σχήμα 3.8: Η apriori-gen function (Join Step) [6]

Η συνάρτηση apriori-gen, όπως παρουσιάζεται στην εικόνα 3.2.1 λαμβάνει ως παράμετρο το σύνολο L_{k-1}, το σύνολο των συχνών (k-1) - στοιχειοσυνόλων [10]. Επιστρέφει αντίστοιχα ένα υπερσύνολο του συνόλου των συχνών στοιχειοσυνόλων μεγέθους k. Αρχικά λαμβάνει χώρα το επονομαζόμενο Join Step, το οποίο παρουσιάζεται στην εικόνα 3.2.2 και αποτελεί την ένωση του συνόλου L_{k-1} με τον εαυτό του. Έπειτα στο Prune Step [10], το οποίο ακολουθεί το Join Step [10] και παρουσιάζεται στην εικόνα 3.2.3, κάθε στοιχειοσύνολο c το οποίο ανήκει στο σύνολο με τα υποψήφια Ck ελέγχεται για το αν κάποιο από τα υποσύνολα του είναι μη συχνό δεν ανήκει δηλαδή στο σύνολο L_{k-1}.

```
forall itemsets c ∈ Ck do
   forall (k−1)-subsets s of c do
      if (s ∉ Lk−1) then
         delete c from Ck;
```

Σχήμα 3.9: Η apriori-gen function (Prune Step) [6]

3.2.2 Παράδειγμα εκτέλεσης του αλγορίθμου Apriori

Μέσα από το παρακάτω σύνολο δεδομένων (DataSet), θα βρεθούν σε πρώτη φάση τα συχνά στοιχειοσύνολα (Frequent Itemsets), τα οποία θα οδηγήσουν, σε δεύτερη φάση, στην παραγωγή των συνδυαστικών κανόνων.

TID	Items
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

Σχήμα 3.10: Η βάση με τις συναλλαγές του παραδείγματος.

Ορίζεται ως ελάχιστη τιμή υποστήριξης (Minimum Support) η τιμή 0.22 ή 22% (για να χαρακτηριστεί δηλαδή ένα υποψήφιο στοιχειοσύνολο μεγέθους k ως συχνό, θα πρέπει να συναντάται σε τουλάχιστον δύο από τις εννέα συναλλαγές της βάσης των συναλλαγών).

Οπότε στην πρώτη επανάληψη για μέγεθος k = 1, αναζητούνται τα συχνά στοιχειοσύνολα μήκους 1.

Πίνακας 3.1: Ο πίνακας με τα υποψήφια 1-στοιχειοσύνολα (C1)

Itemset	Support
I1	6
I2	7
I3	6
I4	2
I5	2

Αρχικά, μέσα από την βάση των συναλλαγών εξάγονται όλα τα στοιχειοσύνολα μήκους 1 και σημειώνεται η τιμή υποστήριξης (support) για κάθε ένα υποψήφιο 1-στοιχειοσύνολο. Κατά συνέπεια, σχηματίζεται ο παραπάνω πίνακας, ο οποίος αποτελεί το σύνολο C1 (Candidate Set).

Πίνακας 3.2: Ο πίνακας με τα συχνά 1-στοιχειοσύνολα (F1)

1-Itemset	Support
I1	6
I2	7
I3	6
I4	2
I5	2

Κατόπιν, για κάθε ένα υποψήφιο 1-στοιχειοσύνολο, συγκρίνεται η τιμή υποστήριξης του με την ελάχιστη τιμή υποστήριξης (minimum support) που έχει οριστεί εξαρχής. Έτσι, προκύπτει ο παραπάνω πίνακας, ο οποίος περιέχει όλα τα συχνά 1-στοιχειοσύνολα, τα στοιχειοσύνολα δηλαδή των οποίων η τιμή υποστήριξης είναι ίση ή και μεγαλύτερη από την ελάχιστη τιμή υποστήριξης που έχει οριστεί και σχηματίζεται το σύνολο F1 (Frequent Set).

Περνώντας στη δεύτερη κατα σειρά επανάληψη για μέγεθος k = 2, αναζητούνται τα συχνά στοιχειοσύνολα μήκους 2.

Προκειμένου, ωστόσο να σχηματιστεί το σύνολο C2 της παρακάτω εικόνας, σύμφωνα και με τα βήματα εκτέλεσης του αλγορίθμου Apriori, λαμβάνει χώρα το Join Step του αλγορίθμου Apriori. Πιο συγκεκριμένα και με βάση τη δημιουργία του συνόλου των υποψηφίων στοιχειοσυνόλων Ck για τον αλγόριθμο Apriori, πραγματοποιείται ένα Self Join για το σύνολο F1 (ένωση του κάθε συχνού στοιχειοσυνόλου μηκους 1, με τα υπόλοιπα συχνά στοιχειοσύνολα μήκους 1), με το οποίο σχηματίζεται το κάθε υποψήφιο στοιχειοσύνολο μήκους 2. Έχοντας τα παραπάνω ως δεδομένα, σχηματίζονται τα υποψήφια στοιχειοσύνολα μήκους 2 και πιο συγκεκριμένα: C2 = ({I1,I2},{I1,I3},{I1,I4},{I1,I5},{I2,I3},{I2,I4},{I2,I5},{I3,I4},{I3,I5},{I4,I5}). Αμέσως μετά υπολογίζεται η τιμή υποστήριξης για κάθε ένα υποψήφιο στοιχειοσύνολο.

Itemset	Support
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

Πίνακας 3.3: Ο πίνακας με τα υποψήφια 2-στοιχειοσύνολα (C2)

Στη συνέχεια για το κάθε ένα υποψήφιο στοιχειοσύνολο μήκους 2, συγκρίνεται η τιμή υποστήριξης του με την ελάχιστη τιμή υποστήριξης (minimum support threshold) που έχει οριστεί και διατηρούνται μόνο εκείνα που έχουν τιμή υποστήριξης ίση ή μεγαλύτερη από την ελάχιστη τιμή υποστήριξης. Κατ' επέκταση, προκύπτει το σύνολο F2 της παρακάτω εικόνας, το οποίο αποτελεί το σύνολο των συχνών 2-στοιχειοσυνόλων (των στοιχειοσυνόλων μήκους 2).

2-Itemset	Support
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

Πίνακας 3.4: Ο πίνακας με τα συχνών 2-στοιχειοσύνολα (F2)

Στην τρίτη επανάληψη για μέγεθος k = 3, αναζητούνται τα συχνά στοιχειοσύνολα μήκους 3.

Αντίστοιχα με την προηγούμενη επανάληψη, για να σχηματιστεί το σύνολο των υποψήφιων στοιχειοσυνόλων C3, πραγματοποιείται το Join Step του αλγορίθμου Apriori, εν προκειμένω πραγματοποιείται ένα Self Join για το σύνολο F2 (ένωση του κάθε συχνού στοιχειοσυνόλου μηκους 2, με τα υπόλοιπα συχνά στοιχειοσύνολα μήκους 2 υπό την συνθήκη ύπαρξης ενός κοινού στοιχείου), με το οποίο σχηματίζεται κάθε ένα υποψήφιο στοιχειοσύνολο μήκους 3. Το παραπάνω οδηγεί στο συμπέρασμα πως το πρώτο στοιχείο πρέπει να είναι κοινό. Με βάση όλα τα παραπάνω σχηματίζονται

τα υποψήφια στοιχειοσύνολα μήκους 3 δηλαδή C3 = ({I1,I2,I3},{I1,I2,I5},{I1,I3,I5},{I2,I3,I4},{I2,I3,I5},{I2,I4,I5}) τα οποία και αποτυπώνονται στην παρακάτω εικόνα.

Itemset	Support
I1,I2,I3	
I1,I2,I5	
I1,I3,I5	
I2,I3,I4	
I2,I3,I5	
I2,I4,I5	

Πίνακας 3.5: Ο πίνακας με τα υποψήφια 3-στοιχειοσύνολα.

Σε αυτό το σημείο όμως, προτού προχωρήσει η εκτέλεση του αλγορίθμου με τον υπολογισμό της τιμής υποστήριξης για κάθε ένα υποψήφιο στοιχειοσύνολο θα ελεγχθεί, εάν κάποιο από τα υποσύνολα για το εκάστοτε υποψήφιο στοιχειοσύνολο μήκους 3 είναι μη συχνό(δεν ικανοποιεί τον περιορισμό της ελάχιστης τιμής υποστήριξης, η τιμή υποστήριξής για το στοιχειοσύνολο αυτό είναι μικρότερη από την ελάχιστη), θα χρησιμοποιηθεί δηλαδή η ιδιότητα Apriori (Prune Step). Πιο συγκεκριμένα, για τα υποψήφια στοιχειοσύνολα ({I1,I2,I3},{I1,I2,I5},{I1,I3,I5},{I2,I3,I4},{I2,I3,I5},{I2,I4,I5}) και λαμβάνοντας υπόψη την εικόνα 3.2.4, προκύπτει ότι για το υποψήφιο στοιχειοσύνολο {I1,I3,I5} το υποσύνολο του {I3,I5} είναι μη συχνό καθώς η τιμή υποστήριξης του είναι ίση με 1, ενώ το κατώφλι της τιμής υποστήριξης για το παράδειγμα είναι 2. Επίσης, για το υποψήφιο στοιχειοσύνολο {I2,I3,I4} αντίστοιχα το υποσύνολο {I3,I4} έχει τιμή υποστήριξης ίση με 0 είναι δηλαδή μη συχνό. Για τα υποψήφια στοιχειοσύνολα {I2,I3,I5} και {I2,I4,I5}, ισχύει κατ' αντιστοιχία ότι τα υποσύνολα τους για το μεν {I3,I5}, του υποψήφιου στοιχειοσυνόλου {I2,I3,I5} και το δε {I4,I5} για το υποψήφιο στοιχειοσύνολο {I2,I4,I5}, έχουν τιμή υποστήριξης ίση με 1 και 0 αντίστοιχα και δεν ικανοποιούν την ελάχιστη τιμή υποστήριξης. Επομένως προκύπτει ότι τα υποψήφια 3-στοιχειοσύνολα είναι τα παρακάτω στην εικόνα 3.2.7 και σημειώνεται η τιμή υποστήριξης για κάθε ένα από αυτά.

Itemset	Support
I1,I2,I3	2
I1,I2,I5	2

Πίνακας 3.6: Ο πίνακας με τα υποψήφια 3-στοιχειοσύνολα (C3)

Με βάση τις παραπάνω τιμές υποστήριξης, κάθε ένα υποψήφιο 3-στοιχειοσύνολο συγκρίνεται με την ελάχιστη τιμή υποστήριξης και διατηρείται μόνο εκείνο που έχει τιμή υποστήριξης μεγαλύτερη ή ίση από την ελάχιστη. Κατά συνέπεια, προκύπτει το σύνολο F3, το οποίο αποτελείται από τα συχνά 3-στοιχειοσύνολα και αποτυπώνεται στην παρακάτω εικόνα.

3-Itemset	Support
I1,I2,I3	2
I1,I2,I5	2

Πίνακας 3.7: Ο πίνακας με τα συχνά 3-στοιχειοσύνολα (F3)

Στην τέταρτη επανάληψη για μέγεθος k = 4, αναζητούνται τα συχνά στοιχειοσύνολα μήκους 4.

Αντίστοιχα και με τις προηγούμενες επαναλήψεις το σύνολο C4 σχηματίζεται από το F3 και το υποψήφιο στοιχειοσύνολο μήκους 4 που προκύπτει είναι το {I1,I2,I3,I5}. Σύμφωνα και με τα όσα έχουν προηγηθεί έως τώρα το παραπάνω υποψήφιο στοιχειοσύνολο, ικανοποιεί τη συνθήκη που υπάρχει προκειμένου να σχηματιστεί το C4 και στην συγκεκριμένη περίπτωση υφίστανται τα κοινά στοιχεία (I1,I2). Πηγαίνοντας στη συνέχεια στον έλεγχο, για κάθε ένα υποσύνολο του υποψήφιου 4-στοιχειοσυνόλου {I1,I2,I3,I5}, προκύπτει ότι τουλάχιστον ένα υποσύνολο {I2,I3,I5} είναι μη συχνό, καθώς έχει τιμή υποστήριξης ίση με 1. Με αυτόν τον τρόπο παύει ο έλεγχος για το συγκεκριμένο υποψήφιο στοιχειοσύνολο και προκύπτει ότι το Σύνολο C4 είναι κενό, καθώς δεν μπορούν να σχηματιστούν νέα υποψήφια στοιχειοσύνολα μήκους 4, και κατα συνέπεια και το σύνολο F4. Ένα γεγονός που προϊδεάζει την μη ύπαρξη συχνού στοιχειοσυνόλου μήκους 4 είναι και το πλήθος των συχνών στοιχειοσυνόλων στο σύνολο F3(το σύνολο των συχνών στοιχειοσυνόλων μήκους 3), καθώς θα έπρεπε να υπάρχουν τουλάχιστον (4,3) = 4! / (3!*1!) = 4 συχνά 3-στοιχειοσύνολα (στοιχειοσύνολα μήκους 3), ενώ στην περίπτωση του παραδείγματος υπάρχουν μόλις δύο. Σε αυτό το σημείο τερματίζει ο αλγόριθμος και σχηματίζεται το σύνολο των συχνών στοιχειοσυνόλων το οποίο περιλαμβάνει:

Τα στοιχειοσύνολα μήκους 1: {I1},{I2},{I3},{I4},{I5}.

Τα στοιχειοσύνολα μήκους 2:{I1,I2},{I1,I3},{I1,I5},{I2,I3},{I2,I4},{I2,I5}.

Τα στοιχειοσύνολα μήκους 3:{I1,I2,I3},{I1,I2,I5}.

Το αμέσως επόμενο βήμα αποτελεί η παραγωγή των συνδυαστικών κανόνων(Association Rules):

1)I1=>I2 Εμπιστοσύνη (Confidence) = Supp(I1&I2) / Supp(I1) = 4 / 6 = 66%.

2)I2=>I1 Εμπιστοσύνη (Confidence) = Supp(I1&I2) / Supp(I2) = 4 / 7 = 57%.

3)I1=>I3 Εμπιστοσύνη (Confidence) = Supp(I1&I3) / Supp(I1) = 4 / 6 = 66%.

4)I3=>I1 Εμπιστοσύνη (Confidence) = Supp(I1&I3) / Supp(I3) = 4 / 6 = 66%.

5)I1=>I5 Εμπιστοσύνη (Confidence) = Supp(I1&I5) / Supp(I1) = 2 / 6 = 33%.

6)I5=>I1 Εμπιστοσύνη (Confidence) = Supp(I1&I5) / Supp(I5) = 2 / 2 = 100%.

7)I2=>I3 Εμπιστοσύνη (Confidence) = Supp(I2&I3) / Supp(I2) = 4 / 7 = 57%.

8)I3=>I2 Εμπιστοσύνη (Confidence) = Supp(I2&I3) / Supp(I3) = 4 / 6 = 66%.

9)I2=>I4 Εμπιστοσύνη (Confidence) = Supp(I2&I4) / Supp(I2) = 2 / 7 = 28%.

10)I4=>I2 Εμπιστοσύνη (Confidence) = Supp(I2&I4) / Supp(I4) = 2 / 2 = 100%.

11)I2=>I5 Εμπιστοσύνη (Confidence) = Supp(I2&I5) / Supp(I2) = 2 / 7 = 28%.

12)I5=>I2 Εμπιστοσύνη (Confidence) = Supp(I2&I5) / Supp(I5) = 2 / 2 = 100%.

13)(I1&I2=>I3) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I1&I2) = 2 / 4 = 50%

14)(I1&I3=>I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I1&I3) = 2 / 4 = 50%

15)(I2&I3=>I1) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I2&I3) = 2 / 4 = 50%

16)(I1=>I2&I3) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I1) = 2 / 6 = 33%

17)(I2=>I1&I3) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I2) = 2 / 7 = 28%

18)(I3=>I1&I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I3) = 2 / 6 = 33%

19)(I1&I2=>I5) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I1&I2) = 2 / 4 = 50%

20)(I1&I5=>I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I1&I5) = 2 / 2 = 100%

21)(I2&I5=>I1) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I2&I5) = 2 / 2 = 100%

22)(I1=>I2&I5) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I1) = 2 / 6 = 33%

23)(I2=>I1&I5) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I2) = 2 / 7 = 28%

24)(I5=>I1&I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I5) = 2 / 2 = 100%

3.2.3 Incremental mining

Σύμφωνα με την έκδοση του αλγορίθμου Apriori η οποία περιγράφεται από τον Agrawal [8] ο συγκεκριμένος αλγόριθμος δεν υποστηρίζει το incremental mining. Ωστόσο σύμφωνα με τη βιβλιογραφία έχουν δημιουργηθεί εκδόσεις του αλγορίθμου Apriori οι οποίες υποστηρίζουν το incremental mining [65].

3.2.4 Συμπεράσματα

Ο αλγόριθμος Apriori, εξετάστηκε σε βάθος μέσα από τον τρόπο με τον οποίο υλοποιείται σε θεωρητικό επίπεδο όσο και μέσα από ένα αναλυτικό παράδειγμα περιγράφοντας βήμα βήμα την πορεία εκτέλεσής του. Παρακάτω στο έκτο κεφάλαιο συγκρίνεται ο αλγόριθμος Apriori με τους υπόλοιπους δύο αλγορίθμους στην πράξη και εξετάζεται η επίδοση του σε διάφορα σενάρια. Ένα χρήσιμο συμπέρασμα που προκύπτει είναι ότι μπορεί να οδηγήσει υπό ορισμένες συνθήκες (ιδιαίτερα χαμηλές τιμές υποστήριξης) σε τεράστιους χρόνους εκτέλεσης και εξάντληση της διαθέσιμης μνήμης, γι'αυτό άλλωστε στην υλοποίηση του αλγορίθμου στο πακέτο arules υπάρχει πρόβλεψη για την αποφυγή τέτοιου είδους γεγονότων μέσα από τον ορισμό συγκεκριμένης παραμέτρου (maxtime) με την κατάλληλη τιμή

3.3 Ο αλγόριθμος Eclat

Ο αλγόριθμος Eclat (Equivalence Class Clustering and bottom-up Lattice Traversal) [27], αποτελεί επίσης έναν από τους δημοφιλέστερους αλγορίθμους στην εύρεση τόσο συχνών στοιχειοσυνόλων (Frequent Itemsets) όσο και συνδυαστικών κανόνων [28]. Συγκριτικά με τον αλγόριθμο Apriori ο αλγόριθμος Eclat, αποτελεί μία πιο αποδοτική και πιο επεκτάσιμη έκδοση έναντι του πρώτου [29]. Αυτό συμβαίνει καθώς ο αλγόριθμος Apriori εκτελείται όπως τέθηκε και πιο πάνω οριζόντια δίνοντας βάση ανα επίπεδο (Level-Wise), με τον ίδιο τρόπο με τον οποίο πραγματοποιείται μία Breadth-First αναζήτηση ενός γράφου. Ο χαρακτηρισμός Breadth First algorithm, αποδίδεται αρκετά σύχνά και για τον ίδιο τον αλγόριθμο Apriori, χαρακτηρισμός που προέρχεται από τον τρόπο με τον οποίο εκτελείται. Αντιθέτως ο αλγόριθμος Eclat εκτελείται κατακόρυφα δίνοντας βάση στο βάθος, όπως αποτυπώνεται και μέσα από το παράδειγμα παρακάτω, κατ'αντιστοιχία με μία Depth-First αναζήτηση ενός γράφου. Η

παραπάνω διαφοροποίηση, δηλαδή η κατακόρυφη προσέγγιση την οποία υιοθετεί ο αλγόριθμος Eclat, είναι αυτή η οποία καθιστά τον προαναφερθέντα αλγόριθμο γρηγορότερο από τον Apriori. Η βασική ιδέα του αλγορίθμου Eclat, είναι η χρήση των Transaction Id Sets (tidsets) [39], τα οποία είναι τα σύνολα με τις συναλλαγές οι οποίες περιέχουν τα εκάστοτε στοιχειοσύνολα. Οι τομές των συνόλων αυτών υπολογίζουν την τιμή υποστήριξης ενός υποψήφιου στοιχειοσυνόλου [9],[29]. Αρχικά με την πρώτη κλήση του αλγορίθμου, χρησιμοποιούνται όλα τα στοιχειοσύνολα μήκους ένα μαζί με τα tidesets τους. Στη συνέχεια ο αλγόριθμος καλείται αναδρομικά και σε κάθε αναδρομική κλήση, κάθε ένα στοιχειοσύνολο το οποίο συνοδεύεται από το tidset του συνδυάζεται μαζί με τα υπόλοιπα στοιχειοσύνολα τα οποία επίσης συνοδεύονται από τα tidsets τους. Ο συνδυασμός αυτός σχηματίζει τα υποψήφια στοιχειοσύνολα για τα οποία υπολογίζεται η υποστήριξή τους μέσα από την τομή των tidsets των δύο στοιχειοσυνόλων που συνθέτουν το υποψήφιο στοιχειοσύνολο. Η διαδικασία αυτή συνεχίζεται έως ότου κανένα άλλο υποψήφιο στοιχειοσύνολο μπορεί να σχηματιστεί [28],[30] .

Αλγόριθμος Eclat για την εύρεση των συχνών στοιχειοσυνόλων:

Input: Η βάση η οποία συμβολίζεται με D, η ελάχιστη τιμή υποστήριξης minsup σ και το στοιχειοσύνολο *I* το οποίο για την αρχική κλίση είναι ίσο με *I* = { } [6].

Output: Το σύνολο των συχνών στοιχειοσυνόλων.

3.3.1 Βήματα εκτέλεσης του αλγορίθμου Eclat

Έχοντας μία δοσμένη βάση συναλλαγών D (Transactional Database), μία ελάχιστη τιμή υποστήριξης σ (Minimum Support) και το στοιχειοσύνολο *I* το οποίο για την αρχική κλίση είναι ίσο με *I* = { } [28].

1. Αρχικά, ο αλγόριθμος μετατρέπει τη βάση των συναλλαγών D από την οριζόντια (horizontal layout) στην κατακόρυφη της μορφή (vertical layout) [40], όπου κάθε ένα στοιχειοσύνολο αποθηκεύεται μαζί με το σύνολο των συναλλαγών στις οποίες εντοπίζεται (Tidset), κάθε στοιχειοσύνολο που δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης δεν λαμβάνεται υπόψη.
2. Κάθε ένα συχνό στοιχειοσύνολο i τοποθετείται στο σύνολο με τα συχνά στοιχειοσύνολα (γραμμή 3).
3. Κατόπιν για κάθε ένα τέτοιο συχνό στοιχειοσύνολο i δημιουργείται η i-projected βάση δεδομένων Di, η οποία περιέχει κάθε ένα στοιχειοσύνολο j το οποίο συναντάται συχνά μαζί με το i (γραμμές 5-9).
4. Στη συνέχεια υπολογίζεται η τιμή υποστήριξης για κάθε ένα τέτοιο σετ {i,j}, η οποία προκύπτει από την τομή των συνόλων των συναλλαγών για τα στοιχειοσύνολα i και j (γραμμή 7). Εφόσον το σύνολο {i,j} είναι συχνό, ικανοποιεί δηλαδή την ελάχιστη τιμή υποστήριξης, τότε στοιχειοσύνολο j μαζί με τα σύνολο των συναλλαγών στις οποίες συναντάται (Tidset), τοποθετείται στην i-projected βάση δεδομένων Di.
5. Έπειτα ο αλγόριθμος καλείται αναδρομικά, προκειμένου να βρεθεί το σύνολο των συχνών στοιχειοσυνόλων στην νέα βάση Di, στην οποία πλέον κάθε υποψήφιο στοιχειοσύνολο αντιπροσωπεύεται από κάθε σετ I {i,j}.

Ορισμός του cover:

Ως cover ενός στοιχειοσυνόλου X στη βάση των συναλλαγών D, ορίζεται ως το σύνολο των αναγνωριστικών των συναλλαγών (tidset-Transaction Identifiers Set) στη βάση D, οι οποίες υποστηρίζουν το στοιχειοσύνολο X, συναντάται δηλαδή το στοιχειοσύνολο X στις συναλλαγές που απαρτίζουν το σύνολο (tidset) [9],[28]. Σύμφωνα με τα παραπάνω ισχύει:

$$\text{cover}(X, D) := \{\text{tid} \mid (\text{tid}, I) \in D, X \subseteq I\}.$$

Επεξήγηση των συμβόλων στον αλγόριθμο Eclat:

D: Η βάση των συναλλαγών η οποία αναπαρίσταται κάθετα, υλοποιεί δηλαδή την κατακόρυφη μορφή αναπαράστασης (Vertical Data Layout).

σ: Η ελάχιστη τιμή υποστήριξης(Minimum Support).

I: Συμβολίζει το στοιχειοσύνολο. Στην αρχική κλίση του αλγορίθμου ισχύει ότι $I = \{\ \}$ [6].

F[I]: Το σύνολο των συχνών στοιχειοσυνόλων.

D^i: Η i-projected βάση Di ή κλάση ισοδυναμίας (equivalence class) για το στοιχειοσύνολο i [28].

Input: $\mathcal{D}, \sigma, I \subseteq \mathcal{I}$
Output: $\mathcal{F}[I](\mathcal{D}, \sigma)$

```
1:  F[I] := {}
2:  for all i ∈ I occurring in D do
3:      F[I] := F[I] ∪ {I ∪ {i}}
4:      // Create D^i
5:      D^i := {}
6:      for all j ∈ I occurring in D such that j > i do
7:          C := cover({i}) ∩ cover({j})
8:          if |C| ≥ σ then
9:              D^i := D^i ∪ {(j, C)}
10:         end if
11:     end for
12:     // Depth-first recursion
13:     Compute F[I ∪ {i}](D^i, σ)
14:     F[I] := F[I] ∪ F[I ∪ {i}]
15: end for
```

Σχήμα 3.11: Ο αλγόριθμος Eclat [28]

3.3.2 Παράδειγμα εκτέλεσης του αλγορίθμου Eclat

Μέσα από το παρακάτω σύνολο δεδομένων (DataSet), θα βρεθούν σε πρώτη φάση τα συχνά στοιχειοσύνολα (Frequent Itemsets), τα οποία θα οδηγήσουν, σε δεύτερη φάση, στην παραγωγή των συνδυαστικών κανόνων.

TID	Items
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

Σχήμα 3.12: Η βάση με τις συναλλαγές του παραδείγματος

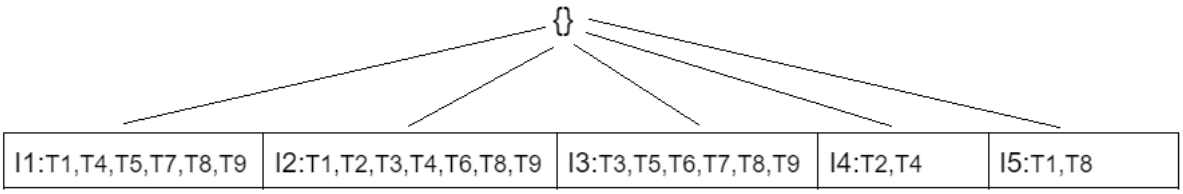
Ορίζεται ως ελάχιστη τιμή υποστήριξης (Minimum Support) η τιμή 0.22 ή 22% (αντίστοιχα με το προηγούμενο παράδειγμα για να χαρακτηριστεί ένα υποψήφιο στοιχειοσύνολο μεγέθους k ως συχνό, θα πρέπει να συναντάται σε τουλάχιστον δύο από τις εννέα συναλλαγές της βάσης των συναλλαγών).

Αρχικά η βάση με τις συναλλαγές από την οριζόντια μορφή (Horizontal Format) μετατρέπεται στην κατακόρυφη της μορφή (Vertical Format). Παρατηρείται, ότι όλα τα υποψήφια στοιχειοσύνολα ικανοποιούν την ελάχιστη τιμή υποστήριξης (minimum support) και σχηματίζεται το σύνολο των συχνών στοιχειοσυνόλων μήκους 1.

Item	TIDset
I1	T1,T4,T5,T7,T8,T9
I2	T1,T2,T3,T4,T6,T8,T9
I3	T3,T5,T6,T7,T8,T9
I4	T2,T4
I5	T1,T8

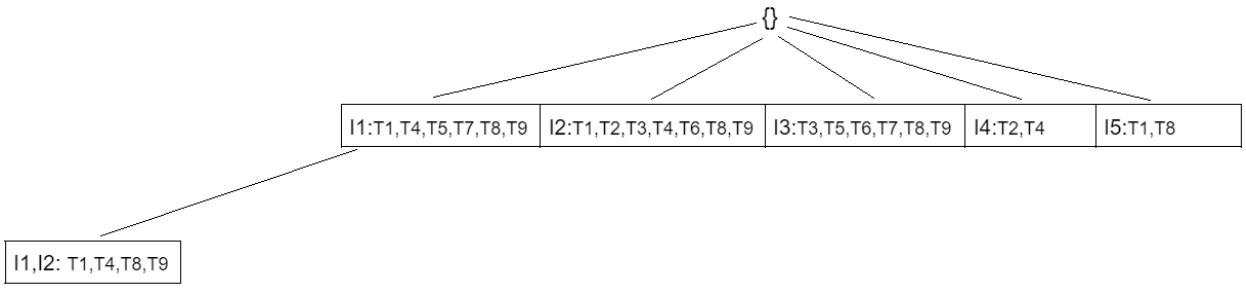
Σχήμα 3.13: Η βάση με τις συναλλαγές του παραδείγματος στην απαιτούμενη μορφή

Σύμφωνα με τον αλγόριθμο Eclat παραπάνω, στην αρχική είσοδο του αλγορίθμου περιλαμβάνεται και το σύνολο των συχνών στοιχειοσυνόλων μήκους 1 (σχήμα 3.13), που έχουν προκύψει από την σάρωση της βάσης. Το συγκεκριμένο σύνολο όπως αποτυπώνεται και από το παρακάτω σχήμα (σχήμα 3.14), έχει ως κοινό πρόθεμα το κενό πρόθεμα. Κάθε ένα στοιχειοσύνολο συνοδεύεται από τις συναλλαγές στις οποίες εντοπίζεται και το πλήθος των συναλλαγών αυτών δίνει την υποστήριξη για το στοιχειοσύνολο αυτό.



Σχήμα 3.14 Η αρχική είσοδος του αλγορίθμου Eclat

Ξεκινώντας αρχικά από το πρώτο συχνό στοιχειοσύνολο μήκους 1, το οποίο είναι το στοιχειοσύνολο {I1}, η τιμή υποστήριξης του είναι ίση με έξι και προκύπτει από το πλήθος των συναλλαγών στις οποίες εντοπίζεται. Στη συνέχεια αυτό που ακολουθεί είναι ο συνδυασμός του στοιχειοσυνόλου {I1} με τα υπόλοιπα συχνά στοιχειοσύνολα μήκους ένα, ώστε να εξεταστεί ποιά από τα υποψήφια στοιχειοσύνολα που σχηματίζονται είναι κι αυτά με τη σειρά τους συχνά, ικανοποιούν δηλαδή την ελάχιατη τιμή υποστήριξης που έχει οριστεί. Ο τρόπος υπολογισμού της υποστήριξης για το κάθε ένα υποψήφιο στοιχειοσύνολο {I1,j} αποτελεί η τομή των συναλλαγών που περιέχουν το ένα στοιχειοσύνολο, με τις συναλλαγές που περιέχουν το άλλο στοιχειοσύνολο όπως φαίνεται και παρακάτω. Οπότε, δημιουργείται πλέον το δεύτερο επίπεδο (στοιχειοσύνολα μήκους 2), το οποίο αρχικά περιέχει το υποψήφιο στοιχειοσύνολο {I1,I2}:



Σχήμα 3.15 Έλεγχος για το στοιχειοσύνολο {I1,I2}

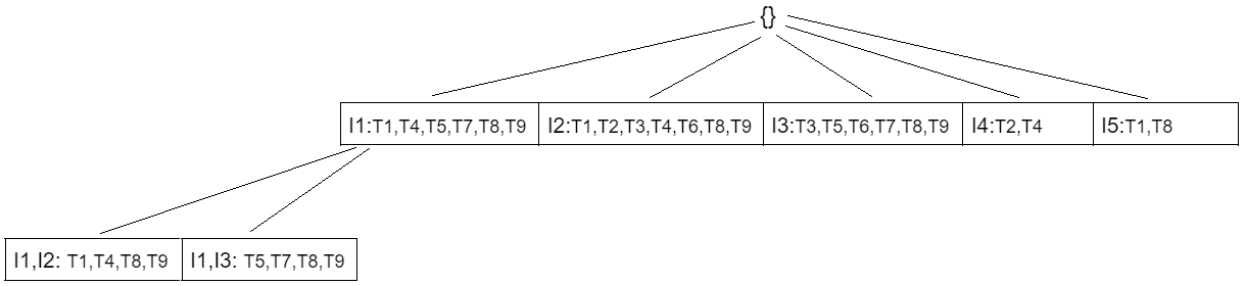
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I1,I2}:

Item	TIDset
I1	T1,T4,T5,T7,T8,T9
I2	T1,T2,T3,T4,T6,T8,T9

Σχήμα 3.16: Τα σύνολα {Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται, ότι οι συναλλαγές που περιέχουν το υποψήφιο στοιχειοσύνολο {I1,I2} είναι: t(I1,I2) = t(I1) ∩ t(I2) = T1,T4,T5,T7,T8,T9 ∩ T1,T2,T3,T4,T6,T8,T9 = T1,T4,T8,T9. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2}. Προκύπτει κατά συνέπεια πως το υποψήφιο στοιχειοσύνολο {I1,I2}, που προέρχεται από τον συνδυασμό των στοιχειοσυνόλων {I1} και {I2}, είναι συχνό καθώς ικανοποιεί την ελάχιστη τιμή υποστήριξης (minimum support).

Κατόπιν, ελέγχεται το επόμενο στη σειρά υποψήφιο στοιχειοσύνολο {I1,I3}:



Σχήμα 3.17: Έλεγχος για το στοιχειοσύνολο{I1,I3}.

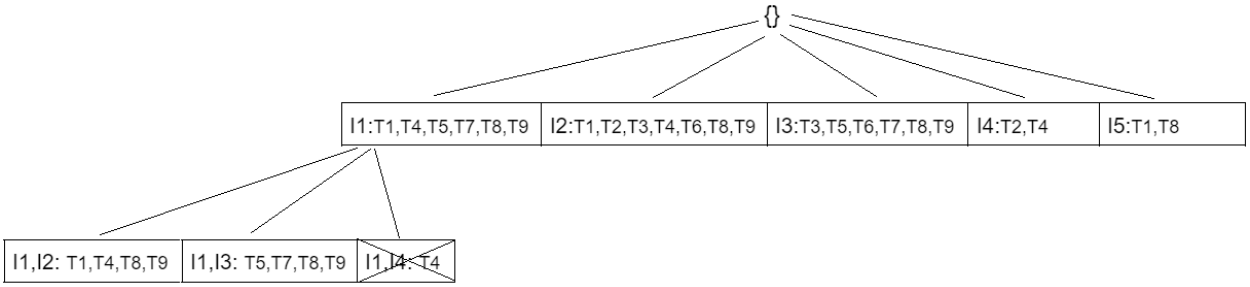
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I1,I3}:

Item	TIDset
I1	T1,T4,T5,T7,T8,T9
I3	T3,T5,T6,T7,T8,T9

Σχήμα 3.18: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται ότι οι συναλλαγές που περιέχουν το υποψήφιο στοιχειοσύνολο {I1,I3} είναι: t(I1,I3) = t(I1) ∩ t(I3) = T1,T4,T5,T7,T8,T9 ∩ T3,T5,T6,T7,T8,T9 = T5,T7,T8,T9. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I3}. Προκύπτει και σε αυτό το σημείο πως το υποψήφιο στοιχειοσύνολο {I1,I3}, που προέρχεται από τον συνδυασμό των στοιχειοσυνόλων {I1} και {I3}, είναι συχνό καθώς ικανοποιεί την ελάχιστη τιμή υποστήριξης (minimum support).

Κατόπιν εξετάζεται το επόμενο υποψήφιο στοιχειοσύνολο {I1,I4}:



Σχήμα 3.19: Έλεγχος για το στοιχειοσύνολο{I1,I4}.

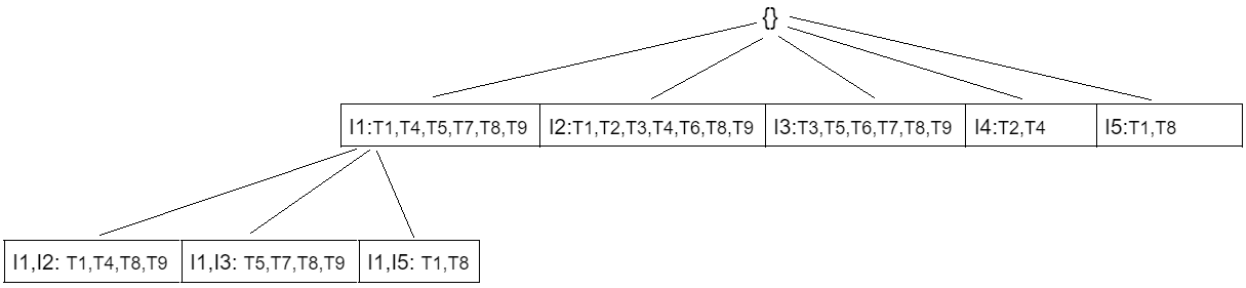
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I1,I4}:

Item	TIDset
I1	T1,T4,T5,T7,T8,T9
I4	T2,T4

Σχήμα 3.20: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται ότι η συναλλαγή που περιέχει το το υποψήφιο στοιχειοσύνολο {I1,I4} είναι: t(I1,I4) = t(I1) ∩ t(I4) = T1,T4,T5,T7,T8,T9 ∩ T2,T4 = T4. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I4}. Στη συγκεκριμένη περίπτωση, το πλήθος των συναλλαγών που περιέχουν το συγκεκριμένο υποψήφιο στοιχειοσύνολο δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης, οπότε και απορρίπτεται.

Κατόπιν ελέγχεται το επόμενο υποψήφιο στοιχειοσύνολο {I1,I5}:



Εικόνα 3.21: Έλεγχος για το στοιχειοσύνολο{I1,I5}.

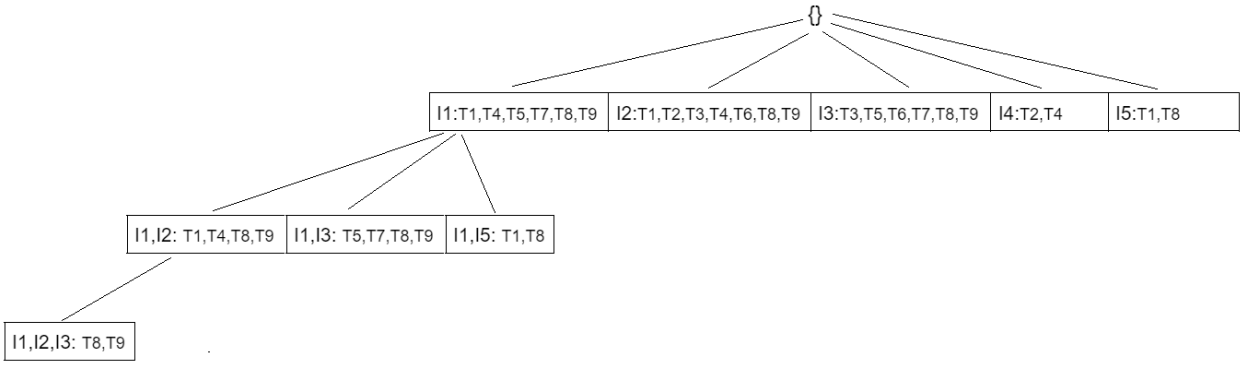
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I1,I5}:

Item	TIDset
I1	T1,T4,T5,T7,T8,T9
I5	T1,T8

Εικόνα 3.22: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται ότι οι συναλλαγές που περιέχουν το υποψήφιο στοιχειοσύνολο {I1,I5} είναι: t(I1,I5) = t(I1) ∩ t(I5) = T1,T4,T5,T7,T8,T9 ∩ T1,T8 = T1,T8. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I5}. Προκύπτει κι εδώ πως το υποψήφιο στοιχειοσύνολο {I1,I5}, που προέρχεται από τον συνδυασμό των στοιχειοσυνόλων {I1} και {I5}, είναι συχνό καθώς ικανοποιεί την ελάχιστη τιμή υποστήριξης (minimum support).

Έως τώρα έχουν βρεθεί τα συχνά στοιχειοσύνολα μήκους 2: {I1,I2}, {I1,I3}, {I1,I5}, οπότε στο επόμενο βήμα εξετάζονται τα υποψήφια στοιχειοσύνολα μήκους 3, τα οποία και θα οδηγήσουν στην εύρεση των συχνών στοιχειοσυνόλων μήκους 3. Τα υποψήφια στοιχειοσύνολα σχηματίζονται όπως και στον αλγόριθμο Apriori με τη χρήση του Join Step [6],[9]. Έτσι τα υποψήφια στοιχειοσύνολα είναι {I1,I2,I3}, {I1,I2,I5}, {I1,I3,I5}. Οπότε, δημιουργείται πλέον το τρίτο επίπεδο (στοιχειοσύνολα μήκους 3), το οποίο αρχικά περιέχει το υποψήφιο στοιχειοσύνολο {I1,I2,I3}. Ελέγχεται δηλαδή η τιμή υποστήριξης για το συγκεκριμένο υποψήφιο στοιχειοσύνολο, ώστε να διαπιστωθεί, εάν το συγκεκριμένο υποψήφιο στοιχειοσύνολο είναι συχνό ή όχι.



Εικόνα 3.23: Έλεγχος για το στοιχειοσύνολο{I1,I2,I3}

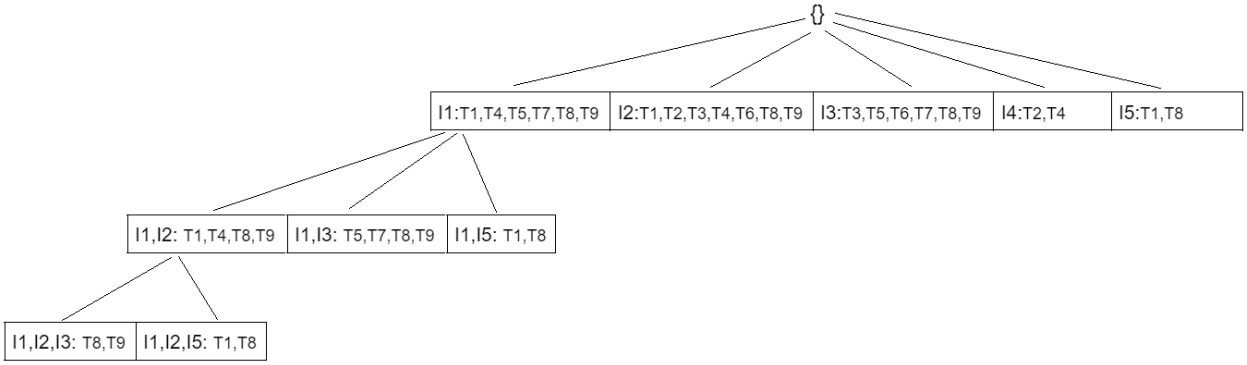
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I1,I2,I3}:

Item	TIDset
I1,I2	T1,T4,T8,T9
I1,I3	T5,T7,T8,T9

Εικόνα 3.24: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα

Παρατηρείται, ότι οι συναλλαγές που περιέχουν το υποψήφιο στοιχειοσύνολο {I1,I2,I3} είναι: t(I1,I2,I3) = t(I1,I2) ∩ t(I1,I3) = T1,T4,T8,T9 ∩ T5,T7,T8,T9 = T8,T9. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο{I1,I2} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1,I3}. Προκύπτει επομένως πως το υποψήφιο στοιχειοσύνολο {I1,I2,I3}, που προέρχεται από τον συνδυασμό των στοιχειοσυνόλων {I1,I2} και {I1,I3}, είναι συχνό καθώς ικανοποιεί την ελάχιστη τιμή υποστήριξης (minimum support), έχει δηλαδή τιμή υποστήριξης ίση με την ελάχιστη.

Κατόπιν ελέγχεται το υποψήφιο στοιχειοσύνολο {I1,I2,I5}:



Σχήμα 3.25: Έλεγχος για το στοιχειοσύνολο {I1,I2,I5}

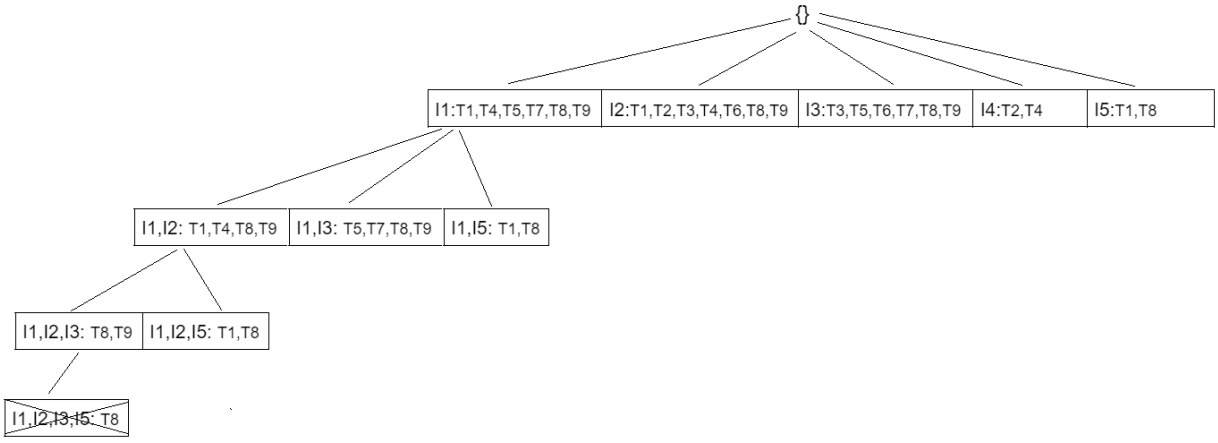
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I1,I2,I5}:

Item	TIDset
I1,I2	T1,T4,T8,T9
I1,I5	T1,T8

Σχήμα 3.26: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα

Παρατηρείται ότι οι συναλλαγές που περιέχουν το υποψήφιο στοιχειοσύνολο {I1,I2,I5} είναι: t(I1,I2,I5) = t(I1,I2) ∩ t(I1,I5) = T1,T4,T8,T9 ∩ T1,T8 = T1,T8. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1,I2} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1,I5}. Στη συγκεκριμένη περίπτωση προκύπτει επίσης πως το υποψήφιο στοιχειοσύνολο {I1,I2,I5}, που προέρχεται από τον συνδυασμό των στοιχειοσυνόλων {I1,I2} και {I1,I5}, είναι συχνό καθώς ικανοποιεί την ελάχιστη τιμή υποστήριξης (minimum support), έχει και σε αυτή την περίπτωση τιμή υποστήριξης ίση με την ελάχιστη. Στο αμέσως επόμενο βήμα του αλγορίθμου εξετάζεται το υποψήφιο στοιχειοσύνολο μήκους 4, το οποίο είναι το βαθύτερο επίπεδο που εξετάζεται στο συγκεκριμένο παράδειγμα, ώστε να διαπιστωθεί εάν το υποψήφιο στοιχειοσύνολο χαρακτηρίζεται ως συχνό και ικανοποιεί την ελάχιστη τιμή υποστήριξης.

Ακολουθεί ο έλεγχος του υποψηφίου στοιχειοσυνόλου {I1,I2,I3,I5}:



Σχήμα 3.27: Έλεγχος για το στοιχειοσύνολο {I1,I2,I3,I5}

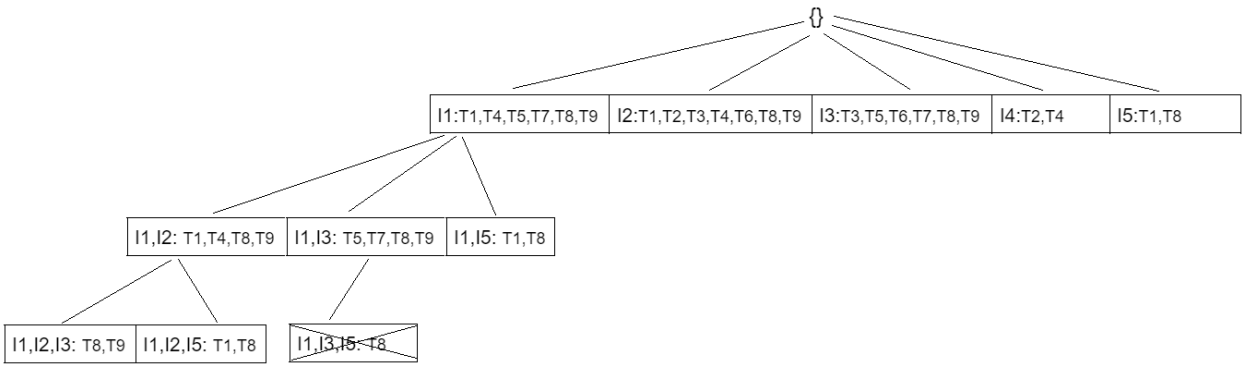
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I1,I2,I3,I5}:

Item	TIDset
I1,I2,I3	T8,T9
I1,I2,I5	T1,T8

Σχήμα 3.28: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται ότι η συναλλαγή που περιέχει το υποψήφιο στοιχειοσύνολο {I1,I2,I3,I5} είναι: t(I1,I2,I3,I5) = t(I1,I2,I3) ∩ t(I1,I2,I5) = T8,T9 ∩ T1,T8 = T8. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1,I2,I3} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1,I2,I5}. Στη συγκεκριμένη περίπτωση το πλήθος των συναλλαγών που περιέχουν το συγκεκριμένο υποψήφιο στοιχειοσύνολο δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης, οπότε και απορρίπτεται. Ο έλεγχος για το παραπάνω υποψήφιο στοιχειοσύνολο πραγματοποιήθηκε εξαιτίας της φύσης του αλγορίθμου Eclat, καθώς ο συγκεκριμένος αλγόριθμος αποτελεί έναν Depth First Search (DFS) αλγόριθμο.

Κατόπιν ελέγχεται το υποψήφιο στοιχειοσύνολο {I1,I3,I5}:



Σχήμα 3.29: Το δεύτερο συχνό στοιχειοσύνολο μήκους 1.

Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I1,I3,I5}:

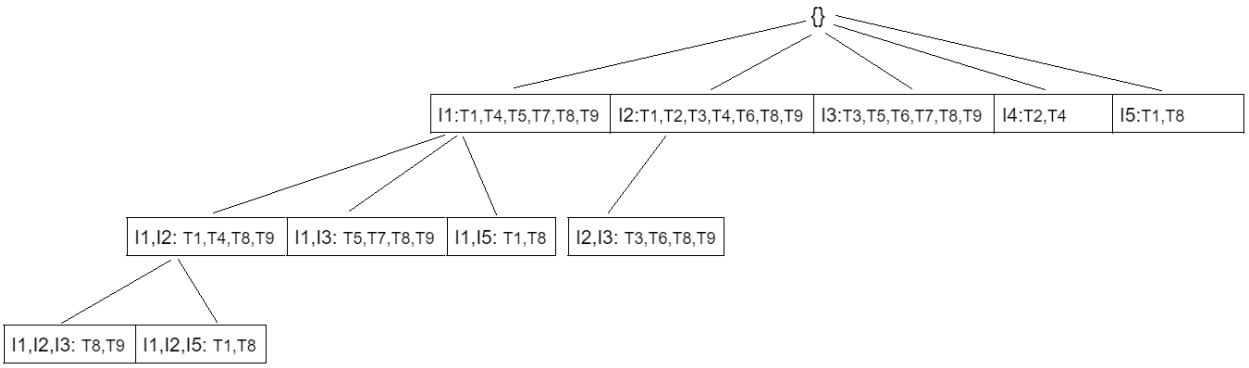
Item	TIDset
I1,I3	T5,T7,T8,T9
I1,I5	T1,T8

Σχήμα 3.30 Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται ότι η συναλλαγή που περιέχει το υποψήφιο στοιχειοσύνολο {I1,I3,I5} είναι: t(I1,I3,I5) = t(I1,I3) ∩ t(I1,I5) = T5,T7,T8,T9 ∩ T1,T8 = T8. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1,I3} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I1,I5}. Στη συγκεκριμένη περίπτωση, το πλήθος των συναλλαγών που περιέχουν το

συγκεκριμένο υποψήφιο στοιχειοσύνολο, δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης, οπότε και απορρίπτεται. Στο σημείο αυτό ολοκληρώθηκε ο πρώτος κύκλος στον οποίο εξετάστηκε το στοιχειοσύνολο {I1}. Τα συχνά στοιχειοσύνολα που βρέθηκαν μέχρι αυτό το σημείο είναι: {I1,I2}, {I1,I3}, {I1,I5}, {I1,I2,I3}, {I1,I2,I5}.

Πλέον, ο έλεγχος περνάει στο συχνό στοιχειοσύνολο {I2}, του οποίου η τιμή υποστήριξης είναι επτά και προκύπτει από το πλήθος των συναλλαγών στις οποίες εντοπίζεται το στοιχειοσύνολο αυτό. Εκ νέου ο έλεγχος βρίσκεται στο πρώτο επίπεδο (στοιχειοσύνολα μήκους 1) και μεταβαίνει αμέσως στο δεύτερο (στοιχειοσύνολα μήκους 2). Ακολουθεί δηλαδή ο συνδυασμός του στοιχειοσυνόλου {I2} με τα υπόλοιπα συχνά στοιχειοσύνολα μήκους ένα, τα οποία έχουν απομείνει, ώστε να εξεταστεί ποιά από τα υποψήφια στοιχειοσύνολα που σχηματίζονται είναι κι αυτά συχνά, ικανοποιούν δηλαδή την ελάχιατη τιμή υποστήριξης που έχει οριστεί. Άμεση συνέπεια για το παραπάνω, είναι η δημιουργία του δευτέρου επιπέδου, που θα περιέχει τους συνδυασμούς του στοιχειοσυνόλου {I2} με τα υπόλοιπα συχνά στοιχειοσύνολα μηκους ένα, και το οποίο αρχικά περιέχει το υποψήφιο στοιχειοσύνολο {I2,I3}. Ο τρόπος υπολογισμού του κάθε υποψήφιου στοιχειοσυνόλου {I2,j} είναι ο ίδιος όπως και πριν η τομή δηλαδή των συναλλαγών που περιέχουν το ένα στοιχειοσύνολο, με τις συναλλαγές που περιέχουν το άλλο στοιχειοσύνολο, καθένα από τα οποία στοιχειοσύνολα συνθέτουν το υποψήφιο στοιχειοσύνολο μήκους δύο {I2,j}.



Σχήμα 3.31: Έλεγχος για το στοιχειοσύνολο {I2,I3}.

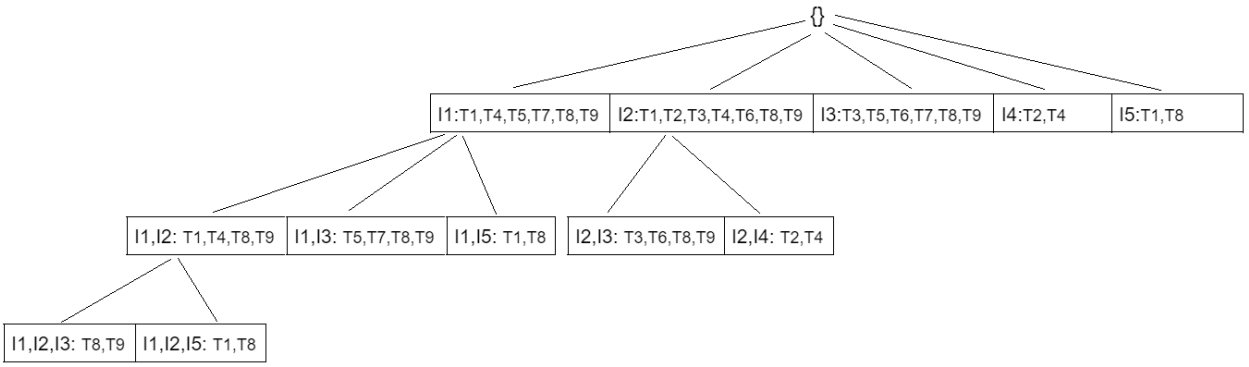
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I2,I3}:

Item	TIDset
I2	T1,T2,T3,T4,T6,T8,T9
I3	T3,T5,T6,T7,T8,T9

Σχήμα 3.32: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται ότι οι συναλλαγές που περιέχουν το υποψήφιο στοιχειοσύνολο {I2,I3} είναι: t(I2,I3) = t(I2) ∩ t(I3) = T1,T2,T3,T4,T6,T8,T9 ∩ T3,T5,T6,T7,T8,T9 = T3,T6,T8,T9. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I3}. Στη συγκεκριμένη περίπτωση προκύπτει πως το υποψήφιο στοιχειοσύνολο {I2,I3}, που προέρχεται από τον συνδυασμό των στοιχειοσυνόλων {I2} και {I3}, είναι συχνό καθώς ικανοποιεί την ελάχιστη τιμή υποστήριξης (minimum support).

Έπειτα ελέγχεται το επόμενο υποψήφιο στοιχειοσύνολο {I2,I4}:



Σχήμα 3.33: Έλεγχος για το στοιχειοσύνολο {I2,I4}

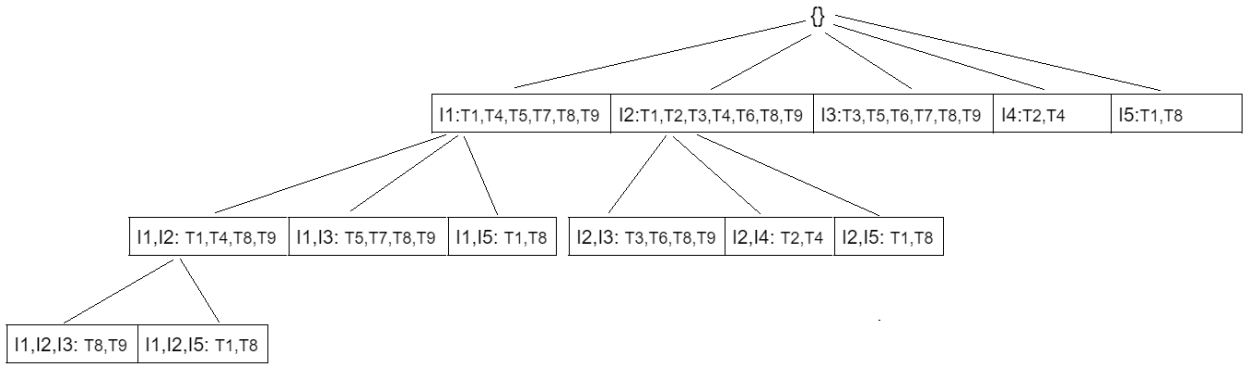
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I2,I4}:

Item	TIDset
I2	T1,T2,T3,T4,T6,T8,T9
I4	T2,T4

Σχήμα 3.34: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα

Παρατηρείται ότι οι συναλλαγές που περιέχουν το υποψήφιο στοιχειοσύνολο {I2,I4} είναι: t(I2,I4) = t(I2) ∩ t(I4) = T1,T2,T3,T4,T6,T8,T9 ∩ T2,T4 = T2,T4. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I4}. Στη συγκεκριμένη περίπτωση, προκύπτει επίσης πως το υποψήφιο στοιχειοσύνολο {I2,I4}, που προέρχεται από τον συνδυασμό των στοιχειοσυνόλων {I2} και {I4}, είναι συχνό καθώς ικανοποιεί την ελάχιστη τιμή υποστήριξης (minimum support).

Κατόπιν ελέγχεται το υποψήφιο στοιχειοσύνολο {I2,I5}:



Σχήμα 3.35: Έλεγχος για το στοιχειοσύνολο {I2,I5}

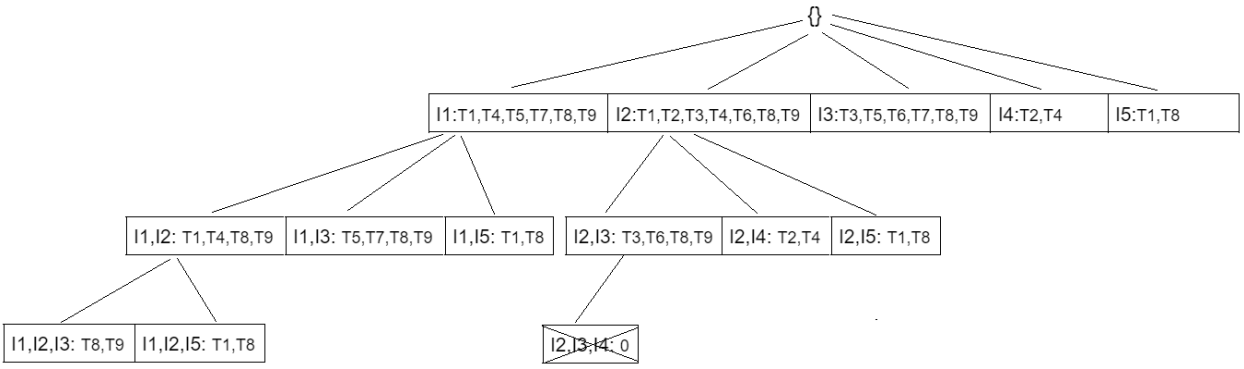
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I2,I5}:

Item	TIDset
I2	T1,T2,T3,T4,T6,T8,T9
I5	T1,T8

Σχήμα 3.36: Τα σύνολα {Item,TIDset} για τα στοιχειοσύνολα

Παρατηρείται ότι οι συναλλαγές που περιέχουν το υποψήφιο στοιχειοσύνολο {I2,I5} είναι: t(I2,I5) = t(I2) ∩ t(I5) = T1,T2,T3,T4,T6,T8,T9 ∩ T1,T8 = T1,T8. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I5}. Στη συγκεκριμένη περίπτωση προκύπτει πως το υποψήφιο στοιχειοσύνολο {I2,I5}, που προέρχεται από τον συνδυασμό των στοιχειοσυνόλων {I2} και {I5}, είναι συχνό καθώς ικανοποιεί την ελάχιστη τιμή υποστήριξης (minimum support). Εκ νέου ο έλεγχος περνάει στο τρίτο επίπεδο και εξετάζει τα υποψήφια στοιχειοσύνολα για το συγκεκριμένο επίπεδο, δηλαδή τα {I2,I3,I4}, {I2,I3,I5}, {I2,I4,I5}.

Αρχικά εξετάζεται το υποψήφιο στοιχειοσύνολο {I2,I3,I4}:



Σχήμα 3.37: Έλεγχος για το στοιχειοσύνολο {I2,I3,I4}.

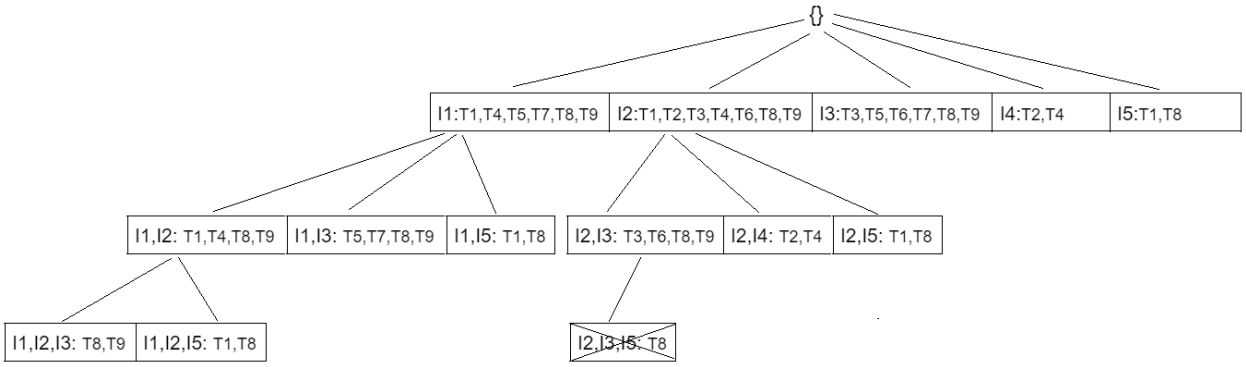
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I2,I3,I4}:

Item	TIDset
I2,I3	T3,T6,T8,T9
I2,I4	T2,T4

Σχήμα 3.38: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα

Παρατηρείται ότι καμία από τις συναλλαγές δεν περιέχει το υποψήφιο στοιχειοσύνολο {I2,I3,I4}, ισχύει δηλαδή ότι: t(I2,I3,I4) = t(I2,I3) ∩ t(I2,I4) = T3,T6,T8,T9 ∩ T2,T4 = 0. Η τομή δηλαδή, του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2,I3} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2,I4}, δεν περιέχει καμία συναλλαγή. Στη συγκεκριμένη περίπτωση, το υποψήφιο στοιχειοσύνολο δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης, οπότε και απορρίπτεται.

Έπειτα ελέγχεται το υποψήφιο στοιχειοσύνολο {I2,I3,I5}:



Σχήμα 3.39: Έλεγχος για το στοιχειοσύνολο {I2,I3,I5}.

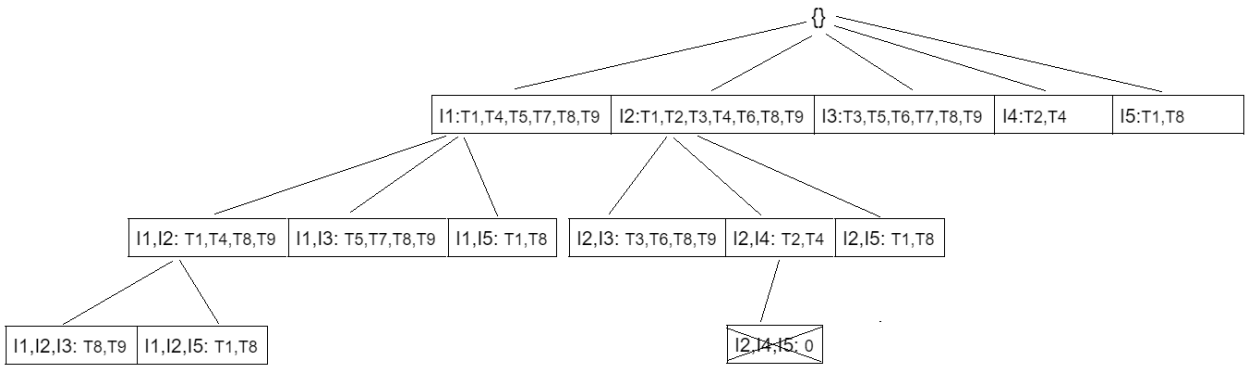
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I2,I3,I5}:

Item	TIDset
I2,I3	T3,T6,T8,T9
I2,I5	T1,T8

Σχήμα 3.40: Τα σύνολα {Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται ότι η συναλλαγή που περιέχει το υποψήφιο στοιχειοσύνολο {I2,I3,I5} είναι: t(I2,I3,I5) = t(I2,I3) ∩ t(I2,I5) = T3,T6,T8,T9 ∩ T1,T8 = T8. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2,I3} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2,I5} περιέχει μία συναλλαγή. Στη συγκεκριμένη περίπτωση, το υποψήφιο στοιχειοσύνολο δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης, οπότε και απορρίπτεται.

Κατόπιν ελέγχεται το υποψήφιο στοιχειοσύνολο {I2,I4,I5}:



Σχήμα 3.41: Έλεγχος για το στοιχειοσύνολο {I2,I4,I5}

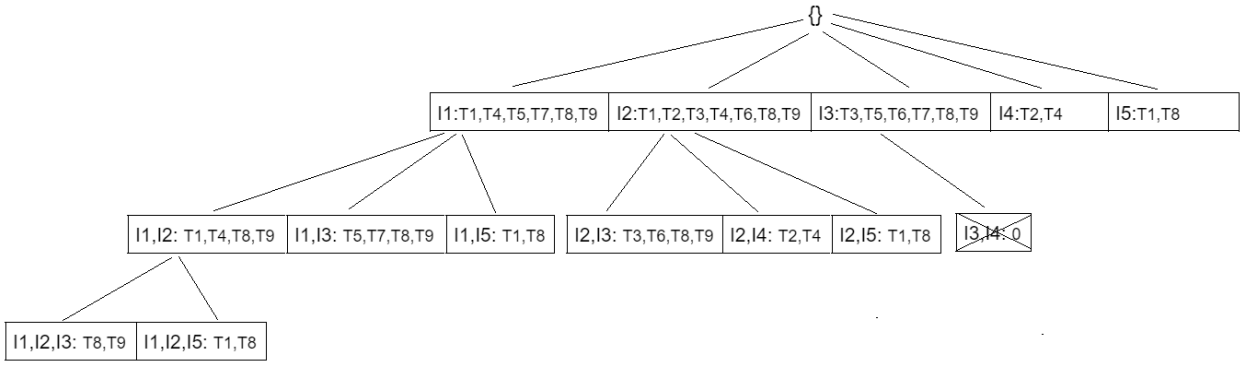
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I2,I4,I5}:

Item	TIDset
I2,I4	T2,T4
I2,I5	T1,T8

Σχήμα 3.42: Τα σύνολα{Item,TIDset} για τα στοιχειοσύνολα

Παρατηρείται ότι καμία από τις συναλλαγές δεν περιέχει το υποψήφιο στοιχειοσύνολο {I2,I4,I5}, ισχύει δηλαδή ότι: t(I2,I4,I5) = t(I2,I4) ∩ t(I2,I5) = T2,T4 ∩ T1,T8 = 0. Η τομή δηλαδή, του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2,I4} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I2,I5}, δεν περιέχει καμία συναλλαγή. Σύμφωνα με τα προηγούμενα, το υποψήφιο στοιχειοσύνολο δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης οπότε και απορρίπτεται. Στο σημείο αυτό ολοκληρώθηκε ο δεύτερος κύκλος, στον οποίο εξετάστηκε το στοιχειοσύνολο {I2}. Τα συχνά στοιχειοσύνολα που βρέθηκαν μέχρι αυτό το σημείο είναι: {I2,I3}, {I2,I4}, {I2,I5}. Πλέον ο έλεγχος βρίσκεται στο συχνό στοιχειοσύνολο {I3}, του οποίου η τιμή υποστήριξης είναι έξι, η οποία προκύπτει από το πλήθος των συναλλαγών στις οποίες εντοπίζεται. Το στοιχειοσύνολο {I3} βρίσκεται στο πρώτο επίπεδο. Έπειτα ακολουθεί ο συνδυασμός του στοιχειοσυνόλου {I3} με τα υπόλοιπα στοιχειοσύνολα μήκους ένα, τα οποία έχουν απομείνει, προκειμένου να διαπιστωθεί ποια από τα υποψήφια στοιχειοσύνολα ικανοποιούν την ελάχιστη τιμή υποστήριξης με συνέπεια να χαρακτηρίζονται συχνά. Ο τρόπος με τον οποίο υπολογίζεται η τιμή υποστήριξης του υποψηφίου στοιχειοσυνόλου {I3, j} παραμένει ο ίδιος και προκύπτει από την τομή των δύο συνόλων καθένα από τα οποία περιέχει τις συναλλαγές στις οποίες εντοπίζεται το κάθε στοιχειοσύνολο που συνθέτει το υποψήφιο στοιχειοσύνολο {I3,j}.

Αρχικά εξετάζεται το υποψήφιο στοιχειοσύνολο {I3,I4}:



Σχήμα 3.43: Έλεγχος για το στοιχειοσύνολο {I3,I4}

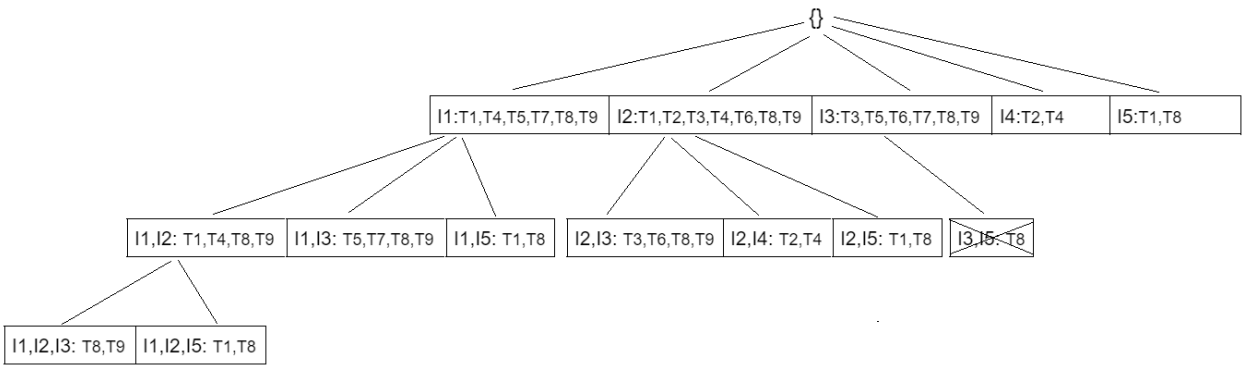
Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I3,I4}:

Item	TIDset
I3	T3,T5,T6,T7,T8,T9
I4	T2,T4

Σχήμα 3.44: Τα σύνολα {Item,TIDset} για τα στοιχειοσύνολα

Παρατηρείται ότι καμία από τις συναλλαγές δεν περιέχει το υποψήφιο στοιχειοσύνολο {I3,I4}, ισχύει δηλαδή ότι: t(I3,I4) = t(I3) ∩ t(I4) = T3,T5,T6,T7,T8,T9 ∩ T2,T4 = 0. Η τομή δηλαδή, του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I3} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I4}, δεν περιέχει καμία συναλλαγή. Στη συγκεκριμένη περίπτωση το υποψήφιο στοιχειοσύνολο δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης, οπότε και απορρίπτεται.

Έπειτα εξετάζεται το υποψήφιο στοιχειοσύνολο {I3,I5}:



Σχήμα 3.45: Έλεγχος για το στοιχειοσύνολο{I3,I5}

Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I3,I5}:

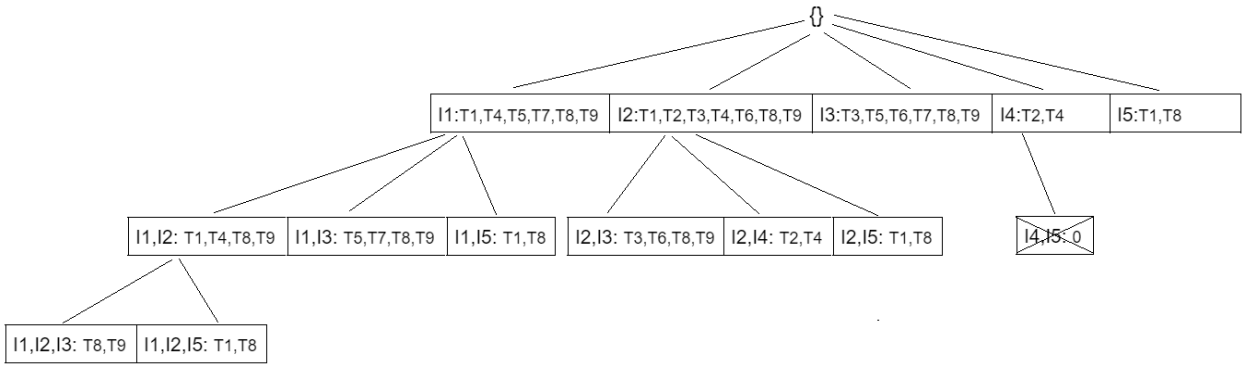
Item	TIDset
I3	T3,T5,T6,T7,T8,T9
I5	T1,T8

Σχήμα 3.46: Τα σύνολα {Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται ότι η συναλλαγή που περιέχει το υποψήφιο στοιχειοσύνολο {I3,I5} είναι: t(I3,I5) = t(I3) ∩ t(I5) = T3,T5,T6,T7,T8,T9 ∩ T1,T8 = T8. Η τομή δηλαδή του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I3} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I5}, περιέχει μία συναλλαγή την T8. Στη συγκεκριμένη περίπτωση, το υποψήφιο στοιχειοσύνολο δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης, οπότε και απορρίπτεται. Στο σημείο αυτό ολοκληρώθηκε ο τρίτος κύκλος στον οποίο εξετάστηκε το στοιχειοσύνολο {I3}. Πλέον ο έλεγχος βρίσκεται στο συχνό στοιχειοσύνολο {I4}, του οποίου η τιμή υποστήριξης είναι δύο και προκύπτει από το πλήθος των

συναλλαγών στις οποίες εντοπίζεται. Το στοιχειοσύνολο {I4} υπενθυμίζεται πως βρίσκεται στο πρώτο επίπεδο. Αμέσως μετά για κάθε επόμενο συχνό στοιχειοσύνολο μήκους ένα το οποίο έχει απομείνει συνδυάζεται με το στοιχειοσύνολο {I4} και σχηματίζονται τα υποψήφια στοιχειοσύνολα, τα οποία θα εξεταστούν αν ικανοποιούν την ελάχιστη τιμή υποστήριξης ωστε να χαρακτηριτούν εν τέλει ως συχνά. Ο υπολογισμός της τιμής υποστήριξης είναι ο ίδιος όπως και με τα προηγούμενα υποψήφια στοιχειοσύνολα, η τομή των συναλλαγών του κάθε στοιχειοσυνόλου που συνθέτει το υποψήφιο στοιχειοσύνολο με τις συναλλαγές του δεύτερου στοιχειοσυνόλου που απαρτίζει το υποψήφιο στοιχειοσύνολο I4, j}.

Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I4,I5}:



Σχήμα 3.47: Έλεγχος για το στοιχειοσύνολο{I4,I5}

Υπολογισμός της τιμής υποστήριξης για το υποψήφιο στοιχειοσύνολο {I4,I5}:

Item	TIDset
I4	T2,T4
I5	T1,T8

Σχήμα 3.48:Τα σύνολα {Item,TIDset} για τα στοιχειοσύνολα.

Παρατηρείται ότι καμία από τις συναλλαγές δεν περιέχει το υποψήφιο στοιχειοσύνολο {I4,I5}, ισχύει δηλαδή ότι: $t(I4,I5) = t(I4) \cap t(I5) = T2,T4 \cap T1,T8 = 0$. Η τομή δηλαδή, του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I4} και του συνόλου των συναλλαγών που περιέχουν το στοιχειοσύνολο {I5}, δεν περιέχει καμία συναλλαγή. Στη συγκεκριμένη περίπτωση το υποψήφιο στοιχειοσύνολο δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης οπότε και απορρίπτεται. Στο σημείο αυτό ολοκληρώθηκε ο τέταρτος κύκλος στον οποίο εξετάστηκε το στοιχειοσύνολο {I4}.

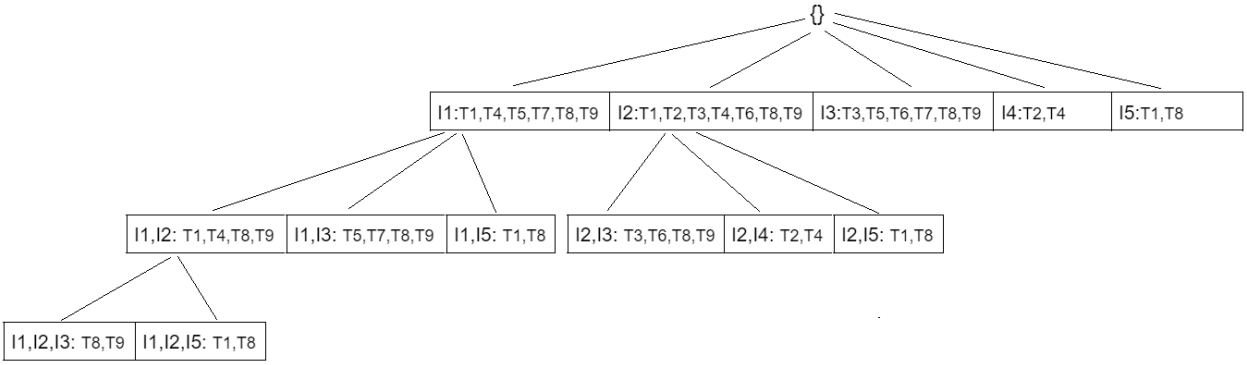
Το τελευταίο συχνό στοιχειοσύνολο μήκους ένα, το οποίο είναι το στοιχειοσύνολο {I5}, ανήκει στο πρώτο επίπεδο και μαζί με τα υπόλοιπα τα οποία εξετάστηκαν έως το σημείο αυτό ολοκληρώνουν την εκτέλεση του αλγορίθμου, καθώς δεν υπάρχει άλλο συχνό στοιχειοσύνολο μήκους ένα στη λίστα με τα συχνά. Το σύνολο των συχνών στοιχειοσυνόλων που προέκυψε από την εκτέλεση του αλγορίθμου είναι:

Συχνά στοιχειοσύνολα μήκους ένα:{I1},{I2},{I3},{I4},{I5}.

Συχνά στοιχειοσύνολα μήκους δύο: {I1,I2},{I1,I3},{I1,I5},{I2,I3},{I2,I4}, {I2,I5}.

Συχνά στοιχειοσύνολα μήκους τρία:{I1,I2,I3},{I1,I2,I5}.

Το σύνολο των συχνών στοιχειοσυνόλων, μπορεί να διαπιστωθεί και από την τελική μορφή που λαμβάνει η ιεραρχία με τα συχνά στοιχειοσύνολα μετά το πέρας της εκτέλεσης του αλγορίθμου:



Σχήμα 3.49: Η τελική μορφή της ιεραρχίας των συχνών στοιχειοσυνόλων

Η δομή, η οποία παρουσιάζεται στην παραπάνω εικόνα πέρα από την αποτύπωση των συχνών στοιχειοσυνόλων μήκους ένα, δύο και τρία του παραδείγματος, καταχωρεί και όλες τις τιμές υποστήριξης των συχνών στοιχειοσυνόλων που προκύπτουν από το πλήθος των συναλλαγών που συνοδεύουν το κάθε συχνό στοιχειοσύνολο. Μπορεί δηλαδή εύκολα ο αναγνώστης διατρέχοντας τη δομή να λάβει και την πληροφορία για την υποστήριξη ενός συχνού στοιχειοσυνόλου πέρα από τα συχνά στοιχειοσύνολα αυτά καθ'αυτά.

Το αμέσως επόμενο βήμα αποτελεί η παραγωγή των συνδυαστικών κανόνων.

Πιο συγκεκριμένα οι συνδυαστικοί κανόνες που προκύπτουν είναι:

1)I1=>I2 Εμπιστοσύνη (Confidence) = Supp(I1&I2) / Supp(I1) = 4 / 6 = 66%.

2)I2=>I1 Εμπιστοσύνη (Confidence) = Supp(I1&I2) / Supp(I2) = 4 / 7 = 57%.

3)I1=>I3 Εμπιστοσύνη (Confidence) = Supp(I1&I3) / Supp(I1) = 4 / 6 = 66%.

4)I3=>I1 Εμπιστοσύνη (Confidence) = Supp(I1&I3) / Supp(I3) = 4 / 6 = 66%.

5)I1=>I5 Εμπιστοσύνη (Confidence) = Supp(I1&I5) / Supp(I1) = 2 / 6 = 33%.

6)I5=>I1 Εμπιστοσύνη (Confidence) = Supp(I1&I5) / Supp(I5) = 2 / 2 = 100%.

7)I2=>I3 Εμπιστοσύνη (Confidence) = Supp(I2&I3) / Supp(I2) = 4 / 7 = 57%.

8)I3=>I2 Εμπιστοσύνη (Confidence) = Supp(I2&I3) / Supp(I3) = 4 / 6 = 66%.

9)I2=>I4 Εμπιστοσύνη (Confidence) = Supp(I2&I4) / Supp(I2) = 2 / 7 = 28%.

10)I4=>I2 Εμπιστοσύνη (Confidence) = Supp(I2&I4) / Supp(I4) = 2 / 2 = 100%.

11)I2=>I5 Εμπιστοσύνη (Confidence) = Supp(I2&I5) / Supp(I2) = 2 / 7 = 28%.

12)I5=>I2 Εμπιστοσύνη (Confidence) = Supp(I2&I5) / Supp(I5) = 2 / 2 = 100%.

13)(I1&I2=>I3) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I1&I2) = 2 / 4 = 50%

14)(I1&I3=>I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I1&I3) = 2 / 4 = 50%

15)(I2&I3=>I1) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I2&I3) = 2 / 4 = 50%

16)(I1=>I2&I3) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I1) = 2 / 6 = 33%

17)(I2=>I1&I3) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I2) = 2 / 7 = 28%

18)(I3=>I1&I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I3) = 2 / 6 = 33%

19)(I1&I2=>I5) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I1&I2) = 2 / 4 = 50%

20)(I1&I5=>I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I1&I5) = 2 / 2 = 100%

21)(I2&I5=>I1) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I2&I5) = 2 / 2 = 100%

22)(I1=>I2&I5) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I1) = 2 / 6 = 33%

23)(I2=>I1&I5) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I2) = 2 / 7 = 28%

24)(I5=>I1&I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I5) = 2 / 2 = 100%

Εάν είχε τεθεί για το παραπάνω παράδειγμα ως ελάχιστη τιμή εμπιστοσύνης (minimum confidence) η τιμή confidence=70%, οι κανόνες οι οποίοι θα χαρακτηρίζονταν ως χρήσιμοι, είναι οι εξής: 6)I5=>I1, 10)I4=>I2, 12)I5=>I2, 20)(I1&I5=>I2), 21) (I2&I5=>I1) και 24)(I5=>I1&I2).

3.3.3 Incremental mining

Στην αρχική του μορφή η οποία εισήχθη από τον Zaki, [27] ο αλγόριθμος Eclat δεν υποστηρίζει το incremental mining. Πέραν τούτου όμως στη βιβλιογραφία μπορεί κάποιος να συναντήσει εκδόσεις του αλγορίθμου Eclat, οι οποίες να υποστηρίζουν το incremental mining [63-64].

3.3.4 Συμπεράσματα

Ο αλγόριθμος Eclat ανήκει στην κατηγορία των αλγορίθμων που υλοποιούν την αναζήτηση σε βάθος (Depth First Search). Χρησιμοποιεί μια κατακόρυφη διάταξη (Vertical Layout) της βάσης των συναλλαγών, δηλαδή αντί να αναφέρονται ρητά όλες οι συναλλαγές και το περιεχόμενό τους καθε στοιχειοσύνολο αποθηκεύεται μαζί με το σύνολο των συναλλαγών στις οποίες εντοπίζεται (tidlist ή tidset) [35]. Επίσης προκειμένου να υπολογίσει την τιμή υποστήριξης ενός στοιχειοσυνόλου χρησιμοποιεί την μέθοδο που βασίζεται στη τομή των tidlists των δύο στοιχειοσυνόλων από τα οποία σχηματίζεται. Παρακάτω, στο έκτο κεφάλαιο ακολουθεί η σύγκριση του αλγορίθμου με τους υπόλοιπους δύο στο περιβάλλον R/RStudio. Μέσα από τα διάφορα σενάρια παρατηρείται σε ποιές περιπτώσεις ο αλγόριθμος Eclat πετυχαίνει τις καλύτερες επιδόσεις, όχι μόνο για τον ίδιο αλλά και έναντι των υπολοίπων δύο. Ακόμα παρατηρείται πως ακριβώς συμπεριφέρεται για ιδιαίτερα χαμηλές τιμές υποστήριξης. Αν είναι δηλαδή κατάλληλότερος για μικρά ή μεγάλα σύνολα δεδομένων και απαιτεί λιγότερο χρόνο για συχνή εύρεση των συχνών στοιχειοσυνόλων από ό,τι οι υπόλοιποι δύο αλγόριθμοι. Οι παραπάνω εικασίες ως προς τις επιδόσεις των αλγορίθμων θα εξεταστούν παρακάτω μέσα από τα πειράματα που συνοδεύουν την παρούσα εργασία.

3.4 Ο αλγόριθμος FP-Growth

Έναν από τους πιο δημοφιλείς αλγορίθμους εξόρυξης συχνών στοιχειοσυνόλων, αποτελεί και ο αλγόριθμος FP-Growth, ο οποίος προτάθηκε από τους Han et al [44],[67]. Με σκοπό την αντιμετώπιση των δύο κύριων μειονεκτημάτων του αλγορίθμου Apriori [16], δημιουργείται μια καινούρια, δομή δεδομένων συμπιεσμένης μορφής η οποία ονομάζεται δενδρική δομή FP-Tree. Η νέα αυτή δομή δεδομένων χαρακτηρίζεται ως μία δομή προθέματος (Prefix) [4] που αποθηκεύει μετρήσιμες πληροφορίες οι οποίες σχετίζονται με τα συχνά στοιχειοσύνολα [67]. Με βάση τη δενδρική δομή FP-Tree, αναπτύχθηκε ένας νέος αλγόριθμος εύρεσης συχνών στοιχειοσυνόλων, ο αλγόριθμος FP-Growth. Ο συγκεκριμένος αλγόριθμος αποτελεί ουσιαστικά μία διαδικασία δύο βημάτων [67]. Στο πρώτο βήμα

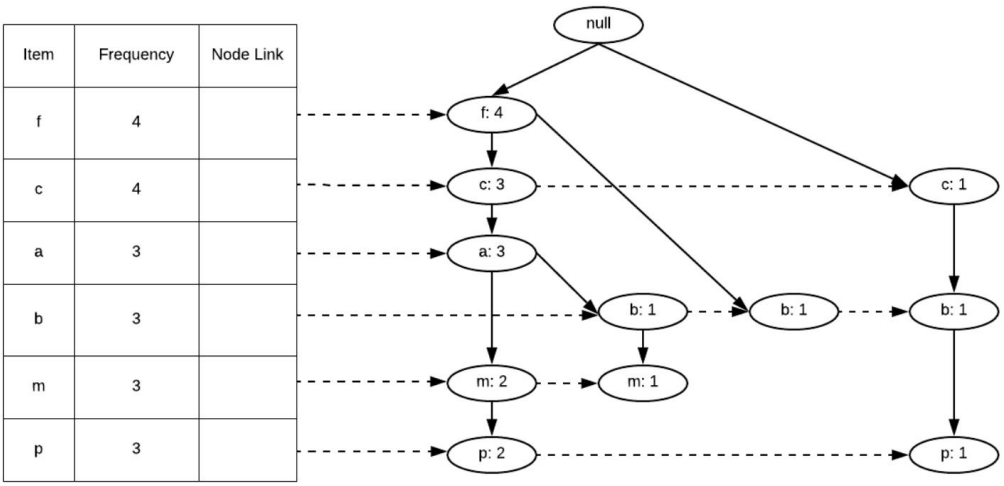
κατασκευάζεται ένα δέντρο συχνών στοιχειοσυνόλων (Frequent Pattern Tree), το οποίο σαρώνει τη βάση των συναλλαγών δύο φορές. Στην πρώτη σάρωση της βάσης με τις συναλλαγές, σαρώνονται τα περιεχόμενα της βάσης και για κάθε στοιχειοσύνολο μήκους ένα υπολογίζεται η τιμή υποστήριξης του. Τα στοιχειοσύνολα που δεν χαρακτηρίζονται ως συχνά αφαιρούνται από τη λίστα ενω τα υπόλοιπα ταξινομούνται σε φθίνουσα τάξη [37]. Έπειτα, στο δεύτερο πέρασμα της βάσης με τις συναλλαγές, δημιουργείται η δενδρική δομή FP-Tree (Frequent Pattern Tree), η οποία περιέχει, σε συνδυασμό με τον πίνακα κεφαλίδας συχνών στοιχειοσυνόλων, όλες τις απαραίτητες πληροφορίες για τα συχνά στοιχειοσύνολα. Αντίστοιχα, στο δεύτερο βήμα μέσα από την εκτέλεση του αλγορίθμου FP-Growth, δημιουργούνται όλα τα συχνά στοιχειοσύνολα τα οποία προκύπτουν από τη δενδρική δομή FP-Tree [33],[44],[67].

Η δενδρική δομή FP-Tree ορίζεται ως εξής:

1.Αποτελείται από τη ρίζα (root), η οποία λαμβάνει την τιμή "null", ένα σύνολο υποδέντρων στοιχείου-προθέματος (item-prefix), τα οποία αποτελούν παιδιά της ρίζας (root) και έναν πίνακα κεφαλίδας συχνών στοιχείων (frequent-item-header table), μέσω του οποίου μπορεί κανείς να αντλήσει χρήσιμες πληροφόρίες για τα συχνά στοιχειοσύνολα μήκους ένα [52].

2.Κάθε κόμβος στο υποδέντρο στοιχείου-προθέματος (item-prefix) αποτελείται συνολικά από τρία πεδία: το στοιχείο-όνομα (item-name), την μέτρηση της υποστήριξης (count) και τον κόμβο-σύνδεσμο (node-link), όπου το στοιχείο-όνομα (item-name) σημειώνει ποιο στοιχειοσύνολο αντιπροσωπεύει αυτός ο κόμβος, η μέτρηση της υποστήριξης (count) σημειώνει τον αριθμό των συναλλαγών που αντιπροσωπεύονται από το τμήμα της διαδρομής που καταλήγει σε αυτόν τον κόμβο και ο κόμβος-σύνδεσμος (node-link) συνδέεται στον επόμενο κόμβο στο δέντρο FP (FP-tree), ο οποίος φέρει το ίδιο στοιχείο-όνομα (item-name) ή δεν συνδέεται σε κάποιον άλλο κόμβο εάν δεν υφίσταται ο τελευταίος [54-55].

3.Κάθε καταχώρηση στον πίνακα κεφαλίδας συχνών στοιχείων (frequent-item-header table) αποτελείται από τρία πεδία: 1) το όνομα του στοιχειοσυνόλου 2) την τιμή υποστήριξης του στοιχειοσυνόλου αυτού και 3) την κεφαλή του κόμβου-συνδέσμου (node-link), η οποία δείχνει προς τον πρώτο κόμβο στη δενδρική δομή (FP-Tree) που φέρει το όνομα του εκάστοτε στοιχειοσυνόλου [51],[67].



Σχήμα 3.50: Ένα τυχαίο FP δέντρο μαζί με τον πίνακα κεφαλίδας συχνών στοιχείων [45]

Αλγόριθμος κατασκευής της δενδρικής δομής FP-Tree:

Input: Μία βάση με συναλλαγές DB και μία ελάχιστη τιμή υποστήριξης σ.

Output: Η δενδρική δομή FP-Tree της βάσης με των συναλλαγών DB.

1.Αρχικά ο αλγόριθμος σαρώνει τη βάση με τις συναλλαγές DB μία φορά. Συλλέγει το F, το σύνολο των συχνών στοιχειοσυνόλων μήκους ένα και την τιμή υποστήριξης κάθε συχνού στοιχειοσυνόλου μήκους ένα. Ταξινομεί το σύνολο F κατα φθίνουσα σειρά με βάση την τιμή υποστήριξης και αποθηκεύεται ως L, η λίστα των συχνών στοιχειοσυνόλων μήκους ένα [67].

2.Έπειτα ο αλγόριθμος δημιουργεί τη ρίζα T της δενδρικής δομής FP-Tree, η οποία λαμβάνει την τιμή "null" [50]. Για κάθε συναλλαγή Trans στη βάση των συναλλαγών DB πραγματοποιεί τα παρακάτω:

Επιλέγει τα συχνά στοιχειοσύνολα στη συναλλαγή Trans και τα ταξινομεί σύμφωνα με τη σειρά της λίστας L. Έστω η ταξινομημένη λίστα συχνών στοιχειοσυνόλων στη συναλλαγή Trans [p | P], όπου το p αναφέρεται στο πρώτο στοιχείο και το P αναφέρεται στην υπόλοιπη λίστα. Ο αλγόριθμος καλεί την συνάρτηση insert tree ([p | P], T), η οποία λαμβάνει ως παραμέτρους την ταξινομημένη λίστα των συχνών στοιχειοσυνόλων [p | P] και την ρίζα του δέντρου T [44],[55],[67].

Η συνάρτηση insert_tree([p | P], T) εκτελείται ως εξής:

Αν η ρίζα T έχει ένα παιδί N τέτοιο ώστε να ισχύει ότι N.item-name = p.item-name, το όνομα δηλαδή του παιδιού N ισούται με το όνομα του p, τότε αυξάνεται η μέτρηση της υποστήριξης (Count) του N κατά 1 [53], εάν δηλαδή ο συγκεκριμένος κόμβος υπάρχει εκ των προτέρων δεν δημιουργείται νέος κόμβος και η τιμή υποστήριξης του συγκεκριμένου κόμβου αυξάνεται κατά 1, διαφορετικά δημιουργείται ένας νέος κόμβος N, με τη μέτρηση της υποστήριξης (Count) του να αρχικοποιείται και να λαμβάνει την τιμή 1, τον γονικό του σύνδεσμο συνδεδεμένο με τη ρίζα T και τον κόμβο-σύνδεσμο (Node-Link) του συνδεδεμένο με τους υπόλοιπους κόμβους οι οποίοι φέρουν το ίδιο στοιχείο-όνομα (Item-Name), μέσω του κόμβου-συνδέσμου (Node-Link). Εάν το P είναι μη κενό, καλείται η συνάρτηση insert_tree (P, N) αναδρομικά [44],[49].

Αμέσως μετά εκτελείται ο αλγόριθμος FP-Growth, προκειμένου να βρεθεί το σύνολο των συχνών στοιχειοσυνόλων. Ως είσοδος δίνεται η δενδρική δομή FP-Tree, η οποία έχει κατασκευαστεί στο αμέσως προηγούμενο βήμα, καθώς και μία ελάχιστη τιμή υποστήριξης σ. Ως έξοδος εξάγεται το σύνολο των συχνών στοιχειοσυνόλων.

Ορισμός της Conditional Pattern Base και του Conditional FP-Tree:

Έστω η διαδρομή (f :4, c:3 a:3 m:2 p:2) και είναι επιθυμητή η εύρεση των συχνών στοιχειοσυνόλων τα οποία για να σχηματιστούν συνδυάζονται με τον κόμβο m, στη διαδρομή αυτή θα πρέπει να εξαχθεί μόνο η υποδιαδρομή προθέματος η οποία αφορά αποκλειστικά τον κόμβο m δηλαδή (f:4, c:3 a:3), καθώς και να οριστεί η τιμή υποστήριξης, για κάθε έναν κόμβο στη συγκεκριμένη διαδρομή, ίση με την τιμή υποστήριξης του κόμβου m. Με αυτόν τον τρόπο προκύπτει η συγκεκριμένη διαδρομή (f:2, c:2 a:2), καθώς η τιμή υποστήριξης για τον κόμβο m είναι ίση με 2. Σύμφωνα με τα παραπάνω, η υποδιαδρομή προθέματος ενός κόμβου ai σε μία διαδρομή P, μπορεί να αποθηκευτεί και να μετατραπεί σε μία προσαρμοσμένη με βάση την τιμή υποστήριξης διαδρομή πρόθεματος, προσαρμόζοντας την τιμή υποστήριξης, για κάθε έναν κόμβο στην υποδιαδρομή προθέματος, η οποία να ισούται με την τιμή υποστήριξης του κόμβου ai. Η επονομαζόμενη διαδρομή προθέματος, ονομάζεται η μετασχηματισμένη διαδρομή προθέματος του κόμβου ai για τη διαδρομή P. Παρατηρείται, πως το σύνολο των μετασχηματισμένων διαδρομών προθέματος οι οποίες αφορούν έναν κόμβο ai, σχηματίζουν μία μικρή

βάση από στοιχειοσύνολα τα οποία συνυπάρχουν μαζί με τον κόμβο ai. Μία τέτοια βάση από στοιχειοσύνολα τα οποία συνυπάρχουν μαζί με τον κόμβο ai ονομάζεται η Conditional Pattern Base του κόμβου ai και συμβολίζεται με "pattern_base | ai". Πιο συγκεκριμένα ονομάζεται conditional, καθώς η συγκεκριμένη βάση σχηματίζεται υπό την συνθήκη ύπαρξης του κόμβου ai. Κατόπιν μπορούν να βρεθούν τα συχνά στοιχειοσύνολα τα οποία συνδέονται με τον συγκεκριμένο κόμβο ai, στην Conditional Pattern Base του κόμβου αυτού, μέσα από τη δημιουργία μιας δενδρικής δομής FP-Tree η οποία ονομάζεται Conditional FP-Tree, η οποία αφορά αποκλειστικά τον κόμβο ai και συμβολίζεται με "FP-Tree | ai". Έπειτα αναζητούνται τα συχνά στοιχειοσύνολα στο συγκεκριμένο δέντρο. Οι παραπάνω έννοιες συναντώνται παρακάτω στο παράδειγμα για τον αλγόριθμο FP-Growth [48],[52],[67].

Ο αλγόριθμος FP-Growth για την εύρεση των συχνών στοιχειοσυνόλων:

Input: Η δενδρική δομή FP-Tree έχοντας ως βάση την DB και ως minsup σ.

Output: Το σύνολο των συχνών στοιχειοσυνόλων.

3.4.1 Βήματα εκτέλεσης του αλγορίθμου FP-Growth

1. Αν η δενδρική δομή FP-Tree περιέχει μία μοναδική διαδρομή (single path), τα στοιχειοσύνολα που δημιουργούνται σχηματίζονται από τους συνδυασμούς των κόμβων στη διαδρομή (path) αυτή, με την τιμή υποστήριξης να είναι η ελάχιστη τιμή υποστήριξης των κόμβων στην υποδιαδρομή [48],[67]. (γραμμές 1-3)
2. Εναλλακτικά, για κάθε συχνό στοιχειοσύνολο ai, κατασκευάζεται η Conditional Pattern Base (της οποίας η έννοια ορίστηκε παραπάνω και η οποία αποτελείται από το σύνολο των διαδρομών προθέματος (Prefix-Paths) στη δενδρική δομή FP-Tree που καταλήγουν στο συχνό στοιχειοσύνολο ai), μέσα από την οποία προκύπτει και ελέγχεται το Conditional FP-Tree από το οποίο και σχηματίζονται τα συχνά στοιχειοσύνολα. (γραμμές 4-6) [48,67]
3. Κατόπιν εάν το δέντρο για το συνδυασμό β είναι μη κενό, ο αλγόριθμος καλείται αναδρομικά. (γραμμές 7-8)

Input: FP-tree constructed based on Algorithm 1, using DB and a minimum support threshold ξ.

Output: The complete set of frequent patterns.

Method: Call FP-growth (FP-tree , $null$).

Procedure FP-growth ($Tree, \alpha$)
{
(1) **if** $Tree$ contains a single path P
(2) **then for each** combination (denoted as β) of the nodes in the path P **do**
(3) generate pattern $\beta \cup \alpha$ with $support =$ $minimum\ support\ of\ nodes\ in\ \beta$;
(4) **else for each** a_i in the header of $Tree$ **do** {
(5) generate pattern $\beta = a_i \cup \alpha$ with $support = a_i.support$;
(6) construct β's conditional pattern base and then β's conditional FP-tree $Tree_\beta$;
(7) **if** $Tree_\beta \neq \emptyset$
(8) **then** call FP-growth ($Tree_\beta, \beta$) }
}

Σχήμα 3.51: Ο αλγόριθμος FP-Growth [44]

3.4.2 Παράδειγμα εκτέλεσης του αλγορίθμου FP-Growth

TID	Items
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

Σχήμα 3.52: Η βάση με τις συναλλαγές του παραδείγματος.

Η πρώτη σάρωση της βάσης των συναλλαγών DB, εξάγει το σύνολο των συχνών στοιχειοσυνόλων μήκους 1 και τις αντίστοιχες τιμές υποστήριξης για κάθε ένα από αυτό. Ορίζεται ως ελάχιστη τιμή υποστήριξης (Minimum Support) η τιμή 0.22 ή 22% (Επιθυμητό άθροισμα των εμφανίσεων ενός στοιχειοσυνόλου τουλάχιστον δύο φορές επί του συνόλου των συναλλαγών). Το σύνολο των συχνών στοιχειοσυνόλων ταξινομείται σε φθίνουσα τάξη με βάση την τιμή υποστήριξης. Αυτό το σύνολο που προκύπτει αποτελεί την λίστα L. Με αυτόν τον τρόπο η λίστα L αποτελείται από L = {{I2: 7}, {I1: 6}, {I3: 6},{I4: 2}, {I5: 2}}. Έπεται η κατασκευή της δενδρικής δομής FP-Tree:

Αρχικά, δημιουργείται η ρίζα του δέντρου, η οποία λαμβάνει την τιμή “null”.

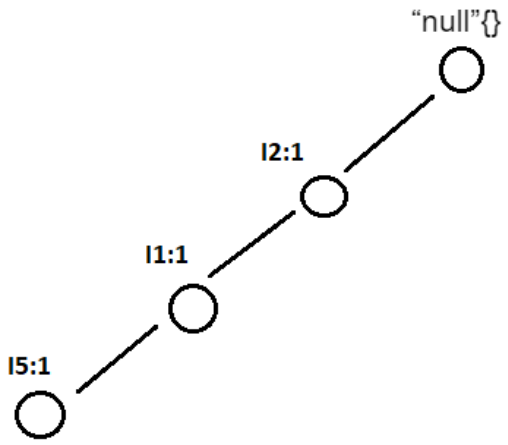
Σχήμα 3.53: Η δημιουργία της ρίζας.

Στη συνέχεια σαρώνεται η βάση των συναλλαγών DB για δεύτερη φορά. Η σάρωση η οποία αφορά την πρώτη συναλλαγή T1, οδηγεί με τη σειρά της στην κατασκευή του πρώτου κλαδιού, της δενδρικής δομής FP-Tree, το οποίο είναι (I2:1,I1:1,I5:1). Σημειώνεται πως τα στοιχεία σε κάθε συναλλαγή ταξινομούνται κατά σειρά L (ταξινόμηση κατα φθίνουσα τάξη με βάση την τιμή υποστήριξης), όπως απεικονίζονται και στην εικόνα 3.4.4 παρακάτω.

TID	Items
T1	I2,I1,I5
T2	I2,I4
T3	I2,I3
T4	I2,I1,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I2,I1,I3,I5
T9	I2,I1,I3

Σχήμα 3.54: Οι συναλλαγές ταξινομημένες ως προς την λίστα L.

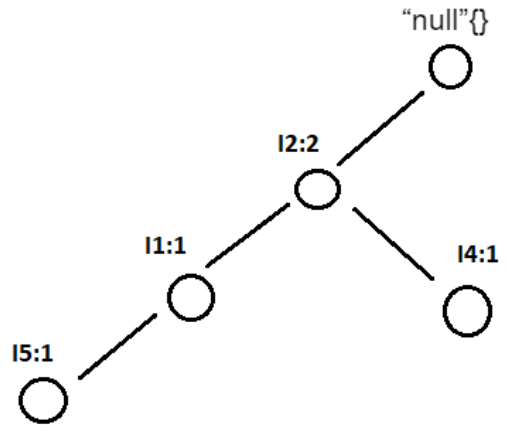
Στην πρώτη συναλλαγή T1:{I2,I1,I5}, η οποία είναι ταξινομημένη σύμφωνα με τη λίστα L, κατασκευάζεται όπως προαναφέρθηκε το πρώτο κλαδί της δενδρικής δομής FP-Tree, το οποίο περιέχει τρεις κόμβους, τον κόμβο (I2:1) ο οποίος αναφέρεται στο στοιχειοσύνολο {I2} και συνδέεται με τη ρίζα, τον κόμβο (I1:1) ο οποίος αναφέρεται στο στοιχειοσύνολο {I1} και συνδέεται με τον κόμβο (I2:1) και τον κόμβο (I5:1) ο οποίος αναφέρεται στο στοιχειοσύνολο {I5} και συνδέεται με τον κόμβο (I1:1). Σε κάθε ένα από τους παραπάνω κόμβους, οι οποίοι αναφέρονται στα αντίστοιχα στοιχειοσύνολα, αποδίδεται η τιμή υποστήριξης 1.



Σχήμα 3.55: Η κατασκευή της δενδρικής δομής FP-Tree.

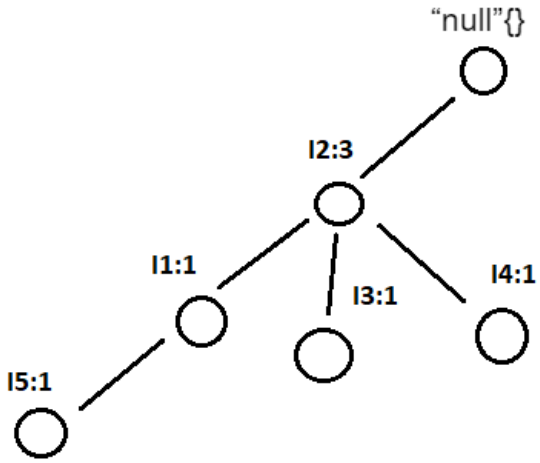
Κατόπιν, στη δεύτερη συναλλαγή T2:{I2,I4}, η οποία είναι ταξινομημένη σύμφωνα με τη λίστα L, αποφεύγεται η κατασκευή ενός δεύτερου κλαδιού για τη δενδρική δομή FP-Tree, καθώς παρατηρείται η ύπαρξη ενός κοινού προθέματος (Prefix), με την προϋπάρχουσα διαδρομή (I2,I1,I5), και πιο συγκεκριμένα το κοινό πρόθεμα είναι το {I2}. Με αυτόν τον τρόπο αυξάνεται η τιμή υποστήριξης του

κόμβου (I2), ο οποίος αφορά το στοιχειοσύνολο {I2}, κατα 1, ισχύει δηλαδή ότι (I2:2). Έπειτα δημιουργείται ο κόμβος (I4), ο οποίος αναφέρεται στο στοιχειοσύνολο {I4}, συνδέεται ως παιδί με τον κόμβο (I2:2) και λαμβάνει ως τιμή υποστήριξης την τιμή 1, με συνέπεια να προκύπτει ότι (I4:1).



Σχήμα 3.56: Η κατασκευή της δενδρικής δομής FP-Tree.

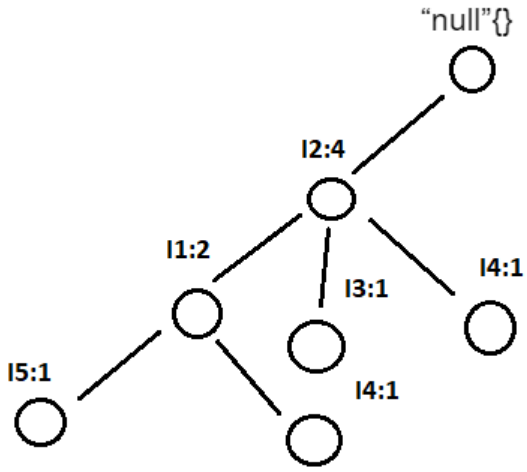
Στην τρίτη συναλλαγή T3:{I2,I3}, η οποία και αυτή με τη σειρά της είναι ταξινομημένη σύμφωνα με τη λίστα L, αντίστοιχα με προηγουμένως, αποφεύγεται η κατασκευή ενός δεύτερου κλαδιού για τη δενδρική δομή FP-Tree, καθώς παρατηρείται η ύπαρξη επίσης ενός κοινού προθέματος (Prefix), με την προϋπάρχουσα διαδρομή (I2,I1,I5), και πιο συγκεκριμένα το κοινό πρόθεμα και σε αυτήν την περίπτωση είναι το {I2}. Με τον ίδιο τρόπο αυξάνεται η τιμή υποστήριξης του κόμβου (I2), ο οποίος αφορά το στοιχειοσυνόλου {I2}, κατά 1 ισχύει δηλαδή ότι (I2:3). Κατόπιν δημιουργείται ο κόμβος (I3), ο οποίος αναφέρεται στο στοιχειοσύνολο {I3}, συνδέεται ως παιδί με τον κόμβο (I2:3) και λαμβάνει ως τιμή υποστήριξης την τιμή 1 και ισχύει ότι (I3:1).



Σχήμα 3.57: Η κατασκευή της δενδρικής δομής FP-Tree.

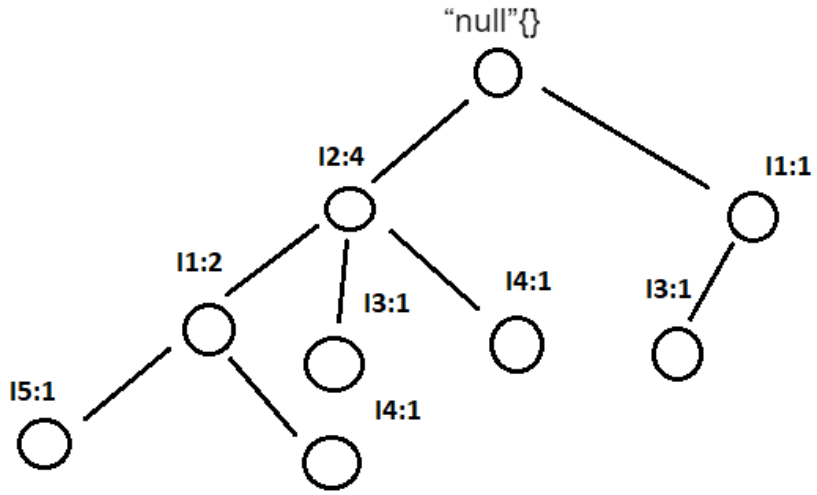
Στην τέταρτη συναλλαγή T4:{I2,I1,I4}, η οποία όπως και οι προηγούμενες είναι ταξινομημένη σύμφωνα με τη λίστα L, παρατηρείται η ύπαρξη επίσης ενός κοινού προθέματος (Prefix), με την προϋπάρχουσα διαδρομή (I2,I1,I5). Πιο συγκεκριμένα το κοινό πρόθεμα και σε αυτήν την περίπτωση είναι το {I2,I1}. Αντίστοιχα με ότι έχει προηγηθεί έως αυτό το σημείο, αυξάνεται η τιμή υποστήριξης των κόμβων (I2) και (I1), οι οποίοι αναφέρονται στα στοιχειοσυνόλων {I2} και {I1} αντίστοιχα, κατά 1 με αποτέλεσμα να προκύπτει ότι (I2:4) και (I1:2). Κατόπιν, ακολουθεί η δημιουργία του κόμβου (I4),

ο οποίος αναφέρεται στο στοιχειοσύνολο {I4}, συνδέεται ως παιδί του κόμβου (I1:2) και λαμβάνει ως τιμή υποστήριξης την τιμή 1, δηλαδή ισχύει (I4:1).



Σχήμα 3.58: Η κατασκευή της δενδρικής δομής FP-Tree.

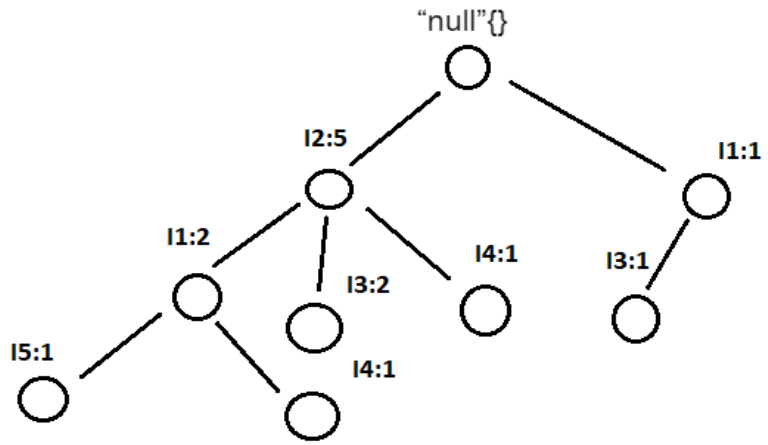
Η πέμπτη συναλλαγή T5:{I1,I3}, η οποία είναι ταξινομημένη σύμφωνα με τη λίστα L, οδηγεί στην κατασκευή του δεύτερου κλαδιού για τη δενδρική δομή FP-Tree, καθώς παρατηρείται ότι δεν υπάρχει κάποιος κόμβος που να έχει ως κοινό πρόθεμα (Prefix), τον κόμβο (I1). Κατ' επέκταση των παραπάνω, δημιουργείται ένα δεύτερο κλαδί όπως και οι δύο νέοι κόμβοι (I1) και (I3), οι οποίοι αναφέρονται στα στοιχειοσύνολα {I1} και {I3} αντίστοιχα, και λαμβάνουν ως τιμή υποστήριξης την τιμή 1, θα ισχύει δηλαδή ότι (I1:1) και (I3:1). Ο κόμβος (I1), θα συνδέεται ως παιδί στην ρίζα και ο κόμβος (I3), συνδέεται ως παιδί του κόμβου (I1).



Σχήμα 3.59: Η κατασκευή της δενδρικής δομής FP-Tree.

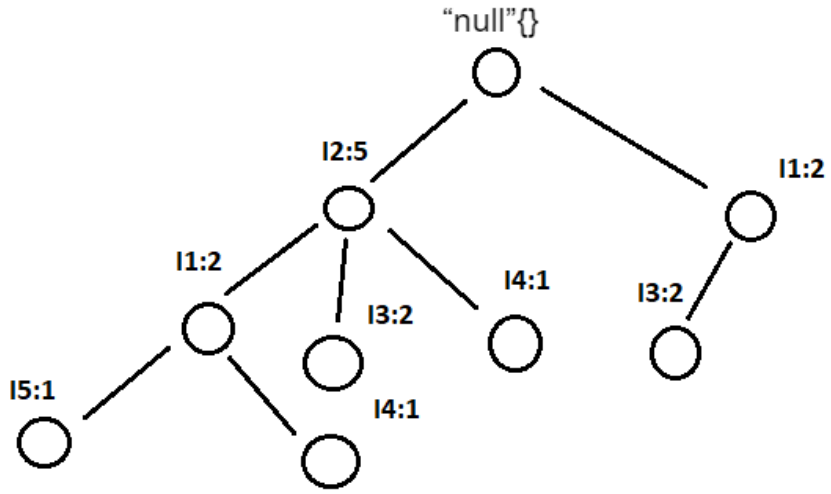
Στην έκτη συναλλαγή T6:{I2,I3}, η οποία είναι ταξινομημένη σύμφωνα με τη λίστα L, προκύπτει ότι η συγκεκριμένη διαδρομή έχει εμφανιστεί σε προηγούμενη διαδρομή. Επομένως καθώς προϋπάρχει εκ των προτέρων η συγκεκριμένη διαδρομή, το μόνο που πραγματοποιείται σε αυτήν την περίπτωση είναι η αύξηση της τιμής υποστήριξης κατά 1 για τους δύο κόμβους (I2) και (I3), οι οποίοι αναφέρονται στα δύο στοιχειοσύνολα {I2} και {I3} κατ' αντιστοιχία και να ισχύει ότι (I2:5) και (I3:2). Η μη κατασκευή

κάποιου νέου κόμβου και η απλή αύξηση της τιμής υποστήριξης κατα 1, για του δύο προαναφερθέντες κόμβους αποτυπώνεται παρακάτω στην εικόνα 3.4.10.



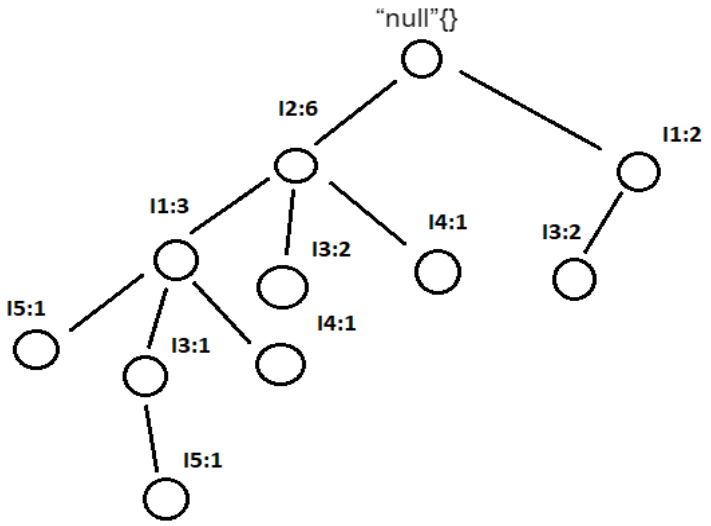
Σχήμα 3.60: Η κατασκευή της δενδρικής δομής FP-Tree.

Στην έβδομη συναλλαγή T7:{I1,I3}, η οποία είναι ταξινομημένη σύμφωνα με τη λίστα L, εφόσον προκύπτει το γεγονός ότι προϋπάρχει η συγκεκριμένη διαδρομή, στην παρούσα περίπτωση, στο δεύτερο κλαδί, όπως και στην αμέσως προηγούμενη συναλλαγή, αυξάνεται η τιμή υποστήριξης των κόμβων (I1) και (I3), οι οποίοι αναφέρονται στα στοιχειοσύνολα {I1} και {I3}, κατά 1 και διαμορφώνονται σε (I1:2) και (I3:2). Όλα τα παραπάνω αποτυπώνονται στην παρακάτω εικόνα 3.4.11.



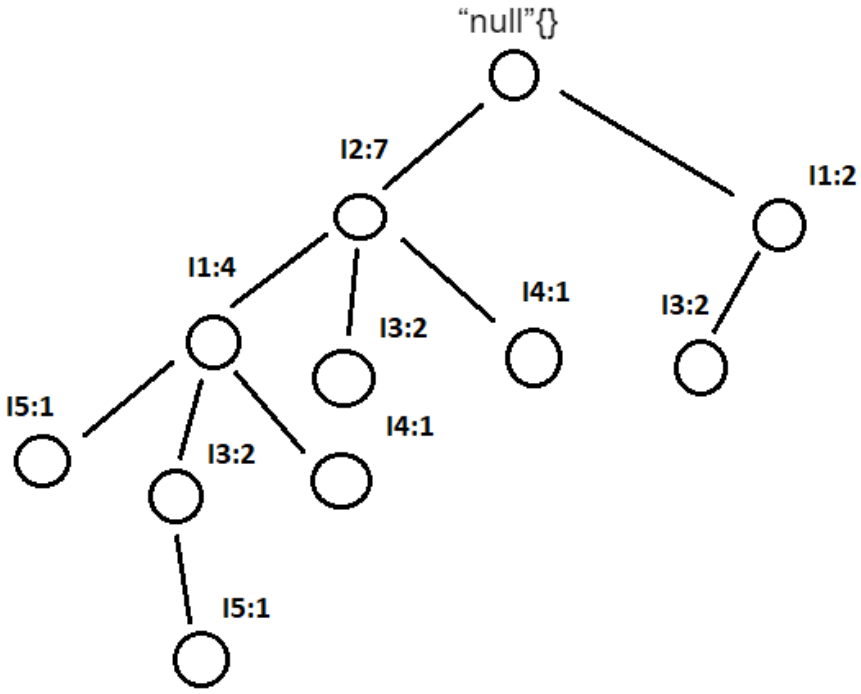
Σχήμα 3.61: Η κατασκευή της δενδρικής δομής FP-Tree.

Στην όγδοη συναλλαγή T8:{I2,I1,I3,I5}, η οποία είναι ταξινομημένη σύμφωνα με τη λίστα L, παρατηρείται η ύπαρξη ενός κοινού προθέματος (Prefix), με την προϋπάρχουσα διαδρομή (I2,I1,I5). Πιο συγκεκριμένα, παρατηρείται ότι το πρόθεμα {I2,I1}, υπάρχει εκ των προτέρων, με αποτέλεσμα να αυξάνεται η τιμή υποστήριξης των κόμβων (I2) και (I1), οι οποίοι αναφέρονται αντίστοιχα στα στοιχειοσύνολα {I2} και {I1}, κατά 1 και να δημιουργούνται οι δύο νέοι κόμβοι (I3) και (I5). Αναλυτικότερα ο κόμβος (I3:1) συνδέεται ως παιδί του του κόμβου (I1:3) και ο κόμβος (I5:1) συνδέεται ως παιδί του κόμβου (I3:1).



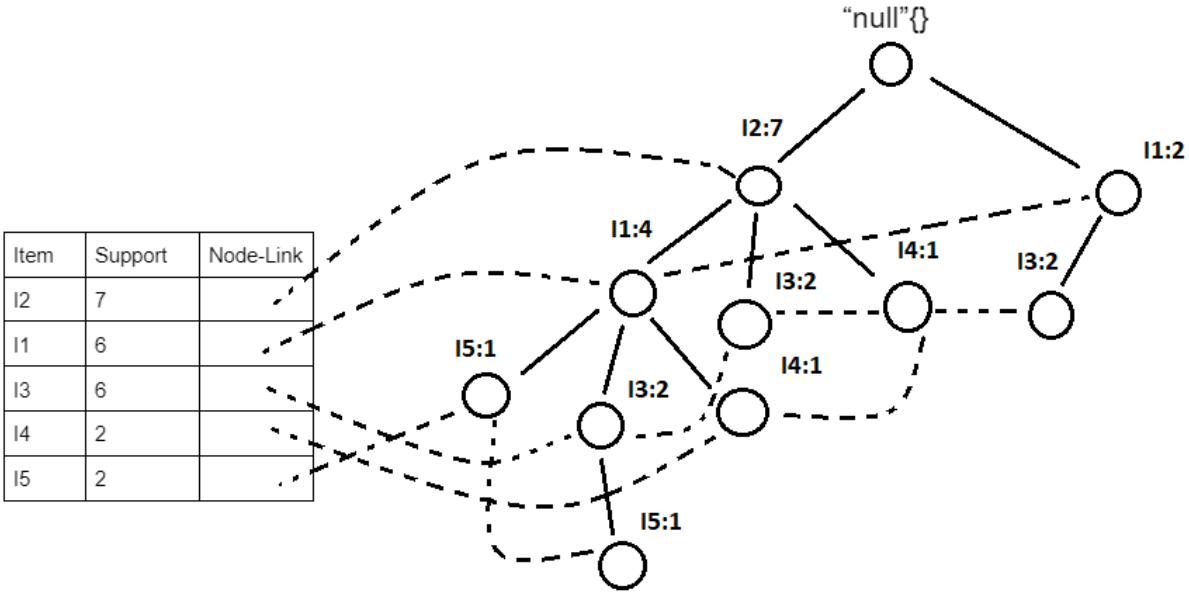
Σχήμα 3.62: Η κατασκευή της δενδρικής δομής FP-Tree.

Τέλος στην ένατη συναλλαγή T9:{I2,I1,I3}, η οποία όπως και το σύνολο των συναλλαγών είναι ταξινομημένη σύμφωνα με τη λίστα L, παρατηρείται ότι η συγκεκριμένη διαδρομή προϋπάρχει, από την αμέσως προηγούμενη συναλλαγή και το μόνο γεγονός που λαμβάνει χώρα είναι η αύξηση της τιμής υποστήριξης των κόμβων (I2),(I1),(I3), οι οποίοι αναφέρονται στα στοιχειοσυνόλων {I2},{I1},{I3} αντίστοιχα, κατά 1. Από τα παραπάνω προκύπτει και η τελική τιμή υποστήριξης των κόμβων (I2:7), (I1:4) και (I3:2).



Σχήμα 3.63: Η κατασκευή της δενδρικής δομής FP-Tree.

Μαζί με το παραπάνω δέντρο δημιουργείται και ο πίνακας κεφαλίδας συχνών στοιχείων (Frequent-Item-Header Table), ο οποίος περιέχει το σύνολο των συχνών στοιχειοσυνόλων μήκους 1, την τιμή υποστήριξης για κάθε ένα στοιχειοσύνολο, καθώς και τον κόμβο-Σύνδεσμο (Node-Link), ο οποίος δείχνει προς τον πρώτο κόμβο στη δενδρική δομή FP-Tree που φέρει το όνομα του στοιχείου.



Σχήμα 3.64: Η τελική μορφή της δενδρικής δομής FP-Tree.

Να σημειωθεί ότι στο παραπάνω σχήμα ο κόμβος I3 συνδέεται μόνο με τους υπόλοιπους κόμβους I3. Παρόλο που φαίνεται να διαπερνά τον κόμβο I4 και να καταλήγει στον τελευταίο κόμβο I3, δεν υπάρχει καμία σύνδεση του I3 με τον I4 και αυτό διότι ο κάθε κόμβος συνδέεται μόνο με κόμβους που φέρουν το ίδιο όνομα.

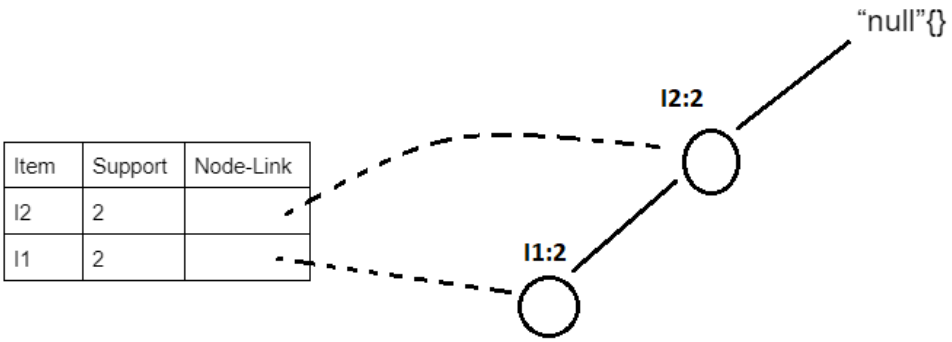
Στη συνέχεια, ακολουθεί η εύρεση του συνόλου των συχνών στοιχειοσυνόλων, τα οποία προκύπτουν μέσα από τη δενδρική δομή FP-Tree.

Αρχικά ελέγχεται το στοιχειοσύνολο {I5}, το οποίο αποτελεί το τελευταίο στοιχειοσύνολο στη λίστα L, γεγονός το οποίο μπορεί να διαπιστωθεί και από τον παραπάνω Frequent-Item-Header Table, καθώς βρίσκεται στο τέλος του παραπάνω πίνακα. Το συγκεκριμένο στοιχειοσύνολο συναντάται σε δύο περιπτώσεις και οι εμφανίσεις του μπορούν εύκολα να ανακτηθούν ακολουθώντας τους κόμβους-συνδέσμους (Node-Links), οι οποίοι συνδέουν τους κόμβους οι οποίοι αφορούν τα στοιχειοσύνολα που φέρουν το ίδιο όνομα. Με αυτόν τον τρόπο, οι διαδρομές οι οποίες σχηματίζονται στη δενδρική δομή FP-Tree που έχει σχηματιστεί προηγουμένως είναι: (I2:7,I1:4,I5:1) και (I2:7,I1:4,I3:2,I5:1). Η πρώτη διαδρομή υποδεικνύει πως το σετ (I2,I1,I5) εμφανίζεται μία φορά στη βάση των συναλλαγών, ενώ αντίστοιχα το σετ (I2,I1,I3,I5) εμφανίζεται κι αυτό μία φορά στη βάση των συναλλαγών, ασχέτως εάν κάποιο από τα υποσύνολα των δύο σετ εμφανίζεται περισσότερες φορές στη βάση των συναλλαγών. Παραδείγματος χάρη για το πρώτο σετ (I2,I1,I5) το υποσύνολο (I2,I1) εμφανίζεται τέσσερις φορές στη βάση των συναλλαγών, όπως και για το σετ (I2,I1,I3,I5) το υποσύνολό του (I2,I1,I3) εμφανίζεται δύο φορές στη βάση των συναλλαγών. Προκειμένου να ανακτηθούν τα ορθά σετ που εμφανίζονται μαζί με το {I5}, αναζητούνται μόνο οι διαδρομές προθέματος για το συγκεκριμένο στοιχειοσύνολο. Έχοντας ως πρόθεμα τον κόμβο (I5), οι δύο διαδρομές προθέματος (Prefix Paths) για το συγκεκριμένο πρόθεμα είναι: (I2,I1:1) και (I2,I1,I3:1), οι οποίες σχηματίζουν την Conditional Pattern Base. Υπενθυμίζεται σε αυτό το σημείο πως η Conditional Pattern Base αποτελεί μία μικρότερη βάση, η οποία σχηματίζεται υπό την συνθήκη ύπαρξης του {I5}, περιέχει δηλαδή τις διαδρομές οι οποίες καταλήγουν στο συγκεκριμένο στοιχείο και η οποία χρησιμοποιείται για την κατασκευή του Conditional FP-Tree. Χρησιμοποιώντας την Conditional Pattern Base ως μία βάση συναλλαγών, κατασκευάζεται η δενδρική δομή Conditional FP-Tree για τη συγκεκριμένη βάση συναλλαγών, η οποία περιέχει μία μόνο διαδρομή.

Τονίζεται επίσης πως το στοιχειοσύνολο {I3:1} στη συγκεκριμένη περίπτωση δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης. Η ολοκλήρωση της κατασκευής για το Conditional FP-Tree, οδηγεί σε ένα μόνο κλαδί όπως αποτυπώνεται και στην εικόνα 3.4.16, Έπειτα ο κάθε συνδυασμός του {I5} με τους κόμβους (I2) και (I1), οι οποίοι αναφέρονται στα αντίστοιχα στοιχειοσύνολα {I2} και {I5}, σχηματίζει το σύνολο των συχνών στοιχειοσυνόλων και πιο συγκεκριμένα: {I2}U{I5} = {I2,I5:2}, {I1}U{I5} = {I1,I5:2}, {I2,I1} U {I5} = {I2,I1,I5:2} όπως μπορεί εύκολα να διαπιστωθεί και από την εικόνα 3.4.15. Η περαιτέρω αναζήτηση συχνών στοιχειοσυνόλων, τα οποία και συνδυάζονται με το {I5}, παύει και ο έλεγχος περνάει στο επόμενο στοιχειοσύνολο.

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent Itemsets
I5	{{I2,I1:1},{I2,I1,I3:1}}	(I2:2,I1:2)	{I2,I5:2},{I1,I5:2},{I2,I1,I5:2}

Σχήμα 3.65: Η Conditional Pattern Base.

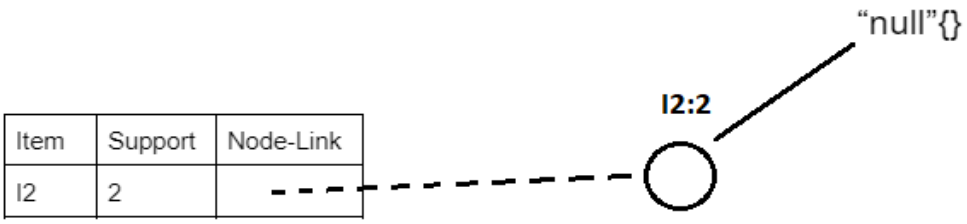


Σχήμα 3.66: Η δενδρική δομή Conditional FP-Tree.

Στη συνέχεια ελέγχεται το στοιχειοσύνολο {I4}, το οποίο βρίσκεται προτελευταίο στη λίστα L. Σε αυτήν την περίπτωση επίσης, το συγκεκριμένο στοιχειοσύνολο συναντάται σε δύο περιπτώσεις και οι εμφανίσεις του μπορούν επίσης εύκολα να ανακτηθούν. Οι διαδρομές οι οποίες σχηματίζονται στη δενδρική δομή FP-Tree, η οποία έχει υλοποιηθεί προηγουμένως, είναι: (I2:7,I1:4,I4:1) και (I2:7,I4:1). Η πρώτη από τις δύο διαδρομές, όπως διατυπώθηκε και στον προηγούμενο έλεγχο για το {I5}, υποδεικνύει ότι το σετ (I2,I1,I4) εμφανίζεται μία φορά στη βάση των συναλλαγών, ενώ και το σετ (I2,I4) εμφανίζεται επίσης μία φορά στη βάση των συναλλαγών. Προκειμένου να ανακτηθούν τα ορθά σετ που εμφανίζονται μαζί με το {I4}, αναζητούνται μόνο οι διαδρομές προθέματος για το συγκεκριμένο στοιχειοσύνολο. Έχοντας ως πρόθεμα το στοιχειοσύνολο {I4} οι δύο διαδρομές προθέματος (Prefix Paths) για το συγκεκριμένο πρόθεμα είναι: (I2,I1:1) και (I2:1), οι οποίες σχηματίζουν την Conditional Pattern Base (όπως ειπώθηκε και πιο πάνω η συγκεκριμένη βάση θα περιέχει τις διαδρομές οι οποίες καταλήγουν στο συγκεκριμένο στοιχείο). Χρησιμοποιώντας στη συνέχεια την Conditional Pattern Base ως μία βάση συναλλαγών, κατασκευάζεται η δενδρική δομή Conditional FP-Tree για τη συγκεκριμένη Βάση, η οποία περιέχει μία μόνο διαδρομή. Το στοιχειοσύνολο {I1:1} δεν ικανοποιεί την ελάχιστη τιμή υποστήριξης με αποτέλεσμα να απορριφθεί. Η ολοκλήρωση της κατασκευής για το Conditional FP-Tree, οδηγεί σε ένα μόνο κλαδί όπως αποτυπώνεται και στην εικόνα 3.4.17, Έπειτα ο συνδυασμός του {I4} με τον κόμβο (I2) ο οποίος αναφέρεται στο στοιχειοσύνολο {I2}, σχηματίζει το σύνολο των συχνών στοιχειοσυνόλων και πιο συγκεκριμένα: {I2} U {I4} = {I2,I4:2}. Η περαιτέρω αναζήτηση συχνών στοιχειοσυνόλων, τα οποία και συνδυάζονται με το {I4}, παύει και ο έλεγχος περνάει στο επόμενο στοιχειοσύνολο.

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent Itemsets
I5	{{I2,I1:1},{I2,I1,I3:1}}	(I2:2,I1:2)	{I2,I5:2},{I1,I5:2},{I2,I1,I5:2}
I4	{{I2,I1:1},{I2:1}}	(I2:2)	{I2,I4:2}

Σχήμα 3.67: Η δενδρική δομή Conditional FP-Tree.

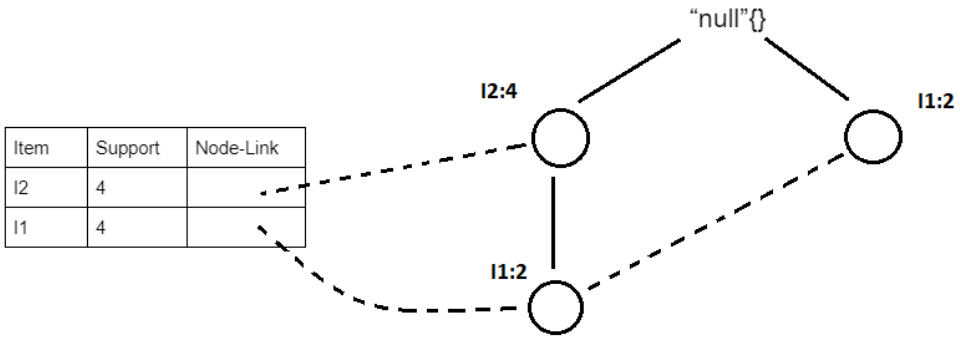


Σχήμα 3.68: Η δενδρική δομή Conditional FP-Tree.

Στη συνέχεια ελέγχεται το στοιχειοσύνολο {I3}, το οποίο βρίσκεται στην τρίτη θέση από το τέλος στη λίστα L. Σε αυτήν την περίπτωση το συγκεκριμένο στοιχειοσύνολο, συναντάται σε τρεις περιπτώσεις και οι εμφανίσεις του μπορούν επίσης εύκολα να ανακτηθούν. Οι διαδρομές οι οποίες σχηματίζονται στη δενδρική δομή FP-Tree είναι: (I2:7,I1:4,I3:2),(I2:7,I3:2) από την αριστερή πλευρά ενώ από τη δεξιά (I1:2,I3:2). Η πρώτη από τις τρεις διαδρομές υποδεικνύει ότι, το σετ (I2,I1,I3) εμφανίζεται δύο φορές στη βάση των συναλλαγών, Με τον ίδιο τρόπο προκύπτει πως τα σετ (I2,I3) και (I1,I3), εμφανίζονται από δύο φορές. Έχοντας σαν πρόθεμα το στοιχειοσύνολο {I3}, οι τρεις διαδρομές προθέματος(Prefix Paths) για το συγκεκριμένο πρόθεμα είναι: (I2,I1:2), (I2:2) και (I1:2), οι οποίες σχηματίζουν την Conditional Pattern Base για το {I3}. Χρησιμοποιώντας την Conditional Pattern Base ως μία βάση συναλλαγών, κατασκευάζεται η δενδρική δομή Conditional FP-Tree για τη συγκεκριμένη Βάση, η οποία θα περιέχει δύο διαδρομές. Η ολοκλήρωση της κατασκευής για το Conditional FP-Tree, οδηγεί σε δύο κλαδιά όπως αποτυπώνεται και στην εικόνα 3.4.19. Έπειτα ο συνδυασμός του {I3} με τον κόμβο (I2) και (I1), οι οποίοι αφορούν τα αντίστοιχα στοιχειοσύνολα, σχηματίζει το σύνολο των συχνών στοιχειοσυνόλων και πιο συγκεκριμένα: {I2} U {I3} = {I2,I3:4}, {I1} U {I3} = {I1,I3:4}, {I2,I1} U {I3} = {I2,I1,I3:2}. Παρατηρείται για το τελευταίο συχνό στοιχειοσύνολο {I2,I1,I3:2}, ότι έχει την τιμή υποστήριξης ίση με 2 και όχι με 4 όπως τα δύο υπόλοιπα συχνά στοιχειοσύνολα. Αυτό συμβαίνει καθώς στο πρώτο Conditional FP-Tree το στοιχειοσύνολο {I1} έχει την τιμή υποστήριξης ίση με 2. Άλλωστε μπορεί να διαπιστωθεί εύκολα και μέσα από την βάση των συναλλαγών του παραδείγματος ότι το συχνό στοιχειοσύνολο {I2,I1,I3} εμφανίζεται δύο φορές στη βάση των συναλλαγών. Η περαιτέρω αναζήτηση συχνών στοιχειοσυνόλων, τα οποία και συνδυάζονται με το {I3}, παύει και ο έλεγχος περνάει στο επόμενο στοιχειοσύνολο.

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent Itemsets
I5	{{I2,I1:1},{I2,I1,I3:1}}	(I2:2,I1:2)	{I2,I5:2},{I1,I5:2},{I2,I1,I5:2}
I4	{{I2,I1:1},{I2:1}}	(I2:2)	{I2,I4:2}
I3	{{I2,I1:2},{I2:2},{I1:2}}	(I2:4,I1:2),(I1:2)	{I2,I3:4},{I1,I3:4},{I2,I1,I3:2}

Σχήμα 3.69: Η δενδρική δομή Conditional FP-Tree.

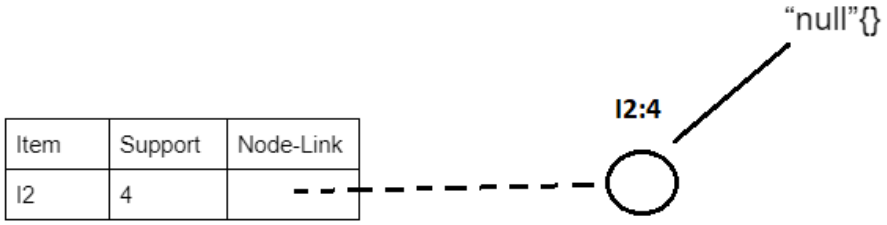


Σχήμα 3.70: Η δενδρική δομή Conditional FP-Tree.

Τέλος ελέγχεται το στοιχειοσύνολο {I1}, το οποίο βρίσκεται στη δεύτερη θέση από το τέλος στη λίστα L. Σε αυτήν την περίπτωση το συγκεκριμένο στοιχειοσύνολο συναντάται σε μία περίπτωση. Η διαδρομή η οποία σχηματίζεται στη δενδρική δομή FP-Tree είναι: (I2,I1:4). Να σημειωθεί σε αυτό το σημείο πως κανονικά οι διαδρομές συμβολίζονται με τον παραπάνω τρόπο, αναφέρεται δηλαδή η τιμή υποστήριξης μόνο του τελευταίου στοιχείου με αυτές των υπολοίπων να παραλείπονται καθώς δεν παίζουν ουσιαστικό ρόλο. Στους προηγούμενους ελέγχους ωστόσο αυτές σημειώνονταν για χάρη παραστατικότητας. Προκειμένου να ανακτηθούν τα ορθά σετ που εμφανίζονται μαζί με το {I1}, αναζητούνται μόνο οι διαδρομές προθέματος για το συγκεκριμένο στοιχειοσύνολο. Έχοντας σαν πρόθεμα το στοιχειοσύνολο {I1}, η διαδρομή προθέματος(Prefix Path), που σχηματίζεται για το συγκεκριμένο πρόθεμα είναι:(I2:4), η οποία και σχηματίζει την Conditional Pattern Base. Χρησιμοποιώντας την Conditional Pattern Base, για το {I1}, ως μία βάση συναλλαγών, κατασκευάζεται η δενδρική δομή Conditional FP-Tree για τη συγκεκριμένη Βάση, η οποία θα περιέχει μία διαδρομή. Η ολοκλήρωση της κατασκευής για το Conditional FP-Tree, οδηγεί σε ένα κλαδί όπως αποτυπώνεται και στην εικόνα 3.4.21. Έπειτα ο συνδυασμός του {I1} με τον κόμβο (I2) οι οποίοι αφορούν το αντίστοιχο στοιχειοσύνολο, σχηματίζει το σύνολο των συχνών στοιχειοσυνόλων και πιο συγκεκριμένα: {I2} U {I1} = {I2,I1:4}.

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent Itemsets
I5	{{I2,I1:1},{I2,I1,I3:1}}	(I2:2,I1:2)	{I2,I5:2},{I1,I5:2},{I2,I1,I5:2}
I4	{{I2,I1:1},{I2:1}}	(I2:2)	{I2,I4:2}
I3	{{I2,I1:2},{I2:2},{I1:2}}	(I2:4,I1:2),(I1:2)	{I2,I3:4},{I1,I3:4},{I2,I1,I3:2}
I1	{{I2:4}}	(I2:4)	{I2,I1:4}

Σχήμα 3.71: Η δενδρική δομή Conditional FP-Tree.



Σχήμα 3.72: Η δενδρική δομή Conditional FP-Tree.

Το σύνολο των συχνών στοιχειοσυνόλων είναι: {I2,I5:2},{I1,I5:2},{I2,I1,I5:2},{I2,I4:2},{I2,I3:4},{I1,I3:4},{I2,I1,I3:2},{I2,I1:4}.

Το αμέσως επόμενο βήμα αποτελεί η παραγωγή των συνδυαστικών κανόνων.

1)I1=>I2 Εμπιστοσύνη (Confidence) = Supp(I1&I2) / Supp(I1) = 4 / 6 = 66%.

2)I2=>I1 Εμπιστοσύνη (Confidence) = Supp(I1&I2) / Supp(I2) = 4 / 7 = 57%.

3)I1=>I3 Εμπιστοσύνη (Confidence) = Supp(I1&I3) / Supp(I1) = 4 / 6 = 66%.

4)I3=>I1 Εμπιστοσύνη (Confidence) = Supp(I1&I3) / Supp(I3) = 4 / 6 = 66%.

5)I1=>I5 Εμπιστοσύνη (Confidence) = Supp(I1&I5) / Supp(I1) = 2 / 6 = 33%.

6)I5=>I1 Εμπιστοσύνη (Confidence) = Supp(I1&I5) / Supp(I5) = 2 / 2 = 100%.

7)I2=>I3 Εμπιστοσύνη (Confidence) = Supp(I2&I3) / Supp(I2) = 4 / 7 = 57%.

8)I3=>I2 Εμπιστοσύνη (Confidence) = Supp(I2&I3) / Supp(I3) = 4 / 6 = 66%.

9)I2=>I4 Εμπιστοσύνη (Confidence) = Supp(I2&I4) / Supp(I2) = 2 / 7 = 28%.

10)I4=>I2 Εμπιστοσύνη (Confidence) = Supp(I2&I4) / Supp(I4) = 2 / 2 = 100%.

11)I2=>I5 Εμπιστοσύνη (Confidence) = Supp(I2&I5) / Supp(I2) = 2 / 7 = 28%.

12)I5=>I2 Εμπιστοσύνη (Confidence) = Supp(I2&I5) / Supp(I5) = 2 / 2 = 100%.

13)(I1&I2=>I3) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I1&I2) = 2 / 4 = 50%

14)(I1&I3=>I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I1&I3) = 2 / 4 = 50%

15)(I2&I3=>I1) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I2&I3) = 2 / 4 = 50%

16)(I1=>I2&I3) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I1) = 2 / 6 = 33%

17)(I2=>I1&I3) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I2) = 2 / 7 = 28%

18)(I3=>I1&I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I3) / Supp(I3) = 2 / 6 = 33%

19)(I1&I2=>I5) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I1&I2) = 2 / 4 = 50%

20)(I1&I5=>I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I1&I5) = 2 / 2 = 100%

21)(I2&I5=>I1) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I2&I5) = 2 / 2 = 100%

22)(I1=>I2&I5) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I1) = 2 / 6 = 33%

23)(I2=>I1&I5) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I2) = 2 / 7 = 28%

24)(I5=>I1&I2) Εμπιστοσύνη (Confidence) = Supp(I1&I2&I5) / Supp(I5) = 2 / 2 = 100%

Εάν είχε τεθεί στο παραπάνω παράδειγμα ως ελάχιστη τιμή εμπιστοσύνης η τιμή 80% οι κανόνες οι οποίοι θα χαρακτηρίζονταν ως χρήσιμοι είναι οι εξής: 6)I5=>I1, 10)I4=>I2, 12)I5=>I2, 20)(I1&I5=>I2), 21) (I2&I5=>I1) και 24) (I5=>I1&I2).

3.4.3 Incremental mining

Σύμφωνα με τους Han et al [44] ο αλγόριθμος FP-Growth υποστηρίζει το incremental mining και πιο συγκεκριμένα την incremental ενημέρωση της δενδρικής δομής FP-Tree. Επίσης για τον λόγο αυτό έχουν δημιουργηθεί και αλγόριθμοι, με βάση τον αλγόριθμο FP-Growth, οι οποίοι να υποστηρίζουν το incremental mining εκμεταλλευόμενοι την ιδιότητα του FP-Growth [66].

3.4.4 Συμπεράσματα

Προηγουμένως, εξετάστηκε ο αλγόριθμος FP-Growth και αναδείχθηκαν οι σημαντικές διαφορές που συνοδεύουν τον συγκεκριμένο αλγόριθμο [56], έναντι των υπολοίπων δύο, όπως για παράδειγμα η σάρωση της βάσης με τις συναλλαγές δύο φορές συνολικά, η μη δημιουργία υποψήφιων στοιχειοσυνόλων [46] και η προσέγγιση-φιλοσοφία του διαίρει και βασίλευε (Divide and Conquer) [12] για τον αλγοριθμο FP-Growth, η οποία καταφέρνει να μετατρέψει το πρόβλημα της αναζήτησης [38] των συχνών στοιχειοσυνόλων από την βάση των συναλλαγών, σε μικρότερα προβλήματα μειώνοντας παράλληλα τον χώρο αναζήτησης (search space). Μέσα από τις πτυχές του αλγορίθμου FP-Growth που περιγράφηκαν σύντομα προηγουμένως αλλά και αναλυτικότερα πιο πάνω εγείρεται το ερώτημα εάν ο συγκεκριμένος αλγόριθμος αξιοποιώντας όλα του τα χαρακτηριστικά καταφέρνει να πετύχει καλύτερες επιδόσεις από τους υπόλοιπους δύο αλγορίθμους. Αυτό αναδεικνύεται στο έκτο κεφάλαιο παρακάτω, όπου τονίζονται οι περιπτώσεις στις οποίες ο αλγόριθμος FP-Growth πετυχαίνει καλύτερες επιδοσείς έναντι των υπολοίπων δύο αλγορίθμων.

3.5 Το πακέτο arules

Το πακέτο arules [57] το οποίο εισήχθη από τους: Michael Hahsler, Christian Buchta, Bettina Gruen, Kurt Hornik, Christian Borgelt, Ian Johnson, Makhlouf Ledmi και συντηρείται από τον Michael Hahsler [57], χρησιμοποιείται στη συγκεκριμένη εργασία για την υλοποίηση των πειραμάτων δηλαδή για την σύγκριση των αλγορίθμων στο περιβάλλον R/RStudio. Ορισμένα σημαντικά κομμάτια του πακέτου θα αποτυπωθούν παρακάτω. Αρχικά, το πακέτο έως και την ημερομηνία δημοσίευσης της συγκεκριμένης εργασίας βρίσκεται στην έκδοση 1.7.6 με ημερομηνία δημοσίευσης την 23η Μαρτίου 2023. Τιτλος του είναι: Mining Association Rules and Frequent Itemsets και σκοπός του πακέτου η παροχή της απαραίτητης υποδομής για την αναπαράσταση, τον χειρισμό και την ανάλυση τόσο των συχνών στοιχειοσυνόλων όσο και των συνδυαστικών κανόνων. Επίσης, παρέχει τις υλοποιήσεις σε C των αλγορίθμων Apriori και Eclat οι οποίοι αποτελούν τους δύο από τους τρεις αλγορίθμους που περιγράφονται στην συγκεκριμένη εργασία [57]. Ξεκινώντας από τον αλγόριθμο Apriori, χρησιμοποιείται η υλοποίηση σε C από τον Christian Borgelt (2003) [57], και όπως φαίνεται από την παρακάτω εικόνα, αποτυπώνονται σύντομα: η περιγραφή του αλγορίθμου, η κλίση του αλγορίθμου στο πακέτο arules οπως και οι παράμετροι που λαμβάνει και επηρεάζουν τη συμπεριφορά καθώς και τα παραγόμενα αποτελέσματα για κάθε κλίση του αλγορίθμου. Άμεσο παράδειγμα αποτελεί η εναλλαγή της τιμής υποστήριξης (support) το οποίο ανήκει στην parameter [57]. Ο αλγόριθμος Apriori επιστρέφει στοιχειοσύνολα ή συνδυαστικούς κανόνες [57].

apriori *Mining Associations with the Apriori Algorithm*

Description

Mine frequent itemsets, association rules or association hyperedges using the Apriori algorithm.

Usage

```
apriori(data, parameter = NULL, appearance = NULL, control = NULL, ...)
```

Σχήμα 3.73: Η υλοποίηση του αλγορίθμου Apriori εντος του arules [57]

Έπειτα για τον αλγόριθμο Eclat χρησιμοποιείται η υλοποίηση σε C του αλγορίθμου από τον Christian Borgelt (2003) [57] για τη δημιουργία των συχνών στοιχειοσυνόλων, η οποία αποτυπώνεται μέσα από την παρακάτω εικόνα:

eclat *Mining Associations with Eclat*

Description

Mine frequent itemsets with the Eclat algorithm. This algorithm uses simple intersection operations for equivalence class clustering along with bottom-up lattice traversal.

Usage

```
eclat(data, parameter = NULL, control = NULL, ...)
```

Σχήμα 3.74: Η υλοποίηση του αλγορίθμου Eclat εντος του arules [57]

Αντίστοιχα με προηγουμένως παρουσιάζεται: η περιγραφή του αλγορίθμου, η κλίση του αλγορίθμου στο πακέτο arules οπως και οι παράμετροι που λαμβάνει και επηρεάζουν τη συμπεριφορά καθώς και τα παραγόμενα αποτελέσματα για κάθε μία κλίση του αλγορίθμου. Σημειώνεται στο σημείο αυτό πως ο αλγόριθμος Eclat επιστρέφει στοιχειοσύνολα ενώ για την παραγωγή των συνδυαστικών κανόνων χρησιμοποιεί την ruleInduction η οποία ανήκει κι αυτή στο πακέτο arules [57].

Τέλος για τον αλγόριθμο FP-Growth δίνεται η δυνατότητα εκτέλεσης του μέσα από το interface fim4r, το οποίο δίνει επίσης τη δυνατότητα εκτέλεσης και αρκετών άλλων αλγορίθμων. Σκοπός της παρούσας εργασίας είναι η εξέταση του αλγορίθμου FP-Growth, οπότε στην παράμετρο method χρησιμοποιείται ο αλγόριθμος fpgrowth. Αναλυτικότερα και σύμφωνα με την παρακάτω εικόνα, αποτυπώνονται: η περιγραφή του αλγορίθμου, η κλίση του αλγορίθμου στο πακέτο μέσω του interface fim4r οπως και οι παράμετροι που λαμβάνει και επηρεάζουν τη συμπεριφορά καθώς και τα παραγόμενα αποτελέσματα για κάθε κλίση του αλγορίθμου [57].

fim4r *Interface to Mining Algorithms from fim4r*

Description

Interfaces the algorithms implemented in fim4r. The algorithms include: Apriori, Eclat, FPgrowth, Carpenter, IsTa, RElim and SaM.

Usage

```
fim4r(
  transactions,
  method = NULL,
  support = 0.1,
  confidence = 0.8,
  target = "frequent",
  originalSupport = TRUE,
  appear = NULL,
  report = NULL,
  verbose = TRUE,
  ...
)
```

Εικόνα 3.75: Το interface fim4r για την υλοποίηση του FP-Growth [57]

Στο σημείο αυτό θα πρέπει να τονιστεί ιδιαίτερα ότι για τον αλγόριθμο FP-Growth από τη στιγμή που έγινε διαθέσιμος προς εκτέλεση, δηλαδή από την έκδοση 1.7.3 του πακέτου arules έως και την έκδοση 1.7.5, η κλίση του αλγορίθμου μέσω του fim4r όπως αποτυπώνεται από την παραπάνω εικόνα δεν ήταν η ίδια. Πιο συγκεκριμένα όπως φαίνεται και από την παρακάτω εικόνα, η οποία περιέχει την κλίση του αλγορίθμου μέσα από το fim4r για την έκδοση του πακέτου arules 1.7.5, οι τιμές υποστήριξης και εμπιστοσύνης ήταν στην κλίμακα [0-100], σε αντίθεση με ότι ισχύει στην τωρινή 1.7.6 στην οποία οι αντίστοιχες τιμές υποστήριξης και εμπιστοσύνης συμβαδίζουν με τους υπόλοιπος δύο αλγορίθμους και βρίσκονται στην κλίμακα [0-1]. Ο συγγραφέας της παρούσας εργασίας εντόπισε, ότι όταν είχε επιλεγεί ως στόχος η παραγωγή των συνδυαστικών κανόνων, το σύνολο των συνδυαστικών κανόνων που προκύπτει από την εκτέλεση του αλγορίθμου FP-Growth, δεν είναι το ίδιο με αυτό των υπολοίπων δύο αλγοριθμων Apriori και Eclat. Έτσι, επικοινωνώντας μέσω email με τον κ.Michael Hahsler, έναν από τους συγγραφείς και παράλληλα ο συντηρητής του πακέτου arules, τέθηκε το παραπάνω θέμα μέσω ενός σύντομου παραδείγματος. Ο κ.Michael Hahsler, εντόπισε κι ο ίδιος το θέμα και κι απάντησε πως το interface fim4r, χρησιμοποιεί αντί για τον αυθεντικό ορισμό της υποστήριξης, την υποστήριξη του αριστερού μέλους (Left-Hand-Side-LHS) με αποτέλεσμα να προκύπτουν περισσότεροι συνδυαστικοί κανόνες οι οποίοι δεν ικανοποιούν τον περιορισμό της ελάχιστης τιμής υποστήριξης. Αμέσως μετά τη διόρθωση του συγκεκριμένου θέματος δοκιμάστηκε εκ νέου ο αλγόριθμος FP-Growth, με τιμές υποστήριξης πλέον στην κλιμακα [0-1], και προέκυψε πως το σύνολο των συνδυαστικών κανόνων είναι το ίδιο με αυτό των υπολοίπων δύο αλγορίθμων.

`fim4r` *Interface to Mining Algorithms from fim4r*

Description

Interfaces the algorithms implemented in fim4r. The algorithms include: Apriori, Eclat, FPgrowth, Carpenter, IsTa, RElim and SaM.

Usage

```
fim4r(
  transactions,
  method = NULL,
  target = "frequent",
  report = NULL,
  appear = NULL,
  ...
)
```

Σχήμα 3.76: Το interface fim4r της προηγούμενης έκδοσης 1.7.5 [57]

Κεφάλαιο 4ο: Τα σύνολα δεδομένων

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται αναλυτικά τα δύο σύνολα δεδομένων τα οποία χρησιμοποιούνται στην υλοποίηση των παραδειγμάτων σύγκρισης των αλγορίθμων Apriori, Eclat και FP-Growth. Αρχικά παρουσιάζεται το κάθε σύνολο δεδομένων μαζί με τα επιμέρους χαρακτηριστικά που το απαρτίζουν και αναφέρεται η μετατροπή τους στην τελική μορφή για την υλοποίηση των παραδειγμάτων παρακάτω.

4.1 Σύνολο δεδομένων Adult

4.1.1 Περιγραφή του συνόλου δεδομένων Adult

Το σύνολο δεδομένων Adult, το οποίο χρησιμοποιείται για την εκτέλεση των παραδειγμάτων προέρχεται από το Kaggle και δημιουργός του είναι ο κ.Brijesh B. Mehta. Η έκδοση του συνόλου δεδομένων, η οποία χρησιμοποιείται στις περιπτώσεις των παραδειγμάτων και αριθμεί 10 εκατομμύρια καταχωρίσεις, προέρχεται από το σύνολο δεδομένων Adult του UCI repository η οποία αριθμεί 33 χιλιάδες καταχωρίσεις. Η έκδοση δηλαδή των 10 εκατομμυρίων καταχωρίσεων, αποτελεί μία συνθετική έκδοση του αρχικού συνόλου δεδομένων από το UCI repository, όπως χαρακτηριστικά σημειώνει και ο κ.Brijesh B. Mehta. Κατά την εισαγωγή του συνόλου δεδομένων των 10 εκατομμυρίων καταχωρίσεων, για την εκτέλεση των παραδειγμάτων, αυτό αποτελεί ένα data frame 10 εκατομμυρίων καταχωρίσεων στις ακόλουθες 15 μεταβλητές:

- **Age**: Η ηλικία η οποία αποτελεί ένα numeric vector. Στη συνέχεια μετατρέπεται σε factor με επίπεδα: Young (0-25), Middle-aged (26-45), Senior (46-65) και Old (66+)

- **Workclass**: Ο τύπος απασχόλησης ο οποίος αποτελεί factor με τα επίπεδα: Federal-gov, Local-gov, Never-worked, Private, Self-emp-inc, Self-emp-not-inc, State-gov, και Without-pay.

- **Fnlwgt**: Το τελικό βάρος (Final Weight), το οποίο αποτελεί ένα numeric vector. Στα παραδείγματα σύγκρισης των αλγορίθμων η συγκεκριμένη μεταβλητή παραλείπεται.

- **Education**: Η εκπαίδευση, η οποία αποτελεί ένα ordered factor με τα επίπεδα να είναι: Preschool < 1st-4th < 5th-6th < 7th-8th < 9th < 10th < 11th < 12th < HS-grad < Prof-school < Assoc-acdm < Assoc-voc < Some-college < Bachelors < Masters < Doctorate.

- **Education-num**: Η αριθμητική αναπαράσταση της εκπαίδευσης, η οποία αποτελεί ένα numeric vector. Στα παραδείγματα σύγκρισης των αλγορίθμων η συγκεκριμένη μεταβλητή παραλείπεται.

- **Marital-Status**: Η οικογενειακή κατάσταση, η οποία αποτελεί έναν factor με τα επίπεδα να είναι: Divorced, Married-AF-spouse, Married-civ-spouse, Married-spouse-absent, Never-married, Separated, and Widowed

- **Occupation**: Το επάγγελμα, το οποίο αποτελεί factor με τα επίπεδα να είναι: Adm-clerical, Armed-Forces, Craft-repair, Exec-managerial, Farming-fishing, Handlers-cleaners, Machine-op-inspct, Other-service, Priv-house-serv, Prof-specialty, Protective-serv, Sales, Tech-support και Transport-moving.

- **Relationship**: Η σχέση, η οποία αποτελεί factor με τα επίπεδα να είναι: Husband, Not-in-family, Other-relative, Own-child, Unmarried και Wife.

- **Race**: Η φυλή, η οποία αποτελεί factor με τα επίπεδα να είναι: Amer-Indian-Eskimo, Asian-Pac-Islander, Black, Other, and White

- **Sex**: Το φύλο, το οποίο αποτελεί factor με τα επίπεδα να είναι: Female και Male.

- **Capital-gain**: Το κεφαλαιακό κέρδος, το οποίο αποτελεί ένα numeric vector. Έπειτα μετατρέπεται σε ordered factor με επίπεδα: None (0), Low (0 < διάμεσος των τιμών μεγαλύτερων του 0 < max) και High (>=max).

- **Capital-loss**: Η απώλεια κεφαλαίου, η οποία αποτελεί ένα numeric vector. Αντίστοιχα με πριν μετατρέπεται σε ordered factor με επίπεδα: None (0), Low (0 < διάμεσος των τιμών μεγαλύτερων του 0 < max) και High (>=max).

- **Hours-per-week**: Οι ώρες εργασίας την εβδομάδα, οι οποίες αποτελούν ένα numeric vector. Κατόπιν μετατρέπονται σε ordered factor με επίπεδα: Part-time (0-25), Full-time (25-40), Over-time (40-60) και Too-much (60+).

- **Native-country**: Η χώρα καταγωγής η οποία αποτελεί ένα factor με τα επίπεδα να είναι: Cambodia, Canada, China, Columbia, Cuba, Dominican-Republic, Ecuador, El-Salvador, England, France, Germany, Greece, Guatemala, Haiti, Holand-Netherlands, Honduras, Hong, Hungary, India, Iran, Ireland, Italy, Jamaica, Japan, Laos, Mexico, Nicaragua, Outlying-US(Guam-USVI-etc), Peru, Philippines, Poland, Portugal, Puerto-Rico, Scotland, South, Taiwan, Thailand, Trinadad&Tobago, United-States, Vietnam, και Yugoslavia.

- **Income**: Το εισόδημα το οποίο αποτελεί factor με τα επίπεδα: <=50K και >50K, όπου το K εννοείται χιλιάδες δολάρια.

	age	workclass	education	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income
1	39	State-gov	Bachelors	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
2	50	Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
3	38	Private	HS-grad	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
4	53	Private	11th	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
5	28	Private	Bachelors	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
6	37	Private	Masters	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	<=50K
7	49	Private	9th	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16	Jamaica	<=50K
8	52	Self-emp-not-inc	HS-grad	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45	United-States	>50K
9	31	Private	Masters	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	United-States	>50K
10	42	Private	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	5178	0	40	United-States	>50K
11	37	Private	Some-college	Married-civ-spouse	Exec-managerial	Husband	Black	Male	0	0	80	United-States	>50K
12	30	State-gov	Bachelors	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	0	0	40	India	>50K
13	23	Private	Bachelors	Never-married	Adm-clerical	Own-child	White	Female	0	0	30	United-States	<=50K

Σχήμα 4.1: Το σύνολο δεδομένων Adult πριν την προεπεξεργασία του

	age	workclass	education	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income
1	Middle-aged	State-gov	Bachelors	Never-married	Adm-clerical	Not-in-family	White	Male	Low	None	Full-time	United-States	<=50K
2	Senior	Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	None	None	Part-time	United-States	<=50K
3	Middle-aged	Private	HS-grad	Divorced	Handlers-cleaners	Not-in-family	White	Male	None	None	Full-time	United-States	<=50K
4	Senior	Private	11th	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	None	None	Full-time	United-States	<=50K
5	Middle-aged	Private	Bachelors	Married-civ-spouse	Prof-specialty	Wife	Black	Female	None	None	Full-time	Cuba	<=50K
6	Middle-aged	Private	Masters	Married-civ-spouse	Exec-managerial	Wife	White	Female	None	None	Full-time	United-States	<=50K
7	Senior	Private	9th	Married-spouse-absent	Other-service	Not-in-family	Black	Female	None	None	Part-time	Jamaica	<=50K
8	Senior	Self-emp-not-inc	HS-grad	Married-civ-spouse	Exec-managerial	Husband	White	Male	None	None	Over-time	United-States	>50K
9	Middle-aged	Private	Masters	Never-married	Prof-specialty	Not-in-family	White	Female	High	None	Over-time	United-States	>50K
10	Middle-aged	Private	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	High	None	Full-time	United-States	>50K
11	Middle-aged	Private	Some-college	Married-civ-spouse	Exec-managerial	Husband	Black	Male	None	None	Too-much	United-States	>50K
12	Middle-aged	State-gov	Bachelors	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	None	None	Full-time	India	>50K
13	Young	Private	Bachelors	Never-married	Adm-clerical	Own-child	White	Female	None	None	Full-time	United-States	<=50K

Σχήμα 4.2: Το σύνολο δεδομένων Adult έπειτα από την προεπεξεργασία του

4.2 Σύνολο δεδομένων City-Isp-Daily-Speeds

4.2.1 Περιγραφή του συνόλου δεδομένων City-Isp-Daily-Speeds

Το σύνολο δεδομένων Ookla Netindex Data on Internet Speed, Estimates of Internet speed by City (2008-2014) and Country & Region (2010-2014), το οποίο χρησιμοποιείται για την εκτέλεση των παραδειγμάτων προέρχεται από το Kaggle και δημιουργός του είναι ο κ. Qasim Khan. Από την ονομασία του συνόλου δεδομένων, προκύπτει πως το συγκεκριμένο σύνολο δεδομένων περιέχει εκτιμήσεις για την ταχύτητα του internet ανα πόλη για τα έτη 2008 έως 2014 και ανα χώρα και περιοχή για τα έτη 2010 έως 2014. Για τα παραδείγματα εκτέλεσης των αλγορίθμων χρησιμοποιείται ένα υποσύνολο του συνόλου δεδομένων, συγκεκριμένα το city_isp_daily_speeds. Το παραπάνω υποσύνολο συμπεριλαμβάνει τους Internet Service Providers(ISP) και αριθμεί 34449499 καταχωρίσεις. Κατά την εισαγωγή του συνόλου δεδομένων city_isp_daily_speeds, με σκοπό την εκτέλεση των παραδειγμάτων, αυτό αποτελεί ένα data frame 34449499 καταχωρίσεων στις ακόλουθες 11 μεταβλητές:

- **Country**: Η ονομασία της χώρας, η οποία αποτελεί έναν factor με κάθε ένα από τα επίπεδα να αντιπροσωπεύει την αντίστοιχη χώρα.

- **Country_code**: Ο κωδικός ISO της αντίστοιχης χώρας, ο οποίος αποτελεί έναν factor με κάθε ένα από τα επίπεδα να αντιπροσωπεύει την αντίστοιχη χώρα όπως αναφέρθηκε και προηγουμένως.

- **Region:** Η περιοχή από την οποία γίνονται οι δοκιμές (tests), η οποία αποτελεί factor με κάθε ένα από τα επίπεδα να αντιπροσωπεύει την εκάστοτε περιοχή. Οι περιοχές αυτές προέρχονται από τις παραπάνω χώρες.

- **Region_code:** Ο κωδικός της περιοχής, ο οποίος μπορεί να αποτελείται μόνο από αριθμούς, μόνο από χαρακτήρες ή και συνδυασμό των παραπάνω, αποτελεί factor με το κάθε επίπεδο να αποτελεί την περιοχή για την οποία έχει ανατεθεί ο εκάστοτε κωδικός.

- **City:** Η πόλη από την οποία γίνονται οι δοκιμές (tests), η οποία αποτελεί factor με κάθε ένα από τα επίπεδα να αντιπροσωπεύει την εκάστοτε πόλη. Οι πόλεις οι οποίες συναντώνται προέρχονται από τις παραπάνω χώρες.

- **Isp_name:** Το όνομα του Internet Service Provider (ISP) αποτελεί factor και κάθε ένα από τα επίπεδα αντιπροσωπεύει έναν Internet Service Provider (ISP). Όπως είναι αναμενόμενο ορισμένοι από τους ISPs συναντώνται μόνο σε ορισμένες χώρες.

- **Date:** Η ημερομηνία πραγματοποίησης των δοκιμών (tests) αποτελεί Date και για κάθε μία καταχώρηση αποτυπώνεται η ακριβής ημερομηνία στην οποία πραγματοποιήθηκαν οι δοκιμές.

- **Download_kbps:** Η μέση ταχύτητα λήψης σε kbps, η οποία έχει προκύψει από τις μετρήσεις επί του συνόλου των δοκιμών. Έπειτα μετατρέπεται σε ένα ordered factor με επίπεδα: "Very Slow","Slow","Fast","Very Fast".

- **Upload_kbps:** Η μέση ταχύτητα μεταφόρτωσης σε kbps, η οποία έχει προκύψει επίσης από τις μετρήσεις επί του συνόλου των δοκιμών. Στη συνέχεια μετατρέπεται σε ένα ordered factor με επίπεδα: "Very Slow","Slow","Fast","Very Fast".

- **Total_tests**: Ο συνολικός αριθμός των δοκιμών (tests) ανα καταχώρηση ο οποίος αποτελεί numeric vector και μετέπειτα ένα ordered factor με επίπεδα: "Few tests","Medium amount of tests","Many tests","A lot of tests"

- **Distance_miles:** Η μέση απόσταση σε μίλια (miles) μεταξύ του πελάτη και του server για το σύνολο των δοκιμών. Έπειτα μετατρέπεται σε ordered factor με επίπεδα: "Few miles","Medium amount of miles","Many miles","A lot of miles".

	country	country_code	region	region_code	city	isp_name	date	download_kbps	upload_kbps	total_tests	distance_miles
1	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-02	535.573	348.254	125	22.35440
2	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-03	547.641	351.456	131	21.75440
3	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-04	548.410	358.015	133	18.29480
4	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-05	550.538	379.174	125	19.36560
5	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-06	543.765	403.867	125	19.36560
6	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-07	516.671	401.441	126	21.63520
7	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-08	512.871	407.145	128	20.48030
8	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-09	505.080	416.974	132	19.09120
9	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-10	490.692	429.234	132	19.09120
10	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-11	505.471	438.907	135	18.70170
11	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-12	503.679	444.371	140	18.08960
12	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-13	501.227	446.801	141	17.97240
13	El Salvador	SV	San Salvador	10	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-14	496.941	450.249	146	17.41040

Εικόνα 4.3: Το σύνολο δεδομένων city_isp_daily_speeds πριν την προεπεξεργασία του

	country	country_code	region	city	isp_name	date	download_kbps	upload_kbps	total_tests	distance_miles
1	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-02	Very Slow	Very Slow	Few tests	Few miles
2	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-03	Very Slow	Very Slow	Few tests	Few miles
3	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-04	Very Slow	Very Slow	Few tests	Few miles
4	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-05	Very Slow	Very Slow	Few tests	Few miles
5	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-06	Very Slow	Very Slow	Few tests	Few miles
6	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-07	Very Slow	Very Slow	Few tests	Few miles
7	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-08	Very Slow	Very Slow	Few tests	Few miles
8	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-09	Very Slow	Very Slow	Few tests	Few miles
9	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-10	Very Slow	Very Slow	Few tests	Few miles
10	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-11	Very Slow	Very Slow	Few tests	Few miles
11	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-12	Very Slow	Very Slow	Few tests	Few miles
12	El Salvador	SV	San Salvador	San Salvador	Telefonica Moviles El Salvador S.A. de C.V.	2008-01-13	Very Slow	Very Slow	Few tests	Few miles

Εικόνα 4.4: Το σύνολο δεδομένων city_isp_daily_speeds έπειτα από την προεπεξεργασία του

Κεφάλαιο 5ο: Ανάλυση Δεδομένων-Αποτελέσματα

Θα πρέπει να τονιστεί σε αυτό το σημείο ότι στο πακέτο arules το οποίο χρησιμοποιείται για την υλοποίηση των πειραμάτων της σύγκρισης των τριών αλγορίθμων, μέχρι και την έκδοση 1.7.2 δεν υπήρχε η δυνατότητα εκτέλεσης του αλγορίθμου FP-Growth. Αντιθέτως από την έκδοση 1.7.3 δίνεται η δυνατότητα χρήσης του συγκεκριμένου αλγορίθμου μέσα από το πακέτο arules καθώς προστέθηκε το interface για το fim4r το οποίο περιλαμβάνει και τον αλγόριθμο FP-Growth.

Για την καταγραφή του χρόνου εκτέλεσης των τριών αλγορίθμων γινεται χρήση του πακέτου bench [58] και πιο συγκεκριμένα της συνάρτησης mark. Για την κάθε μία εκτέλεση των αλγορίθμων καταγράφεται μία πληθώρα αποτελεσμάτων, όπως για παράδειγμα ο ελάχιστος χρόνος που καταγράφηκε, ο διάμεσος, ο οποίος χρησιμοποιείται στα σενάρια σύγκρισης των αλγορίθμων παρακάτω, ο συνολικός χρόνος εκτέλεσης, το αποτέλεσμα της εκτέλεσης και αρκετά άλλα χρήσιμα αποτελέσματα.

Για την καταγραφή της κατανάλωσης μνήμης των τριών αλγορίθμων χρησιμοποιείται η συνάρτηση peakRAM, η οποία ανήκει στο πακέτο peakRAM [59]. Εναλλακτικά μπορεί κάποιος να χρησιμοποιήσει τις συναρτήσεις Rprof ή Rprofmem του πακέτου utils [60] ή τη συνάρτηση profvis του πακέτου profvis [61] ή τη συνάρτηση profmem του πακέτου profmem [62].

Παρακάτω αναφέρονται τα παραγόμενα αποτελέσματα των αλγορίθμων για τα δύο σύνολα δεδομένων Adult και City_isp_daily_speeds τα οποία περιέχουν ένα, πέντε και δέκα εκατομμύρια συναλλαγές. Για συντομία θα αναφέρεται το καθένα από αυτά με το όνομά του συνοδευόμενο από τον αριθμό των συναλλαγών.

5.1 Μεθοδολογία σύγκρισης των αλγορίθμων Apriori, Eclat και FP-Growth

Αρχικά για το εκάστοτε σενάριο επιλέγεται το κατάλληλο σύνολο δεδομένων το οποίο θα χρησιμοποιηθεί για τη σύγκριση των αλγορίθμων. Κατόπιν, εισάγεται το παραπάνω σύνολο δεδομενων και αρχίζει η τελική διαμόρφωση του για την εκκίνηση της σύγκρισης των αλγορίθμων στις διάφορες τιμές υποστήριξης που περιλαμβάνει το σενάριο. Μόλις έρθει το σύνολο δεδομένων στην απαιτούμενη μορφή, μετατρέπεται στη βάση των συναλλαγών και είναι έτοιμο για την εκκίνηση της σύγκρισης των αλγορίθμων. Αρχικά θα βρεθεί το σύνολο των συχνών στοιχειοσυνόλων το οποίο έπειτα θα τροφοδοτήσει την συνάρτηση ruleInduction, του πακέτου arules, για την παραγωγή των συνδυαστικών κανόνων. Παρόλο που οι αλγόριθμοι Apriori και FP-Growth μπορούν απευθείας να δημιουργήσουν συνδυαστικούς κανόνες θέτοντας στην παράμετρο target = “frequent itemsets” και target = “frequent” αντίστιχα, κατα την κλίση των αλγορίθμων, θεωρείται εγκυρότερο να υπάρχει ο διαχωρισμός για την εύρεση των συχνών στοιχειοσυνόλων και την παραγωγή των συνδυαστικών κανόνων. Στο πρώτο σενάριο επιλέγεται το αντίστοιχο σύνολο δεδομένων το οποίο περιλαμβάνει 1 εκατομμύριο συναλλαγές στο δεύτερο πείραμα περιλαμβάνεται το σύνολο δεδομένων το οποίο αριθμεί 5 εκατομμύρια συναλλαγές και στο τρίτο παράδειγμα είναι το σύνολο δεδομένων το οποίο περιλαμβάνει 10 εκατομμύρια συναλλαγές. Εφόσον έχει οριστεί η ελάχιστη τιμή υποστήριξης, εκτελείται ο αλγόριθμος Apriori με τις κατάλληλες παραμέτρους, την ελάχιστη τιμή υποστήριξης και την επιλογή εύρεσης των συχνών στοιχειοσυνόλων target = “frequent itemsets” και αποθηκεύεται το αποτέλεσμα το οποίο περιέχει το σύνολο των συχνών στοιχειοσυνόλων το οποίο. Στη συνέχεια εκτελείται ο αλγόριθμος Eclat με την κατάλληλη παράμετρο, την ελάχιστη τιμή υποστήριξης δηλαδή, και αποθηκεύεται το αποτέλεσμα της εκτέλεσης το οποίο περιέχει το σύνολο των συχνών στοιχειοσυνόλων για τον αλγόριθμο Eclat. Σημειώνεται πως δεν δίνεται κάποια άλλη παράμετρος στον αλγόριθμο Eclat καθώς

από την φύση του ο αλγόριθμος, στην υλοποίηση του η οποία περιλαμβάνεται στο πακέτο arules, αναζητά τα συχνά στοιχειοσύνολα. Τέλος εκτελείται ο αλγόριθμος FP-Growth με τις κατάλληλες παραμέτρους, την ελάχιστη τιμή υποστήριξης, τον στόχο ο οποίος στη συγκεκριμένη περίπτωση είναι η εύρεση των συχνών στοιχειοσυνόλων και αποθηκεύεται το αποτέλεσμα το οποίο περιλαμβάνει τα συχνά στοιχειοσύνολα του αλγορίθμου FP-Growth. Έπειτα επιλέγεται η επόμενη στη σειρά ελάχιστη τιμή υποστήριξης και επαναλαμβάνεται η παραπάνω διαδικασία. Αφού ολοκληρωθεί η εκτέλεση των αλγορίθμων και για την τελευταία ελάχιστη τιμή υποστήριξης, στη συνέχεια εκτελείται ο εκάστοτε αλγόριθμος εντός της peakRAM προκειμένου να καταγραφεί η κατανάλωση της μνήμης για την αντίστοιχη ελάχιστη τιμή υποστήριξης και αποθηκεύεται το αποτέλεσμα. Αφου ολοκληρωθεί η εκτέλεση των αλγορίθμων εντός της peakRAM για όλες τις ελάχιστες τιμές υποστήριξης ακολουθεί η εκτέλεση της ruleInduction η οποία τροφοδοτείται με τα συχνά στοιχειοσύνολα τα οποία εντόπισε ο κάθε αλγόριθμος για την αντίστοιχη ελάχιστη τιμή υποστήριξης και καταγράφεται ο χρόνος εκτέλεσης της για τον κάθε αλγόριθμο. Τέλος εκτελείται ξανά η peakRAM αυτή τη φορά για την ruleInduction και καταγράφεται η κατανάλωση της μνήμης. Από την συλλογή των παραπάνω αποτελεσμάτων έπειτα προκύπτουν τα γραφήματα τα οποία αποτυπώνουν την συμπεριφορά των αλγορίθμων κατα τη διάρκεια της εκτέλεσης τους.

5.2 Παραγόμενα αποτελέσματα αλγορίθμου Apriori

5.2.1 Εύρεση συχνών στοιχειοσυνόλων

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων Adult1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	11542	623	151	67	39

Πίνακας 5.1: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων Adult5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	11499	645	154	66	39

Πίνακας 5.2: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων Adult10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	11498	650	156	66	39

Πίνακας 5.3: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων City_isp_daily_speeds1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	1215	142	34	20	18

Πίνακας 5.4: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων City_isp_daily_speeds5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	1219	145	34	20	18

Πίνακας 5.5: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων City_isp_daily_speeds10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	1220	145	34	20	18

Πίνακας 5.6: Το πλήθος των συχνών στοιχειοσυνόλων.

5.2.2 Παραγωγή συνδυαστικών κανόνων

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων Adult1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	7550	264	65	20	14

Πίνακας 5.7: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων Adult5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	7106	327	77	23	17

Πίνακας 5.8: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων Adult10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	7294	346	83	25	19

Πίνακας 5.9: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων City_isp_daily_speeds1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	1073	47	8	6	2

Πίνακας 5.10: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων City_isp_daily_speeds5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	1080	49	8	6	2

Πίνακας 5.11: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου Apriori για το σύνολο δεδομένων City_isp_daily_speeds10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	1082	49	8	6	2

Πίνακας 5.12: Το πλήθος των συνδυαστικών κανόνων.

5.3 Παραγόμενα αποτελέσματα αλγορίθμου Eclat

5.3.1 Εύρεση συχνών στοιχειοσυνόλων

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων Adult1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	11542	623	151	67	39

Πίνακας 5.13: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων Adult5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	11499	645	154	66	39

Πίνακας 5.14: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων Adult10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	11498	650	156	66	39

Πίνακας 5.15: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων City_isp_daily_speeds1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	1215	142	34	20	18

Πίνακας 5.16: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων City_isp_daily_speeds5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	1219	145	34	20	18

Πίνακας 5.17: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων City_isp_daily_speeds10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	1220	145	34	20	18

Πίνακας 5.18: Το πλήθος των συχνών στοιχειοσυνόλων.

5.3.2 Παραγωγή συνδυαστικών κανόνων

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων Adult1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	7550	264	65	20	14

Πίνακας 5.19: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων Adult5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	7106	327	77	23	17

Πίνακας 5.20: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων Adult10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	7294	346	83	25	19

Πίνακας 5.21: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων City_isp_daily_speeds1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	1073	47	8	6	2

Πίνακας 5.22: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων City_isp_daily_speeds5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	1080	49	8	6	2

Πίνακας 5.23: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων City_isp_daily_speeds10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	1082	49	8	6	2

Πίνακας 5.24: Το πλήθος των συνδυαστικών κανόνων

5.4 Παραγόμενα αποτελέσματα αλγορίθμου FP-Growth

5.4.1 Εύρεση συχνών στοιχειοσυνόλων

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων Adult1m

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	11542	623	151	67	39

Πίνακας 5.25: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων Adult5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	11499	645	154	66	39

Πίνακας 5.26: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων Adult10m

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	11498	650	156	66	39

Πίνακας 5.27: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου Eclat για το σύνολο δεδομένων City_isp_daily_speeds1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	1215	142	34	20	18

Πίνακας 5.28: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων City_isp_daily_speeds5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	1219	145	34	20	18

Πίνακας 5.29: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων City_isp_daily_speeds10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Frequent Itemsets	1220	145	34	20	18

Πίνακας 5.30: Το πλήθος των συχνών στοιχειοσυνόλων.

5.4.2 Παραγωγή συνδυαστικών κανόνων

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων Adult1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	7550	264	65	20	14

Πίνακας 5.31: Το πλήθος των συχνών στοιχειοσυνόλων.

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων Adult5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	7106	327	77	23	17

Πίνακας 5.32: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων Adult10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	7294	346	83	25	19

Πίνακας 5.33: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων City_isp_daily_speeds1m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	1073	47	8	6	2

Πίνακας 5.34: Το πλήθος των συνδυαστικών κανόνων.

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων City_isp_daily_speeds5m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	1080	49	8	6	2

Πίνακας 5.35: Το πλήθος των συνδυαστικών κανόνων.\

- Εκτέλεση του αλγορίθμου FP-Growth για το σύνολο δεδομένων City_isp_daily_speeds10m:

Support	0.01	0.05	0.1	0.15	0.2
Number of Association Rules	1082	49	8	6	2

Πίνακας 5.36: Το πλήθος των συνδυαστικών κανόνων

Σύγκριση των παραγόμενων αποτελεσμάτων για τους τρεις αλγόριθμους(αν τα συχνά στοιχειοσύνολα τα οποία βρίσκει ο κάθε αλγόριθμος είναι τα ίδια με αυτά των υπολοίπων δύο):

Για τον σκοπό αυτό μπορεί να γίνει χρήση της συνάρτησης match του πακέτου arules [57]. Πιο συγκεκριμένα αρχικά θα χρειαστεί να αποθηκευτεί τα σύνολα των συχνών στοιχειοσυνόλων, που έχουν προκύψει από την εκτέλεση του αλγορίθμου Apriori, του αλγορίθμου Eclat και του αλγορίθμου FP-Growth, για την ίδια τιμή υποστήριξης. Έπειτα εισάγοντας το ένα σύνολο ως την μία παράμετρο στην συνάρτηση match και ένα δεύτερο σύνολο ως την δεύτερη παράμετρο στην ίδια συνάρτηση, το αποτέλεσμα που προκύπτει είναι η ένδειξη για το που βρίσκεται το εκάστοτε στοιχειοσύνολο της πρώτης στο σύνολο της δεύτερης παραμέτρου. Εφόσον υπάρχει εμφανίζεται η θέση του εκάστοτε στοιχειοσυνόλου, εναλλακτικά εμφανίζεται η ένδειξη NA. Για να γίνει ευκολότερη η κατανόηση έστω το σύνολο των συχνών στοιχειοσυνόλων A το οποίο περιέχει 10 συχνά στοιχειοσύνολα και ένα δεύτερο σύνολο που περιέχει επίσης 10 συχνά στοιχειοσύνολα. Εισάγοντας στην συνάρτηση match τα σύνολα A και B ως την πρώτη και την δεύτερη παράμετρο της συνάρτησης match θα προκύψει ένας integer vector με 10 τιμές ο οποίος θα περιέχει τις θέσεις των 10 στοιχειοσυνόλων του συνόλου A στο σύνολο B εάν υπάρχουν τα συγκεκριμένα στοιχειοσύνολα του A στο B ή NA εφόσον δεν υπάρχουν. Για τους τρεις αλγορίθμους των παραδειγμάτων τα σύνολα μπορούν να εναλλάσονται ως παράμετροι της συνάρτησης match και να σημειώνονται τα αποτελέσματα.

Σύγκριση των παραγόμενων αποτελεσμάτων για τους τρεις αλγόριθμους(αν οι συνδυαστικοί κανόνες τους οποίους παράγει η ruleInduction για κάθε αλγόριθμο είναι οι ίδιοι με αυτούς των υπολοίπων δύο):

Η προηγούμενη συνάρτηση match μπορεί να χρησιμοποιηθεί και για την σύγκριση των παραγόμενων συνδυαστικών κανόνων με τον ίδιο τρόπο όπως τα συχνά στοιχειοσύνολα με την αποθήκευση των συνόλων των παραγόμενων συνδυαστικών κανόνων ανα αλγόριθμο και την σύγκριση των παραπάνω συνόλων.

Εναλλακτικά προς επιβεβαίωση των παραγόμενων αποτελεσμάτων μπορεί κανείς με βάση τα αποτελέσματα που προκύπτουν από την συνάρτηση match να συγκρίνει, πέρα από τα στοιχειοσύνολα και τους συνδυαστικούς κανόνες, τις τιμές υποστήριξης και εμπιστοσύνης ανα περίπτωση

Κεφάλαιο 6ο: Μετρήσεις Επιδόσεων και Συμπεράσματα

Παρακάτω αναφέρονται οι συγκρίσεις των αλγορίθμων για τα δύο σύνολα δεδομένων Adult και City_isp_daily_speeds τα οποία περιέχουν ένα, πέντε και δέκα εκατομμύρια συναλλαγές. Για συντομία θα αναφέρεται το καθένα από αυτά με το όνομά του συνοδευόμενο από τον αριθμό των συναλλαγών. Ειδικότερα για το σύνολο δεδομένων Adult συναντάται κι ένα δεύτερο παράδειγμα στο οποίο υπάρχει μια διαφοροποίηση η οποία αναλύεται περαιτέρω παρακάτω πριν την αρχή του παραδείγματος αυτού.

Αναλυτική αποτύπωση της σύγκρισης των αλγορίθμων

Στο σύνολο δεδομένων Adult για την παραγωγή των συνδυαστικών κανόνων επιλέγεται η τιμή εμπιστοσύνης 0.5, η οποία διαφέρει από την τιμή εμπιστοσύνης του δευτέρου συνόλου δεδομένων, ωστόσο με την επιλογή της συγκεκριμένης τιμής στις εκτελέσεις των αλγορίθμων παράγονται συνδυαστικοί κανόνες κάτι το οποίο θα πραγματοποιούνταν τμηματικά εάν είχε επιλεχθεί η ίδια τιμή εμπιστοσύνης με το δεύτερο σύνολο δεδομένων.

- **Πρώτο παράδειγμα για το σύνολο δεδομένων Adult:**

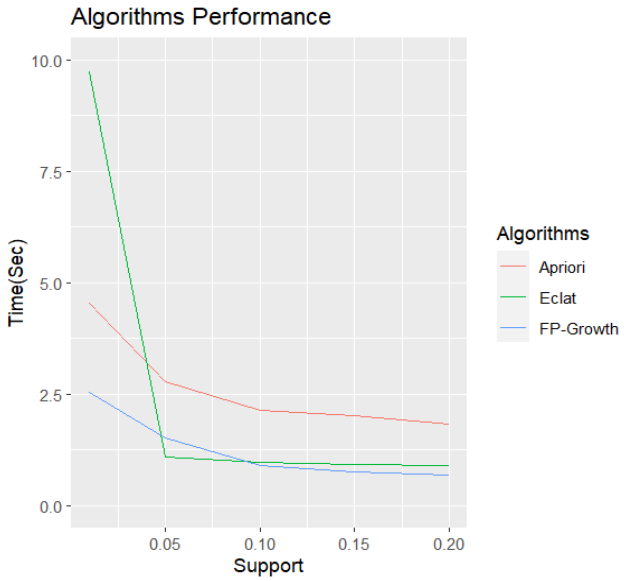
6.1 Δεδομένα: Adult 1M

Αρχικά για το σύνολο δεδομένων Adult1m, παρατηρούνται οι παρακάτω χρόνοι εκτέλεσης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.1.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Time(Sec)	Algorithms
0.01	4.53	Apriori
0.01	9.74	Eclat
0.01	2.54	FP-Growth
0.05	2.78	Apriori
0.05	1.09	Eclat
0.05	1.52	FP-Growth
0.1	2.13	Apriori
0.1	0.961	Eclat
0.1	0.902	FP-Growth
0.15	2.01	Apriori
0.15	0.929	Eclat
0.15	0.749	FP-Growth
0.2	1.83	Apriori
0.2	0.898	Eclat
0.2	0.686	FP-Growth

Πίνακας 6.1: Πίνακας με τους χρόνους εκτέλεσης



Σχήμα 6.1: Γραφική αναπαράσταση των χρόνων εκτέλεσης

Σύμφωνα με το παραπάνω γράφημα ο αλγόριθμος που καταγράφει τους μικρότερους χρόνους εκτέλεσης για τις περισσότερες τιμές υποστήριξης, είναι ο αλγόριθμος FP-Growth. Όπως γίνεται εύκολα αντιληπτό ο αλγόριθμος Eclat για τιμή υποστήριξης ίση με 0.05 καταγράφει τον μικρότερο χρόνο εκτέλεσης και καταλήγει στην συνέχεια να βρίσκεται στη δεύτερη θέση πίσω από τον FP-Growth. Ο αλγόριθμος Apriori σημειώνει τους μεγαλύτερους χρόνους εκτέλεσης εκτός από τον χρόνο εκτέλεσης που προκύπτει για τιμή υποστήριξης ίση με 0.01.

Για την παραγωγή των συνδυαστικών κανόνων η εκτέλεση της συνάρτησης ruleInduction καταγράφει κοινούς χρόνους εκτέλεσης και κοινή κατανάλωση μνήμης και για τους τρεις αλγορίθμους (Apriori,Eclat,FP-Growth), καθώς δέχεται ως είσοδο τα ίδια συχνά στοιχειοσύνολα.

Support	Time(ms)	Memory(MB)
0.01	21.4	8
0.05	9.59	0.4
0.1	5.89	0.2
0.15	5.74	0.1
0.2	8.61	0.1

Πίνακας 6.2: Πίνακας με τους χρόνους εκτέλεσης και την κατανάλωση της μνήμης

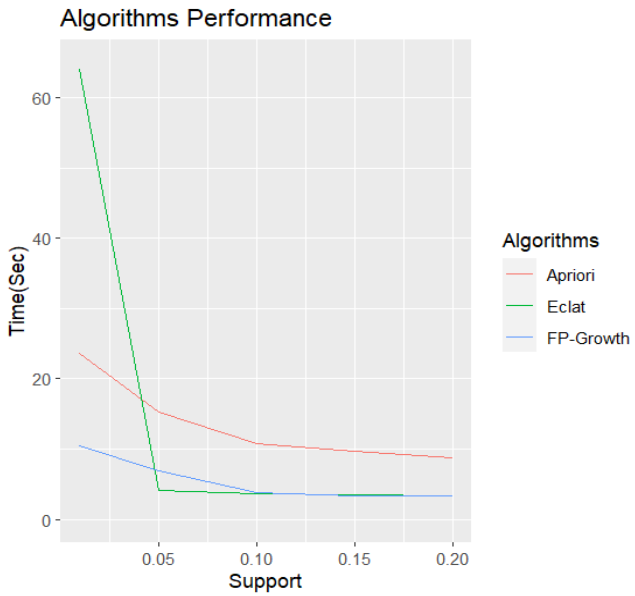
6.2 Δεδομένα: Adult 5M

Στη συνέχεια για το σύνολο δεδομένων Adult5m, παρατηρούνται οι παρακάτω χρόνοι εκτέλεσης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.2.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Time(Sec)	Algorithms
0.01	23.6	Apriori
0.01	64	Eclat
0.01	10.4	FP-Growth
0.05	15.2	Apriori
0.05	4.07	Eclat
0.05	6.96	FP-Growth
0.1	10.7	Apriori
0.1	3.6	Eclat
0.1	3.85	FP-Growth
0.15	9.7	Apriori
0.15	3.49	Eclat
0.15	3.38	FP-Growth
0.2	8.74	Apriori
0.2	3.33	Eclat
0.2	3.37	FP-Growth

Πίνακας 6.3: Πίνακας με τους χρόνους εκτέλεσης.



Σχήμα 6.2: Γραφική αναπαράσταση των χρόνων εκτέλεσης.

Με βάση το παραπάνω γράφημα παρατηρείται πως οι αλγόριθμοι Eclat και FP-Growth καταγράφουν ιδιαίτερα κοντινούς χρόνους εκτέλεσης. Υπάρχει μια εναλλαγή μεταξύ των δύο για το ποιός σημειώνει τον μικρότερο χρόνο εκτέλεσης όσο αυξάνεται η τιμή υποστήριξης ενώ αυτός που καταλήγει να καταγράφει τον μικρότερο χρόνο εκτέλεσης είναι ο αλγόριθμος Eclat. Για τιμές υποστήριξης από 0.05 έως 0.2 ο αλγόριθμος Apriori καταλήγει να έχει τους μεγαλύτερους χρόνους εκτέλεσης.

Για την παραγωγή των συνδυαστικών κανόνων η εκτέλεση της συνάρτησης ruleInduction καταγράφει κοινούς χρόνους εκτέλεσης και κοινή κατανάλωση μνήμης και για τους τρεις αλγορίθμους (Apriori,Eclat,FP-Growth), καθώς δέχεται ως είσοδο τα ίδια συχνά στοιχειοσύνολα.

Support	Time(ms)	Memory(MB)
0.01	20.3	7.5
0.05	6.57	0.4
0.1	5.88	0.2
0.15	6.23	0.1
0.2	5.96	0.1

Πίνακας 6.4: Πίνακας με τους χρόνους εκτέλεσης και την κατανάλωση της μνήμης

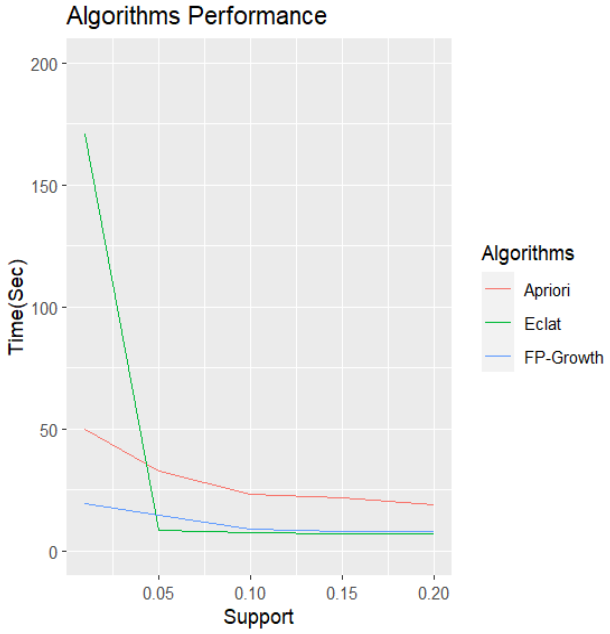
6.3 Δεδομένα: Adult 10M

Τέλος για το σύνολο δεδομένων Adult10m, παρατηρούνται οι παρακάτω χρόνοι εκτέλεσης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.3.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Time(Sec)	Algorithms
0.01	49.9	Apriori
0.01	171	Eclat
0.01	19.3	FP-Growth
0.05	32.9	Apriori
0.05	8.39	Eclat
0.05	14.8	FP-Growth
0.1	23.3	Apriori
0.1	7.42	Eclat
0.1	8.77	FP-Growth
0.15	21.7	Apriori
0.15	7.13	Eclat
0.15	8.06	FP-Growth
0.2	18.8	Apriori
0.2	6.87	Eclat
0.2	8.12	FP-Growth

Πίνακας 6.5: Πίνακας με τους χρόνους εκτέλεσης.



Σχήμα 6.3: Γραφική αναπαράσταση των χρόνων εκτέλεσης

Με βάση το παραπάνω γράφημα για το σύνολο δεδομένων Adult10m, πέρα από την τιμή υποστήριξης 0.01 ο αλγόριθμος Eclat σημειώνει τους μικρότερους χρόνους εκτέλεσης έναντι των υπολοίπων δύο αλγορίθμων.

Για την παραγωγή των συνδυαστικών κανόνων η εκτέλεση της συνάρτησης ruleInduction καταγράφει κοινούς χρόνους εκτέλεσης και κοινή κατανάλωση μνήμης και για τους τρεις αλγορίθμους (Apriori,Eclat,FP-Growth), καθώς δέχεται ως είσοδο τα ίδια συχνά στοιχειοσύνολα.

Support	Time(ms)	Memory(MB)
0.01	19.3	7.6
0.05	6.17	0.4
0.1	5.66	0.1
0.15	6.04	0.1
0.2	5.86	0.1

Πίνακας 6.6: Πίνακας με τους χρόνους εκτέλεσης και την κατανάλωση της μνήμης

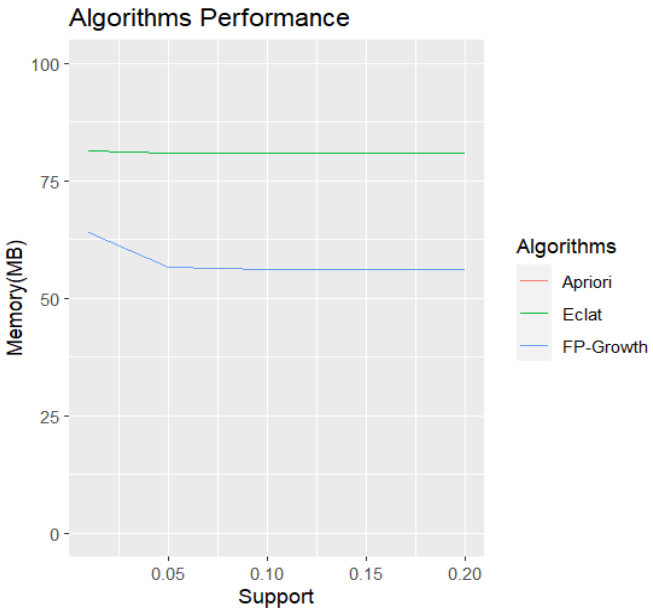
6.4 Δεδομένα: Adult 1M

Αντίστοιχα για το σύνολο δεδομένων Adult1m, παρατηρείται η κατανάλωση της μνήμης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.4.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Memory(MB)	Algorithms
0.01	81.3	Apriori
0.01	81.3	Eclat
0.01	64	FP-Growth
0.05	80.8	Apriori
0.05	80.8	Eclat
0.05	56.6	FP-Growth
0.1	80.8	Apriori
0.1	80.8	Eclat
0.1	56.1	FP-Growth
0.15	80.8	Apriori
0.15	80.8	Eclat
0.15	56	FP-Growth
0.2	80.8	Apriori
0.2	80.8	Eclat
0.2	56	FP-Growth

Πίνακας 6.7: Πίνακας με την κατανάλωση της μνήμης



Σχήμα 6.4: Γραφική αναπαράσταση της κατανάλωσης της μνήμης

Σύμφωνα με το προηγούμενο γράφημα, ο αλγόριθμος που σημειώνει την μικρότερη κατανάλωση είναι ο FP-Growth. Παρόλο που φαίνεται πως υπάρχουν μόνο δύο αλγόριθμοι, κάτι τέτοιο δεν ισχύει καθώς οί αλγόριθμοι Apriori και Eclat σημειώνουν παρόμοιες τιμές κατανάλωσης μνήμης, με αποτέλεσμα ο αλγόριθμος Eclat να επικαλύπτει τον αλγόριθμο Apriori.

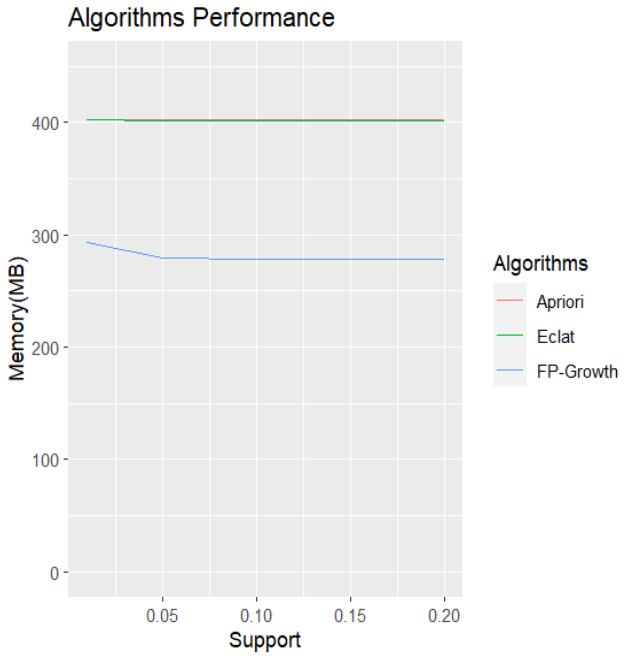
6.5 Δεδομένα: Adult 5M

Στη συνέχεια για το σύνολο δεδομένων Adult5m, παρατηρείται η κατανάλωση της μνήμης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.5.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Memory(MB)	Algorithms
0.01	402.6	Apriori
0.01	403	Eclat
0.01	293.1	FP-Growth
0.05	402.1	Apriori
0.05	402	Eclat
0.05	279.1	FP-Growth
0.1	402.1	Apriori
0.1	402	Eclat
0.1	278.6	FP-Growth
0.15	402.1	Apriori
0.15	402	Eclat
0.15	278.5	FP-Growth
0.2	402.1	Apriori
0.2	402	Eclat
0.2	278.4	FP-Growth

Πίνακας 6.8: Πίνακας με την κατανάλωση της μνήμης



Σχήμα 6.5: Γραφική αναπαράσταση της κατανάλωσης της μνήμης

Με βάση το προηγούμενο γράφημα και σε αυτήν την περίπτωση ο αλγόριθμος FP-Growth καταναλώνει την μικρότερη μνήμη. Οι κατανάλωση μνήμης των υπολοίπων δύο αλγορίθμων κινείται στα ίδια επίπεδα.

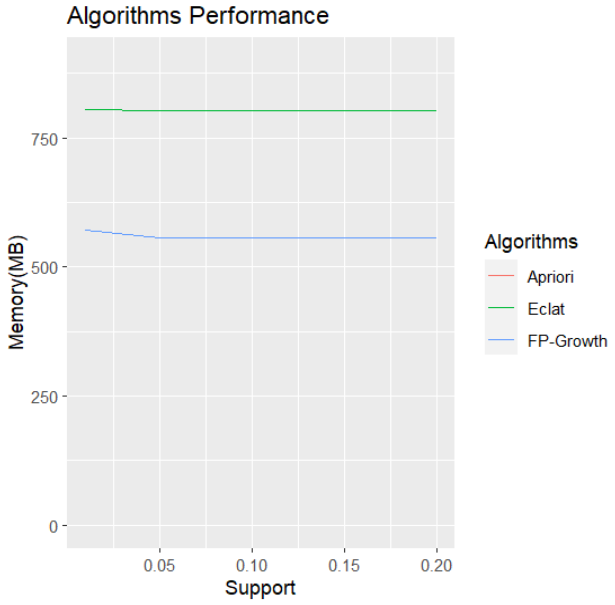
6.6 Δεδομένα: Adult 10M

Τέλος, για το σύνολο δεδομένων Adult10m, παρατηρείται η κατανάλωση της μνήμης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.6.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Memory(MB)	Algorithms
0.01	804.3	Apriori
0.01	804.2	Eclat
0.01	571.2	FP-Growth
0.05	803.8	Apriori
0.05	803.7	Eclat
0.05	557.2	FP-Growth
0.1	803.8	Apriori
0.1	803.7	Eclat
0.1	556.6	FP-Growth
0.15	803.8	Apriori
0.15	803.7	Eclat
0.15	556.5	FP-Growth
0.2	803.8	Apriori
0.2	803.7	Eclat
0.2	556.5	FP-Growth

Πίνακας 6.9: Πίνακας με την κατανάλωση της μνήμης



Σχήμα 6.6: Γραφική αναπαράσταση της κατανάλωσης της μνήμης

Για το σύνολο δεδομένων Adult10m ο αλγόριθμος FP-Growth καταναλώνει την μικρότερη μνήμη έναντι των υπολοίπων δύο αλγορίθμων οι οποίοι καταναλώνουν με μικρή διαφορά την ίδια μνήμη.

- **Δεύτερο παράδειγμα για το σύνολο δεδομένων Adult:**

Στο συγκεκριμένο παράδειγμα αντί για την τιμή υποστήριξης η οποία παραμένει σταθερή και ίση με 0.1, εναλλάσεται η τιμή εμπιστοσύνης και λαμβάνει τις τιμές στο διάστημα από 0.2 έως 0.5. Ο λόγος που επιλέγονται οι συγκεκριμένες χαμηλές τιμές εμπιστοσύνης είναι οτι για τιμές εμπιστοσύνης 0.6 και πάνω δεν παράγονται συνδυαστικοί κανόνες. Μέσα από αυτό το παράδειγμα παρατηρείται εάν με την εναλλαγή της τιμής υποστήριξης υπάρχει κάποια σημαντική διαφοροποίηση για τους τρεις αλγορίθμους.

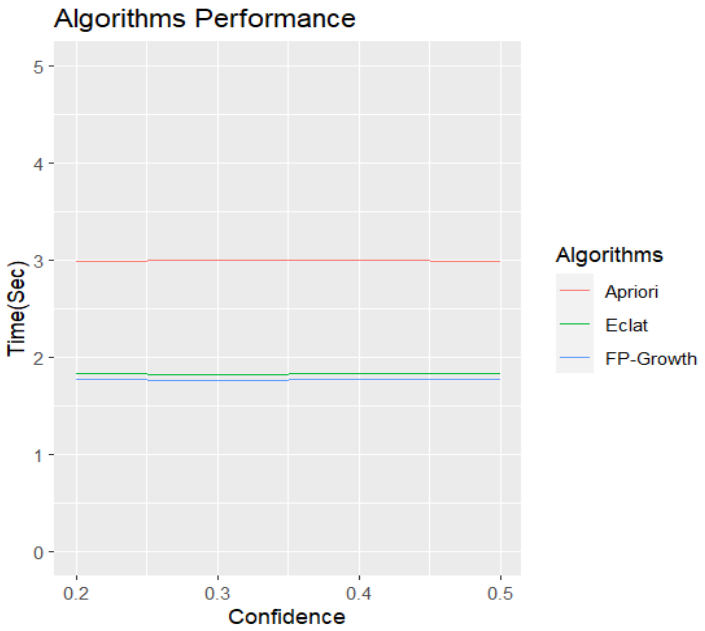
6.7 Δεδομένα: Adult 1M

6.7.1 Χρόνοι Εκτέλεσης

Οι χρόνοι εκτέλεσης για το σύνολο δεδομένων Adult1m:

Confidence	Time(Sec)	Algorithms
0.2	2.989	Apriori
0.2	1.833	Eclat
0.2	1.767	FP-Growth
0.3	3.001	Apriori
0.3	1.82	Eclat
0.3	1.762	FP-Growth
0.4	2.997	Apriori
0.4	1.833	Eclat
0.4	1.768	FP-Growth
0.5	2.988	Apriori
0.5	1.832	Eclat
0.5	1.768	FP-Growth

Πίνακας 6.10: Πίνακας με τους χρόνους εκτέλεσης



Σχήμα 6.7: Γραφική αναπαράσταση των χρόνων εκτέλεσης

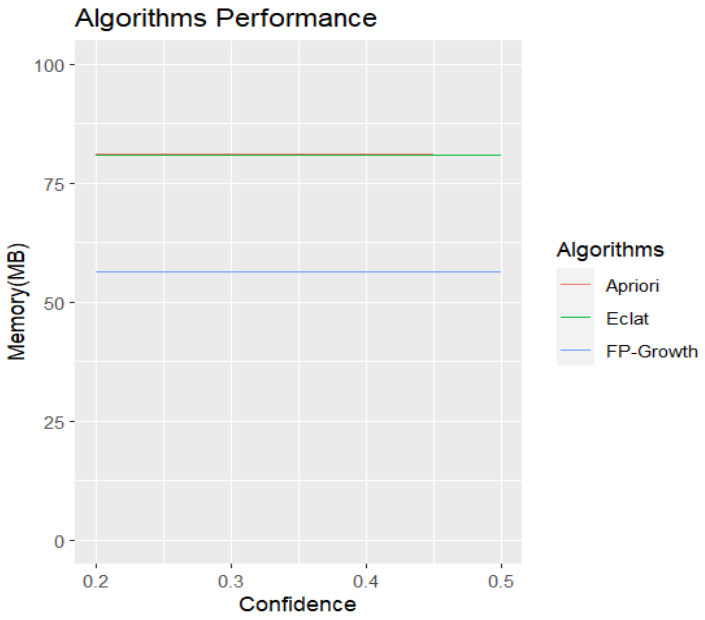
Παρατηρώντας το παραπάνω γράφημα αυτό που προκύπτει είναι ότι δεν καταγράφεται κάποια ιδιαίτερη διαφοροποίηση καθώς αλλάζει η τιμή εμπιστοσύνης. Ο αλγόριθμος FP-Growth σημειώνει τους μικρότερους χρόνους εκτέλεσης, έχοντας από κοντά τον αλγόριθμο Eclat. Ο αλγόριθμος Apriori σημειώνει τους μεγαλύτερους χρόνους εκτέλεσης.

6.7.2 Κατανάλωση της μνήμης

Η κατανάλωση της μνήμης για το σύνολο δεδομένων Adult1m:

Confidence	Memory(MB)	Algorithms
0.2	81.1	Apriori
0.2	81	Eclat
0.2	56.4	FP-Growth
0.3	81.1	Apriori
0.3	81	Eclat
0.3	56.4	FP-Growth
0.4	81.1	Apriori
0.4	81	Eclat
0.4	56.4	FP-Growth
0.5	81	Apriori
0.5	81	Eclat
0.5	56.3	FP-Growth

Πίνακας 6.11: Πίνακας με την κατανάλωση της μνήμης



Σχήμα 6.8: Γραφική αναπαράσταση της κατανάλωσης της μνήμης

Ως προς την κατανάλωση μνήμης επίσης παρατηρείται πως δεν προκύπτει κάποια σημαντική διαφοροποίηση για όσο διάστημα αλλάζει η τιμή εμπιστοσύνης. Ο αλγόριθμος με την μικρότερη κατανάλωση μνήμης είναι ο αλγόριθμος FP-Growth.

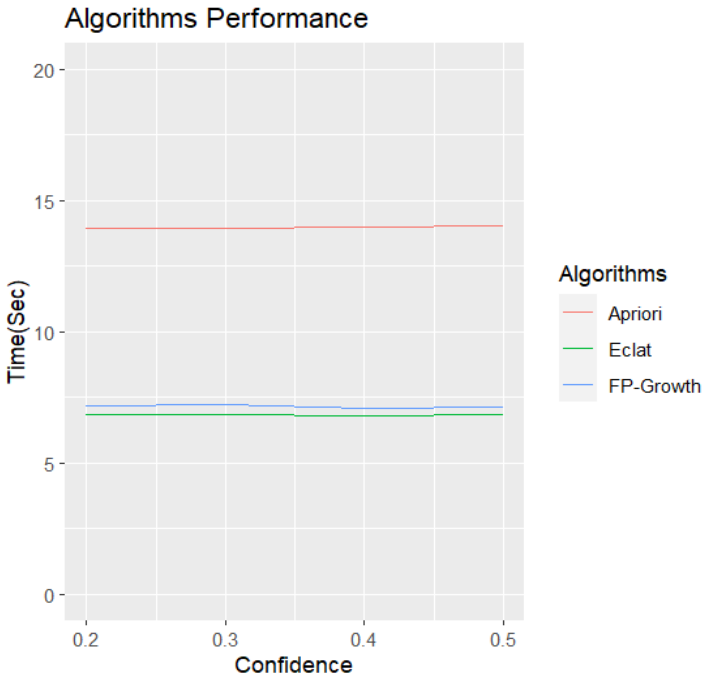
6.8 Δεδομένα: Adult 5M

6.8.1 Χρόνοι Εκτέλεσης

Οι χρόνοι εκτέλεσης για το σύνολο δεδομένων Adult5m:

Confidence	Time(Sec)	Algorithms
0.2	13.95	Apriori
0.2	6.82	Eclat
0.2	7.15	FP-Growth
0.3	13.94	Apriori
0.3	6.82	Eclat
0.3	7.24	FP-Growth
0.4	13.98	Apriori
0.4	6.81	Eclat
0.4	7.1	FP-Growth
0.5	14.01	Apriori
0.5	6.82	Eclat
0.5	7.13	FP-Growth

Πίνακας 6.12: Πίνακας με τους χρόνους εκτέλεσης



Σχήμα 6.9: Γραφική αναπαράσταση των χρόνων εκτέλεσης

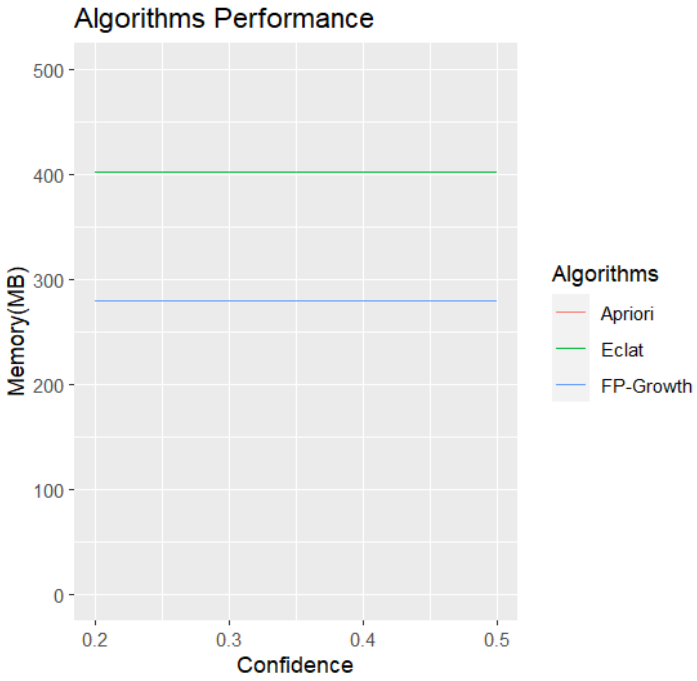
Σύμφωνα με το προηγούμενο γράφημα για το σύνολο δεδομένων Adult5m, δεν προκύπτει κάποια ιδιαίτερη διαφοροποίηση (μεταβολή του χρόνου εκτέλεσης με αύξησή ή μείωσή του) για τις αυξανόμενες τιμές εμπιστοσύνης. Ο αλγόριθμος που καταγράφει τους μικρότερους χρόνους εκτέλεσης, είναι ο αλγόριθμος Eclat. Αυτή είναι η μοναδική διαφορά σε σχέση με το σύνολο δεδομένων Adult1m.

6.8.2 Κατανάλωση της μνήμης

Η κατανάλωση της μνήμης για το σύνολο δεδομένων Adult5m:

Confidence	Memory(MB)	Algorithms
0.2	402.4	Apriori
0.2	402.3	Eclat
0.2	278.9	FP-Growth
0.3	402.4	Apriori
0.3	402.3	Eclat
0.3	278.9	FP-Growth
0.4	402.4	Apriori
0.4	402.3	Eclat
0.4	278.9	FP-Growth
0.5	402.3	Apriori
0.5	402.2	Eclat
0.5	278.8	FP-Growth

Πίνακας 6.13: Πίνακας με την κατανάλωση της μνήμης



Σχήμα 6.10: Γραφική αναπαράσταση της κατανάλωσης της μνήμης

Με βάση το παραπάνω γράφημα, ο αλγόριθμος με την μικρότερη κατανάλωση μνήμης είναι ο αλγόριθμος FP-Growth.

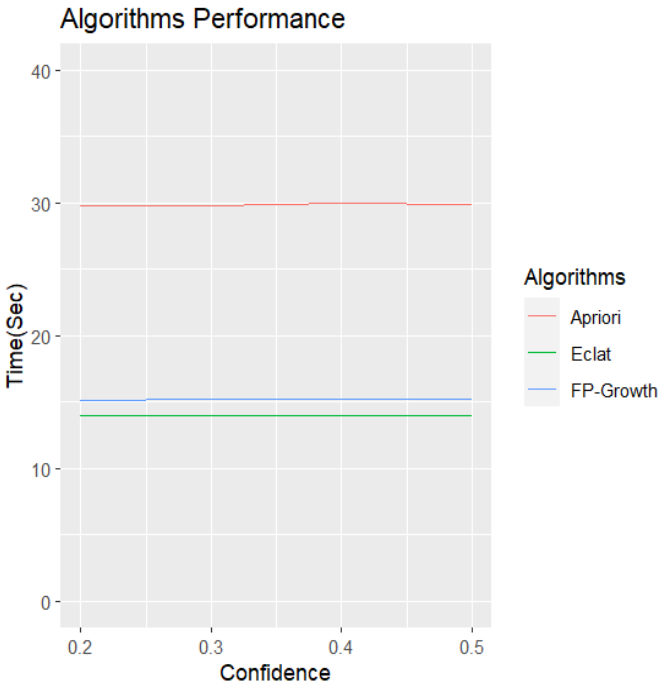
6.9 Δεδομένα: Adult 10M

6.9.1 Χρόνοι Εκτέλεσης

Οι χρόνοι εκτέλεσης για το σύνολο δεδομένων Adult10m:

Confidence	Time(Sec)	Algorithms
0.2	29.81	Apriori
0.2	13.96	Eclat
0.2	15.08	FP-Growth
0.3	29.82	Apriori
0.3	13.93	Eclat
0.3	15.22	FP-Growth
0.4	29.93	Apriori
0.4	14	Eclat
0.4	15.25	FP-Growth
0.5	29.88	Apriori
0.5	13.96	Eclat
0.5	15.22	FP-Growth

Πίνακας 6.14: Πίνακας με τους χρόνους εκτέλεσης



Σχήμα 6.11: Γραφική αναπαράσταση των χρόνων εκτέλεσης

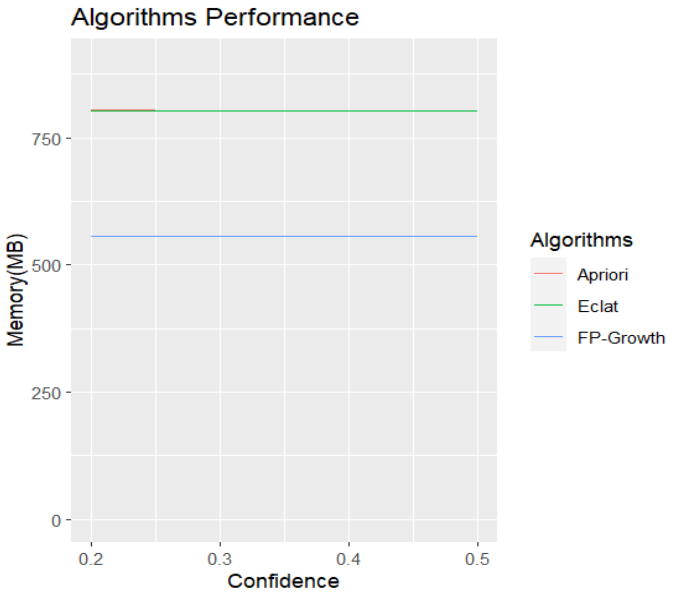
Σύμφωνα με το προηγούμενο γράφημα δεν παρατηρείται κάποια ιδιαίτερη μεταβολή των χρόνων εκτέλεσης. Όπως σημειώθηκε και πιο πάνω για το σύνολο δεδομένων Adult5m, εξακολουθεί ο αλγόριθμος Eclat να καταγράφει τους μικρότερους χρόνους εκτέλεσης.

6.9.2 Κατανάλωση της μνήμης

Η κατανάλωση της μνήμης για το σύνολο δεδομένων Adult10m:

Confidence	Memory(MB)	Algorithms
0.2	804.1	Apriori
0.2	804	Eclat
0.2	556.8	FP-Growth
0.3	804	Apriori
0.3	804	Eclat
0.3	556.9	FP-Growth
0.4	804	Apriori
0.4	804	Eclat
0.4	556.9	FP-Growth
0.5	804	Apriori
0.5	803.9	Eclat
0.5	556.9	FP-Growth

Πίνακας 6.15: Πίνακας με την κατανάλωση της μνήμης



Εικόνα 6.12: Γραφική αναπαράσταση της κατανάλωσης της μνήμης

Τέλος και στην κατανάλωση της μνήμης δεν παρατηρείται κάποια έντονη αλλαγή για τους τρεις αλγορίθμους. Αυτός με την μικρότερη κατανάλωση είναι ο αλγόριθμος FP-Growth.

Σε αυτό το σημείο έχει ολοκληρωθεί η διαδικασία σύγκρισης των αλγορίθμων για το σύνολο δεδομένων Adult. Η σύγκριση των τριών αλγορίθμων πραγματοποιείται εφεξής στο σύνολο δεδομένων City_isp_daily_speeds. Για την παραγωγή των συνδυαστικών κανόνων επιλέγεται η τιμή εμπιστοσύνης 0.6.

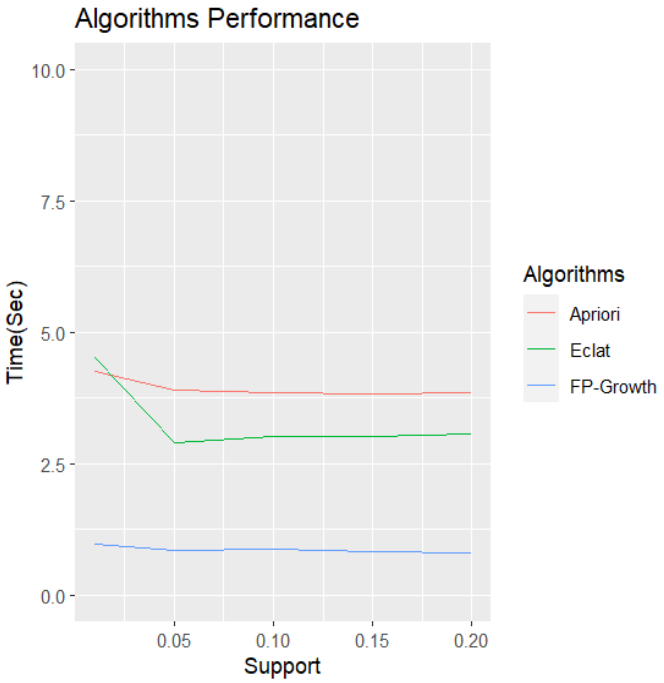
6.10 Δεδομένα: City_isp_daily_speeds 1M

Αρχικά για το σύνολο δεδομένων City_isp_daily_speeds1m, παρατηρούνται οι παρακάτω χρόνοι εκτέλεσης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.10.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Time(Sec)	Algorithms
0.01	4.26	Apriori
0.01	4.51	Eclat
0.01	0.957	FP-Growth
0.05	3.89	Apriori
0.05	2.89	Eclat
0.05	0.853	FP-Growth
0.1	3.85	Apriori
0.1	3.02	Eclat
0.1	0.867	FP-Growth
0.15	3.82	Apriori
0.15	3.02	Eclat
0.15	0.829	FP-Growth
0.2	3.85	Apriori
0.2	3.07	Eclat
0.2	0.794	FP-Growth

Πίνακας 6.16: Πίνακας με τους χρόνους εκτέλεσης



Σχήμα 6.13: Γραφική αναπαράσταση των χρόνων εκτέλεσης

Με βάση το παραπάνω γράφημα ο αλγόριθμος FP-Growth καταγράφει τους μικρότερους χρόνους εκτέλεσης. Ο αλγόριθμος Eclat ακολουθεί εκτός από τον χρόνο που προκύπτει για τιμή υποστήριξης ίση με 0.01.

Για την παραγωγή των συνδυαστικών κανόνων η εκτέλεση της συνάρτησης ruleInduction καταγράφει κοινούς χρόνους εκτέλεσης και κοινή κατανάλωση μνήμης και για τους τρεις αλγορίθμους (Apriori,Eclat,FP-Growth), καθώς δέχεται ως είσοδο τα ίδια συχνά στοιχειοσύνολα.

Support	Time(ms)	Memory(MB)
0.01	14.2	4.3
0.05	12.7	3.4
0.1	12.8	3.3
0.15	12.7	3.3
0.2	12.7	3.3

Πίνακας 6.17: Πίνακας με τους χρόνους εκτέλεσης και την κατανάλωση της μνήμης

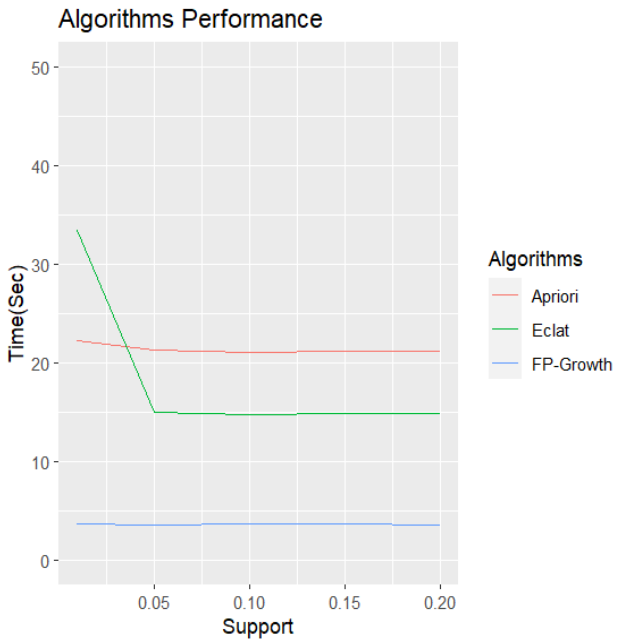
6.11 Δεδομένα: City_isp_daily_speeds 5M

Στη συνέχεια για το σύνολο δεδομένων City_isp_daily_speeds5m, παρατηρούνται οι παρακάτω χρόνοι εκτέλεσης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.11.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Time(Sec)	Algorithms
0.01	22.2	Apriori
0.01	33.4	Eclat
0.01	3.69	FP-Growth
0.05	21.3	Apriori
0.05	15	Eclat
0.05	3.52	FP-Growth
0.1	21	Apriori
0.1	14.7	Eclat
0.1	3.59	FP-Growth
0.15	21.1	Apriori
0.15	14.8	Eclat
0.15	3.68	FP-Growth
0.2	21.1	Apriori
0.2	14.8	Eclat
0.2	3.49	FP-Growth

Πίνακας 6.18: Πίνακας με τους χρόνους εκτέλεσης



Σχήμα 6.14: Γραφική αναπαράσταση των χρόνων εκτέλεσης

Παρατηρείται πως ο αλγόριθμος FP-Growth σημειώνει τους μικρότερους χρόνους εκτέλεσης έναντι των υπολοίπων δύο αλγορίθμων. Ο αλγόριθμος Eclat με μοναδική εξαίρεση για τιμή υποστήριξης ίση με 0.01, ακολουθεί τον αλγόριθμο FP-Growth.

Για την παραγωγή των συνδυαστικών κανόνων η εκτέλεση της συνάρτησης ruleInduction καταγράφει κοινούς χρόνους εκτέλεσης και κοινή κατανάλωση μνήμης και για τους τρεις αλγορίθμους (Apriori,Eclat,FP-Growth), καθώς δέχεται ως είσοδο τα ίδια συχνά στοιχειοσύνολα.

Support	Time(ms)	Memory(MB)
0.01	26.5	4.3
0.05	24.8	3.3
0.1	25.2	3.3
0.15	27	3.3
0.2	26.6	3.3

Πίνακας 6.19: Πίνακας με τους χρόνους εκτέλεσης και την κατανάλωση της μνήμης

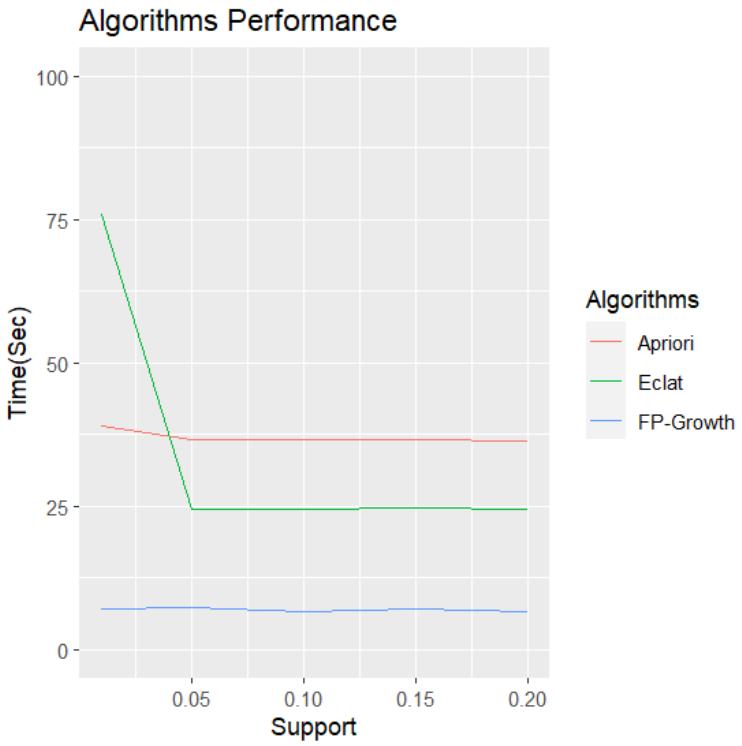
6.12 Δεδομένα: City_isp_daily_speeds 10M

Τέλος για το σύνολο δεδομένων City_isp_daily_speeds10m, παρατηρούνται οι παρακάτω χρόνοι εκτέλεσης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.12.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Time(Sec)	Algorithms
0.01	39	Apriori
0.01	76	Eclat
0.01	7.16	FP-Growth
0.05	36.6	Apriori
0.05	24.5	Eclat
0.05	7.38	FP-Growth
0.1	36.5	Apriori
0.1	24.5	Eclat
0.1	6.61	FP-Growth
0.15	36.6	Apriori
0.15	24.7	Eclat
0.15	6.99	FP-Growth
0.2	36.3	Apriori
0.2	24.5	Eclat
0.2	6.55	FP-Growth

Πίνακας 6.20: Πίνακας με τους χρόνους εκτέλεσης



Σχήμα 6.15: Γραφική αναπαράσταση των χρόνων εκτέλεσης

Με βάση το προηγούμενο γράφημα ο αλγόριθμος με τους μικρότερους χρόνους εκτέλεσης είναι ο αλγόριθμος FP-Growth, όπως χαρακτηριστικά φαίνεται στο συγκεκριμένο γράφημα.

Για την παραγωγή των συνδυαστικών κανόνων η εκτέλεση της συνάρτησης ruleInduction καταγράφει κοινούς χρόνους εκτέλεσης και κοινή κατανάλωση μνήμης και για τους τρεις αλγορίθμους (Apriori,Eclat,FP-Growth), καθώς δέχεται ως είσοδο τα ίδια συχνά στοιχειοσύνολα.

Support	Time(ms)	Memory(MB)
0.01	28.8	4.3
0.05	26.1	3.4
0.1	24.4	3.3
0.15	24.3	3.3
0.2	26.4	3.3

Πίνακας 6.21: Πίνακας με τους χρόνους εκτέλεσης και την κατανάλωση της μνήμης

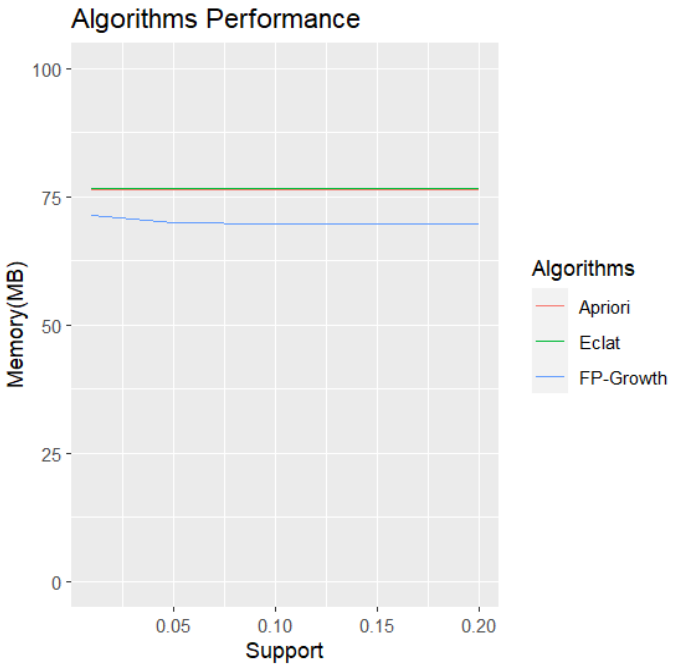
6.13 Δεδομένα: City_isp_daily_speeds 1M

Αντίστοιχα για το σύνολο δεδομένων City_isp_daily_speeds1m, παρατηρείται η κατανάλωση της μνήμης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.13.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Memory(MB)	Algorithms
0.01	76.4	Apriori
0.01	76.7	Eclat
0.01	71.4	FP-Growth
0.05	76.3	Apriori
0.05	76.7	Eclat
0.05	70	FP-Growth
0.1	76.3	Apriori
0.1	76.7	Eclat
0.1	69.8	FP-Growth
0.15	76.3	Apriori
0.15	76.7	Eclat
0.15	69.8	FP-Growth
0.2	76.3	Apriori
0.2	76.7	Eclat
0.2	69.8	FP-Growth

Πίνακας 6.22: Πίνακας με την κατανάλωση της μνήμης



Σχήμα 6.16: Γραφική αναπαράσταση της κατανάλωσης της μνήμης

Με βάση το παραπάνω γράφημα ο αλγόριθμος με την μικρότερη κατανάλωση μνήμης είναι ο αλγόριθμος FP-Growth. Οι δύο υπόλοιποι αλγόριθμοι σημειώνουν τιμές ιδιαίτερα κοντινές.

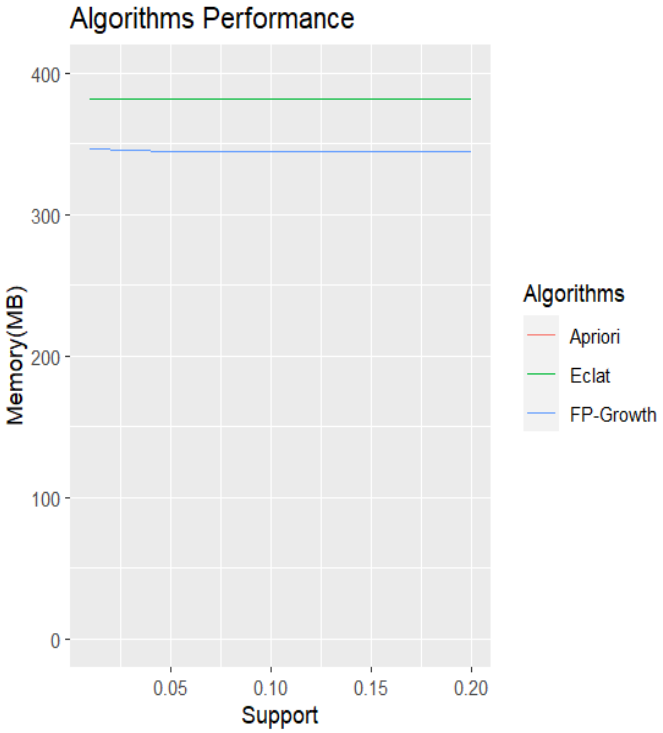
6.14 Δεδομένα: City_isp_daily_speeds 5M

Στη συνέχεια για το σύνολο δεδομένων City_isp_daily_speeds5m, παρατηρείται η κατανάλωση της μνήμης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.14.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Memory(MB)	Algorithms
0.01	381.5	Apriori
0.01	381.7	Eclat
0.01	346.1	FP-Growth
0.05	381.4	Apriori
0.05	381.6	Eclat
0.05	344.6	FP-Growth
0.1	381.4	Apriori
0.1	381.6	Eclat
0.1	344.5	FP-Growth
0.15	381.4	Apriori
0.15	381.6	Eclat
0.15	344.5	FP-Growth
0.2	381.4	Apriori
0.2	381.6	Eclat
0.2	344.5	FP-Growth

Πίνακας 6.23: Πίνακας με την κατανάλωση της μνήμης



Σχήμα 6.17: Γραφική αναπαράσταση της κατανάλωσης της μνήμης

Στη συγκεκριμένη περίπτωση επίσης ο αλγόριθμος FP-Growth καταγράφει την μικρότερη κατανάλωση έναντι των υπολοίπων δύο αλγορίθμων.

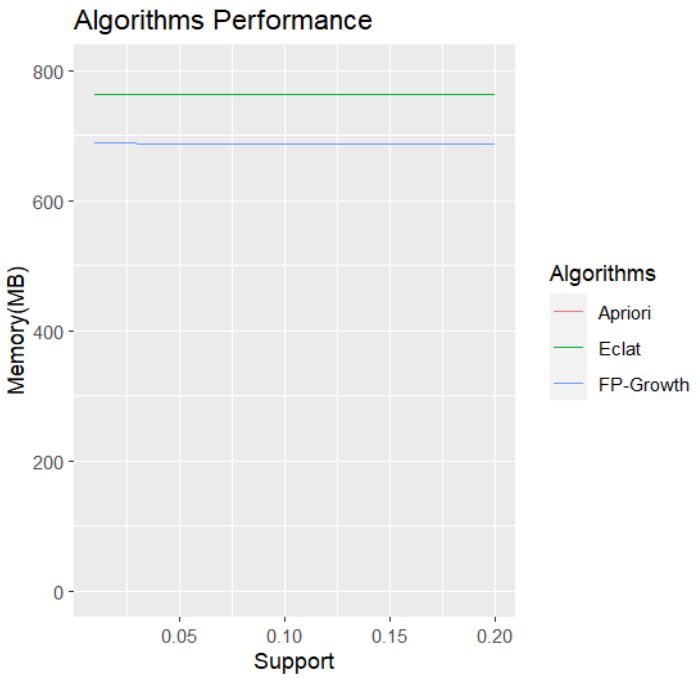
6.15 Δεδομένα: City_isp_daily_speeds 10M

Τέλος για το σύνολο δεδομένων City_isp_daily_speeds το οποίο περιέχει 10m, παρατηρείται η κατανάλωση της μνήμης των αλγορίθμων Apriori, Eclat και FP-Growth.

6.15.1 Δημιουργία Συχνών Στοιχειοσυνόλων:

Support	Memory(MB)	Algorithms
0.01	762.7	Apriori
0.01	763	Eclat
0.01	689.4	FP-Growth
0.05	762.7	Apriori
0.05	763	Eclat
0.05	687.9	FP-Growth
0.1	762.7	Apriori
0.1	763	Eclat
0.1	687.8	FP-Growth
0.15	762.7	Apriori
0.15	763	Eclat
0.15	687.8	FP-Growth
0.2	762.7	Apriori
0.2	763	Eclat
0.2	687.8	FP-Growth

Πίνακας 6.24: Πίνακας με την κατανάλωση της μνήμης



Σχήμα 6.18: Γραφική αναπαράσταση της κατανάλωσης της μνήμης

Στη συγκεκριμένη περίπτωση εξακολουθεί να ισχύει ότι ο αλγόριθμος FP-Growth καταγράφει την μικρότερη κατανάλωση μνήμης.

Κεφάλαιο 7ο: Επίλογος και Προτεινόμενες Επεκτάσεις

Ως προτεινόμενες επεκτάσεις μπορούν να θεωρηθούν ορισμένες αποδοτικότερες εκδόσεις, των τριών αλγορίθμων που εξετάστηκαν στην παρούσα εργασία, οι οποίες να υποστηρίζουν παράλληλη εκτέλεση των αλγορίθμων. Οι παραπάνω προτεινόμενες εκδόσεις θα μπορούσαν να συγκριθούν, με τους αλγοριθμους που περιγράφονται στο τρίτο κεφάλαιο της εργασίας αυτής, και να διαπιστωθεί αν και κατα πόσο είναι αποδοτικότερες οι εκδόσεις αυτές. Η σύγκριση να μην αφορά μόνο τον τρόπο με τον οποίο υλοποιούνται οι προτεινόμενες εκδόσεις σε σχέση με τις αρχικές εκδόσεις των αλγορίθμων αλλά και η επί του πρακτέου σύγκρισή τους με τα αντίστοιχα αποτελέσματα σε χρόνους εκτέλεσης και κατανάλωσης της μνήμης έναντι βασικών αλγορίθμων. Επίσης μία άλλη προτεινόμενη επέκταση συνδέεται με το incremental mining και το αν και σε ποιό βαθμό οι υλοποιήσεις των τριών αλγορίθμων στο πακέτο arules θα μπορούσαν να το εφαρμόσουν ή την προσθήκη αλγορίθμων που να το υποστηρίζουν.

Η παρούσα πτυχιακή εργασία συνιστά μία πλήρη, εμπειρικού τύπου, συγκριτική αξιολόγηση των τριών αλγορίθμων ARM (Apriori, Eclat, και FP-Growth) και, ως τέτοια, θεωρείται ότι θα αποβεί χρήσιμη στον ενδιαφερόμενο ερευνητή ο οποίος εστιάζει στη συγκεκριμένη περιοχής εξειδίκευσης (εξόρυξη πληροφορίας μέσω συνδυαστικών κανόνων). Αναλυτικότερα, παρουσιάστηκαν λεπτομερώς και σε βάθος οι αλγόριθμοι Apriori, Eclat και FP-Growth. Για κάθε έναν ξεχωριστά μελετήθηκε ο τρόπος υλοποίησής του μέσα από την επεξήγηση ψευδοκώδικα, η οποία συνοδεύεται και από ένα άμεσο παράδειγμα. Επίσης τονίστηκε τι ακριβώς ισχύει για τον κάθε αλγόριθμο στο περιβάλλον R/RStudio και πως αξιοποιείται το πακέτο arules [57] μέσω του οποίου πραγματοποιήθηκε η σύγκρισή των τριών στο περιβάλλον R/RStudio επάνω σε δύο σύνολα δεδομένων. Ακόμη, αναλύθηκε ο τρόπος με τον οποίο υλοποιήθηκαν οι συγκρίσεις και τα αποτελέσματα που προέκυψαν τόσο στο τεχνικό κομμάτι (χρόνος εκτέλεσης και κατανάλωση μνήμης) όσο και τα άμεσα αποτελέσματα (στοιχειοσύνολα ή συνδυαστικοί κανόνες) από την εκτέλεση των αλγορίθμων. Επίσης τονίστηκε ο τρόπος με τον οποίο μπορούν να συγκριθούν τόσο τα αποτελέσματα που σχετίζονται με το τεχνικό κομματι (πίνακες και γραφήματα) όσο και τα άμεσα αποτελέσματα (χρήση της συνάρτησης match). Μέσα από τα παραπάνω, αναδείχθηκε η υπεροχή του FP-Growth έναντι των άλλων δύο επί του συνόλου των τριών κριτηρίων συγκριτικής αξιολόγησης που έχουν τεθεί: (α) ταχύτητα επεξεργασίας, (β) δέσμευση κύριας μνήμης, και (γ) χρήση σε δυναμικά, βαθμωτά (incremental) περιβάλλοντα συλλογής δεδομένων, καθώς όπως τονίστηκε ο συγκεκριμένος αλγόριθμος υποστηρίζει βαθμωτές (incremental) ενημερώσεις του δέντρου FP (FP-Tree). Τέλος, μέσα από την μελέτη αντίστοιχων περιπτώσεων σύγκρισης των τριών αλγορίθμων, αυτό που προκύπτει είναι η συμφωνία των αποτελεσμάτων της πτυχιακής εργασίας με εκείνα σχετικών ερευνητικών δημοσιεύσεων της διεθνούς βιβλιογραφίας [68-69].

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] The R Project for Statistical Computing Getting Started https://www.r-project.org/ .

[2] Tibco What is R analytics? https://www.tibco.com/reference-center/what-is-r-analytics .

[3] Verykios, V., Kagklis, V., & Stavropoulos, E. (2015). *Η επιστήμη των δεδομένων μέσα από τη γλώσσα R* [Undergraduate textbook]. Kallipos, Open Academic Editions. https://hdl.handle.net/11419/2965 .

[4] Pang-Ning Tan, Michael Steinbach,Vipin Kumar, Introduction to Data Mining. Boston: Pearson Education, 2006, lSBN 0-321-42052.

[5] Michael R.Berthold, Christian Borgelt, Frank Hoppner, Frank Klawonn, Guide to Intelligent Data Analysis. Cham, Switzerland: Springer 2011, ISBN 978-3-030-45574-3.

[6] Mohammed J. Zaki, Wagner Meira Jr, Data Mining and Analysis, Fundamental Concepts and Algorithms. New York: Cambridge University Press, 2014, ISBN 978-0-521-76633-3.

[7] Gunjan Mehta, Deepa Sharma and Ekta Chauhan. Article: Application of Incremental Mining and Apriori Algorithm on Library Transactional Database. International Journal of Computer Applications 73(8):12-18, July 2013, https://doi.org/10.5120/12760-9336 .

[8] Rakesh Agrawal Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules," VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases September, 1994, Pages 487–499, Santiago, Chile, 1994 https://dl.acm.org/doi/10.5555/645920.672836 .

[9] Pramod S, O.P. Vyas, "Survey on Frequent Item set Mining Algorithms," International Journal of Computer Applications (0975 - 8887), Volume 1 – No. 15, Feb 2010, DOI:10.5120/316-484 .

[10] Rakesh Agrawal John C. Shafer, "Parallel Mining of Association Rules: Design, Implementation and Experience," IEEE Transactions on Knowledge and Data Engineering Volume 8, Issue 6, December 1996, pp 962–969 https://doi.org/10.1109/69.553164 .

[11] Charu C. Aggarwal, Mansurul A. Bhuiyan and Mohammad Al Hasan, Chapter 2 Frequent Pattern Mining Algorithms: A Survey. Switzerland: Springer International Publishing, 2014, DOI https://doi.org/10.1007/978-3-319-07821-2_2 .

[12] Anita Wasilewska Apriori Algorithm Lecture Notes https://www.academia.edu/23894291/APRIORI_Algorithm_Professor_Anita_Wasilewska_Lecture_Notes .

[13] Cristian Aflori, Mitica Craus, “Grid implementation of the Apriori algorithm,” Advances in Engineering Software, Volume 38, Issue 5, May 2007, Pages 295-300 https://doi.org/10.1016/j.advengsoft.2006.08.011 .

[14] Jayshree Jha and Leena Ragha, “Educational Data Mining using Improved Apriori Algorithm,” International Journal of Information and Computation Technology, ISSN 0974-2239, Volume 3, Number 5 (2013), pp. 411-418 https://www.ripublication.com/irph/ijict_spl/08_ijictv3n5spl.pdf .

[15] Jiao Yabing, "Research of an Improved Apriori Algorithm in Data Mining Association Rules," International Journal of Computer and Communication Engineering vol. 2, no. 1, pp. 25-27 , 2013 DOI: https://doi.org/10.7763/IJCCE.2013.V2.128 .

[16] Mohammed Al-Maolegi, Bassam Arkok, “ An Improved Apriori Algorithm For Association Rules,” International Journal on Natural Language Computing (IJNLC), Vol. 3, No.1, February 2014 DOI: https://doi.org/10.5121/ijnlc.2014.3103 .

[17] Xiuli Yuan, “An Improved Apriori Algorithm for Mining Association Rules” ADVANCES IN MATERIALS, MACHINERY, ELECTRONICS I: Proceedings of the International Conference on Advances in Materials, Machinery, Electronics (AMME 2017), 25–26 February 2017 Wuhan, China https://doi.org/10.1063/1.4977361 .

[18] C. Yadav, S. Wang and M. Kumar, "An approach to improve apriori algorithm based on association rule mining," 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, India, 2013, pp. 1-9, doi: https://doi.org/10.1109/ICCCNT.2013.6726678 .

[19] M. Hamdani Santoso, “Application of Association Rule Method Using Apriori Algorithm to Find Sales Patterns Case Study of Indomaret Tanjung Anom,” Brilliance: Research of Artificial Intelligence, Volume 1, Number 2, November 2021 pp: 54-66 https://doi.org/10.47709/briliance.v1i2.1228 .

[20] M.Kavitha, Ms.S.T.Tamil Selvi, “Comparative Study on Apriori Algorithm and Fp Growth Algorithm with Pros and Cons,” International Journal of Computer Science Trends and Technology (IJCS T) – Volume 4 Issue 4, Jul - Aug 2016 http://www.ijcstjournal.org/volume-4/issue-4/IJCST-V4I4P28.pdf .

[21] M. Ilayaraja and T. Meyyappan, "Mining medical data to identify frequent diseases using Apriori algorithm," 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, Salem, India, 2013, pp. 194-199, doi: https://doi.org/10.1109/ICPRIME.2013.6496471 .

[22] J. Du, X. Zhang, H. Zhang and L. Chen, "Research and improvement of Apriori algorithm," 2016 Sixth International Conference on Information Science and Technology (ICIST), Dalian, China, 2016, pp. 117-121, doi: https://doi.org/10.1109/ICIST.2016.7483396 .

[23] Y. Liu, "Study on Application of Apriori Algorithm in Data Mining," 2010 Second International Conference on Computer Modeling and Simulation, Sanya, China, 2010, pp. 111-114, doi: https://doi.org/10.1109/ICCMS.2010.398 .

[24] Markus Hegland, The Apriori Algorithm – a Tutorial. Mathematics and Computation in Imaging Science and Information Processing, pp. 209-262 (2007), World Scientific Publishing Co., Inc.1060 Main Street Suite 1B River Edge, NJ, United States (2007), ISBN:978-981-270-905-9, DOI: https://doi.org/10.1142/9789812709066_0006 .

[25] J. Dongre, G. L. Prajapati and S. V. Tokekar, "The role of Apriori algorithm for finding the association rules in Data mining," 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, 2014, pp. 657-660, https://doi.org/10.1109/ICICICT.2014.6781357 .

[26] Edastama, P., Bist, A., & Prambudi, A, "Implementation Of Data Mining On Glasses Sales Using The Apriori Algorithm," International Journal of Cyber and IT Service Management (IJCITSM), 1(2), 159-172, October 2021, https://doi.org/10.34306/ijcitsm.v1i1.46 .

[27] Zaki, Mohammed & Parthasarathy, Srinivasan & Ogihara, Mitsunori & li, Wei. (1997). New Algorithms for Fast Discovery of Association Rules.. 283-286.

[28] Goethals, Bart. (2003). Survey on Frequent Pattern Mining, http://adrem.uantwerpen.be/~goethals/software/survey.pdf .

[29] Christian Borgelt, Frequent Item set mining, WIREs Data Mining Knowl Discov 2012, Volume 2, November/December 2012: 437–456 doi: https://doi.org/10.1002/widm.1074 .

[30] Hsu Hmwaye Aung, Khin Mar Myo, "ECLAT-Based Association Rules Mining for Education Training Centre," presented at Fourth Local Conference on Parallel and Soft Computing, Yangon Myanmar , 2009 http://onlineresource.ucsy.edu.mm/handle/123456789/1872 .

[31] M. Sinthuja, P. Aruna and N. Puviarasan, “Experimental Evaluation Of Apriori And Equivalence Class Clustering And Bottom Up Lattice Traversal (ECLAT) Algorithms,” Pak. J. Biotechnol. Vol. 13 special issue II (International Conference on Engineering and Technology Systems (ICET’16) Pp. 77 - 82 (2016) https://pjbt.org/index.php/pjbt/article/download/704/681 .

[32] M. Nair and F. Kayaalp , "Performance Comparison of Association Rule Algorithms with SPMF on Automotive Industry Data", Düzce Üniversitesi Bilim ve Teknoloji Dergisi, vol. 7, no. 3, pp. 1985-2000, Jul. 2019, DOI: http://dx.doi.org/10.29130/DUBITED.581931 .

[33] Shamila Nasreen, Muhammad Awais Azam, Khurram Shehzada, Usman Naeemc, Mustansar Ali Ghazanfar, “Frequent Pattern Mining Algorithms for Finding Associated Frequent Patterns for Data Streams: A Survey”, The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014), Procedia Computer Science 37 (2014) 109 – 116, doi: http://dx.doi.org/10.1016/j.procs.2014.08.019 .

[34] Mael Gueguen. (2020) Improving the performance and energy efficiency of complex heterogeneous manycore architectures with on-chip data mining.

[35] Gunawan, I.K.E Purnama, “Efficiency Comparison on Eclat, FP-Tree, and Top-Down Algorithms in Search L-Itemsets,” presented at The First International Conference on Green Computing and the second aun/seed-net regional, Yogyakarta Indonesia 2 – 3 March 2010 https://lppm.istts.ac.id/publication/download?id=185 .

[36] A.Meenakshi, “Survey of frequent Pattern Mining Algorithms in Horizontal and Vertical Data Layouts,” International Journal of Advances in Computer Science and Technology, Volume 4 No.4, April 2015, http://warse.org/ijacst/static/pdf/file/ijacst03442015.pdf .

[37] K.Vani / (IJCSIT) “International Journal of Computer Science and Information Technologies,” Vol. 6 (4) , 2015, 3980-3985 http://www.ijcsit.com/docs/Volume%206/vol6issue04/ijcsit20150604150.pdf .

[38] Goethals, Bart & Zaki, Mohammed. (2003). “FIMI'03: Workshop on Frequent Itemset Mining Implementations,” In Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations Melbourne, Florida, USA, November 19, 2003 pp 1-13 https://cs.fit.edu/~pkc/icdm03/printing/workshops/itemsets/fimi-workshop.pdf .

[39] Rana Ishita, Amit Rathod, “Frequent Itemset Mining in Data Mining: A Survey,” International Journal of Computer Applications (0975 – 8887), Volume 139 – No.9, April 2016 http://dx.doi.org/10.5120/ijca2016909219 .

[40] Rana Ishita, Amit Rathod, “Eclat with Large Data base Parallel Algorithm and Improve its Efficiency” International Journal of Computer Applications (0975 – 8887), Volume 143 – No.13, June 2016 http://dx.doi.org/10.5120/ijca2016910462 .

[41] Chee, Chin-Hoong & Jaafar, Jafreezal & Izzatdin, Abdul & Hasan, Mohd Hilmi & Yeoh, William, (2018), “Algorithms for frequent itemset mining: a literature review. Artificial Intelligence Review,” 52, pp 2603–2621, http://dx.doi.org/10.1007/s10462-018-9629-z .

[42] Vaishnav, Hritika and Choudhary, Anamika, Observational Studies on Algorithms of Frequent Pattern Mining (November 21, 2020). Proceedings of the 2nd International Conference on IoT, Social, Mobile, Analytics & Cloud in Computational Vision & Bio-Engineering (ISMAC-CVB 2020), Available at SSRN: https://ssrn.com/abstract=3734733 or http://dx.doi.org/10.2139/ssrn.3734733

[43] M. J. Zaki, "Scalable algorithms for association mining," in IEEE Transactions on Knowledge and Data Engineering, vol. 12, no. 3, pp. 372-390, May-June 2000, doi: http://dx.doi.org/10.1109/69.846291 .

[44] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. SIGMOD Rec. 29, 2 (June 2000), 1–12, https://doi.org/10.1145/335191.335372 .

[45] MachineX: Frequent Itemset generation with the FP-Growth algorithm https://blog.knoldus.com/machinex-frequent-itemset-generation-with-the-fp-growth-algorithm/ .

[46] T Andi and E Utami, “Association rule algorithm with FP growth for book search,” IOP Conf. Ser.: Mater. Sci. Eng. 434 012035, presented at 3rd Annual Applied Science and Engineering Conference (AASEC 2018) Curran Associates, Inc. (2018), 18 April 2018, Bandung, Indonesia, DOI: https://doi.org/10.1088/1757-899X/434/1/012035 .

[47] N. Mou, H. Wang, H. Zhang and X. Fu, "Association Rule Mining Method Based on the Similarity Metric of Tuple-Relation in Indoor Environment," in IEEE Access, vol. 8, pp. 52041-52051, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2980952 .

[48] J. Heaton, "Comparing dataset characteristics that favor the Apriori, Eclat or FP-Growth frequent itemset mining algorithms," SoutheastCon 2016, Norfolk, VA, USA, 2016, pp. 1-7, doi: https://doi.org/10.1109/SECON.2016.7506659 .

[49] Nataliya Boyko, Oleksandr Tkachyk, “Model for Finding Frequent Sets in FP-growth for Multimodal Data,” presented at the Fifth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2022), CEUR Workshop Proceedings (CEUR-WS.org) Proceedings, May 12, 2022, Zaporizhzhia, Ukraine https://ceur-ws.org/Vol-3137/paper5.pdf .

[50] M.S. Mythili, A.R. Mohamed Shanavas, “Performance Evaluation of Apriori and FP-Growth Algorithms” International Journal of Computer Applications (0975 – 8887) Volume 79 – No10, October 2013 DOI:10.5120/13779-1650 .

[51] P.Naresh, R.Suguna, “Implementation of Improved Association Rule Mining Algorithms for Fast Mining with Efficient Tree Structures on Large Datasets,” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019 https://www.ijeat.org/wp-content/uploads/papers/v9i2/B3876129219.pdf .

[52] Ke Wang, Liu Tang, Jiawei Han, and Junqiang Liu. 2002. Top Down FP-Growth for Association Rule Mining. In Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD '02). Springer-Verlag, Berlin, Heidelberg, 334–340, https://doi.org/10.1007/3-540-47887-6_34 .

[53] Yi Zeng, Shiqun Yin, Jiangyue Liu, Miao Zhang, "Research of Improved FP-Growth Algorithm in Association Rules Mining," Hindawi Publishing Corporation Scientific Programming, vol. 2015, Article ID 910281, 6 pages, 2015, https://doi.org/10.1155/2015/910281 .

[54] Aiman Moyaid Said, P D D. Dominic, Azween B Abdullah, “A Comparative Study of FP-growth Variations,” IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.5, May 2009 pp: 266-272 https://www.academia.edu/2077489/A_comparative_study_of_FP_growth_variations .

[55] K. Dharmaraajan and M. A. Dorairangaswamy, "Analysis of FP-growth and Apriori algorithms on pattern discovery from weblog data," 2016 IEEE International Conference on Advances in Computer Applications (ICACA), Coimbatore, India, 2016, pp. 170-174, doi: https://doi.org/10.1109/ICACA.2016.7887945 .

[56] Shivam Sidhu Upendra Kumar Meena Aditya Nawani Himanshu Gupta Narina Thakur, “FP Growth Algorithm Implementation”, International Journal of Computer Applications (0975 – 8887) Volume 93 – No.8, May 2014 pp:6-10, https://doi.org/10.5120/16233-5613 .

[57] Hahsler M, Buchta C, Gruen B, Hornik K (2023). *arules: Mining Association Rules and Frequent Itemsets*. R package version 1.7-6, https://CRAN.R-project.org/package=arules .

[58] Jim Hester, Davis Vaughan, Drew Schmidt,Posit Software, PBC bench: High Precision Timing of R Expressions https://cran.r-project.org/package=bench .

[59] Thomas Quinn peakRAM: Monitor the Total and Peak RAM Used by an Expression or Function https://cran.r-project.org/web/packages/peakRAM/

[60] Henrik Bengtsson R.utils: Various Programming Utilities, https://cran.r-project.org/package=R.utils .

[61] Winston Chang, Javier Luraschi, Timothy Mastny, RStudio, jQuery Foundation (jQuery library), jQuery contributors , Mike Bostock, Ivan Sagalaev (highlight.js library), profvis: Interactive Visualizations for Profiling R Code https://cran.r-project.org/web/packages/profvis/ .

[62] Henrik Bengtsson profmem: Simple Memory Profiling for R https://cran.r-project.org/package=profmem

[63] Wan Abu Bakar, Wan Aezwani & Man, Mustafa & Man, Mahadi & Abdullah, Zailani. (2020). i-Eclat: performance enhancement of eclat via incremental approach in frequent itemset mining. TELKOMNIKA (Telecommunication Computing Electronics and Control). 18. 562. https://doi.org/10.12928/telkomnika.v18i1.13497 .

[64] Wan Abu Bakar, Wan Aezwani & Abdullah, Zailani & Saman, Md & Jalil, Masita & Man, Mustafa & Herawan, Tutut & Hamdan, Abdul. (2016). Incremental-Eclat Model: An Implementation via Benchmark Case Study, https://doi.org/10.1007/978-3-319-32213-1_4 .

[65] Sharma, Neeraj & Nagwani, Naresh. (2011). Study and Analysis of Incremental Apriori Algorithm, Conference: High Performance Architecture and Grid Computing - International Conference, HPAGC 2011, Chandigarh, India, July 19-20, 2011. pp: 470-472, https://doi.org/10.1007/978-3-642-22577-2_64 .

[66] W. Thurachon and W. Kreesuradej, "Incremental Association Rule Mining With a Fast Incremental Updating Frequent Pattern Growth Algorithm," in IEEE Access, vol. 9, pp. 55726-55741, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3071777 .

[67] Han, J., Pei, J., Yin, Y. et al. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery 8, 53–87 (2004). https://doi.org/10.1023/B:DAMI.0000005258.31418.83 .

[68] A. Nogo, E. zunic and D. Donko, "Identification of association rules in orders of distribution companies' clients," IEEE EUROCON 2019 -18th International Conference on Smart Technologies, Novi Sad, Serbia, 2019, pp. 1-4, doi: https://doi.org/10.1109/EUROCON.2019.8861951 .

[69] R. Sivakumar and J. G. R. Sathiaseelan, "A performance based empirical study of the frequent itemset mining algorithms," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India, 2017, pp. 1627-1631, doi: https://doi.org/10.1109/ICPCSI.2017.8391988 .