数据挖掘导论

Introduction to Data Mining

Frequent Pattern Mining

数据智能实验室
DATA INTELLIGENCE LABORATORY
DI-LAB
浙江大学
Zhejiang University

Agenda

- Applications
- Apriori Algorithm
- Eclat Algorithm
- FP-Tree and FP-Growth
- Summary

Frequent Item Set Mining

- **A data record corresponds to a set of items**
  - ✓ Find the items that are bought together for at least 100 times
  - ✓ Find the items that are frequently bought together with {Beer}

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

Frequent Item Set Mining

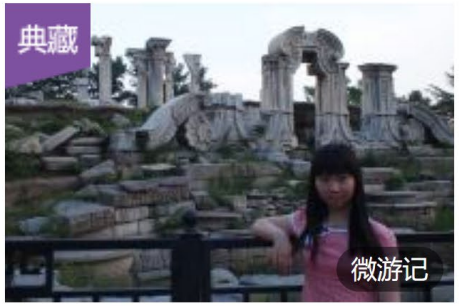
Frequent Attraction Set Mining

京郊野趣自驾，一起抓住夏天的尾巴（附北京周边景点路线及自...

Helen晓世界发表于 2020-09-03　　最新回复 2020-09-03

2020注定是每个人都难以忘却的，从年初开始，大家的生活就被迫做出了改变，我们也不例外。本来双双辞职是为.....

4天 2020-08-18 00:00:00， ￥1500， 夫妻

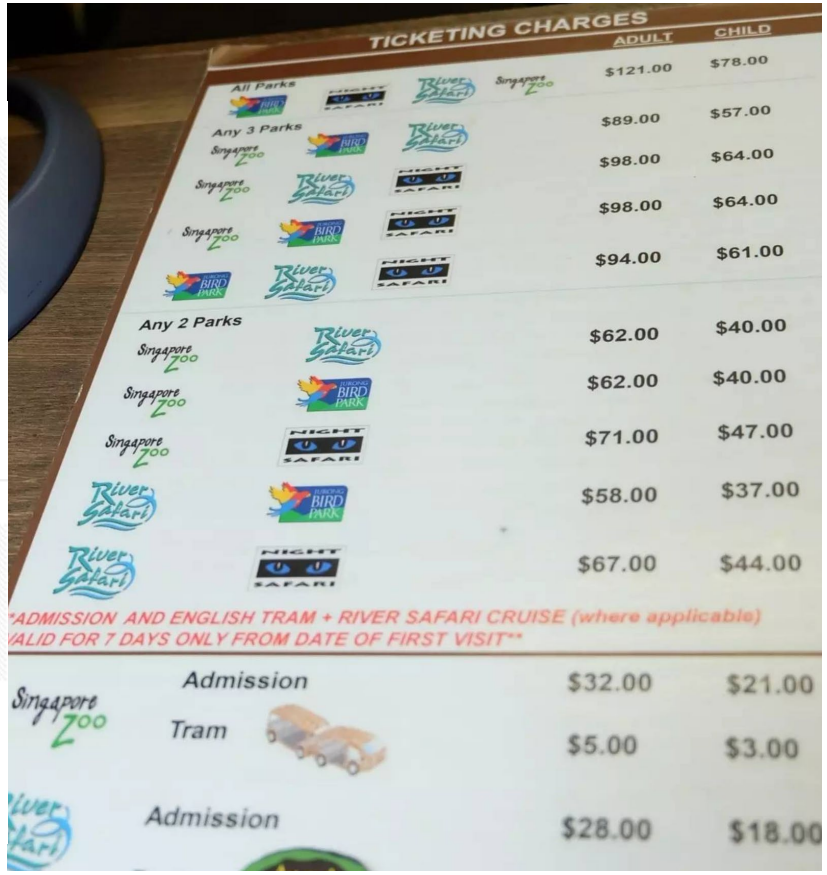


北京5日经典线路美食自由行（长城+十三陵+毛主席纪念堂+人...

兔兔200发表于 2014-01-28　　最新回复 2018-07-28

去北京前，从携程订了一日游和汉庭海友酒店（大栅栏店） 订单编号：803598103 产品名称：北.....

1天 2013-06-01 00:00:00， 情侣



Frequent Sequence Mining

DNA sequence mining

ID	S_i
1	<ACGCAG>
2	<AGCGAC>
3	<CGCACG>
4	<AGCACG>

Support=2

Frequent sequence set	Patterns
L1	A, C, G
L2	AC, CG, GC, CA, AG
L3	ACG, CGC, GCA, AGC, CAC
L4	CGCA, GCAC, CACG
L5	GCACG

之一。生物由于进化等目的对基因进行复制，产生大量重复序列，在遗传分析中起重要作用。过去一直认为被称为“垃圾DNA”的非基因序列没有功能，近年来的研究揭示它们具有重要的功能，并且其中很大部分是重复序列[2]。2003 年 5 月，Makalowski 在《Science》上发表文章表示重复序列在进化过程中可以用于帮助形成新基因[3]，International Human Genome Sequencing Consortium 在《Nature》上的文章中提到很多人类疾病是由重复序列的突变所引起的，如 Williams 综合症、Charcot-Marie-Tooth 病（肌骨肌萎缩症）等[4]。研究和寻找 DNA 序列中的重复序列对于了解基因的演化、进化历史和基因变异的原因有重要意义。采用数据挖掘技术，发现并分析 DNA 重复序列是必要的。

Image source: https://www.jsjkx.com/CN/article/openArticlePDF.jsp?id=8242

Frequent Sequence Mining

Frequent Travel Route Mining

江浙沪4日经典线路

29% 蜂蜂会选择这条线路

游玩时间：4天　最佳季节：全年

杭州 → 南浔 → 乌镇 → 上海 → 苏州

查看路线详情

路线亮点

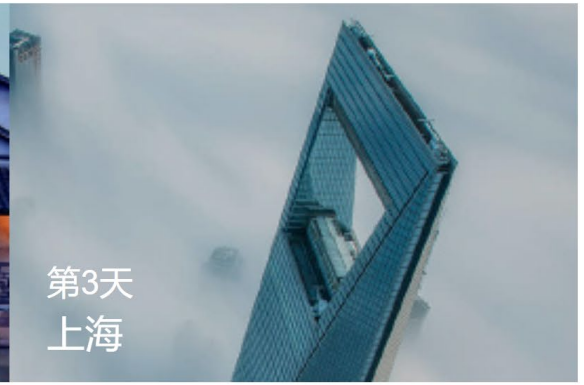
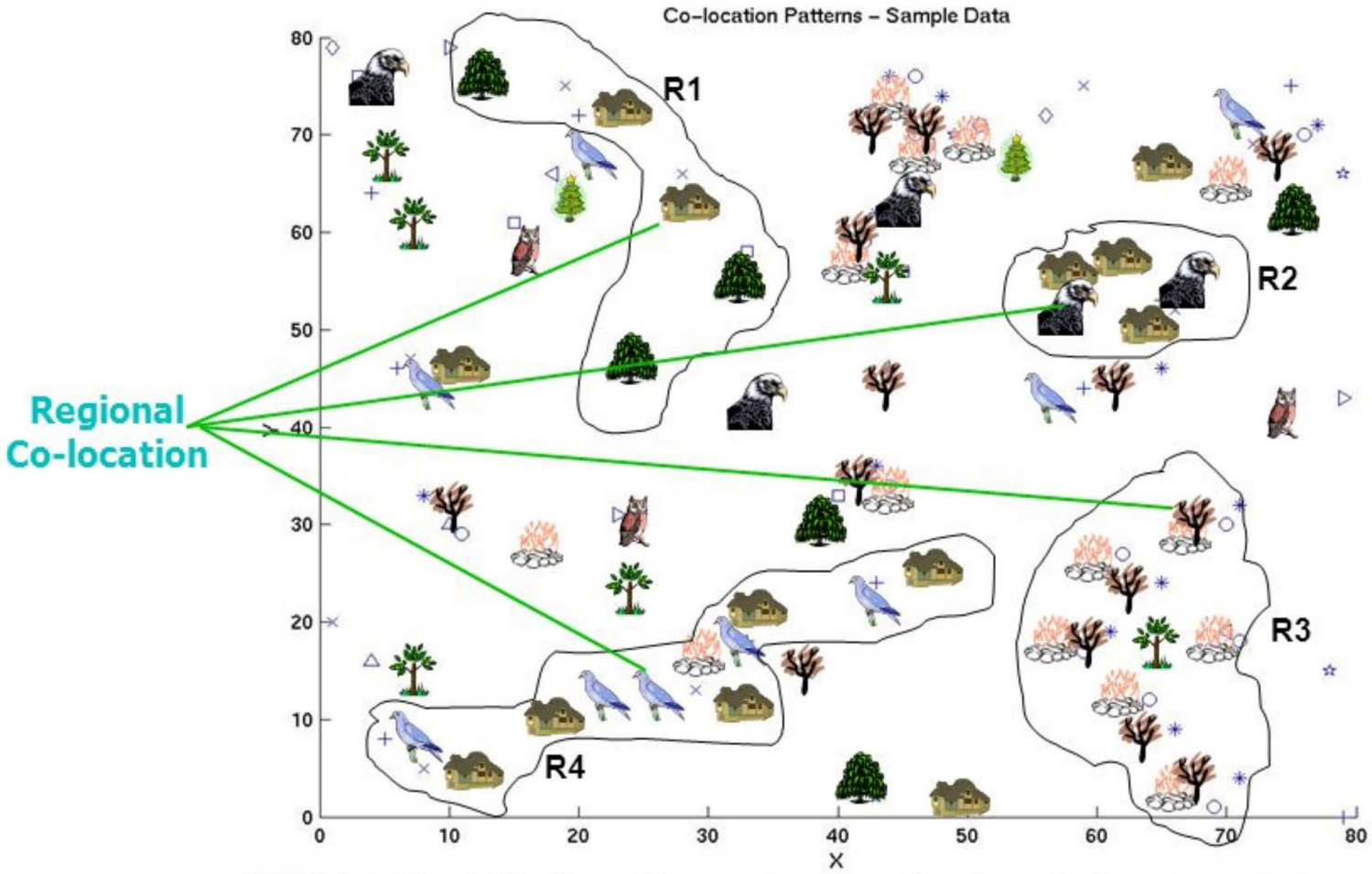


Image source: http://www.mafengwo.cn/mdd/route/10156.html

Frequent Co-Location Mining



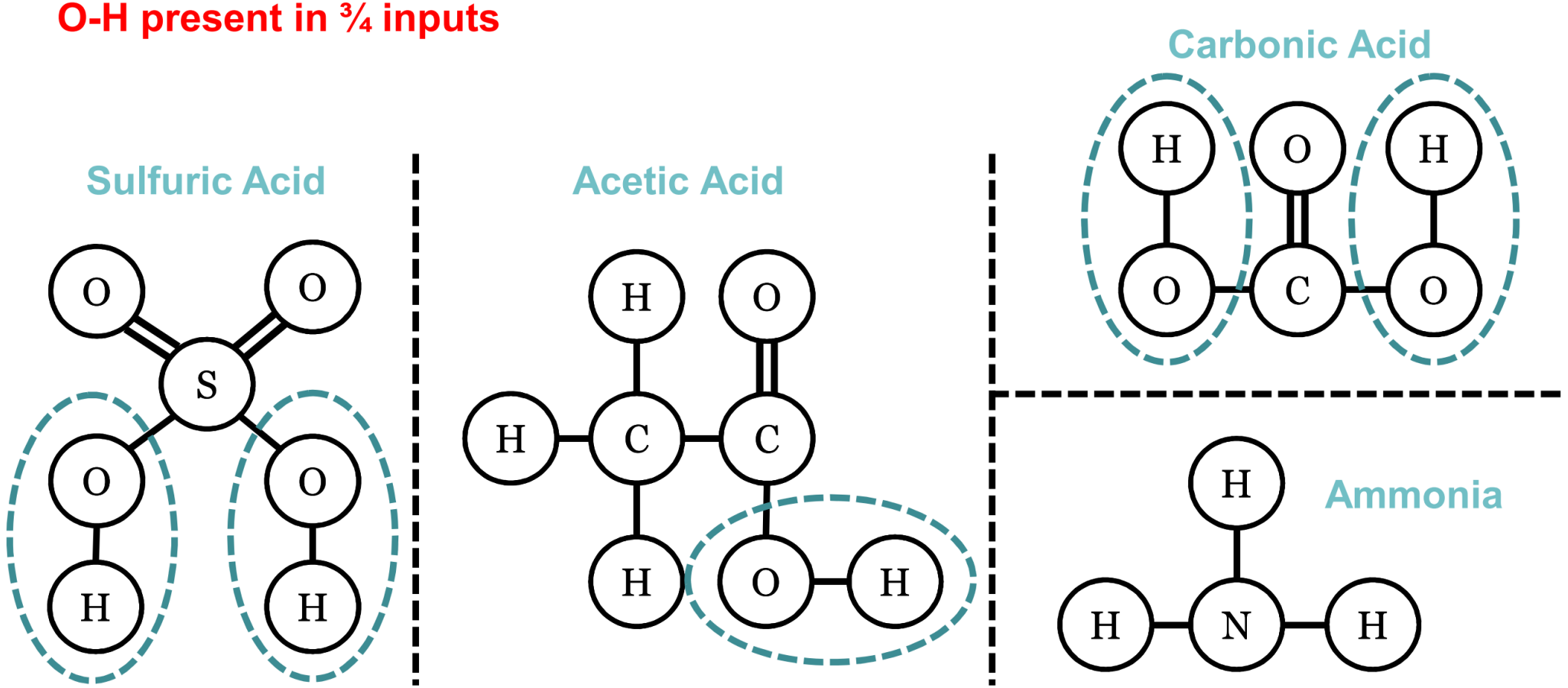
Task: Find Co-location patterns for the following data-set.

Global Co-location: and are co-located in the whole dataset

Frequent Subgraph Mining

- Discovery of subgraph structures that occur frequently across a set of graphs

O-H present in ¾ inputs



Agenda

- Concepts and Applications
- Apriori Algorithm
- Eclat Algorithm
- FP-Tree and FP-Growth
- Summary

Apriori Algorithm

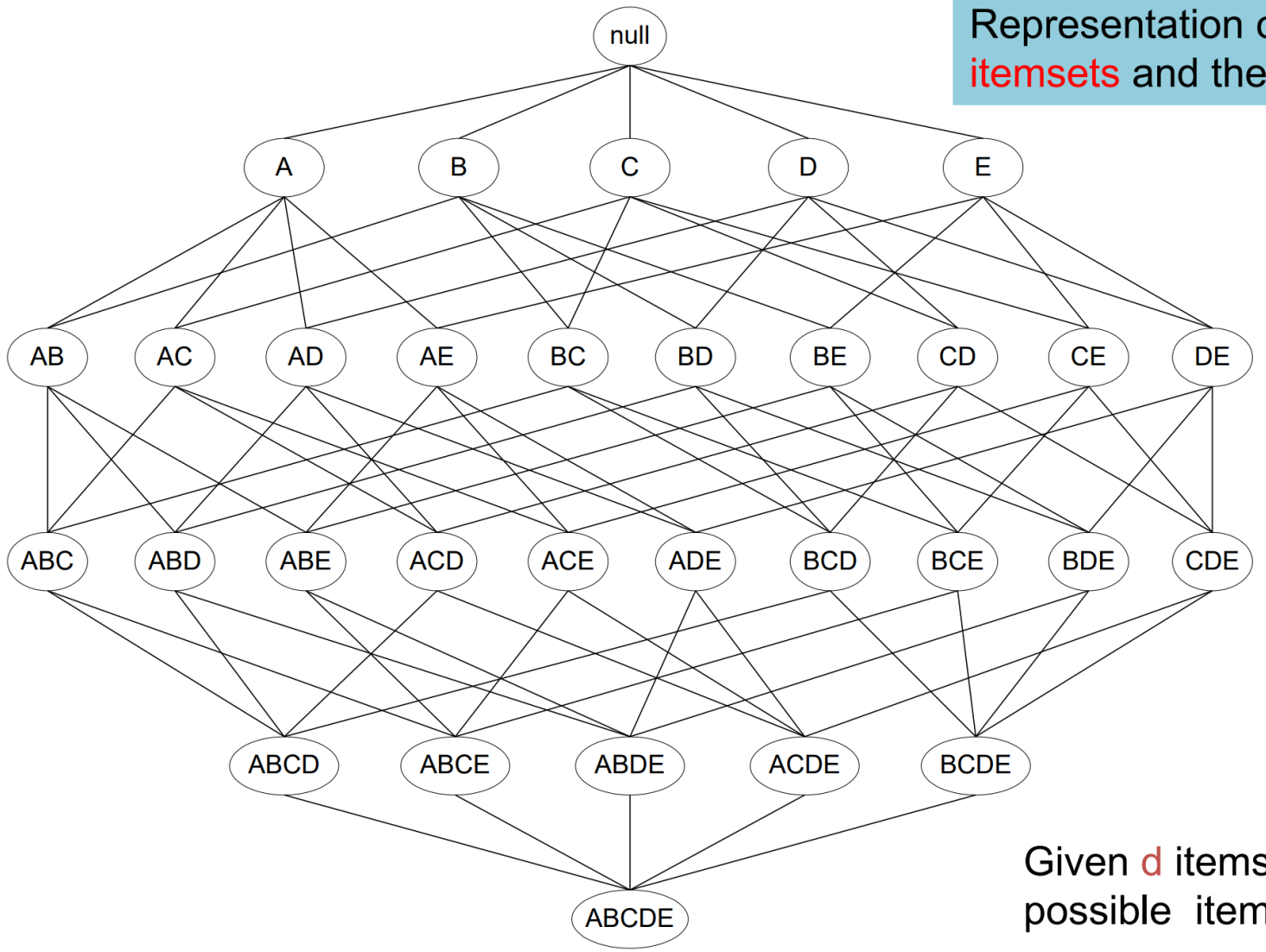
Transaction No	Products
1	beer, wine, cheese
2	beer, potato chips
3	eggs, flour, butter, cheese
4	eggs, flour, butter, beer, potato chips
5	wine, cheese
6	potato chips
7	eggs, flour, butter, wine, cheese
8	eggs, flour, butter, beer, potato chips
9	wine, beer
10	beer, potato chips
11	butter, eggs
12	beer, potato chips
13	flour, eggs
14	beer, potato chips
15	eggs, flour, butter, wine, cheese
16	beer, wine, potato chips, cheese
17	wine, cheese
18	beer, potato chips
19	wine, cheese
20	beer, potato chips

Find the set of items appear together for at least 7 times!

Product	Number of occurrences
Wine	8
Cheese	8
Beer	11
Potato Chips	10
Eggs	7
Flour	6
Butter	6

Image source: https://towardsdatascience.com/the-apriori-algorithm-5da3db9aea95

Search Space – Item Set Lattice



Representation of all possible itemsets and their relationships

Given d items, there are 2^d possible itemsets

Image source: https://www.cs.uoi.gr/~tsap/teaching/2016-cse012/slides/datamining-lect3.pdf

Frequent Item Set Mining

- **Item Set**
  - ✓ A set of one or multiple items {beer, diaper}
- **K-item Set**
  - ✓ An item set with k items. {beer, diaper} is a 2-item set
- **Support**
  - ✓ Frequency of an item set. Sometimes, support is represented as fraction
- **Frequent Item Set**
  - ✓ The support of the item set is no smaller than threshold minSup

Definitions for Frequent Item Set Mining

- If minSup=3, the following item sets are frequent
  - ✓ 1-item sets: {Beer}, {Diaper}, {Eggs}, {Nuts}
  - ✓ 2-item set: {Beer, Diaper}

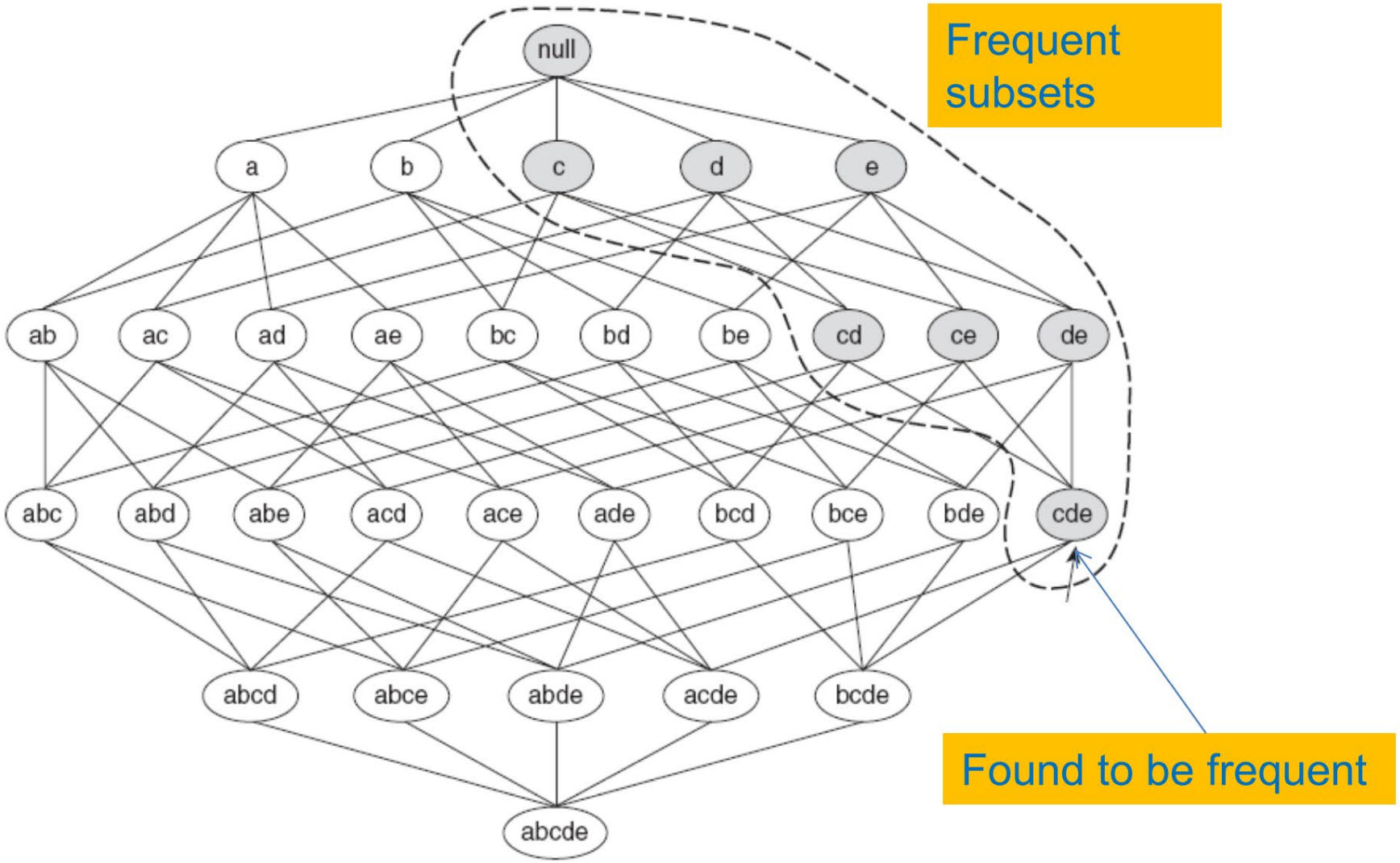
Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

Key Observations

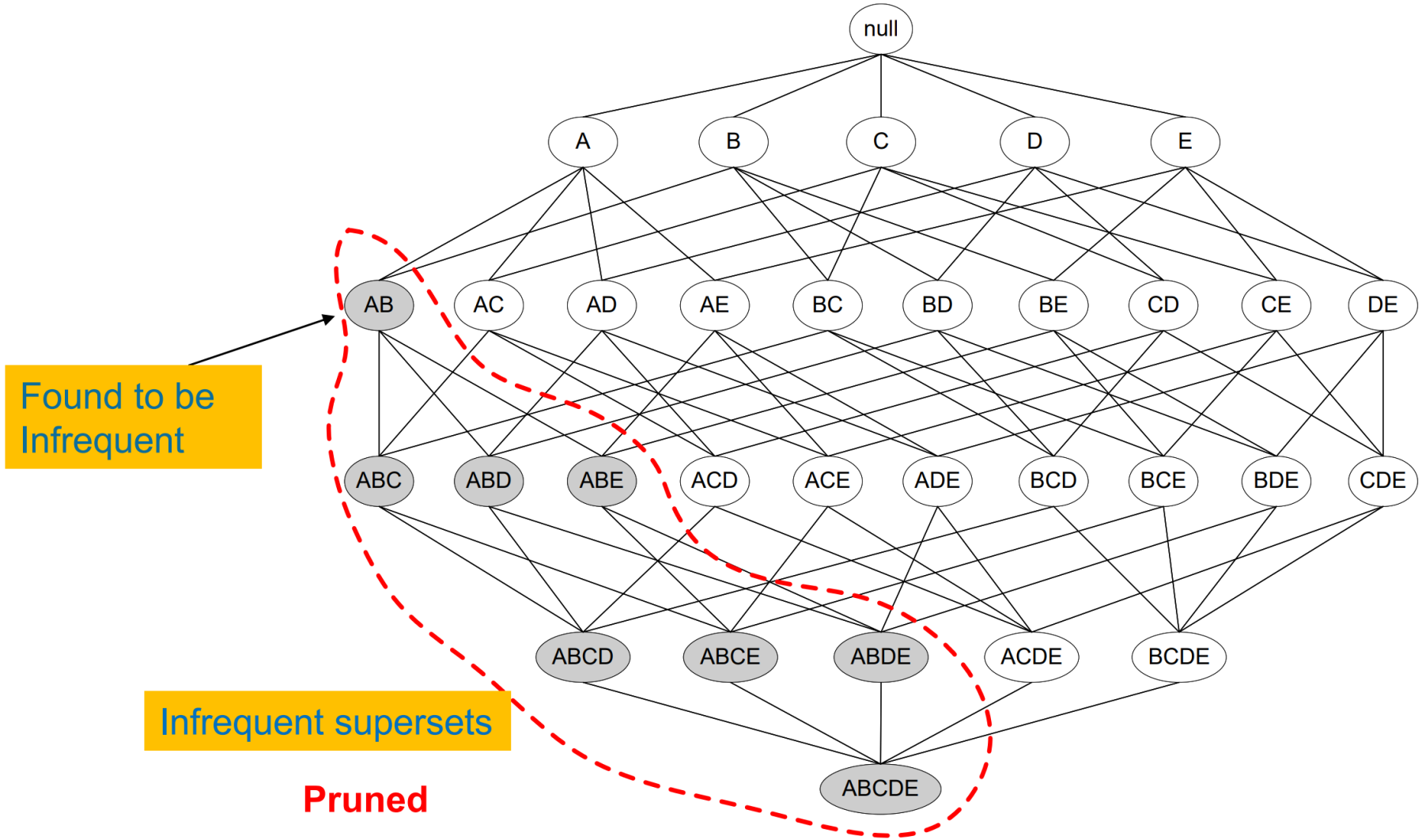
- **Observation 1:**
  - ✓ If an item set is frequent, its subsets are frequent
  - ✓ Sup({Beer}) >= Sup({Beer, Diaper})
  - ✓ Sup({Beer, Diaper}) >= Sup({Beer, Diaper, Nuts})
- **Observation 2:**
  - ✓ If an item set is infrequent, its supersets are infrequent
  - ✓ {Beer, Eggs} is infrequent
  - ✓ {Beer, Eggs, Diaper} is also infrequent

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

Observation 1



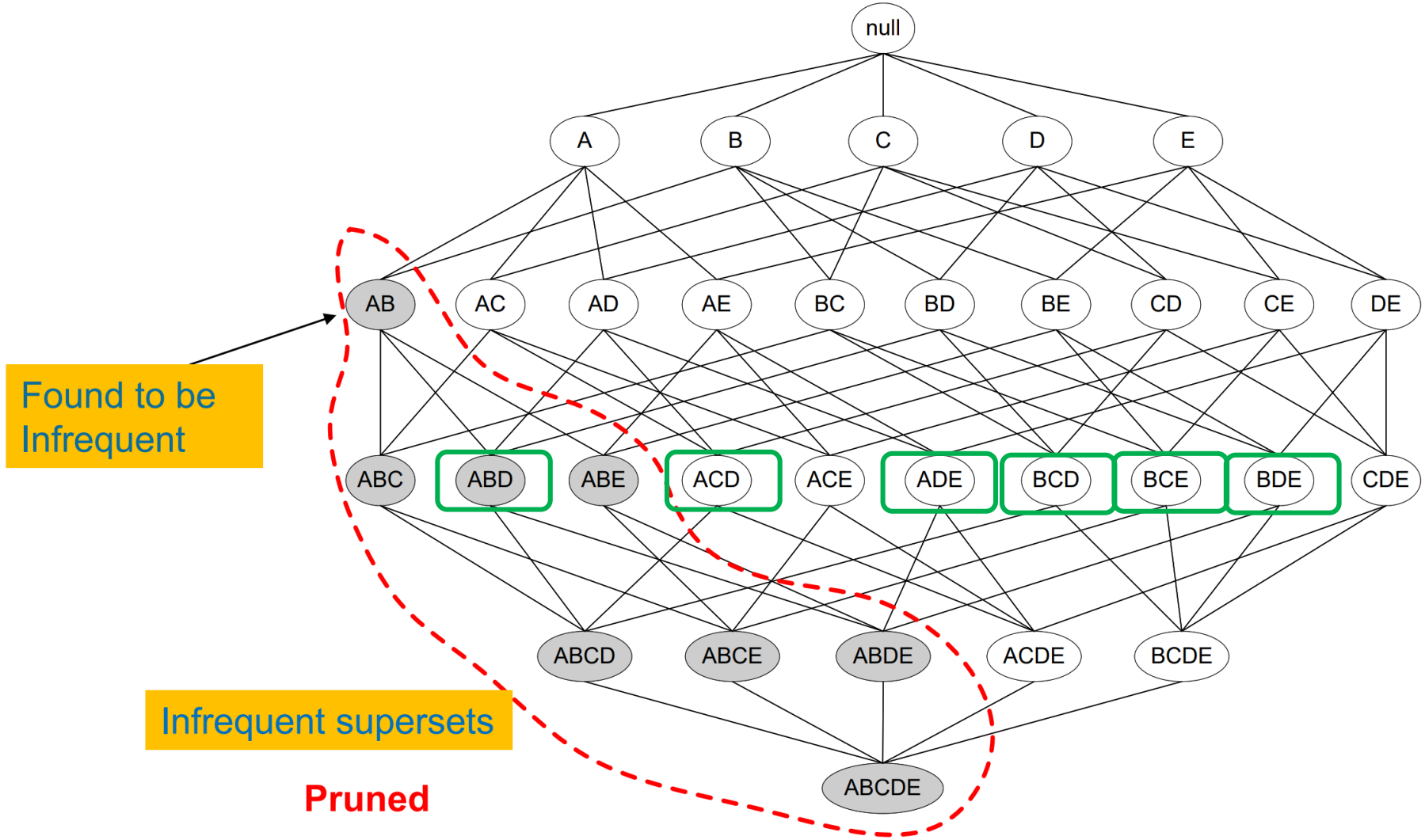
Observation 2



Apriori Algorithm

- **Level-by-level in the lattice**
  - ✓ First generate the candidates with only 1 item (e.g., {A}, {B}, {C},…)
  - ✓ Then, generate frequent 2-item sets (e.g., {AB}, {AC}, {BC},…)
  - ✓ Then, 3-item sets (e.g., {ABC}, {ABD},…)
- **Candidates in the (i)-th level are generated from the frequent candidates in the (i-1)-th level**
  - ✓ If i=2, enumerate all the valid candidates in the first level
  - ✓ If i>2, two candidates in the (i-1)-th level can be joined if they share the same prefix
  - ✓ ABC can join with ABD to get ABCD
- **Prune based on the monotonic property**
  - ✓ If a candidate is not frequent, we can prune all its supersets in the next level

Apriori Algorithm



Apriori Algorithm

C_k: Candidate itemsets of size k

L_k : frequent itemsets of size k

L_1 = {frequent items};

for (k = 2; L_{k-1} !=∅; k++) **do begin**

  C_k = candidates generated from L_{k-1};

  for each transaction t in database do

    increment the count of all candidates in C_k that are contained in t

  L_k = candidates in C_k with min_support

end

return $\cup_k L_k$;

Apriori Algorithm

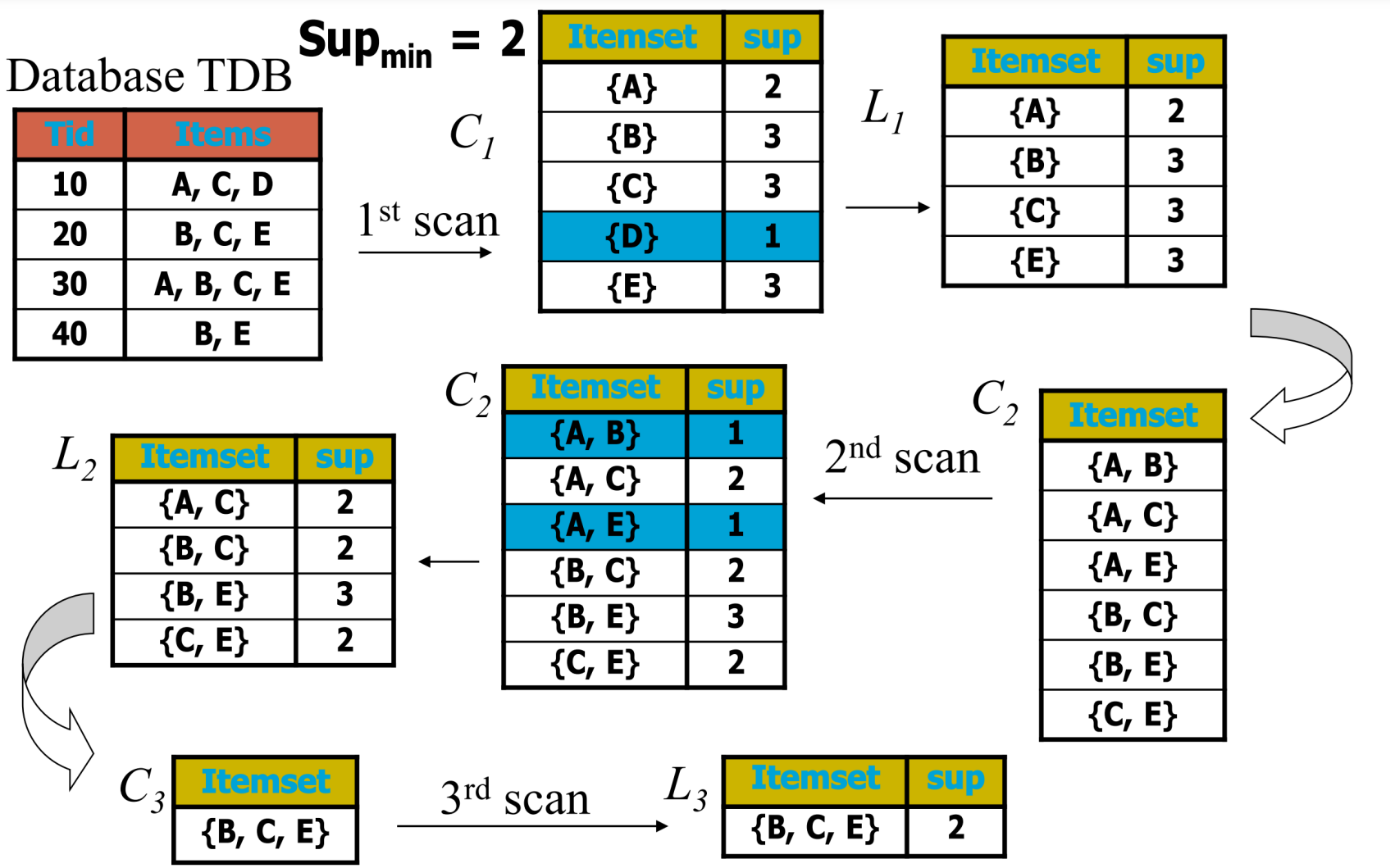


Image from: http://web.cs.ucla.edu/~yzsun/classes/2017Fall_CS145/Slides/13Set_Data.pdf

Association Rule Mining

- **Association Rule X->Y**
  - ✓ If item set X is purchased, Y is also purchased
- **Confidence**
  - ✓ The conditional probability of P(Y|X)
- **{Beer}->{Diaper}**
  - ✓ Support of {Beer} is 3
  - ✓ Confidence of the association rule is 3/3=100%
- **{Diaper}->{Beer}**
  - ✓ Support of {Diaper} is 4
  - ✓ Confidence of the association rule is 3/4=75%
- **Association Rule Mining**
  - ✓ Support of (X,Y) >= minSup, Confidence of X->Y >= minConf

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

Agenda

- Concepts and Applications
- Apriori Algorithm
- Eclat Algorithm
- FP-Tree and FP-Growth
- Summary

Vertical Representation of Transaction Database

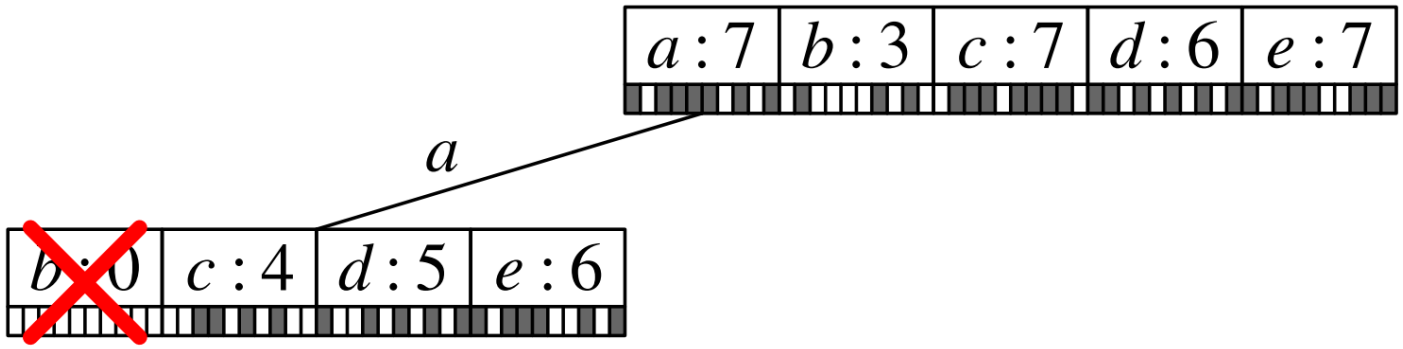
1:	a, d, e
2:	b, c, d
3:	a, c, e
4:	a, c, d, e
5:	a, e
6:	a, c, d
7:	b, c
8:	a, c, d, e
9:	b, c, e
10:	a, d, e

a	b	c	d	e
1	2	2	1	1
3	7	3	2	3
4	9	4	4	4
5		6	6	5
6		7	8	8
8		8	10	9
10		9		10

vertical representation

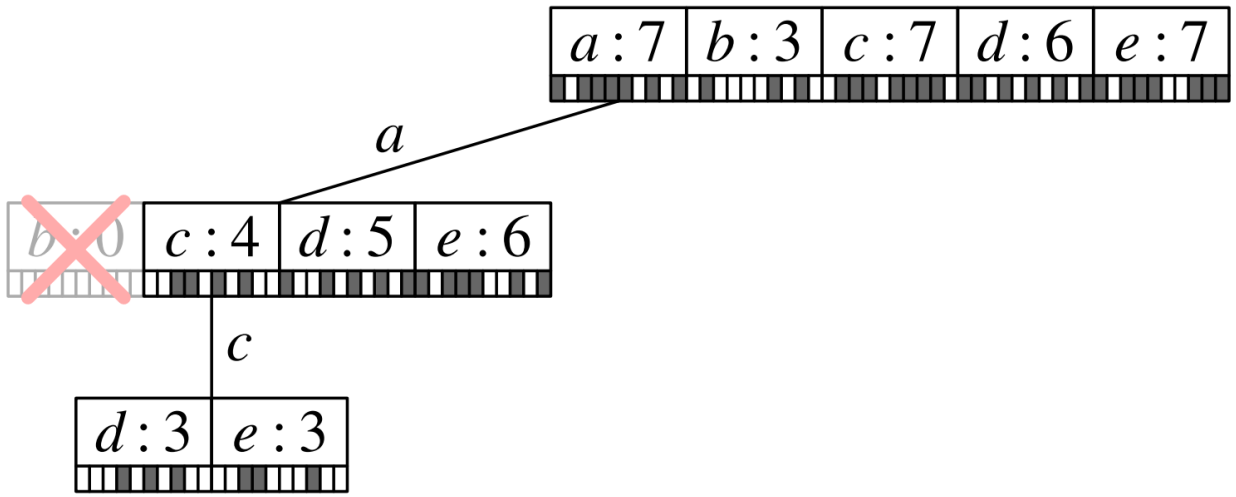
Eclat: Depth-First Search

- Construct the root node using the vertical representation
- Processing patterns start with {a}
  - ✓ Only need to access the 7 transactions containing {a}
  - ✓ Count the frequency for each item in these 7 transactions
  - ✓ {a,b} is pruned



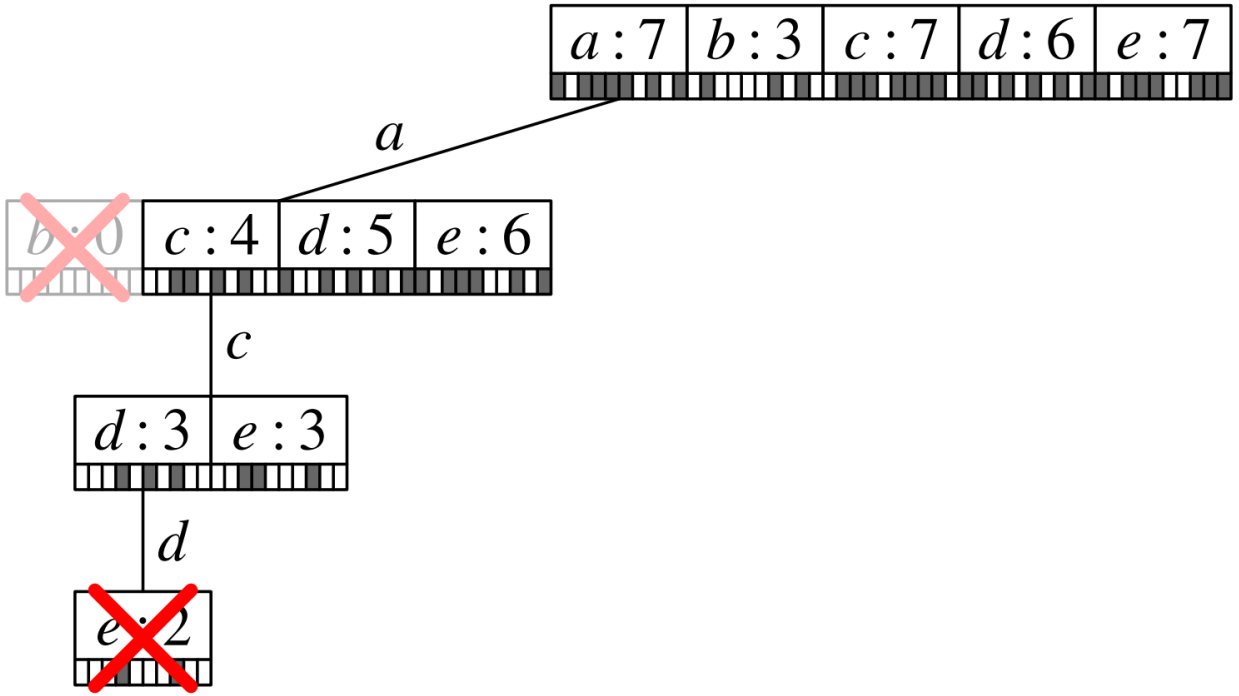
Eclat: Depth-First Search

- **Processing patterns start with {a,c}**
  - ✓ Only need to access the 4 transactions containing {a,c}
  - ✓ Count the frequency for d and e in these 4 transactions



Eclat: Depth-First Search

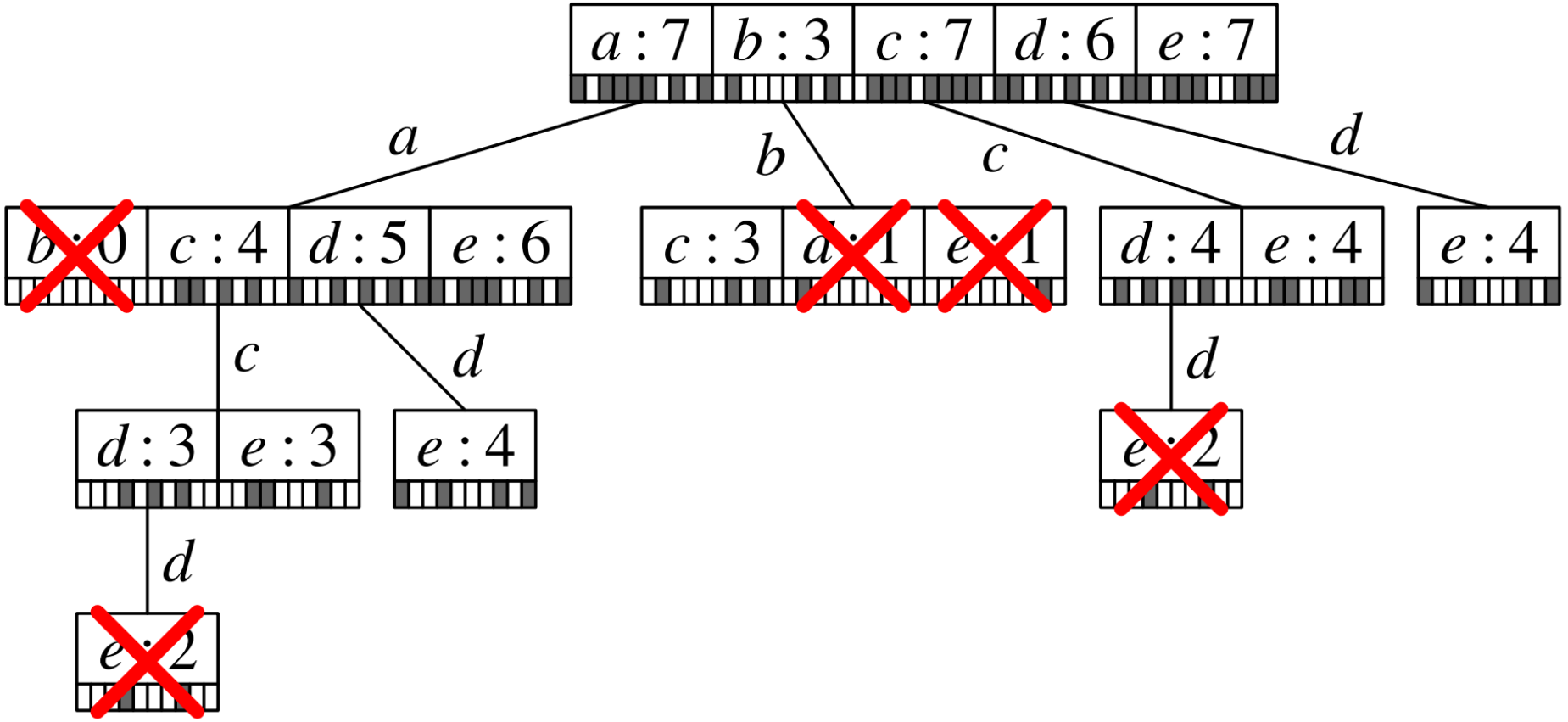
- **Processing patterns start with {a,c,d}**
  - ✓ Only need to access the 3 transactions containing {a,c,d}
  - ✓ Count the frequency for e in these 3 transactions
  - ✓ {a,c,d,e} is not frequent and pruned



1:	a, d, e
2:	b, c, d
3:	a, c, e
4:	a, c, d, e
5:	a, e
6:	a, c, d
7:	b, c
8:	a, c, d, e
9:	b, c, e
10:	a, d, e

Eclat: Depth-First Search

- The process is repeated and all the cases are considered



Eclat v.s. Apriori

- The Eclat algorithm does not involve in the repeated scanning of the data in order to calculate the individual support values
- Eclat algorithm scans the currently generated dataset unlike Apriori which scans the original dataset
- The Eclat algorithm is naturally faster compared to the Apriori algorithm while the size of data set is small or medium
- In case of large dataset there is a chance that Apriori performs ore faster, because intermediate Tid sets which are created in Eclat algorithm consumes more space in memory than Apriori

Conclusion from: https://medium.com/machine-learning-and-artificial-intelligence/3-4-association-rule-mining-using-eclat-algorithm-b6e50aab2147

Agenda

- Concepts and Applications
- Apriori Algorithm
- Eclat Algorithm
- FP-Tree and FP-Growth
- Summary

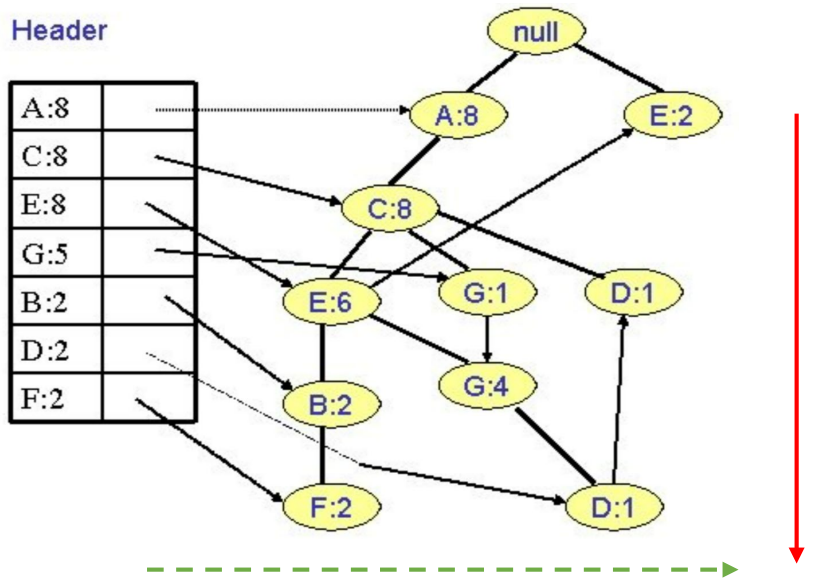
Frequent Pattern-Growth Approach

- **The FPGrowth Approach (J. Han, J. Pei, and Y. Yin, SIGMOD' 00)**
  - ✓ Avoid repeated scan of the database
  - ✓ Stage 1: Represent the database using a compressed representation, called FP-Tree.
  - ✓ Stage 2: Once an FP-tree has been constructed, it uses a divide-and-conquer approach to mine the frequent itemsets

What is FP-Tree?

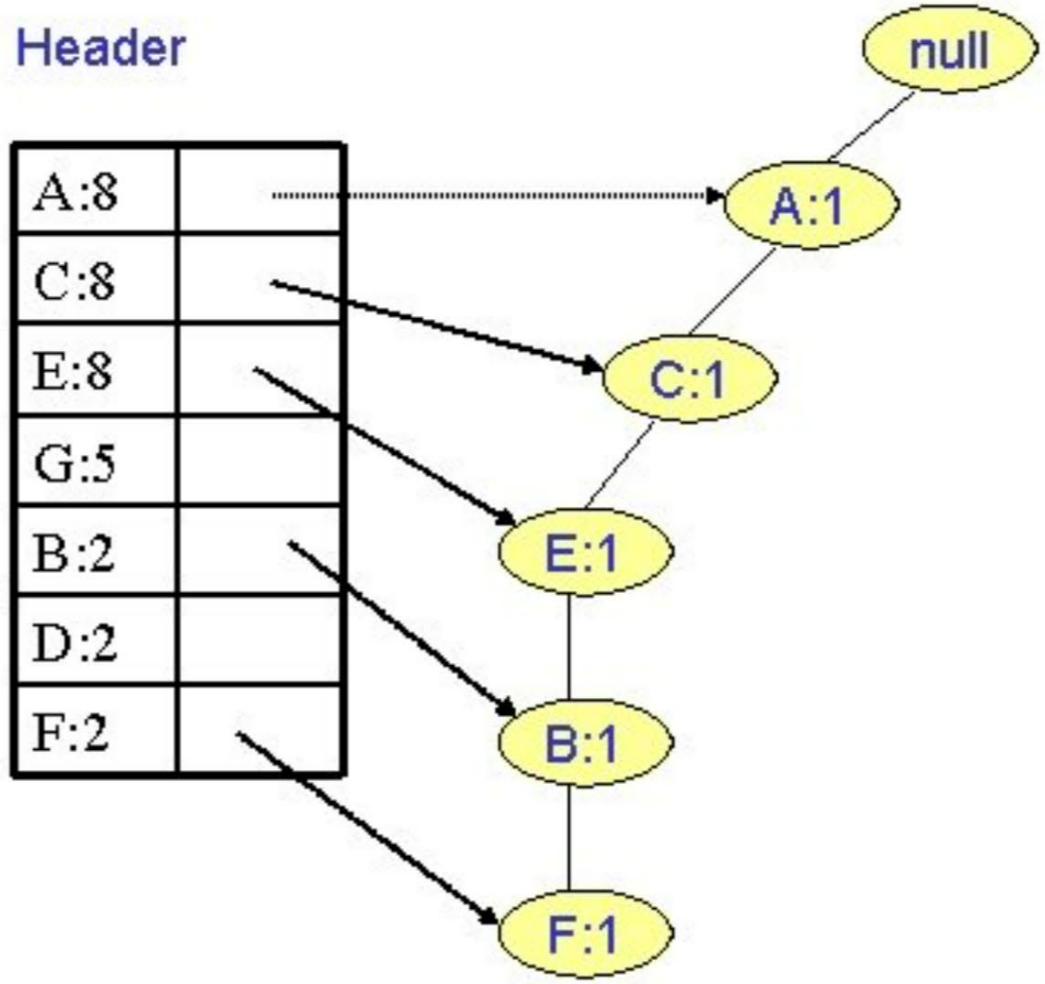
- **FP-tree is a compressed representation of the transaction database**
  - ✓ Each transaction is mapped onto a path in the tree
  - ✓ Each node contains an item and the support count corresponding to the number of transactions with the prefix corresponding to the path from root
  - ✓ Nodes having the same item label are cross-linked: this helps find the frequent item sets ending with a particular item

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



How to Construct FP-Tree?

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



1. **Scan DB once, find frequent 1-itemset (single item pattern)**
2. **Sort frequent items in frequency descending order, f-list**
3. **Scan DB again, construct FP-tree**

Example source: https://slidetodoc.com/association-analysis-3-alternative-methods-for-frequent-itemset/

How to Construct FP-Tree?

A C E B F

A C G

E

A C E G D

A C E G

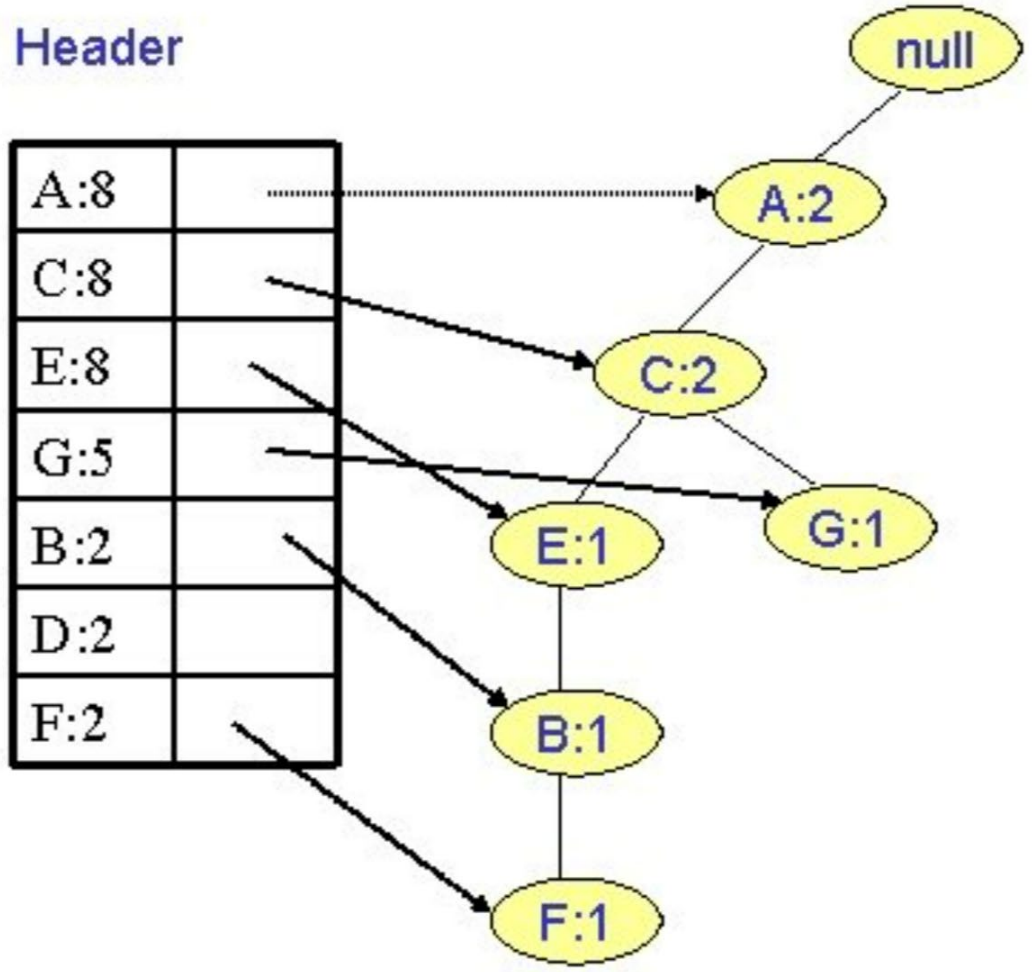
E

A C E B F

A C D

A C E G

A C E G



How to Construct FP-Tree?

A C E B F

A C G

E

A C E G D

A C E G

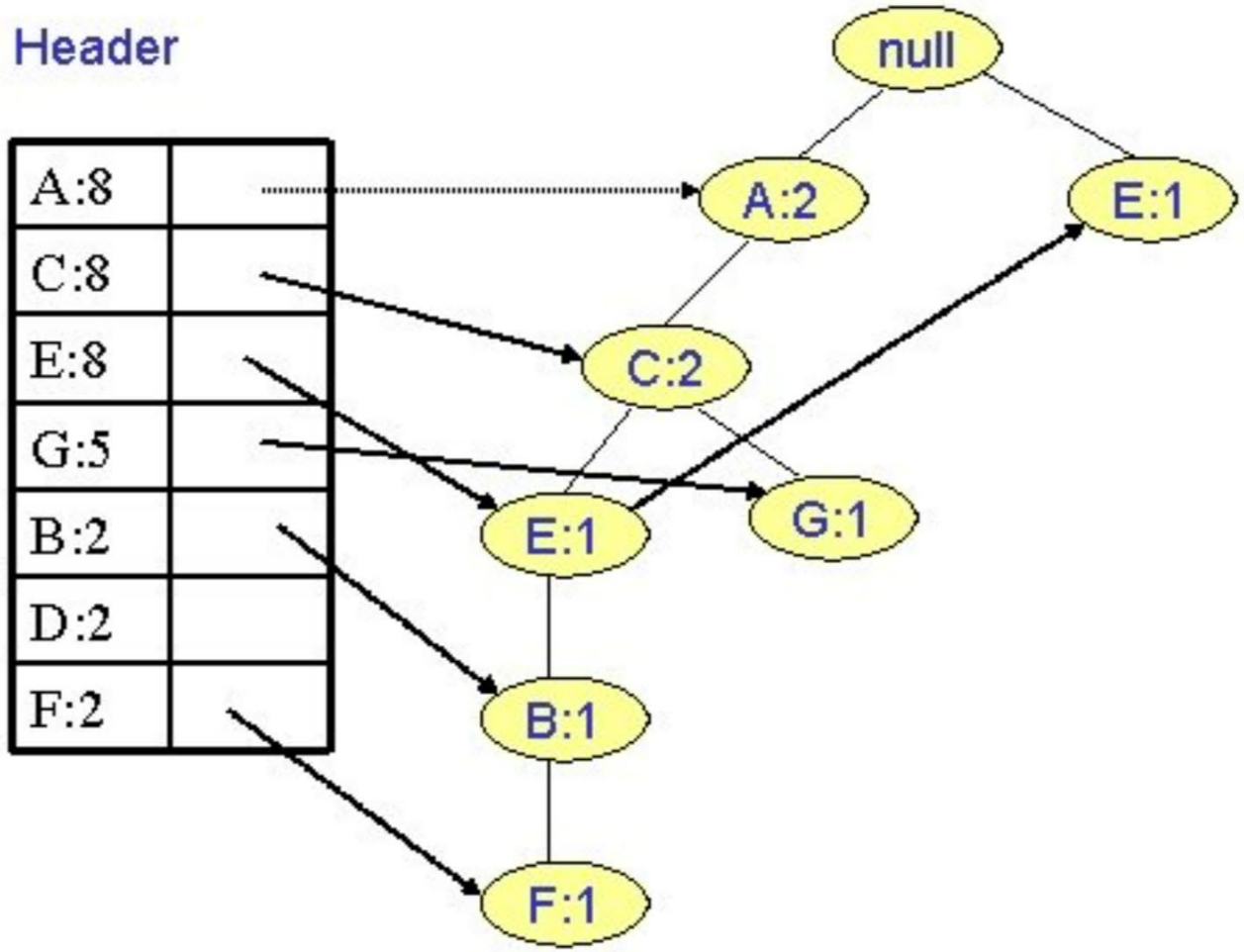
E

A C E B F

A C D

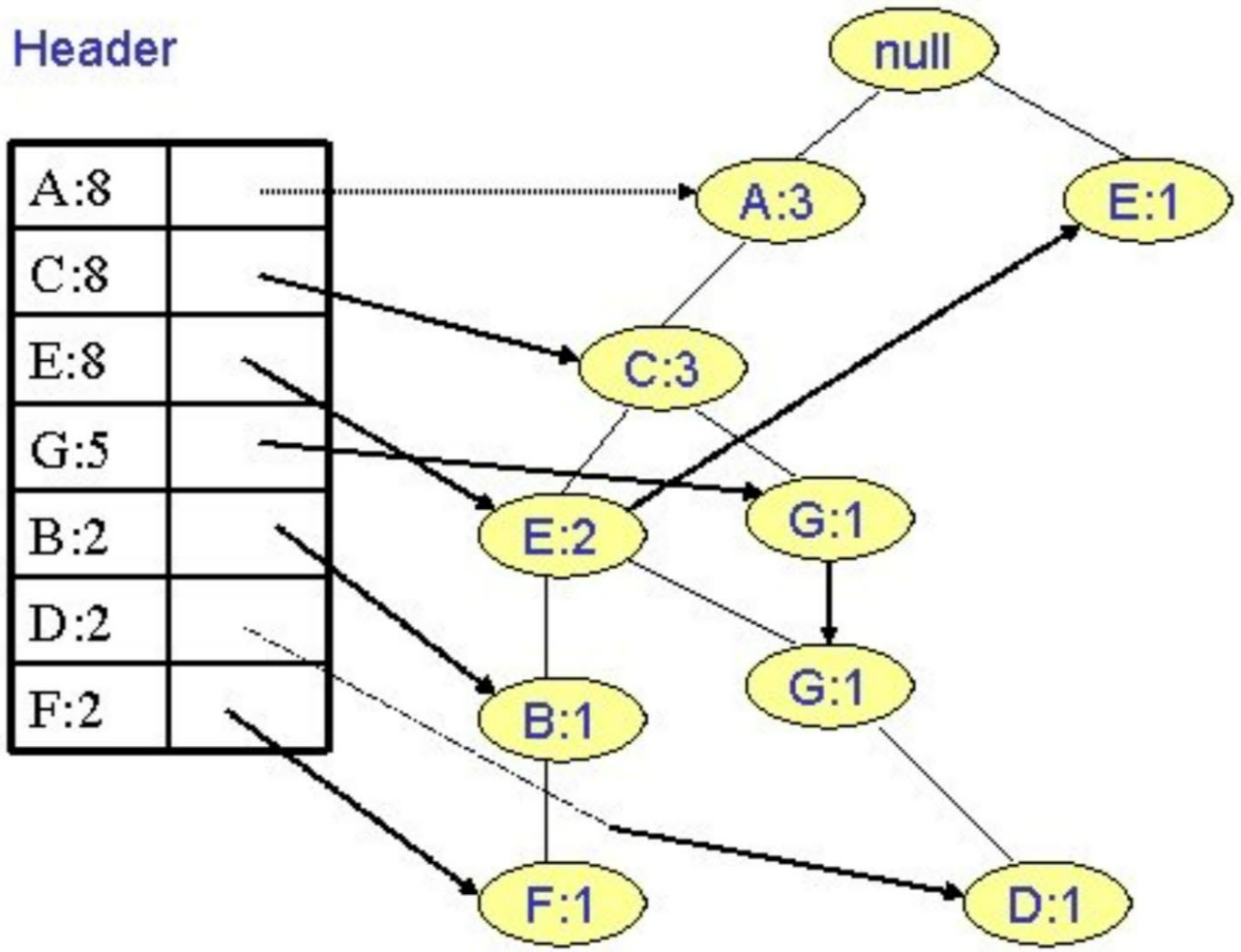
A C E G

A C E G



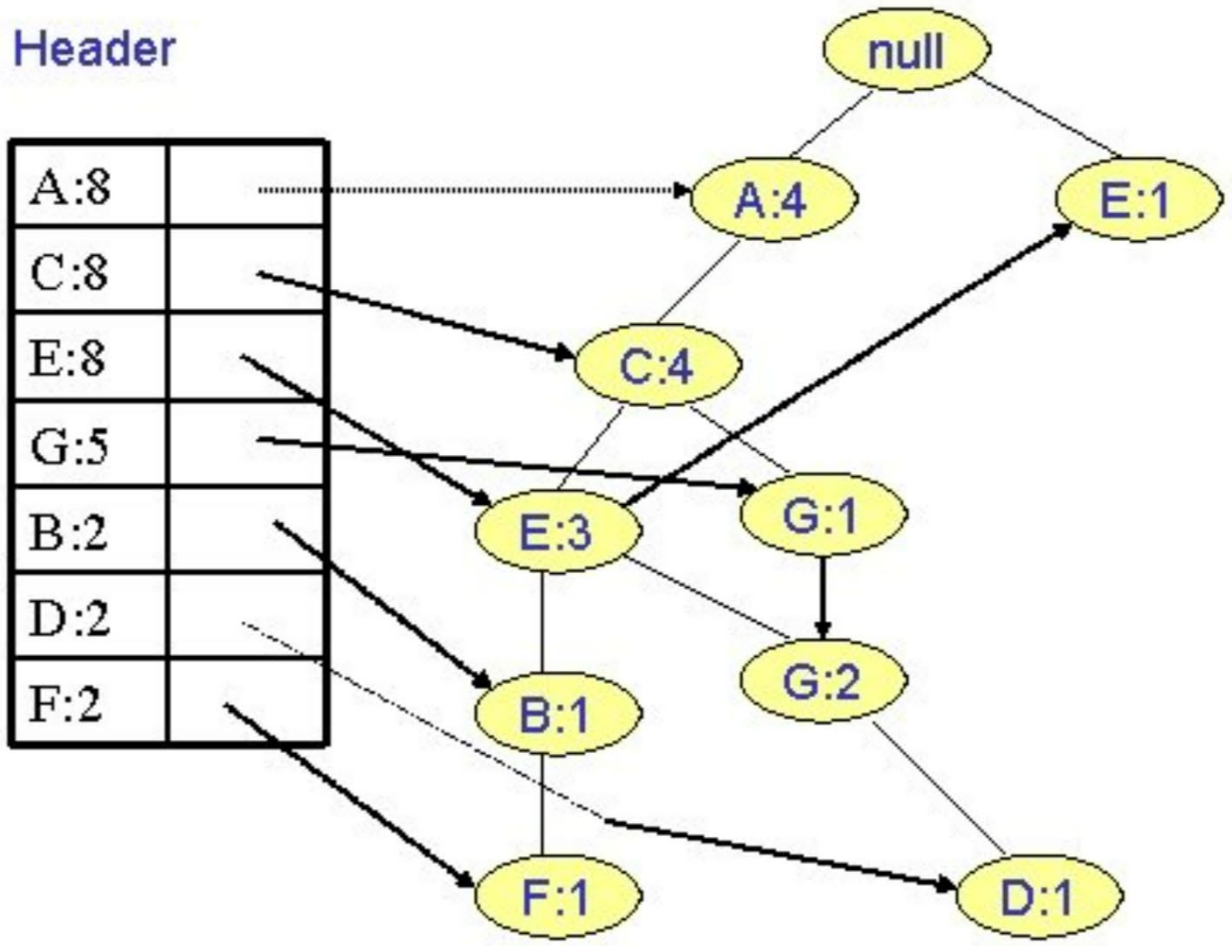
How to Construct FP-Tree?

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



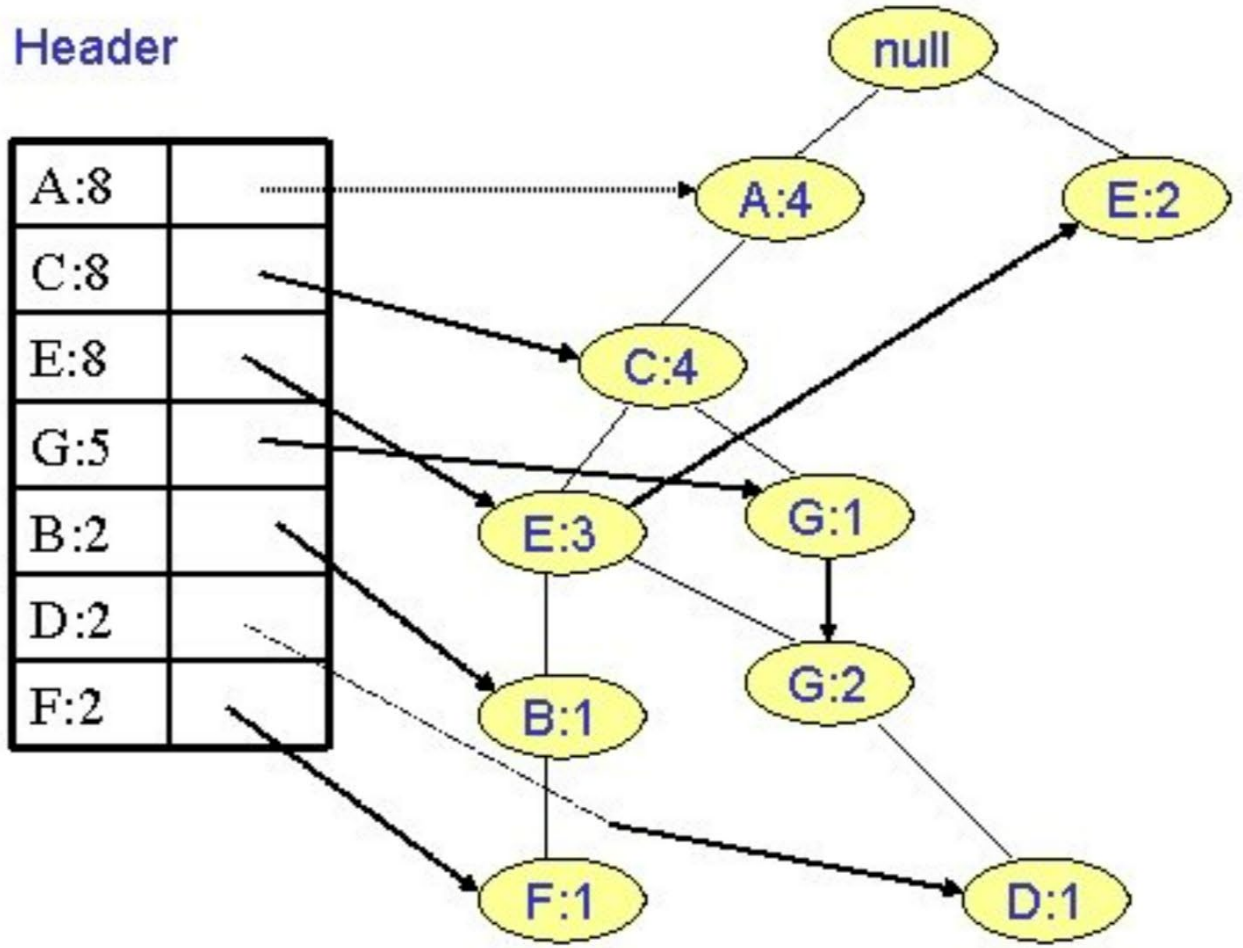
How to Construct FP-Tree?

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



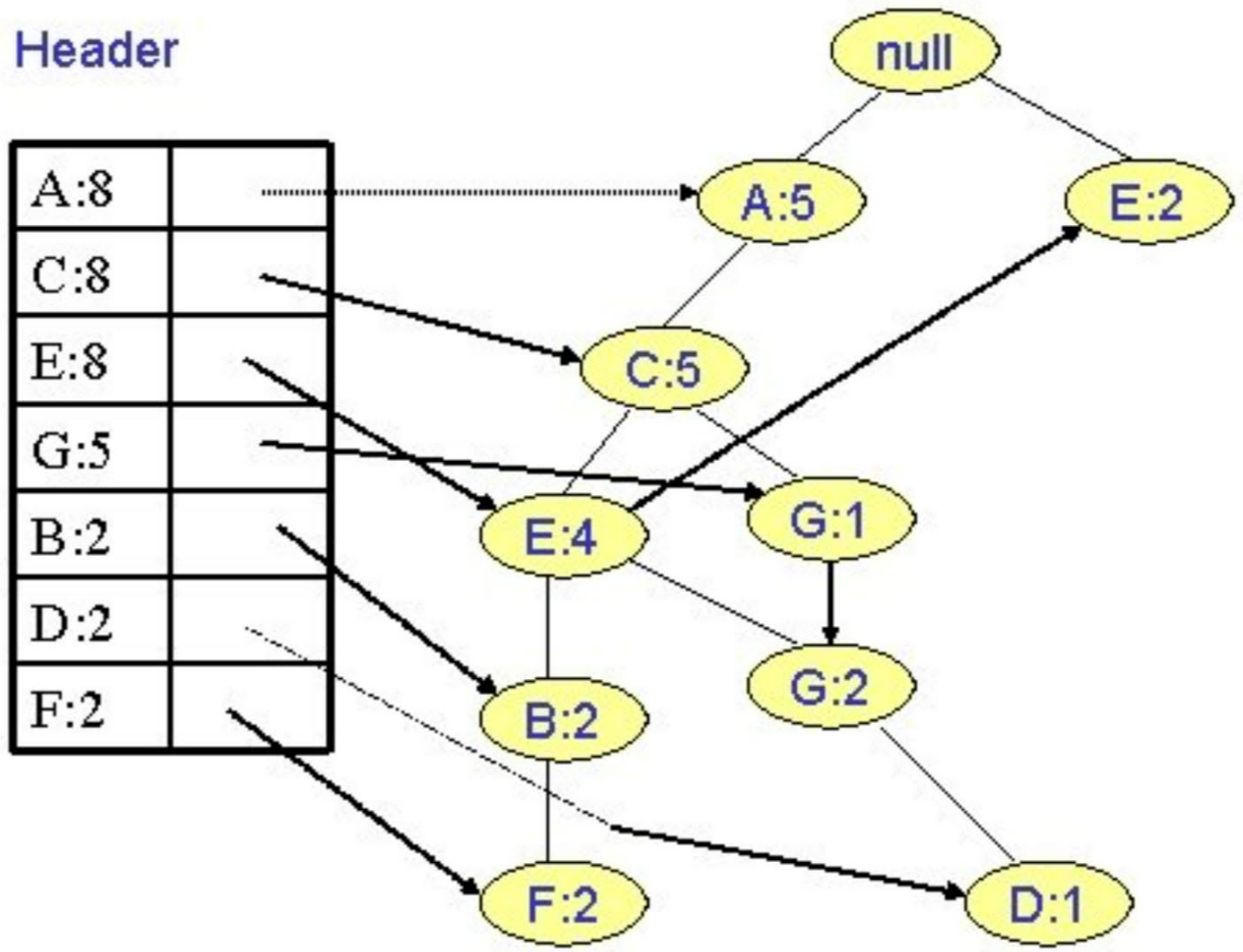
How to Construct FP-Tree?

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



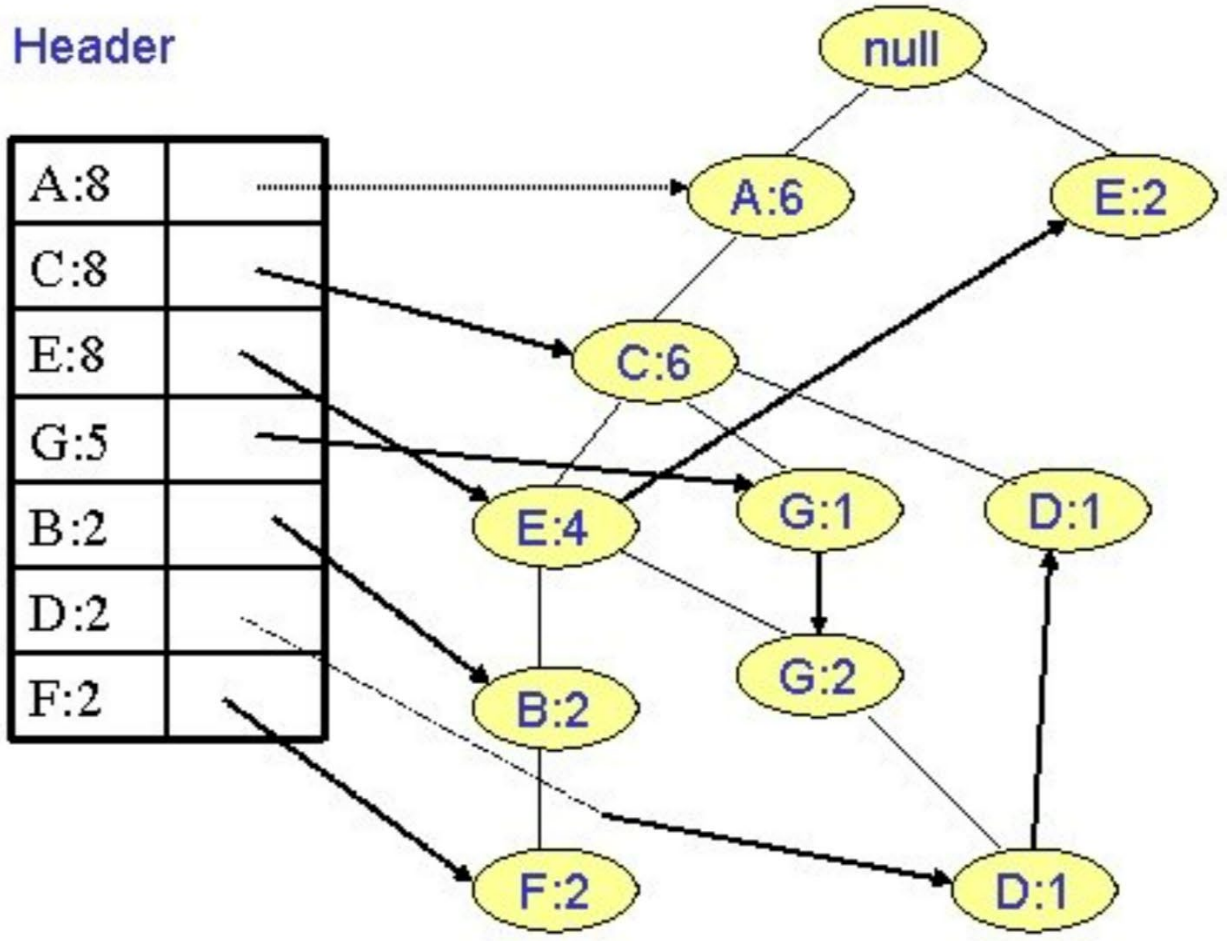
How to Construct FP-Tree?

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



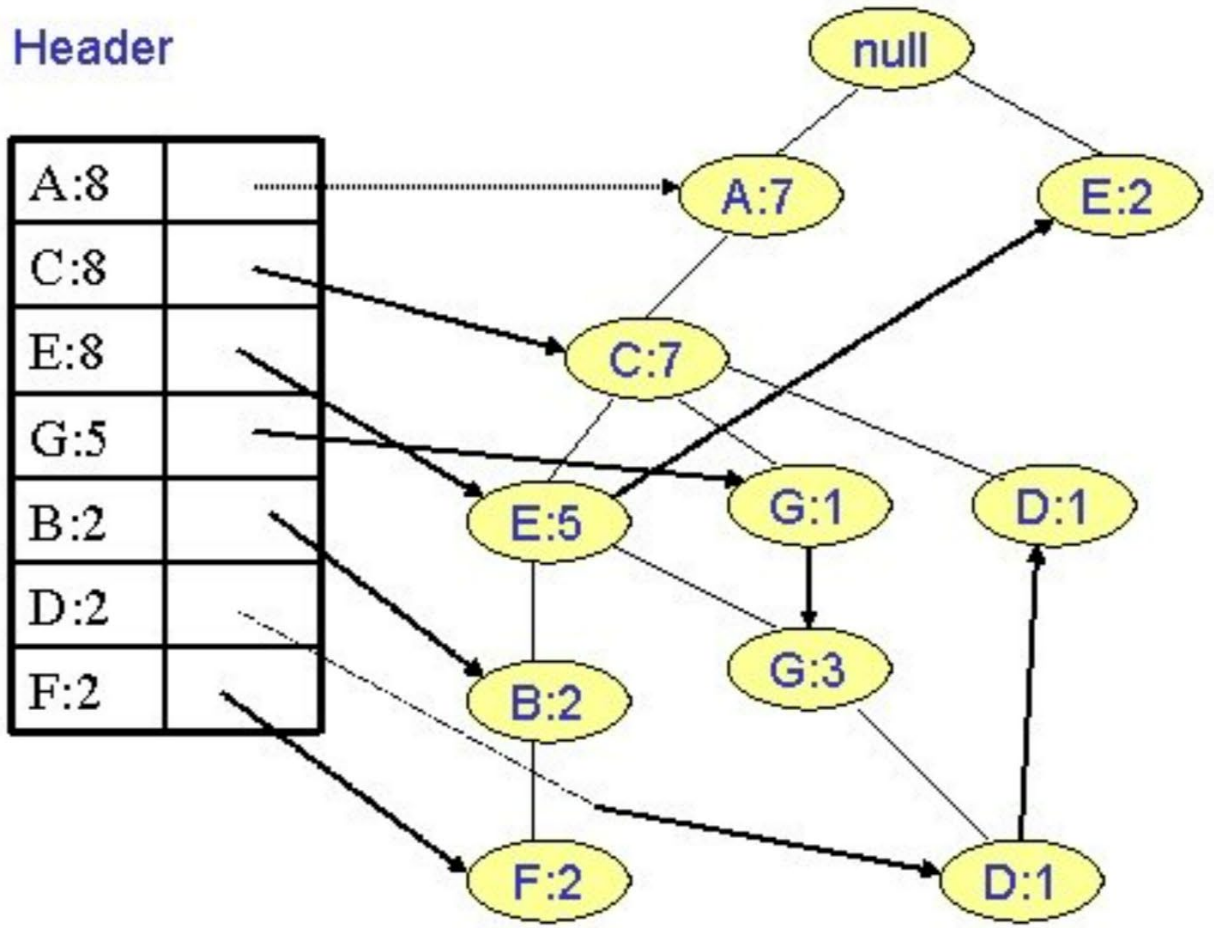
How to Construct FP-Tree?

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



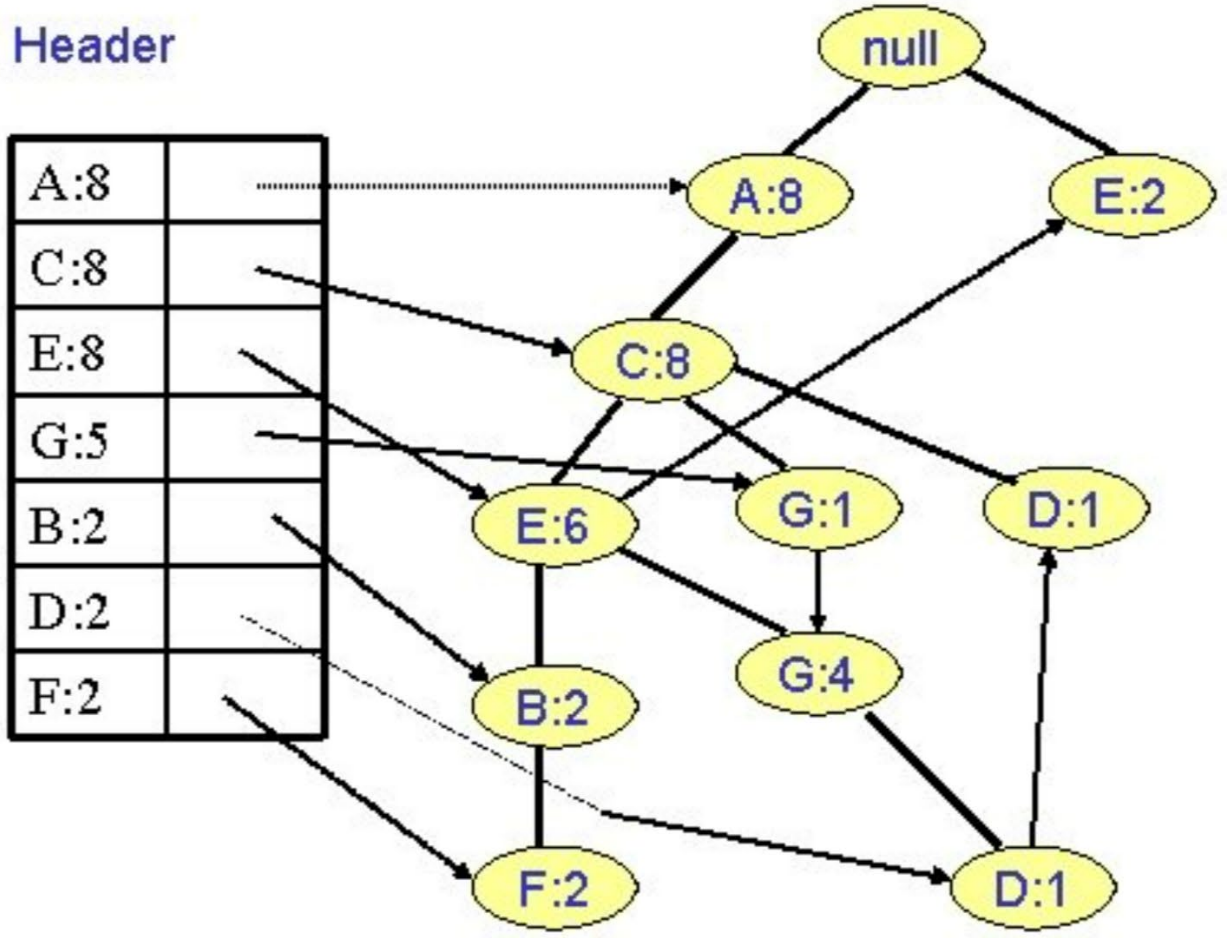
How to Construct FP-Tree?

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



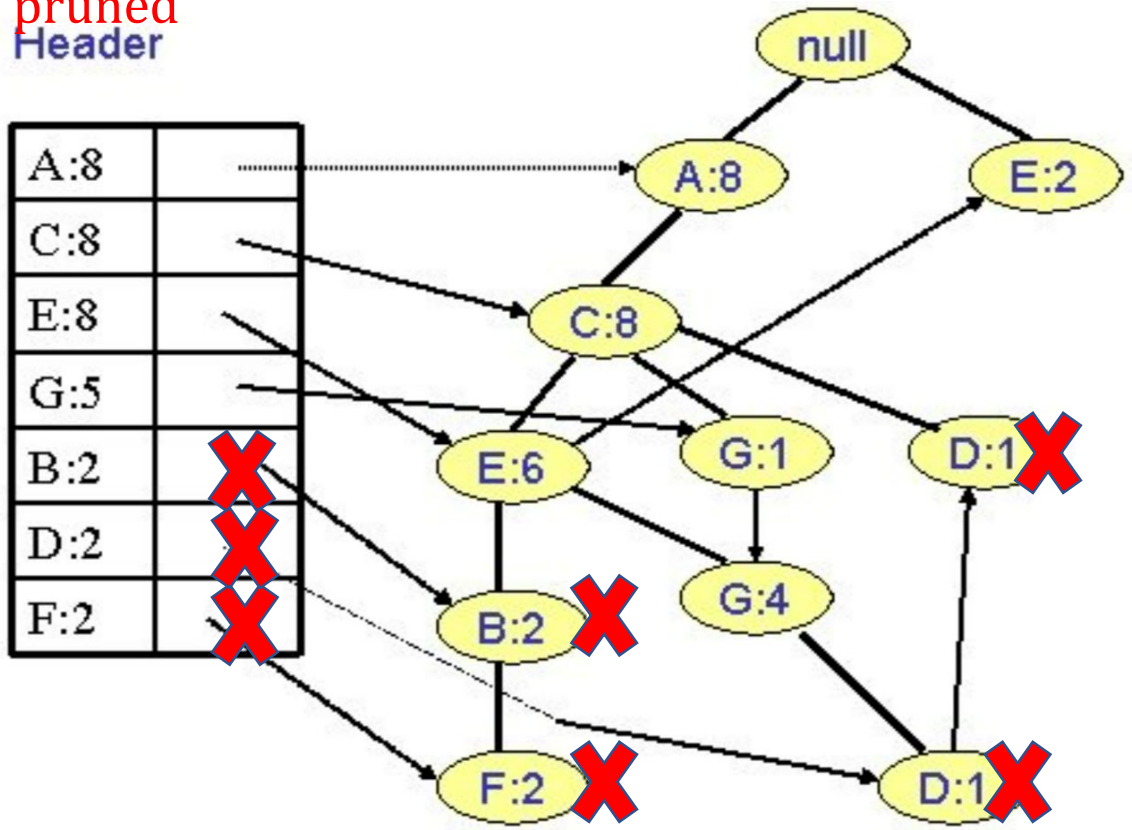
How to Construct FP-Tree?

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



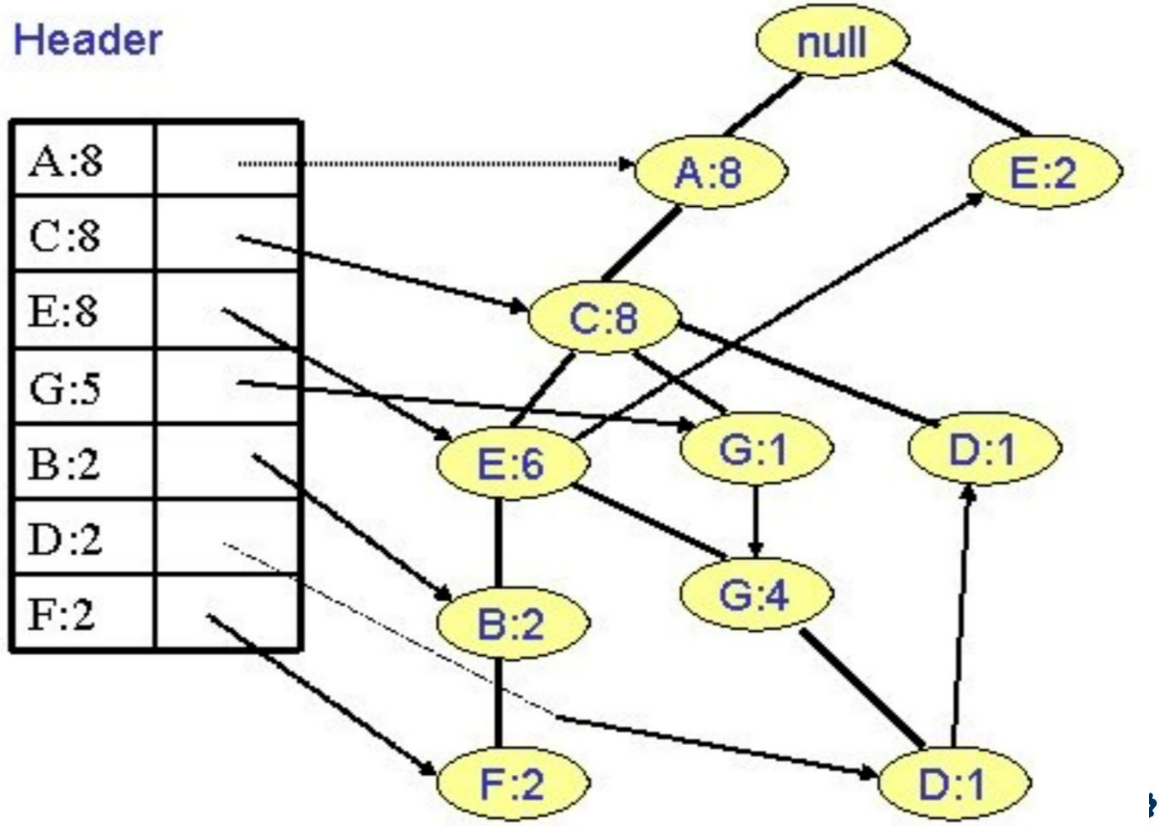
How to Mine Frequent Patterns From FP-Tree?

- Let minSup=5
  - ✓ F-list: A-C-E-G-B-D-F
  - ✓ Candidates that end with B, D, F are pruned



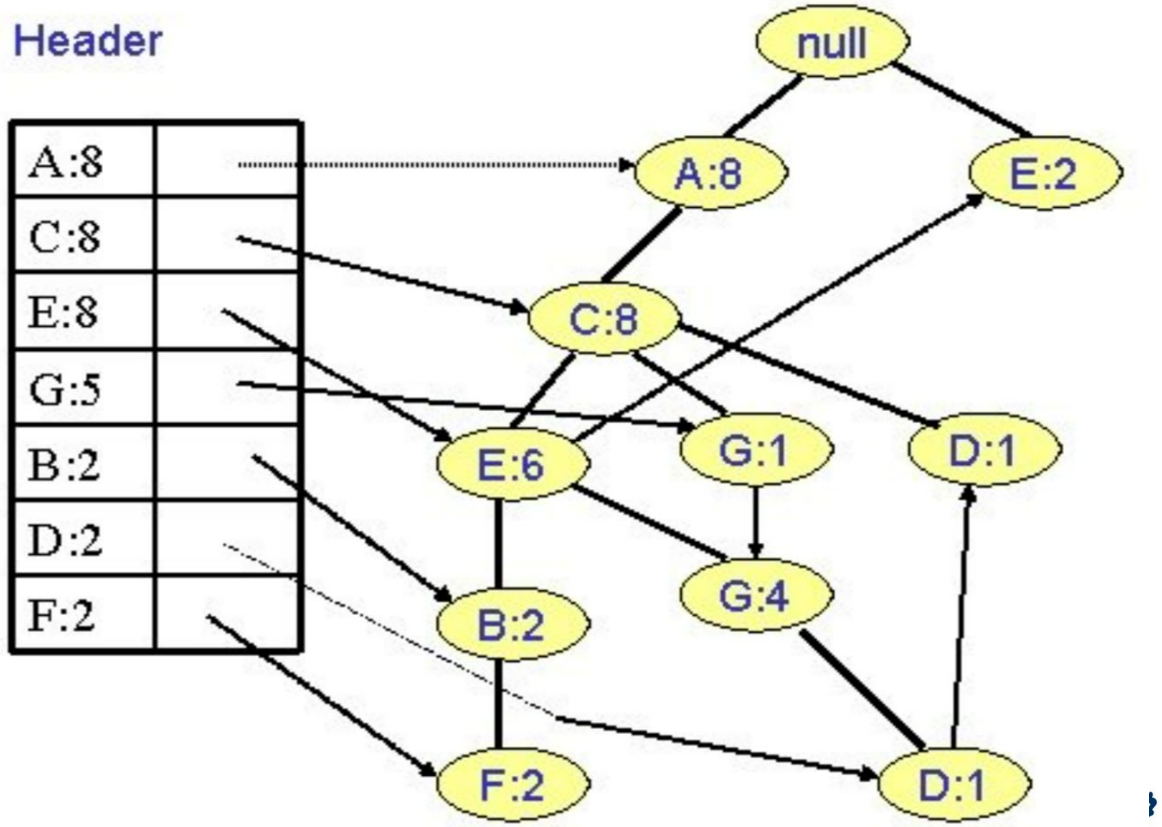
How to Mine Frequent Patterns From FP-Tree?

- **Let minSup=5**
  - ✓ Patterns end with G
  - ✓ Conditional pattern: {A:4, C:4, E:4} and {A:1, C:1}
  - ✓ Frequent patterns:
    - {G}:5
    - {A,G}:5
    - {C,G}:5
    - {A,C,G}:5



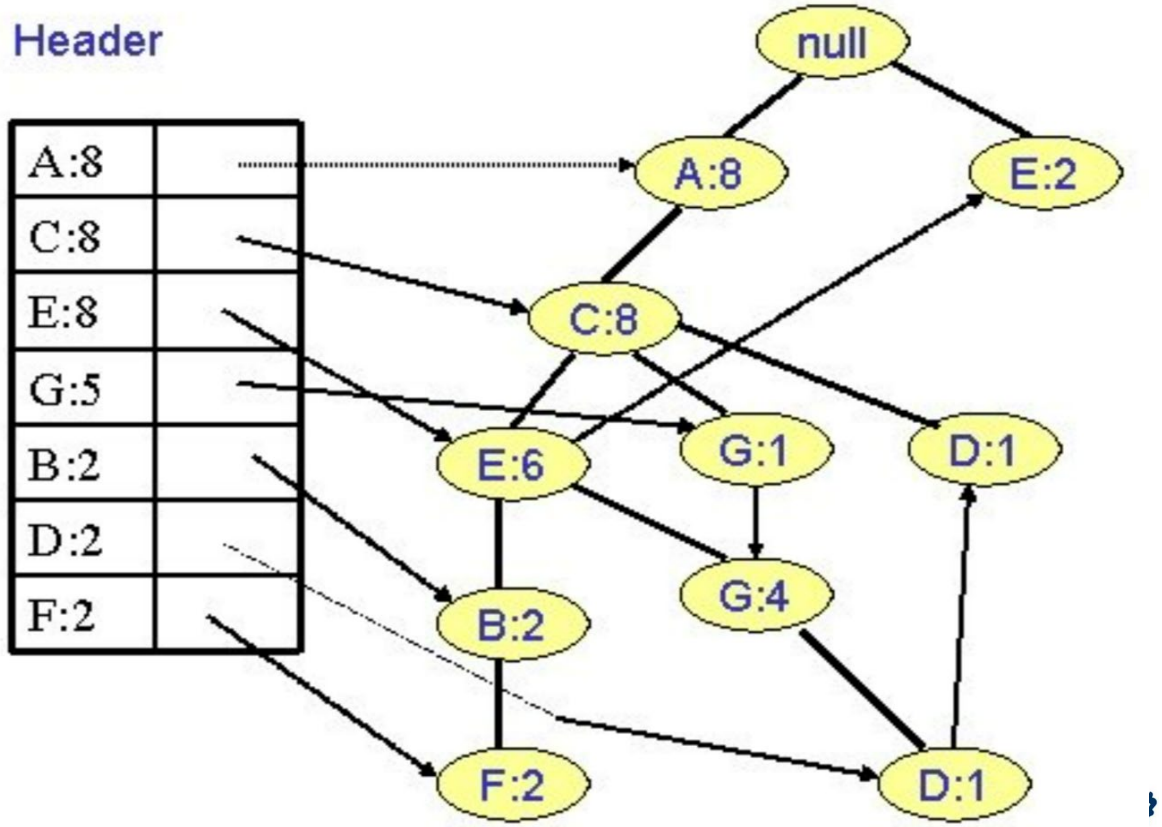
How to Mine Frequent Patterns From FP-Tree?

- **Let minSup=5**
  - ✓ Patterns end with E
  - ✓ Conditional pattern: {A:6, C:6}
  - ✓ Frequent patterns:
    - {E}:8
    - {A,E}:6
    - {C,E}:6
    - {A,C,E}:6



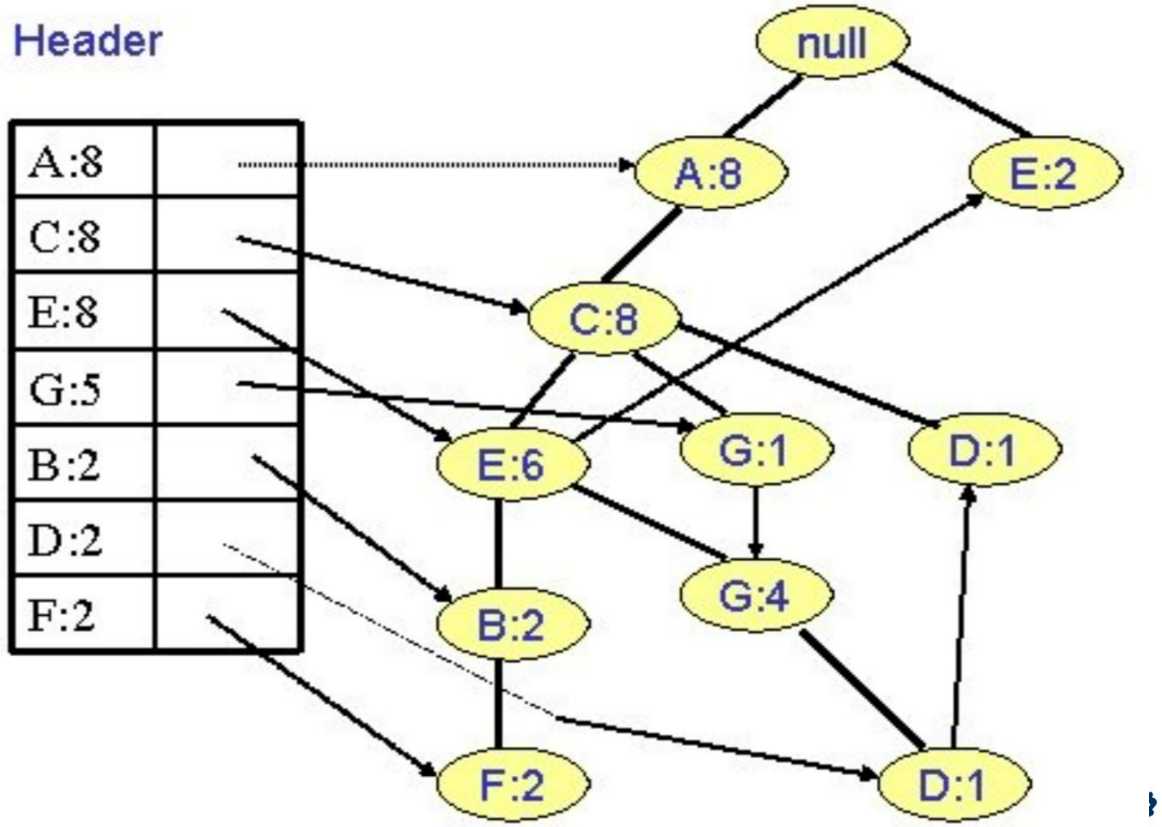
How to Mine Frequent Patterns From FP-Tree?

- **Let minSup=5**
  - ✓ Patterns end with C
  - ✓ Conditional pattern: {A:8}
  - ✓ Frequent patterns:
    - {C}:8
    - {A,C}:8

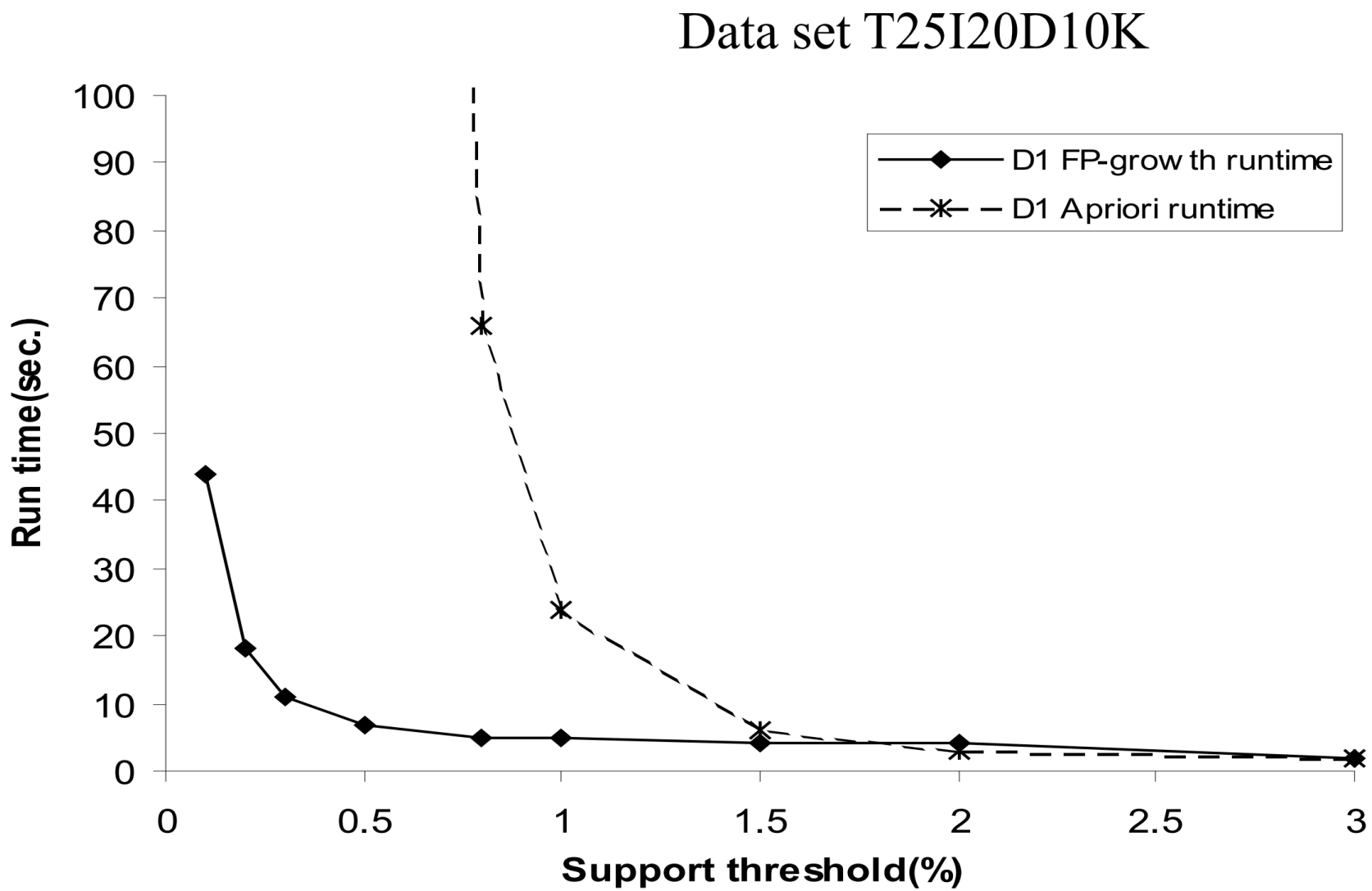


How to Mine Frequent Patterns From FP-Tree?

- **Let minSup=5**
  - ✓ Patterns end with A
  - ✓ Conditional pattern: null
  - ✓ Frequent patterns:
    - {A}:8



Performance of FP-tree and Apriori



Agenda

- Concepts and Applications
- Apriori Algorithm
- Eclat Algorithm
- FP-Tree
- Summary