EXPLORING FREQUENT ITEMSET GENERATION

- Present by Wei Li

- In lecture talk:
  - Association rule mining and Frequent itemset generation problem
  - ECLAT to generate frequent itemsets in a vertical database
- My project:
  - Implement ECLAT and two algorithms: FP-Growth and Apriori for frequent itemset generation
  - Compare three algorithms in two dataset
- Agenda:
  - Refresh the problem
  - Apriori
  - FP-Grwoth
  - Project Experiments and Results

THE PROBLEM

- Discover association rules
  - if-then relationship based on co-occurrence
  - {Bread, Butter} −> {Milk}
- Break Association Rules Mining into 2 tasks
  - **Generating frequent itemsets**
  - Generating rules from frequent itemsets
- Three sequential, exhaustive methods
  - **ECLAT**
  - **Apriori**
  - **FP-growth**

TID	List of item_IDs
T10	Coke,Slice,Kurkure
T20	Coke, Kurkure
T30	Slice, pizza

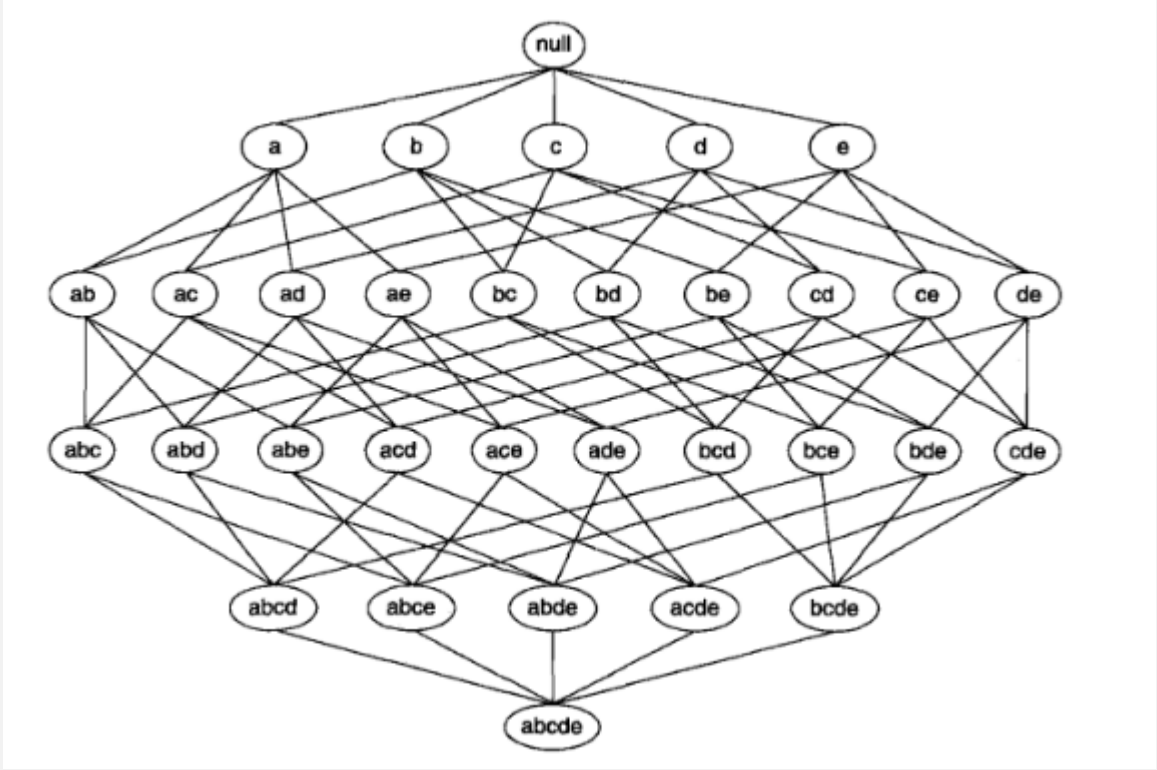
- Horizontal database

Itemsets	TID_set
Coke	T10,T20
Slice	T10,T30
Kurkure	T10,T20
Pizza	T30

- Vertical database

TERMS/ GENERATE ALL FREQUENT SET

- **Commodity set** $I = \{i_1, i_2, \ldots, i_n\}$
- **transaction set** $\Omega = \{R_1, R_2, \ldots, R_m\}$ where $R_i \subseteq I$.
- **Itemset** $X \subseteq I$ **K- Itemset**: $X \subseteq I$ and $|X| = k$
- **Support count :** $\delta(X) = |\{R_i | X \subseteq R_i \& R_i \in \Omega\}|$



- **Frequent Itemset:** itemset whose $\delta(X)$ is greater than a threshold (eg, 3)
- **Maximal frequent Itemset:** An frequent itemset is maximal frequent Itemset if none of its supersets are frequent.
- **Problem: frequent itemset generation**
  - **Input: commodity set** I **, transaction set** Ω **, frequent threshold** $minsup$
  - **Output: all frequent itemset** $\delta(X) >$ $minsup$

APRIORI

- Apriori Property:
  - If an itemset is frequent, all of its subset are frequent
  - If an itemset is not frequent, all of its superset must not be frequent

Algorithm 1: Apriori(Ω, $minsup$)

Result: F

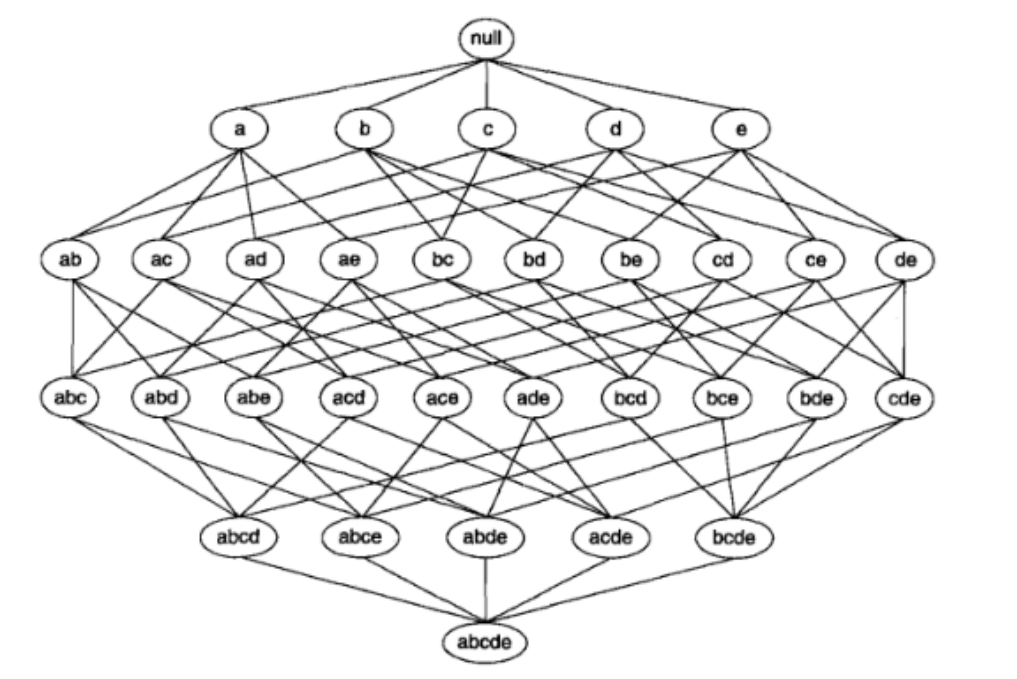
```
1  GLOBAL VAR, estimate;
2  k ← 1;
3  F_1 ← all frequent 1-itemsets;
4  F ← {(f, δ(f)) | f ∈ F_1};
5  while F_k ≠ φ do
6  |   k ← k + 1;
7  |   C_k ← candidate(F_{k-1}, ...);
8  |   for R_i ∈ Ω do
9  |   |   D_i ← {c | c ∈ C_k and c ⊆ R_i};
10 |   |   for d ∈ D_i do
11 |   |   |   δ(d) ← δ(d) + 1;
12 |   |   end
13 |   end
14 |   F_k = {c | c ∈ C_k and δ(c) ≥ minsup};
15 |   F ← F ∪ {(f, δ(f)) | f ∈ F_k};
16 end
17 return F;
```

K=1

K=2

K=3

K=4



FP-GROWTH

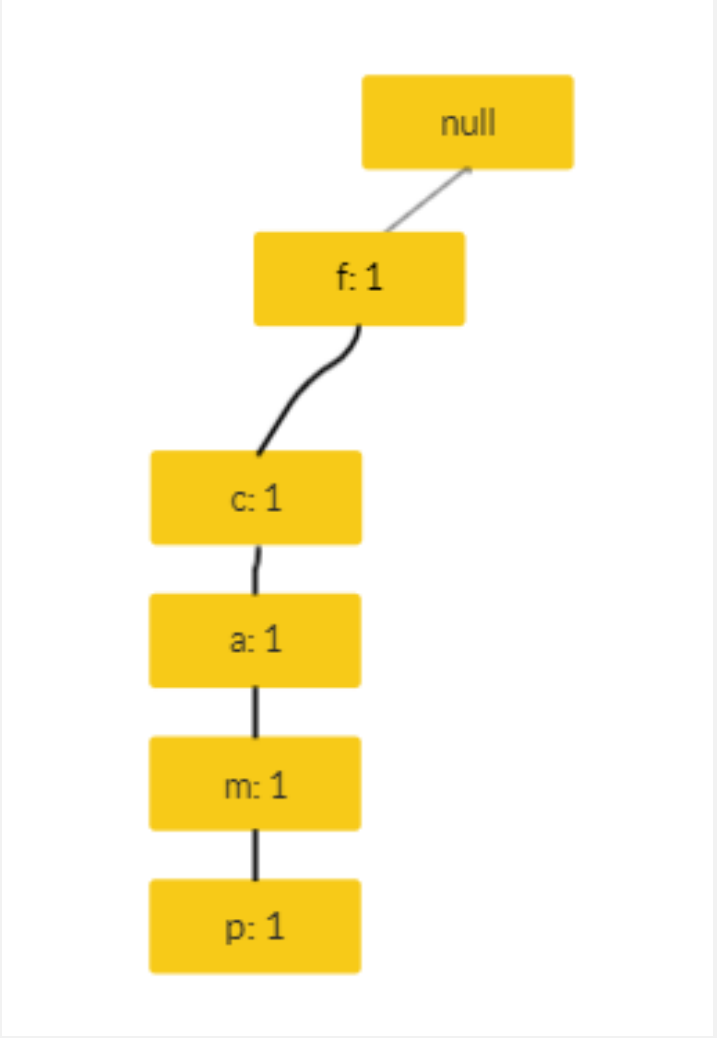
Minsup = 3

TID	Items bought
100	{a, c, d, f, g, i, m, p}
200	{a, b, c, f, i, m, o}
300	{b, f, h, j, o}
400	{b, c, k, s, p}
500	{a, c, e, f, l, m, n, p}

item	freq	rank id
f	4	1
c	4	2
a	3	3
b	3	4
m	3	5
p	3	6

TID = 100

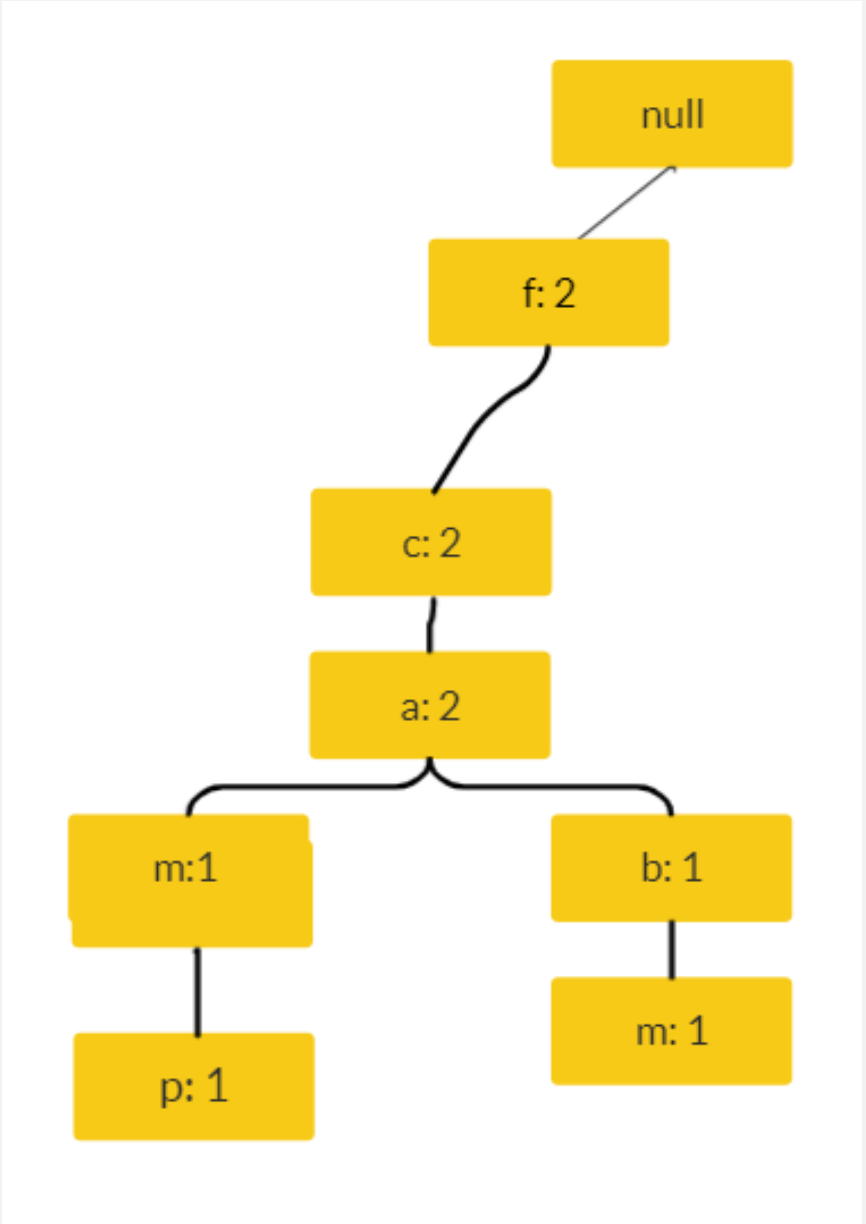
TID	Item bought
100	{f, c, a, m, p}
200	{f, c, a, b, m}
300	{f, b}
400	{c, b, p}
500	{f, c, a, m, p}



FP-GROWTH

TID = 200

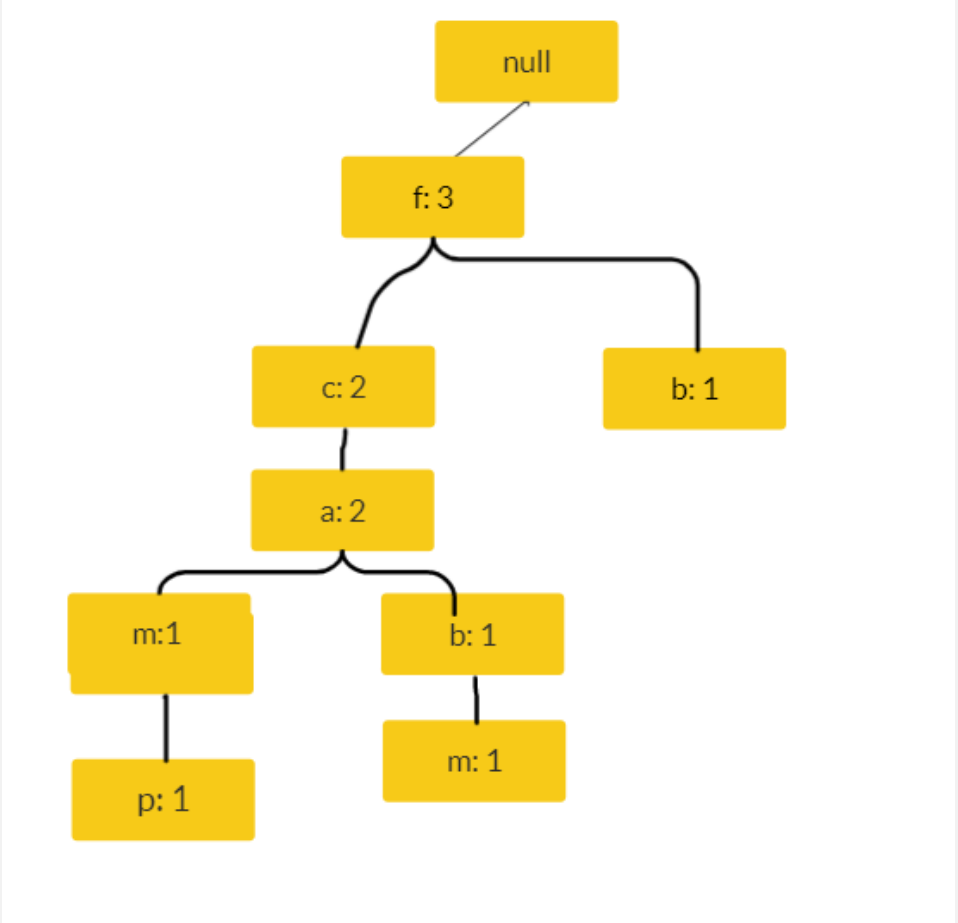
TID	Item bought
100	{f, c, a, m, p}
200	{f, c, a, b, m}
300	{f, b}
400	{c, b, p}
500	{f, c, a, m, p}



FP-GROWTH

TID = 300

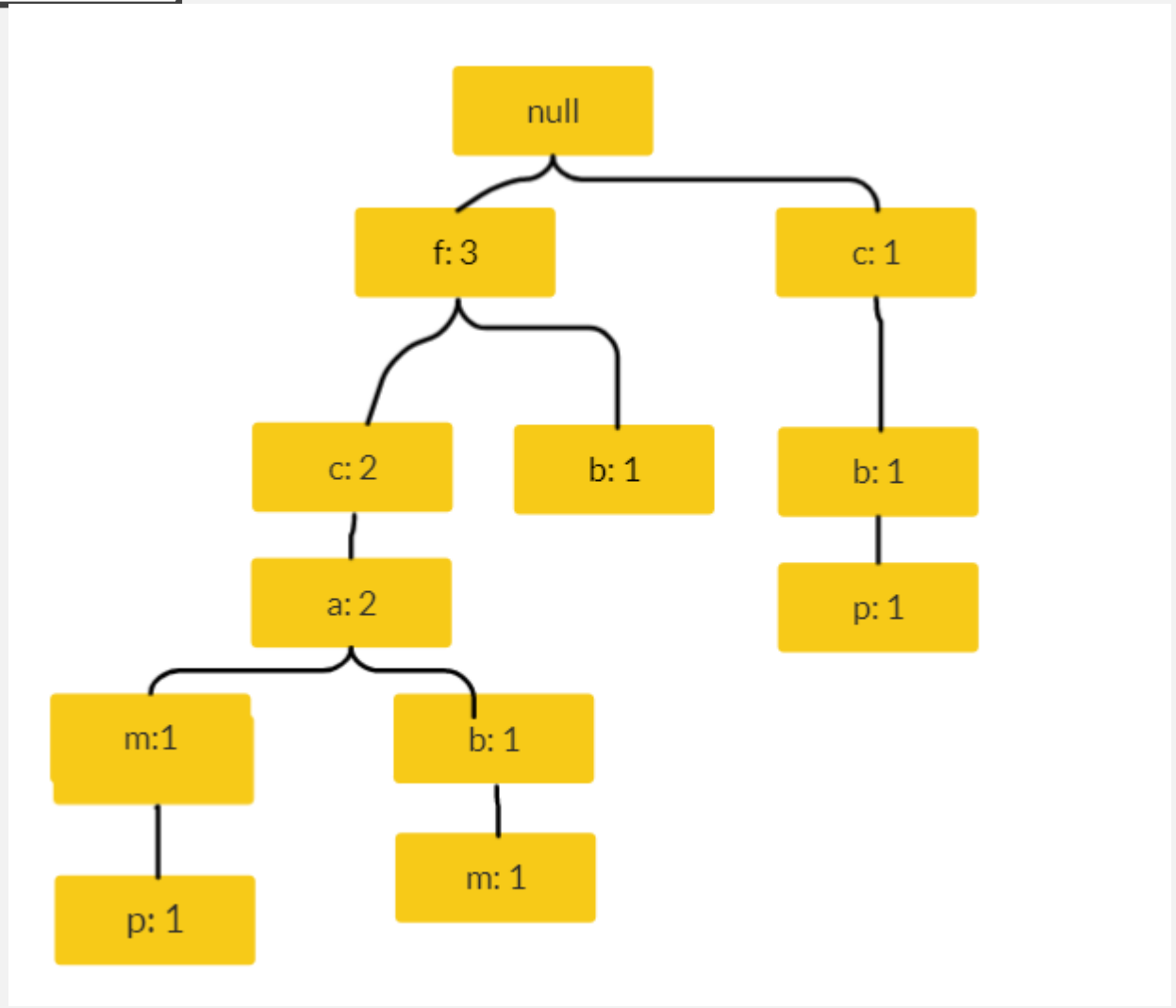
TID	Item bought
100	{f, c, a, m, p}
200	{f, c, a, b, m}
300	{f, b}
400	{c, b, p}
500	{f, c, a, m, p}



FP-GROWTH

TID = 400

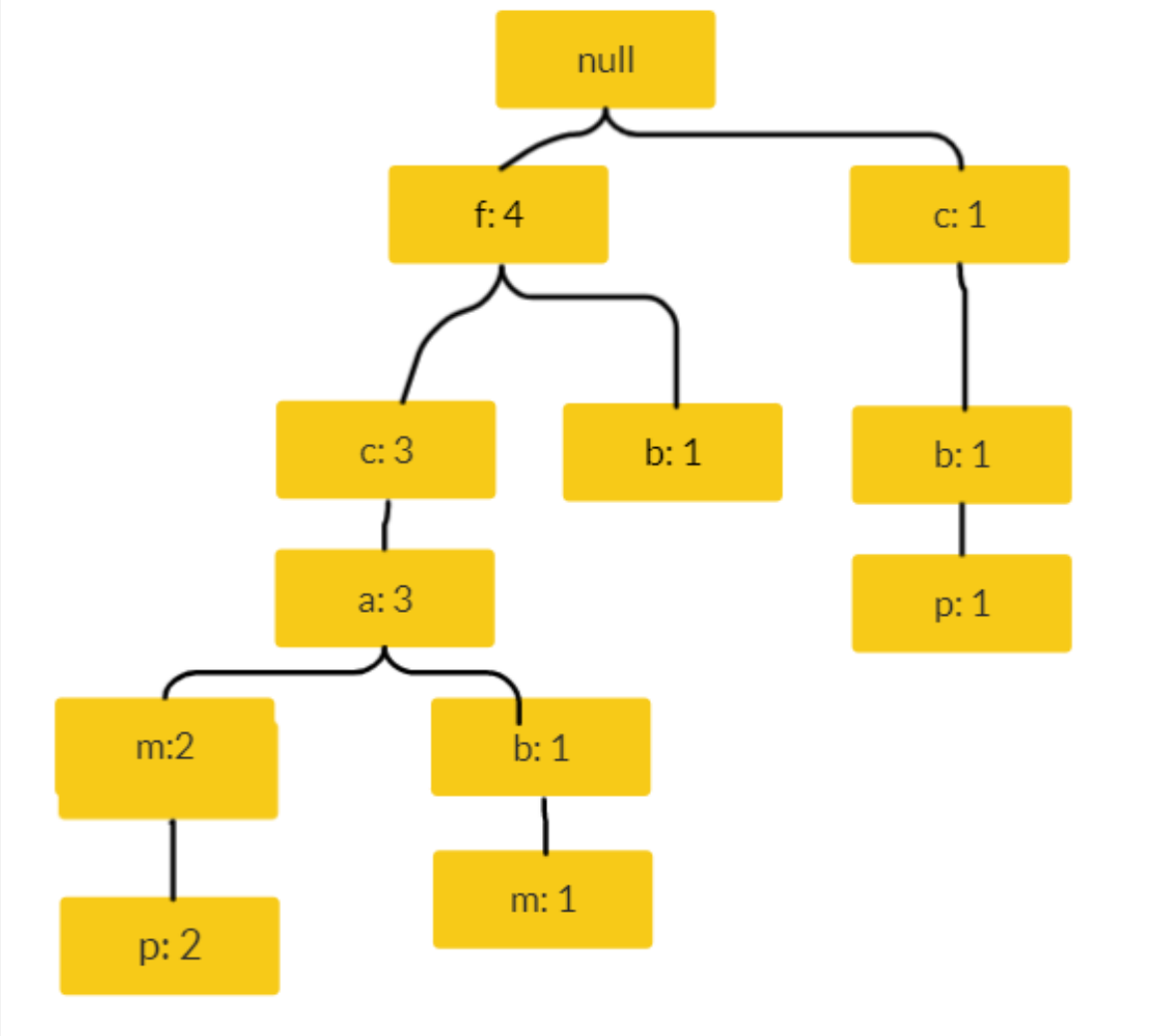
TID	Item bought
100	{f, c, a, m, p}
200	{f, c, a, b, m}
300	{f, b}
400	{c, b, p}
500	{f, c, a, m, p}



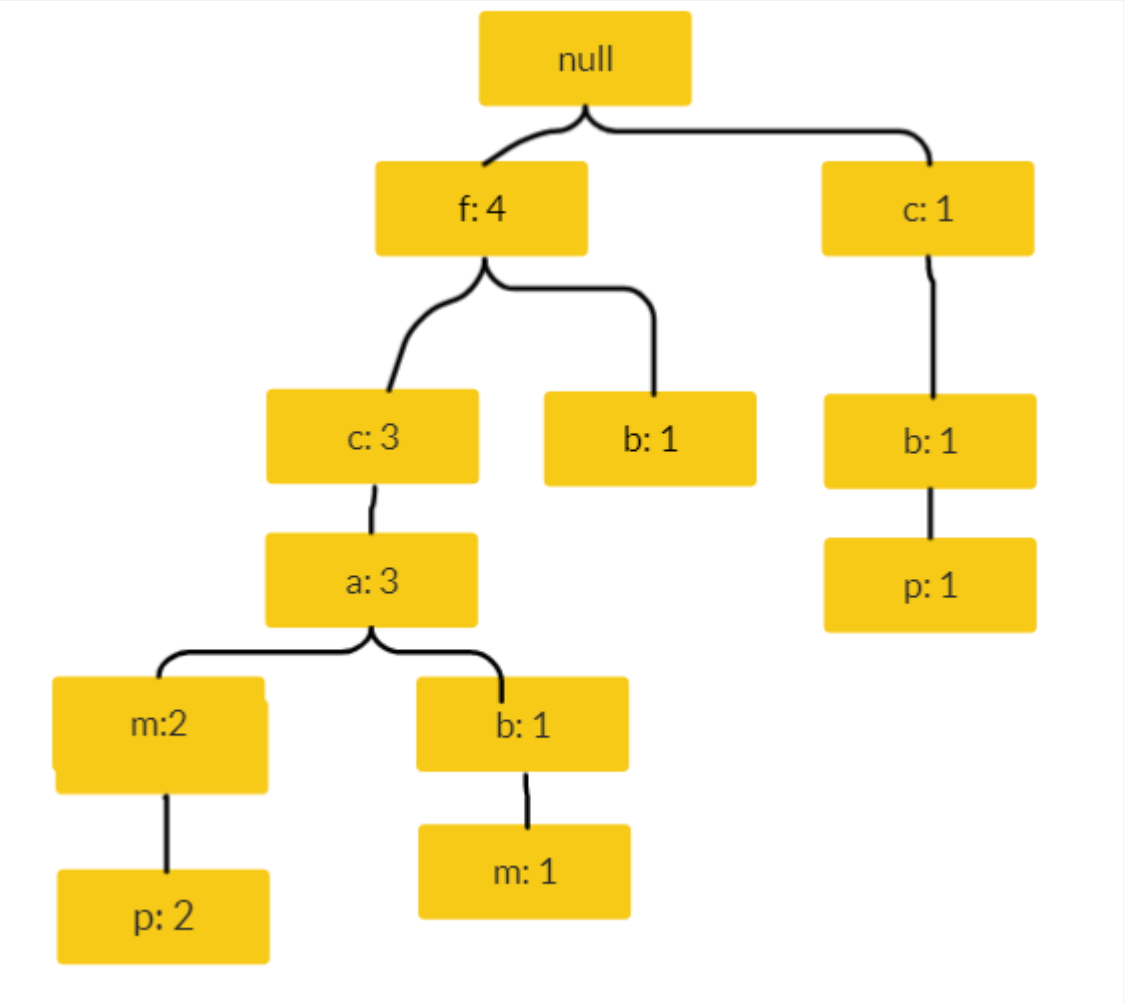
FP-GROWTH

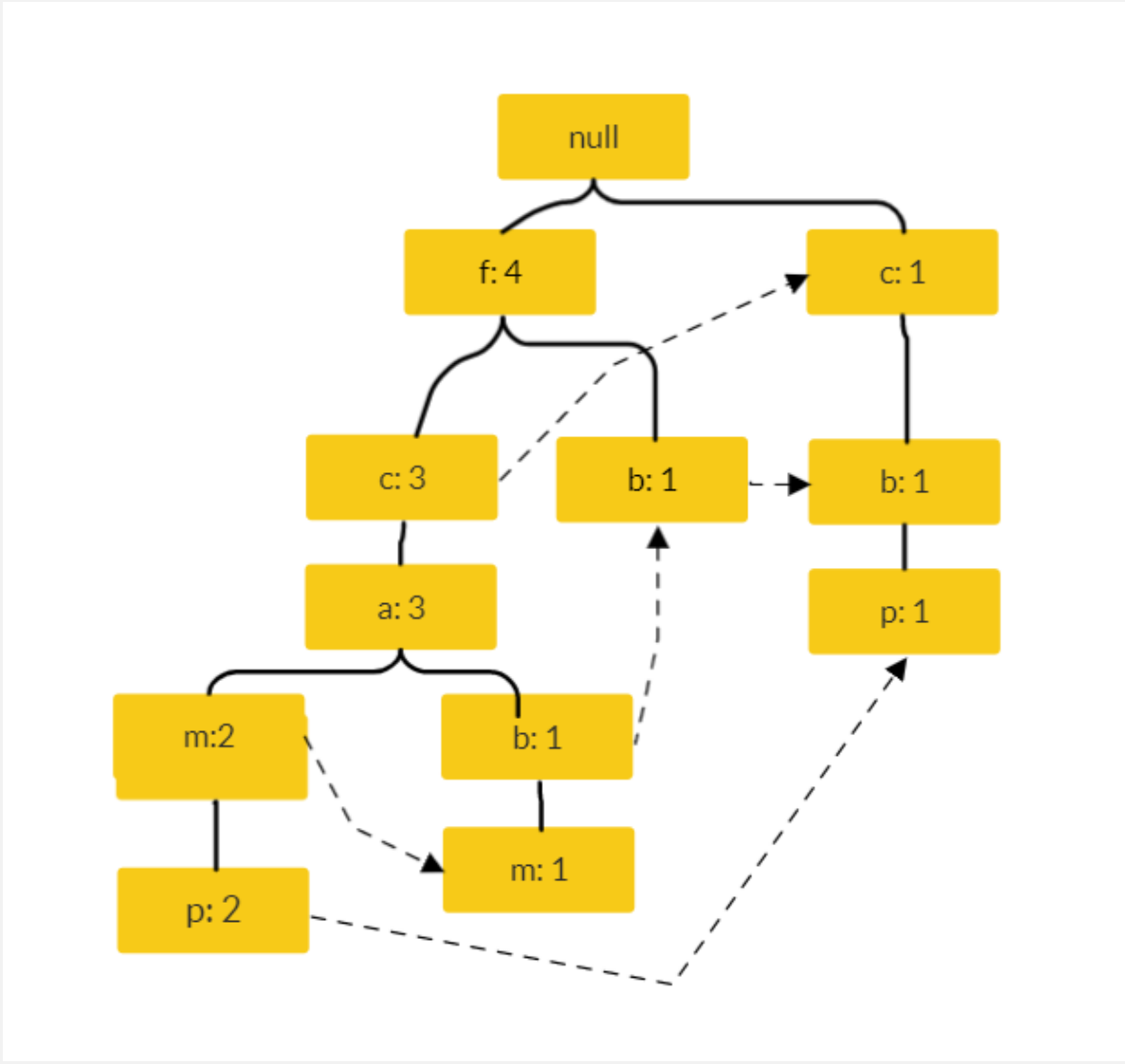
TID = 500

TID	Item bought
100	{f, c, a, m, p}
200	{f, c, a, b, m}
300	{f, b}
400	{c, b, p}
500	{f, c, a, m, p}



FP-GROWTH


null
f: 4
c: 1
c: 3
b: 1
b: 1
a: 3
p: 1
m:2
b: 1
p: 2
m: 1


null
f: 4
c: 1
c: 3
b: 1
b: 1
a: 3
p: 1
m:2
b: 1
m: 1
p: 2

FP-GROWTH

item	freq	rank id
f	4	1
c	4	2
a	3	3
b	3	4
m	3	5
p	3	6

Find frequent itemset {f}

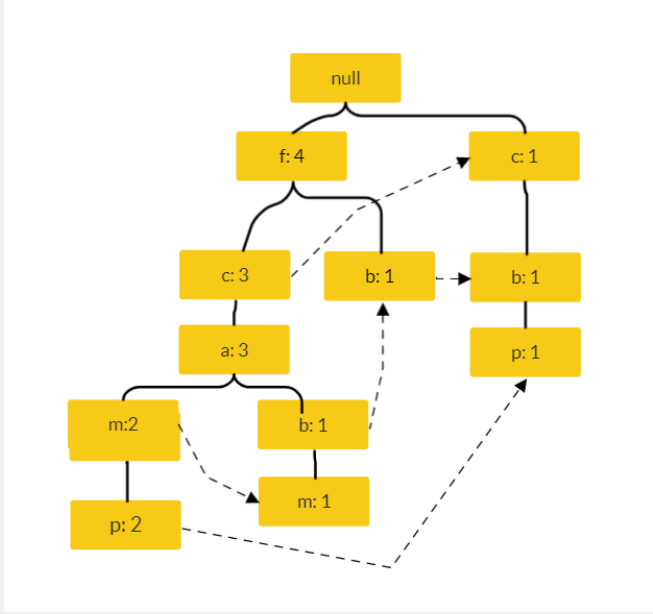
Find frequent itemset {f, c}, {c}

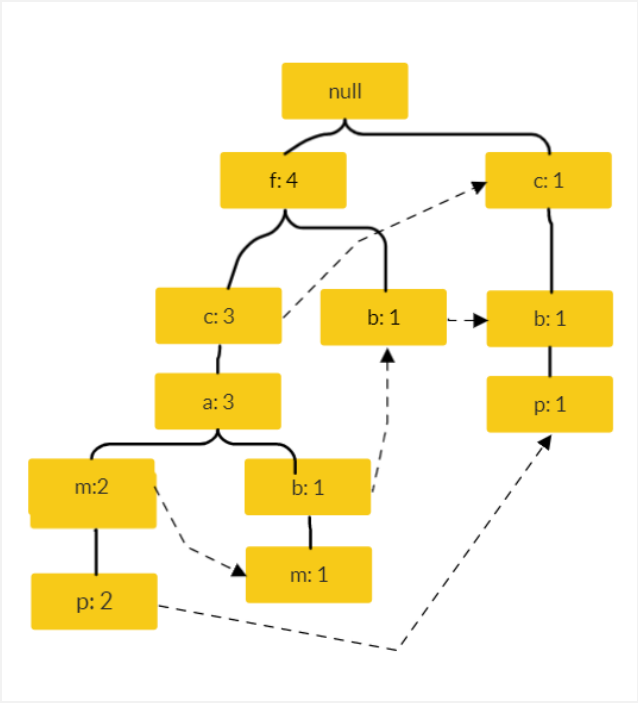
…

Find frequent itemset end with ‘m’

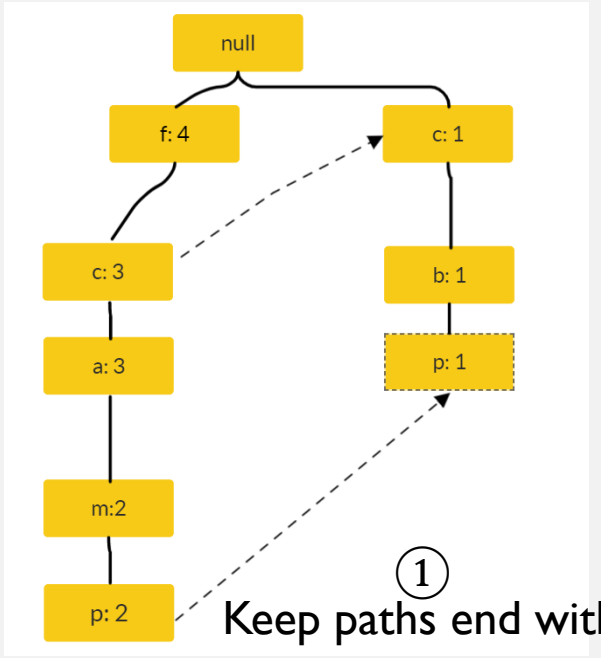
Find frequent itemset end with ‘p’

- Suffix based Divide-and-Conquer
- Find frequent itemset end with ‘p’
  - frequent itemset end with ‘mp’ /(bp, ap, cp, fp)
  - frequent itemset end with ‘bmp’……

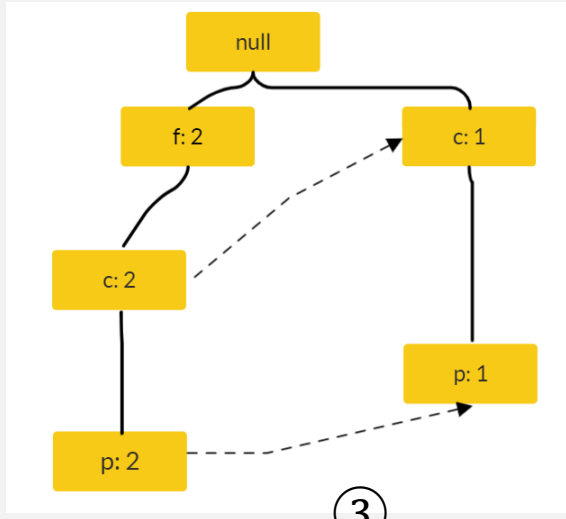




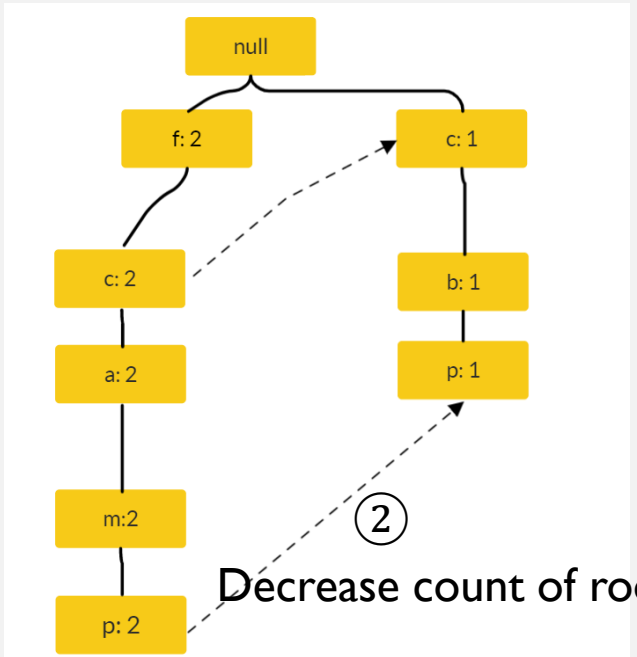
generate frequent itemset {P}



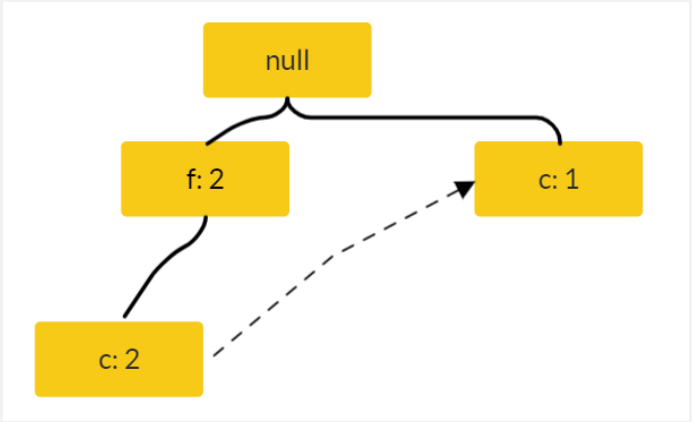
① Keep paths end with {P}



③ Remove non-frequent items



② Decrease count of roots



④ FP-tree condition on ‘p’

Recursively solve the conditional FP-tree

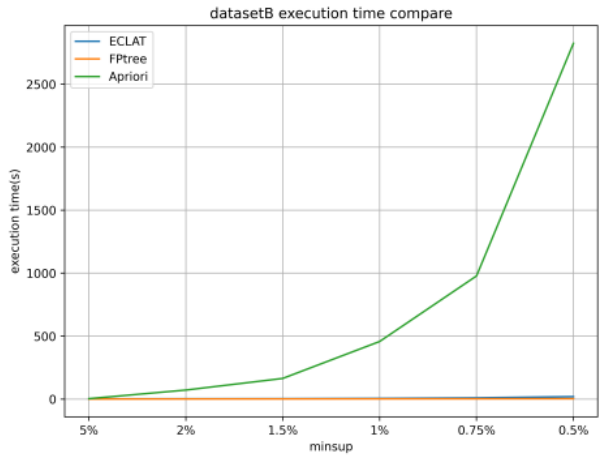
EXPERIMENT

- Data set
  - datasetA: realistic groceries data set, 14963 transactions, 167 unique commodities, average transaction length is 2.54
  - datasetB: synthetic data set with 540455 transactions and 2603 unique items. Average transaction length is 4.37
- Minsup:
  - 5%, 2%, 1.5%, 1%, 0.75%, 0.5%

$$support = \frac{\delta(X)}{|\Omega|}$$

RESULT

(a) Comparing three algorithms on DatasetA



(b) Comparing three algorithms on DatasetB

(a) Comparing ECLAT and FP-tree on DatasetA

(b) Comparing ECLAT and FP-tree on DatasetB

- Thanks

[Scalable Algorithms for Association Mining]