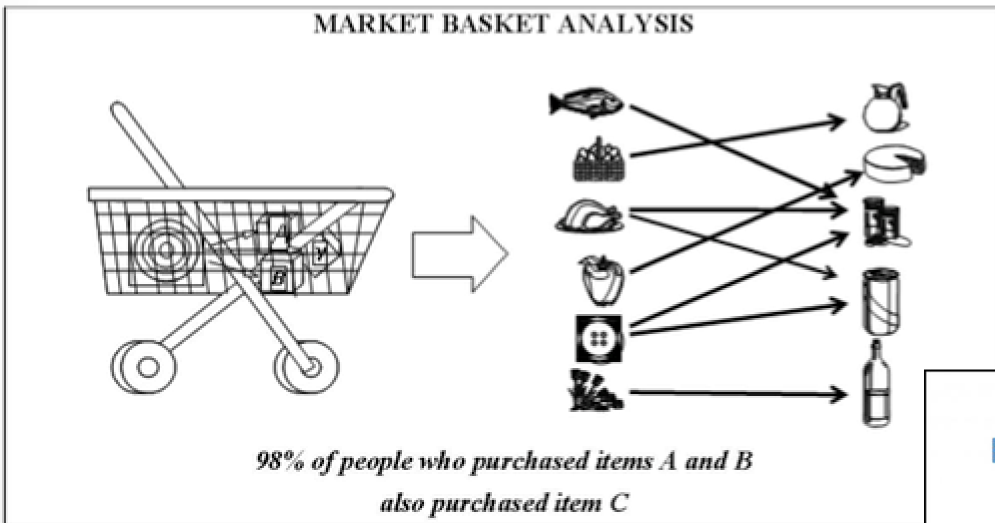
CSX4202/ITX4202: Data Mining
Lecture 9

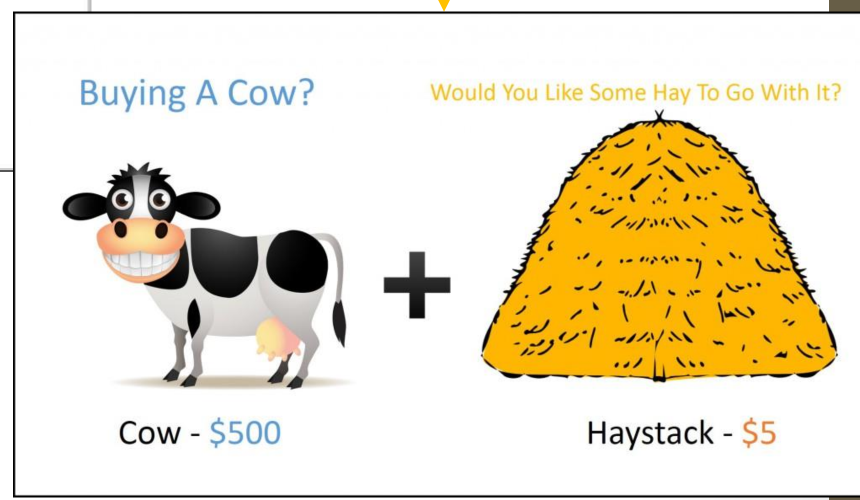
Asst. Prof. Dr. Rachsuda Setthawong
Computer Science Department
Assumption University

What s Association Rule Mining?

- A rule-based machine learning method for ***discovering interesting relations between variables*** in large databases.



The technique is applied to increase business's revenue.



https://en.wikipedia.org/wiki/Association_rule_learning

Image sources: https://vwo.com/blog/use-upsell-cross-sell/
https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/42541/versions/3/screenshot.jpg

Application: Cross Sell

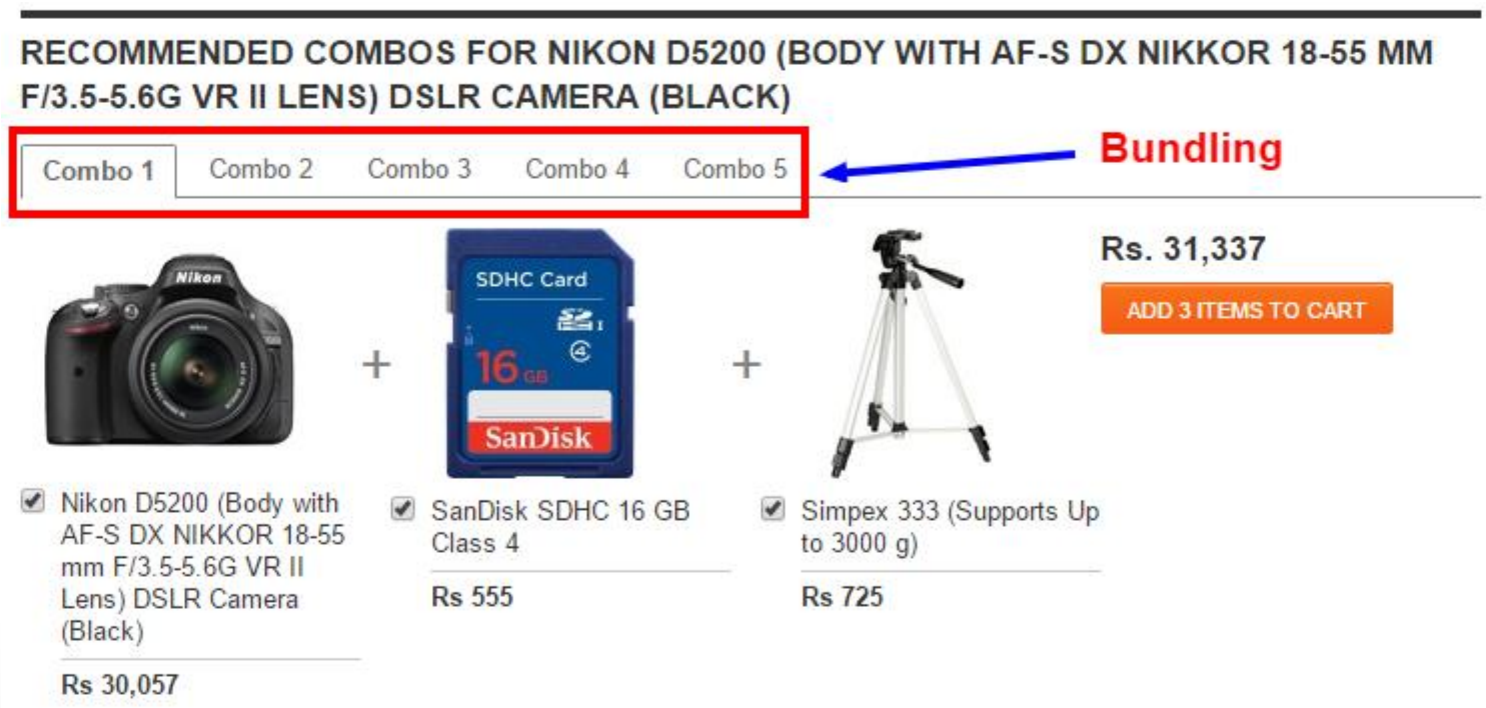


Image source: https://vwo.com/blog/use-upsell-cross-sell/

Application: Recommender System

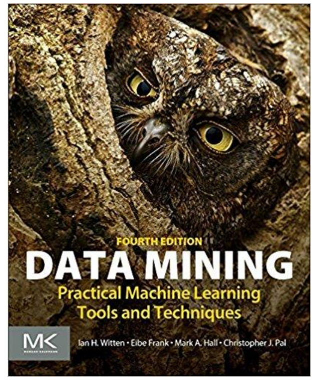


Image source: https://www.amazon.com

A Concept of Association Rule Mining

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

Market-Basket transactions

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Association Rules

{Diaper} $\rightarrow$ {Beer}
{Milk, Bread} $\rightarrow$ {Eggs,Coke}
{Beer, Bread} $\rightarrow$ {Milk}

Implication means co-occurrence, not causality!

Definition: Frequent Itemset

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
    - E.g. σ({Milk, Bread,Diaper}) = 2
- **Support (s)**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is ***greater than or equal to*** a *minsup* threshold

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Asst. Prof. Dr. Rachsuda Setthawong

Definition: Association Rule

- **Association Rule**
  - An implication expression of the form
    $X \rightarrow Y$,

    where X and Y are disjoint itemsets
  - Example:
    {Milk, Diaper} → {Beer}

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Rule Evaluation Metrics**
  - Support (s) $s = \frac{\sigma(x \cap y)}{N}$
    - Fraction of transactions that contain **both** X and Y
  - Confidence (c) $c = \frac{\sigma(x \cap y)}{\sigma(x)}$
    - Measures how often items in Y **appear in** transactions that contain X

Example:

$$\overset{X \rightarrow Y}{\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Asst. Prof. Dr. Rachsuda Setthawong

Association Rule Mining Task

- Given a set of transactions T,
  - the goal of association rule mining is to find all rules having
    - support ≥ *minsup* threshold
    - confidence ≥ *minconf* threshold
- How to find all rules?

How to find all rules? - 1

- Brute-force approach:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Prune rules that fail the *minsup* and *minconf* thresholds

$\Rightarrow$ Computationally prohibitive (2^n where n denotes no. of items)!

How to find all rules? - 2

$$s = \frac{\sigma(x \cap y)}{N} \qquad c = \frac{\sigma(x \cap y)}{\sigma(x)}$$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

Observations:

- All the above rules are binary partitions of the same itemset: {Milk, Diaper, Beer}
- Rules originating from the same itemset have **identical support** but can have **different confidence**

Since any rule will satisfy the 2 conditions:

1) All items in the rule must be frequent (**s** ≥ minsup)

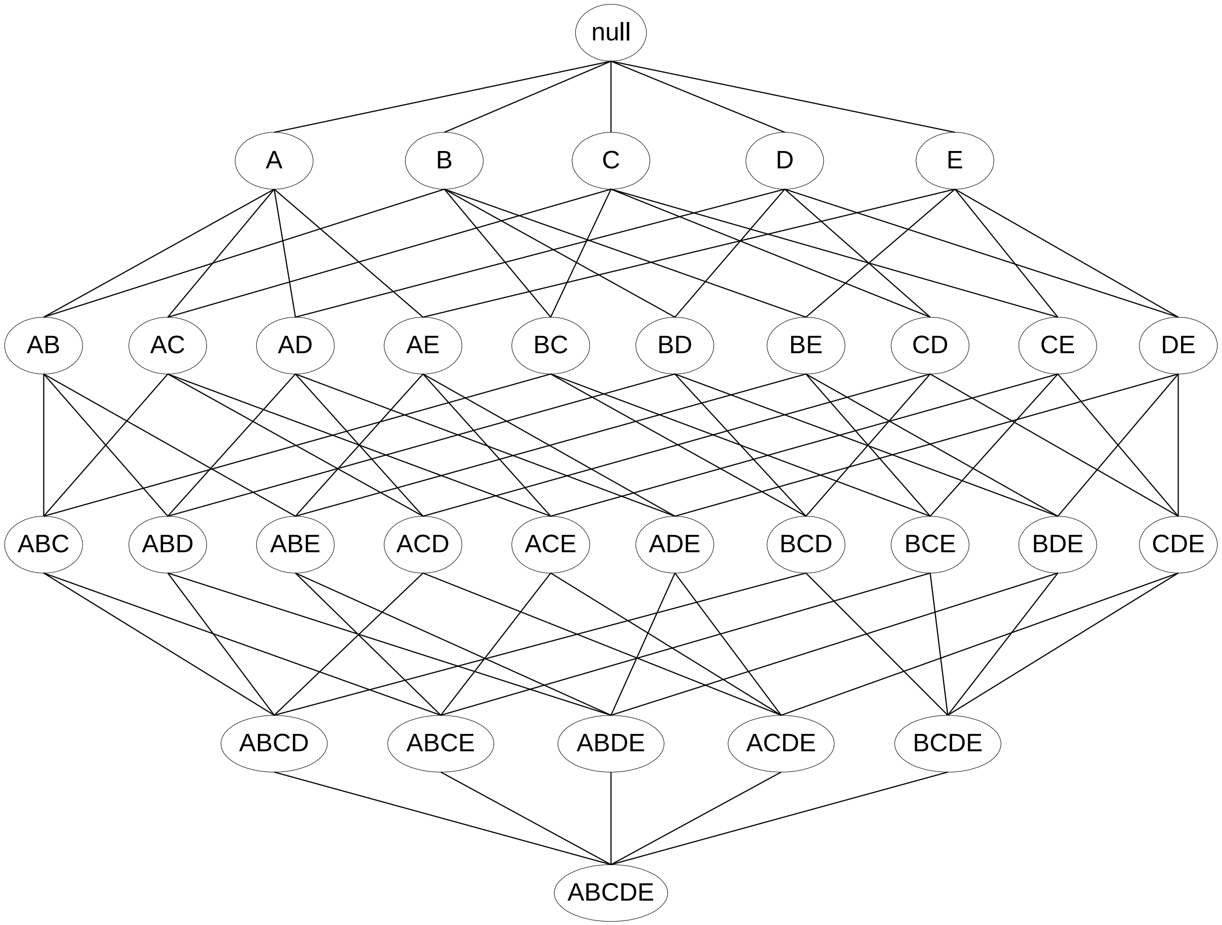
2) The rule must have good enough confidence (**c** ≥ minconf)

→ **decouple** the support and confidence **requirements**

Mining Association Rules

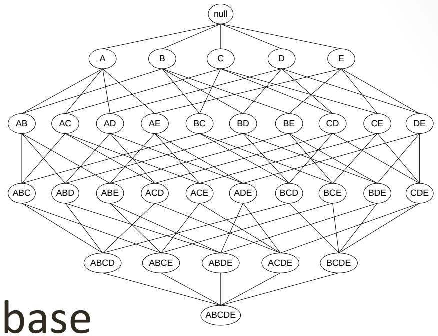
- Two-step approach:
  1. Frequent Itemset Generation
     – Generate all itemsets whose **support** $\geq$ **minsup**
  2. Rule Generation
     – Generate **high confidence** rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

Frequent Itemset Generation - 1



- Frequent itemset generation is computationally expensive.
  - Given d items, there are 2^d possible candidate itemsets

Frequent Itemset Generation - 2



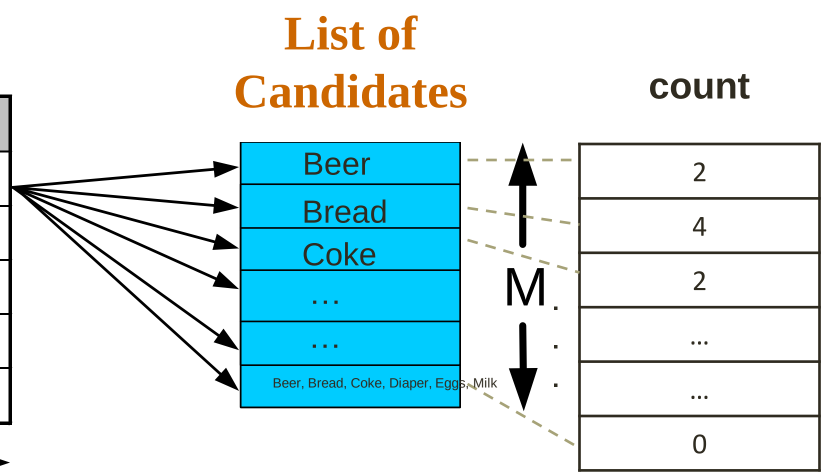
- Brute-force approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database

Transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

N

w



  - Match each transaction against every candidate
  - Complexity ~ O(NMw) => Expensive since $M = 2^d$!!!

Asst. Prof. Dr. Rachsuda Setthawong

Candidate Generation: Brute-force method

Items

Item
Beer
Bread
Cola
Diapers
Milk
Eggs

Candidate Generation

Itemset
{Beer, Bread, Cola}
{Beer, Bread, Diapers}
{Beer, Bread, Milk}
{Beer, Bread, Eggs}
{Beer, Cola, Diapers}
{Beer, Cola, Milk}
{Beer, Cola, Eggs}
{Beer, Diapers, Milk}
{Beer, Diapers, Eggs}
{Beer, Milk, Eggs}
{Bread, Cola, Diapers}
{Bread, Cola, Milk}
{Bread, Cola, Eggs}
{Bread, Diapers, Milk}
{Bread, Diapers, Eggs}
{Bread, Milk, Eggs}
{Cola, Diapers, Milk}
{Cola, Diapers, Eggs}
{Cola, Milk, Eggs}
{Diapers, Milk, Eggs}

Candidate Pruning

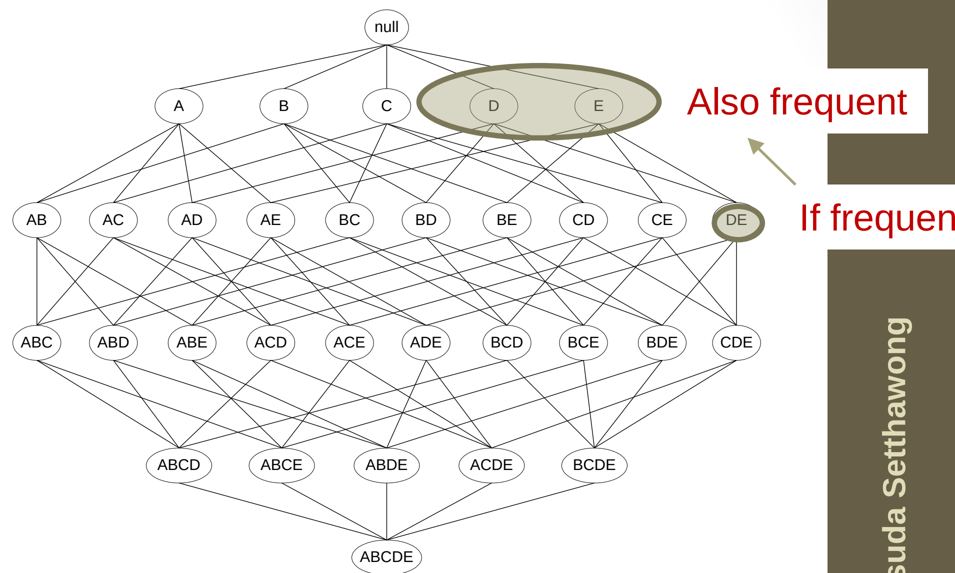
Itemset
{Bread, Diapers, Milk}

Figure 6.6. A brute-force method for generating candidate 3-itemsets.

Frequent Itemset Generation Strategies

- Strategy 1: Reduce the number of candidates (M)
  - Complete search: $M=2^d$
  - Use pruning techniques to reduce M
- Strategy 2: Reduce the number of comparisons (NM)
  - Use efficient data structures to store the candidates or transactions
  - No need to match every candidate against every transaction

Strategy 1: Reducing Number of Candidates



- Apriori principle:
  - *If an **itemset** is **frequent** then*

    ***all of its subsets** must also be **frequent**.*

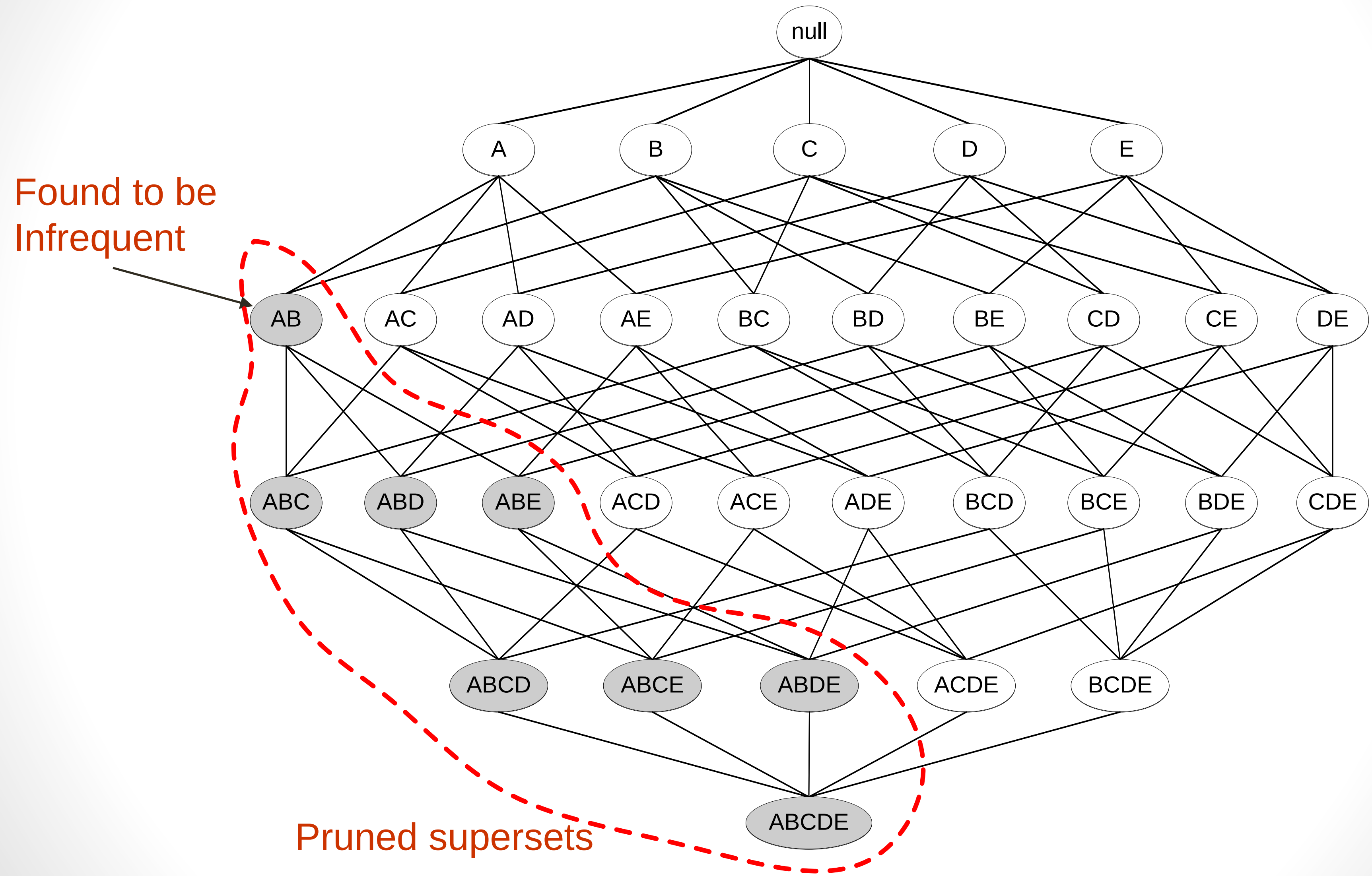
- Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

  - Support of an itemset *never exceeds* the support of its subsets.
  - This is known as the anti-monotone property of support.

Asst. Prof. Dr. Rachsuda Setthawong

Use Apriori Principle to Find Frequent Itemset



(No need to continue finding frequent Itemset having AB as its subset)

Asst. Prof. Dr. Rachsuda Setthawong

Apriori Algorithm

Given that

- F_k: frequent k-itemsets
- L_k: candidate k-itemsets

- **Algorithm**
  - Let k=1
  - Generate F_1 = {frequent 1-itemsets}
  - Repeat until F_k is empty
    - **Candidate Generation**: *Generate* L_{k+1} from F_k
    - **Candidate Pruning:** *Prune* candidate itemsets in L_{k+1} containing subsets of length k that are *infrequent*
    - **Support Counting:** *Count the support* of each candidate in L_{k+1} by scanning the DB
    - **Candidate Elimination:** *Eliminate* candidates in L_{k+1} that are *infrequent*, leaving only those that are frequent => F_{k+1}

Asst. Prof. Dr. Rachsuda Setthawong

Illustrating Apriori Principle

TID	Items
1	Bread, Diaper, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Coke, Diaper, Milk

Suppose that

1. **minsup = 60% → (minsup count = 3)**
2. **Items are listed in alphabetical order (a – z)**

Item	Count
Bread	**4**
Coke	**2**
Milk	**4**
Beer	**3**
Diaper	**5**
Eggs	**1**

Items (1-itemsets)

Itemset	Count
{Bread,Milk}	**3**
{Bread,Beer}	**2**
{Bread,Diaper}	**4**
{Milk,Beer}	**2**
{Milk,Diaper}	**4**
{Beer,Diaper}	**3**

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Itemset	Count
{ Beer, Diaper, Milk}	**2**
{ Beer,Bread, Diaper}	**2**
{Bread, Diaper, Milk}	**3**
{Beer, Bread, Milk}	**1**

Triplets (3-itemsets)

(No need to generate candidates involving {Bread, Beer} or {Milk, Beer})

Comparing no. of candidates generated:

Bruce force (every subset is considered):

$$^6C_1 + {}^6C_2 + {}^6C_3$$

$$6 + 15 + 20 = 41$$

Apriori (with support-based pruning):

$$6 + 6 + 4 = 16$$

Combination (C): $^nC_r = n! / r!\,(n - r)!$

Asst. Prof. Dr. Rachsuda Setthawong

Candidate Generation: Merge F_{k-1} and F_1 itemsets

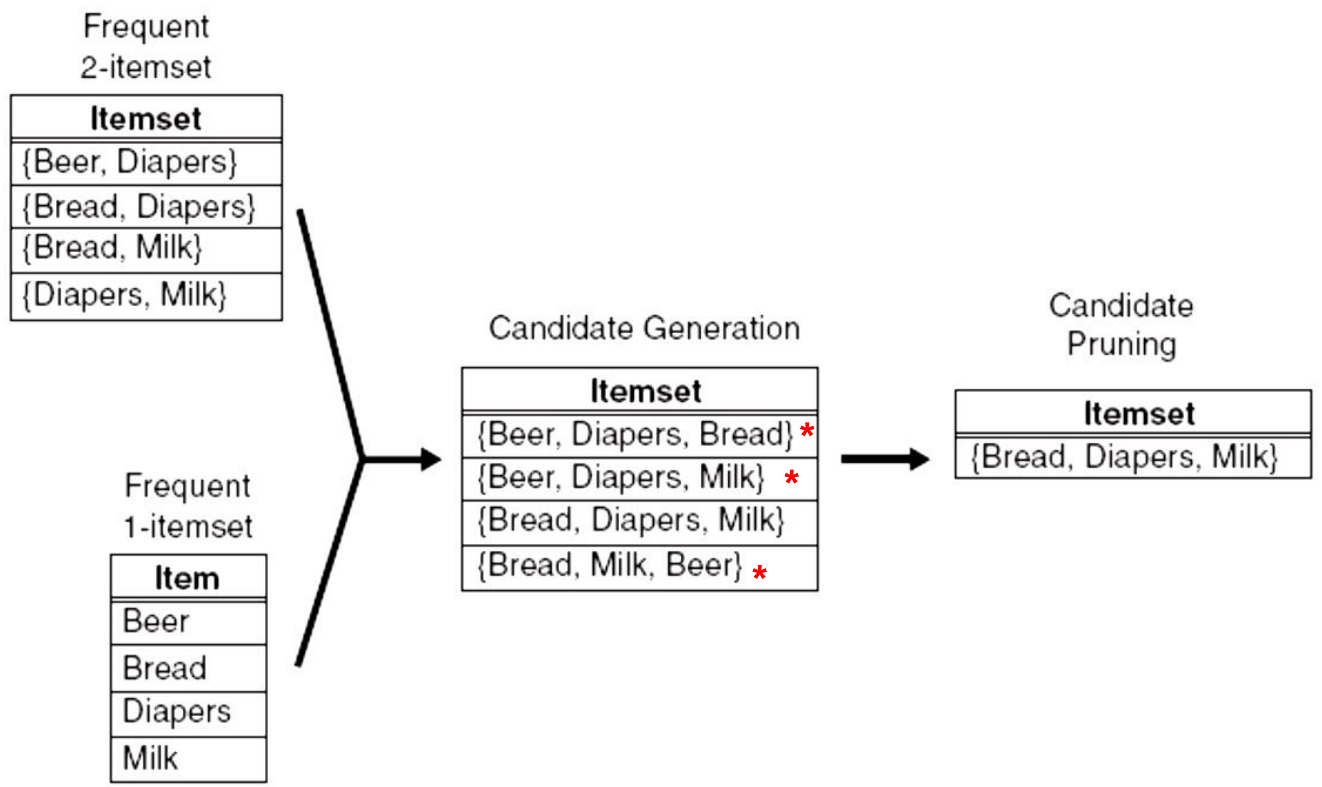


Figure 6.7. Generating and pruning candidate k-itemsets by merging a frequent $(k-1)$-itemset with a frequent item. Note that some of the candidates are unnecessary because their subsets are infrequent.*

Asst. Prof. Dr. Rachsuda Setthawong

Candidate Generation: F_{k-1} x F_{k-1} Method

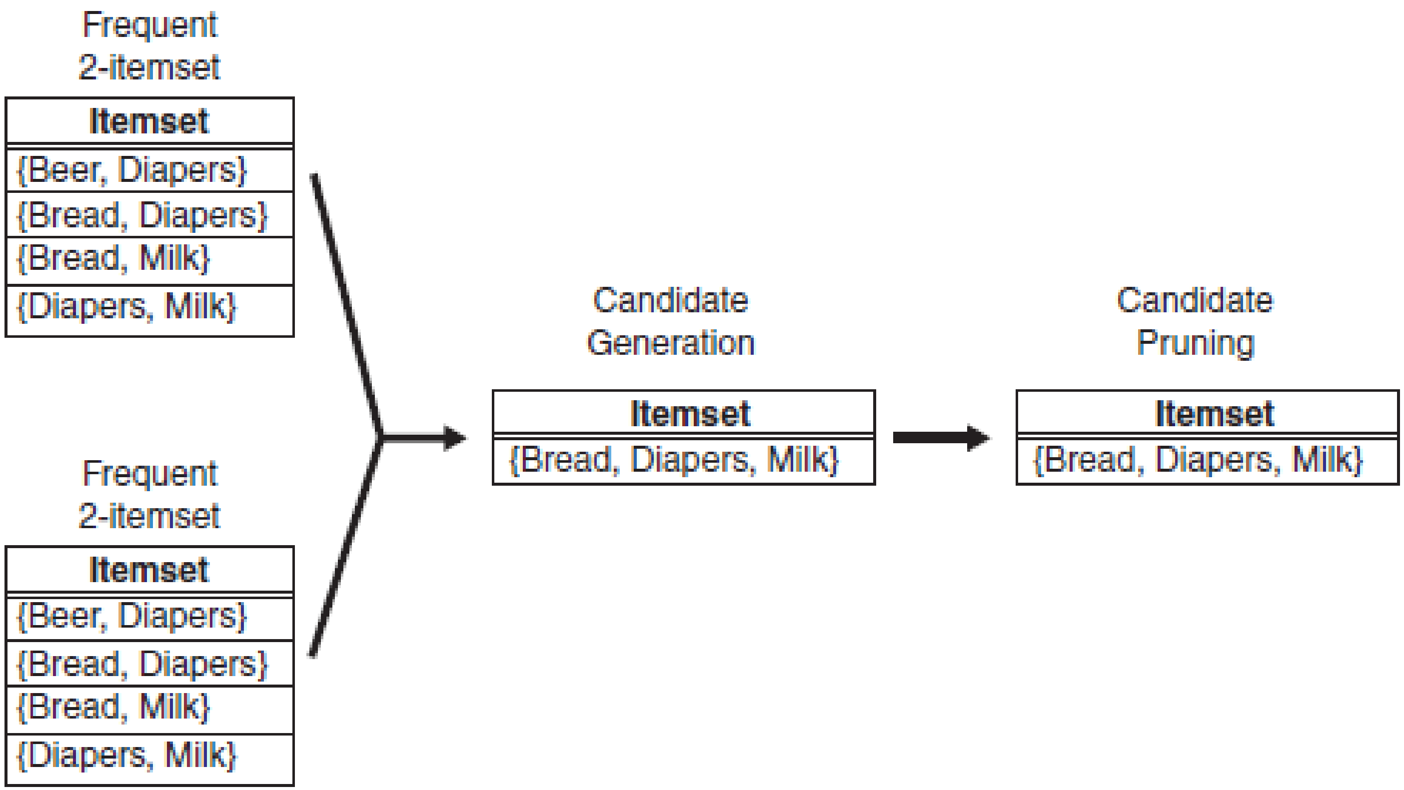


Figure 6.8. Generating and pruning candidate k-itemsets by merging pairs of frequent $(k-1)$-itemsets.

Use $F_{k-1} \times F_{k-1}$ Method in Apriori Algorithm

- $F_{k-1} \times F_{k-1}$ Method
  - The candidate generation procedure in the apriori-gen function merges a pair of frequent (k-1)-itemsets only if their first k-2 items are identical.

Illustrating Apriori Principle (Use $F_{k-1} \times F_{k-1}$ Method)

TID	Items
1	Bread, Diaper, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Coke, Diaper, Milk

Items (1-itemsets)

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	5
Eggs	1

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	4
{Milk,Beer}	2
{Milk,Diaper}	4
{Beer,Diaper}	3

Triplets (3-itemsets)

Itemset	Count
{Bread, Diaper, Milk}	3

Use of $F_{k-1}xF_{k-1}$ method for candidate generation results in only one 3-itemset. This is eliminated after the support counting step.

Comparing no. of candidates generated:

Bruce force (every subset is considered):

$^6C_1 + {}^6C_2 + {}^6C_3$

$6 + 15 + 20 = 41$

Apriori (with Fk-1 × Fk-1 Method):

$6 + 6 + \mathbf{1} = 13$

Asst. Prof. Dr. Rachsuda Setthawong

Strategy 2: Reducing Number of Comparisons

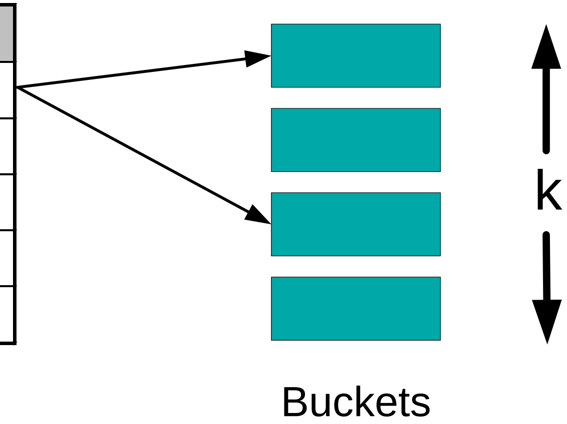
- Candidate counting:
  - Scan the database of transactions to determine the support of each candidate itemset
  - To reduce the number of comparisons, store the **candidates** in a **hash structure**
    - Instead of *matching each transaction* against every candidate, match it against candidates *contained in* the hashed buckets

Transactions

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

N

Hash Structure



Support Counting Using Hash Tree: Initial Idea for Matching

- Suppose you have 15 candidate itemsets of length 3:

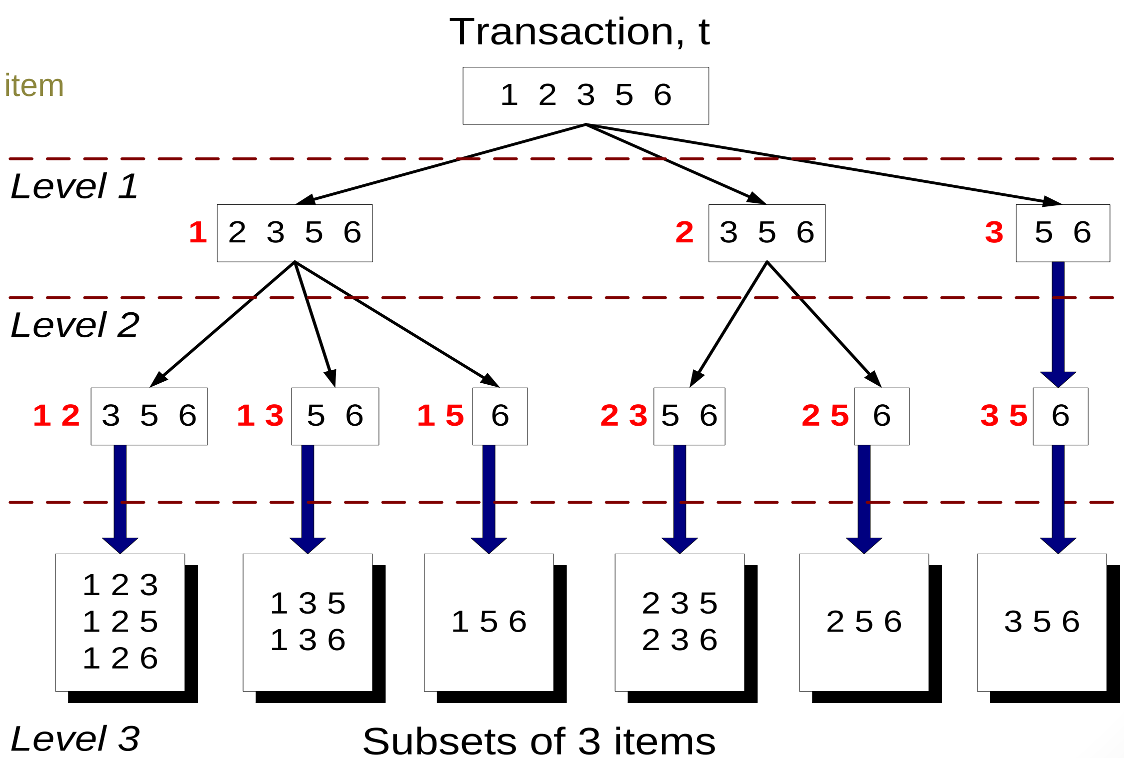
  {1 4 5}, {1 2 4}, {4 5 7}, **{1 2 5},** {4 5 8}, {1 5 9}, **{1 3 6},** {2 3 4}, {5 6 7}, {3 4 5}, **{3 5 6},** {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}

- How many of these itemsets are supported by the transaction (1,2,3,5,6)?
  - Can enumerate subsets of *k*-itemsets from a transaction using *prefix tree structure*

Observation: All 3-itemsets contained in t must begin with item 1, 2, or 3.

Prefix of level 1: 1, 2, or 3

Prefix of level 2:
1 2, 1 3, 1 5
2 3, 2 5
3 5



Asst. Prof. Dr. Rachsuda Setthawong

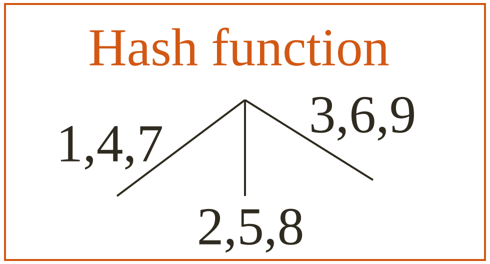
Support Counting Using Hash Tree: An Example

Using the previous example,

1. Create candidate hash tree for *k*-itemsets candidates
2. For each transaction in a dataset, determine whether each enumerated 3-itemset corresponds to an existing candidate item. (If one of them matched, then support count of the corresponding candidate is incremented.)

To create candidate hash tree, we need:

1. **Hash function**: e.g., h(p) = (p-1) mod 3



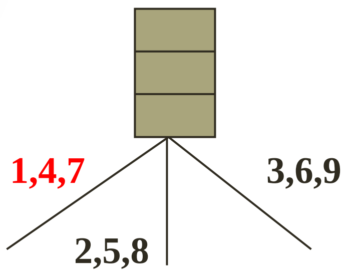
2. **Max leaf size**: max number of itemsets stored in a leaf node (if number of candidate itemsets exceeds max leaf size, split the node)

- E.g., max = 3

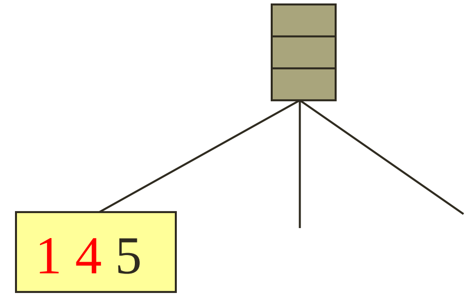
Asst. Prof. Dr. Rachsuda Setthawong

Insert an itemset to the hash tree: {1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}

Hash Function

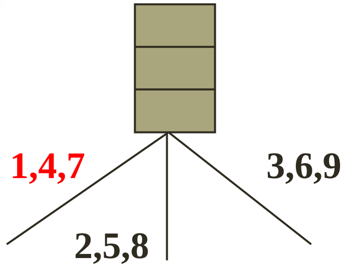


Candidate Hash Tree

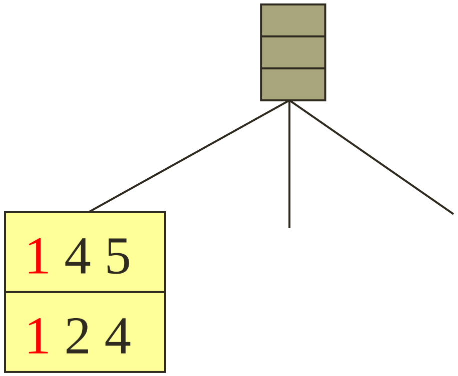


Insert an itemset to the hash tree: {1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}

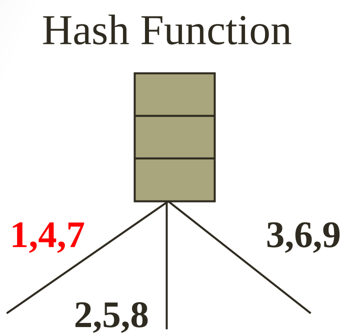
Hash Function

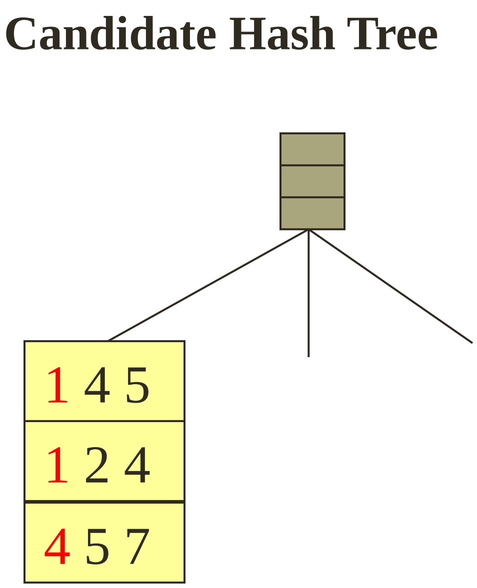


Candidate Hash Tree

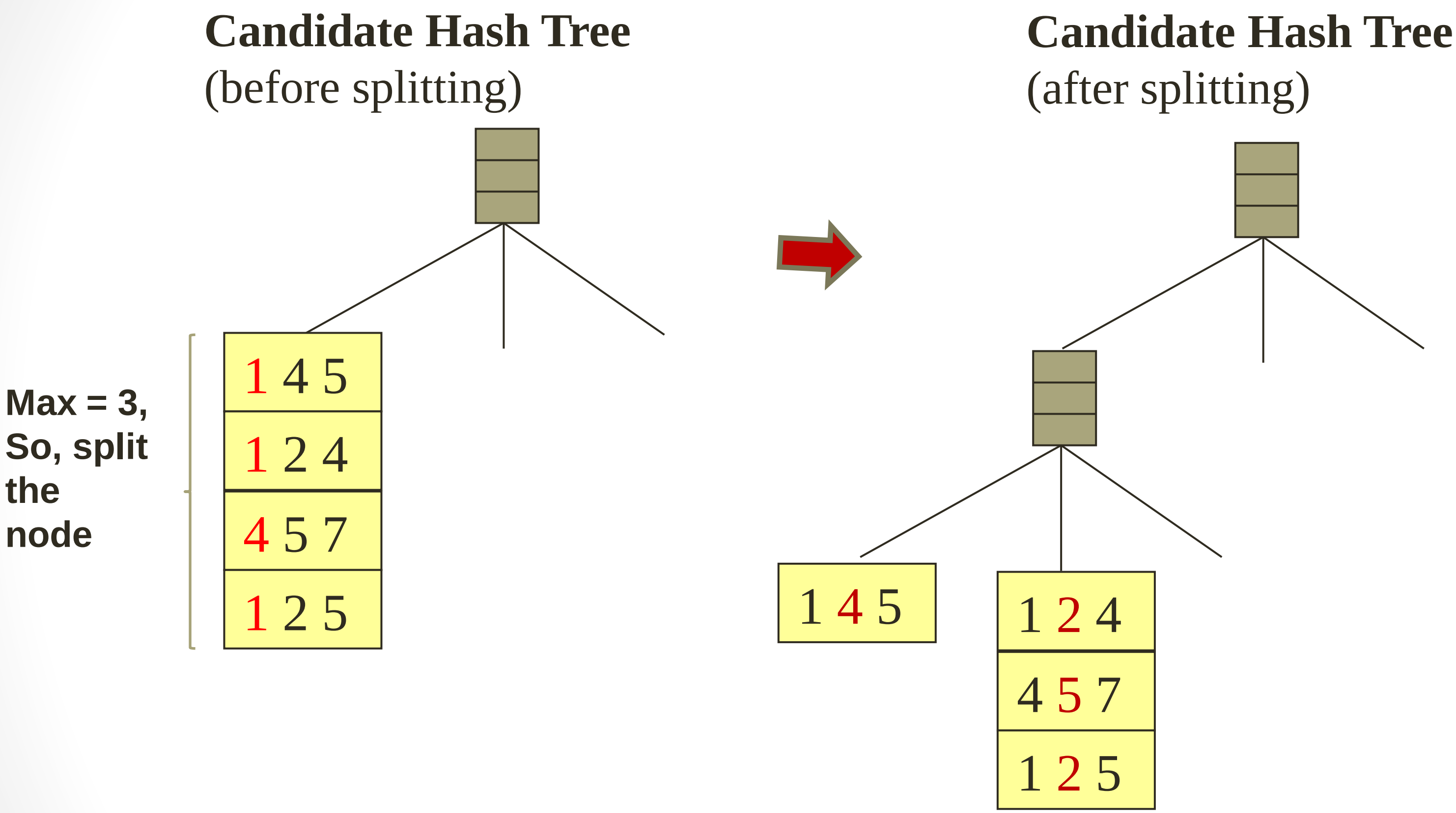


Insert an itemset to the hash tree: {1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}





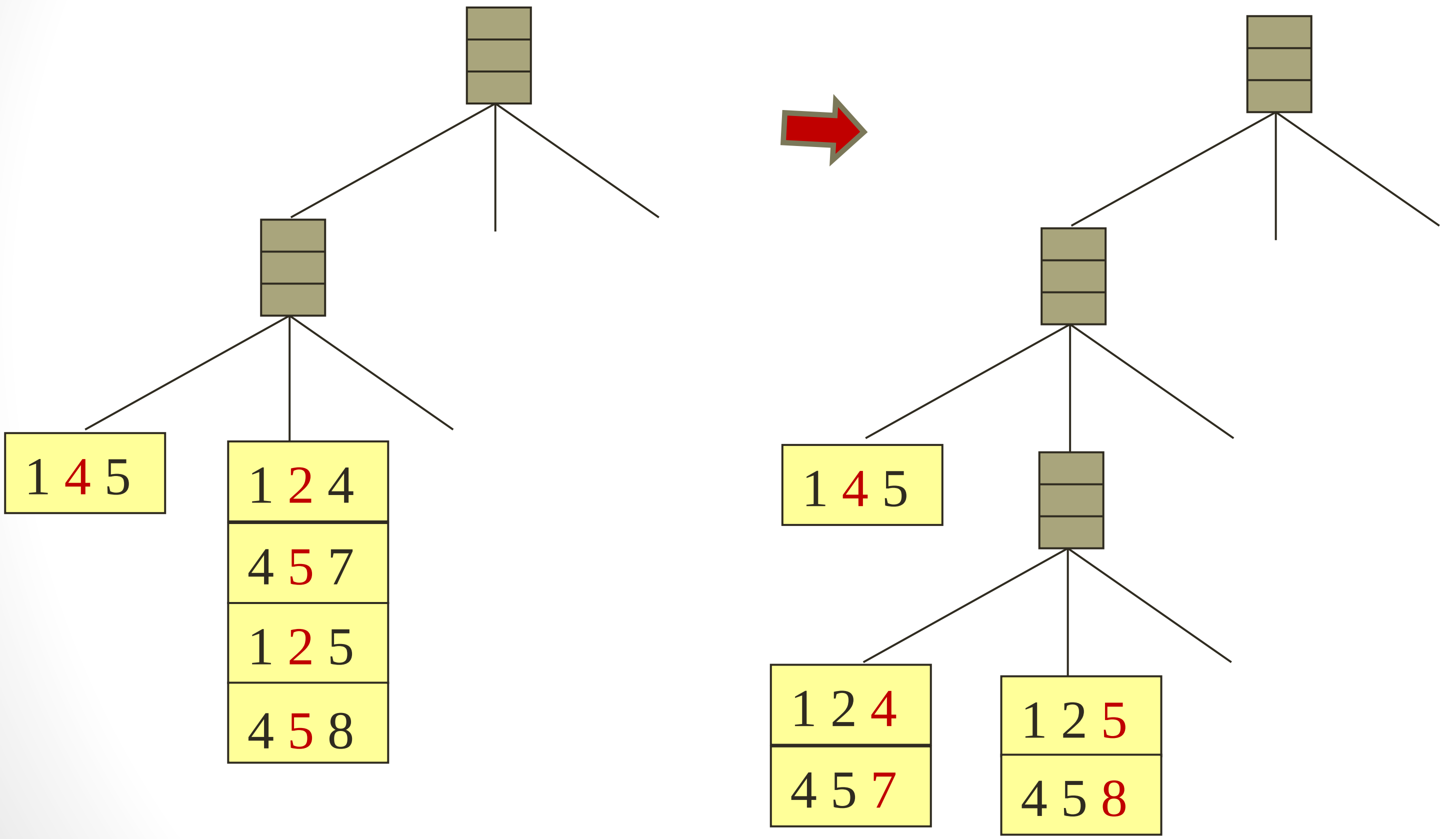
Insert an itemset to the hash tree: {1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}



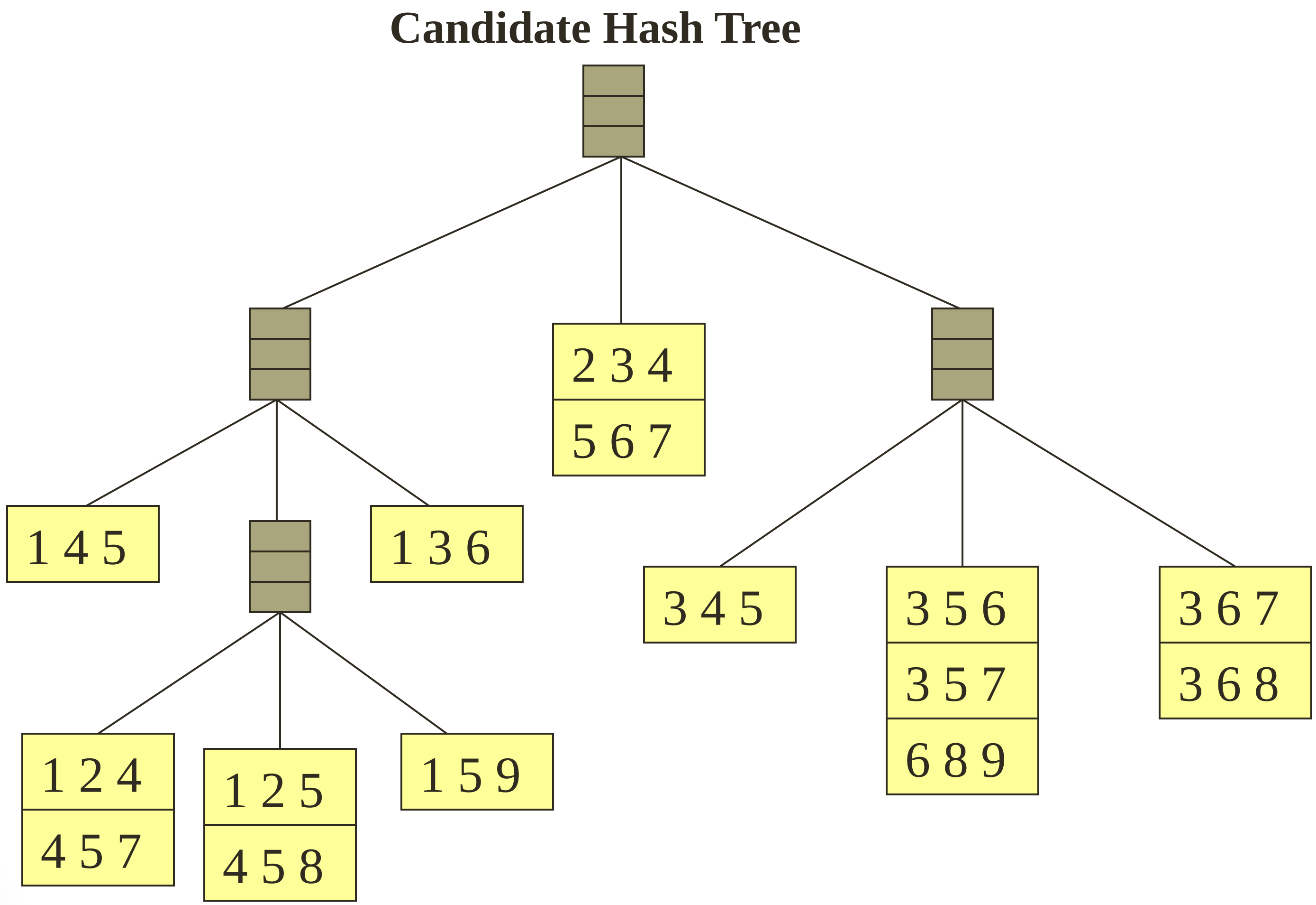
Insert an itemset to the hash tree: {1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}

Candidate Hash Tree
(before splitting)

Candidate Hash Tree
(after splitting)

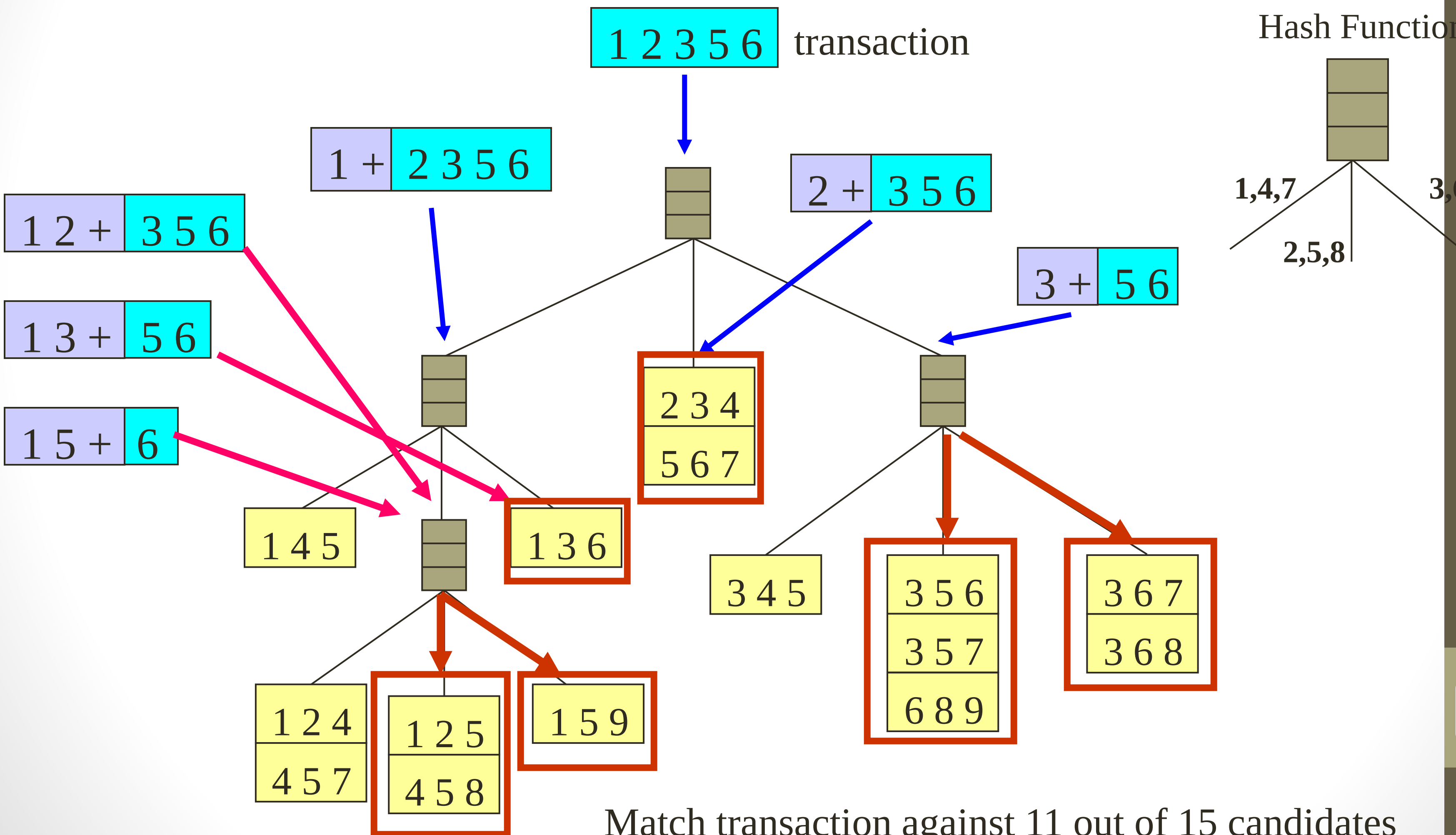


Insert an itemset to the hash tree: {1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}



Support Counting Using a Hash Tree

Step 2. For each transaction in a dataset, determine whether each enumerated 3-itemset corresponds to an existing candidate item. (If one of them matched, then support count of the corresponding candidate is incremented.)

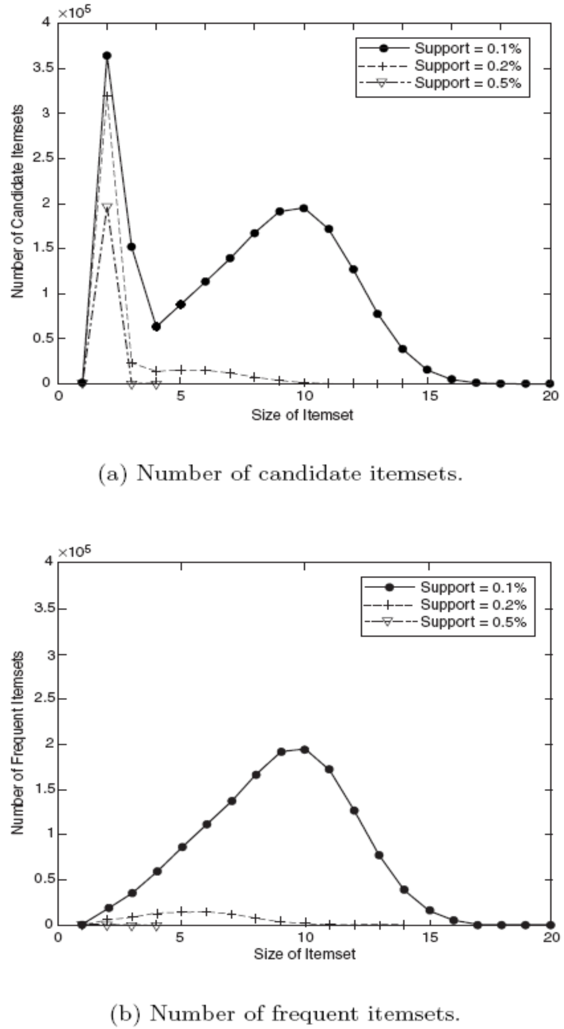


Asst. Prof. Dr. Rachsuda Setthawong

Factors Affecting Complexity of Apriori

- Choice of minimum support threshold
  - lowering support threshold results in more frequent itemsets
    - this may increase number of candidates and max length of frequent itemsets
- Dimensionality (number of items) of the data set
  - more space is needed to store support count of each item
    - if number of frequent items also increases, both computation and I/O costs may also increase
- Size of database
  - run time of algorithm may increase with number of transactions
- Average transaction width
  - transaction width increases with denser data sets
    - this may increase max length of frequent itemsets and traversals of hash tree (number of subsets in a transaction increases with its width)

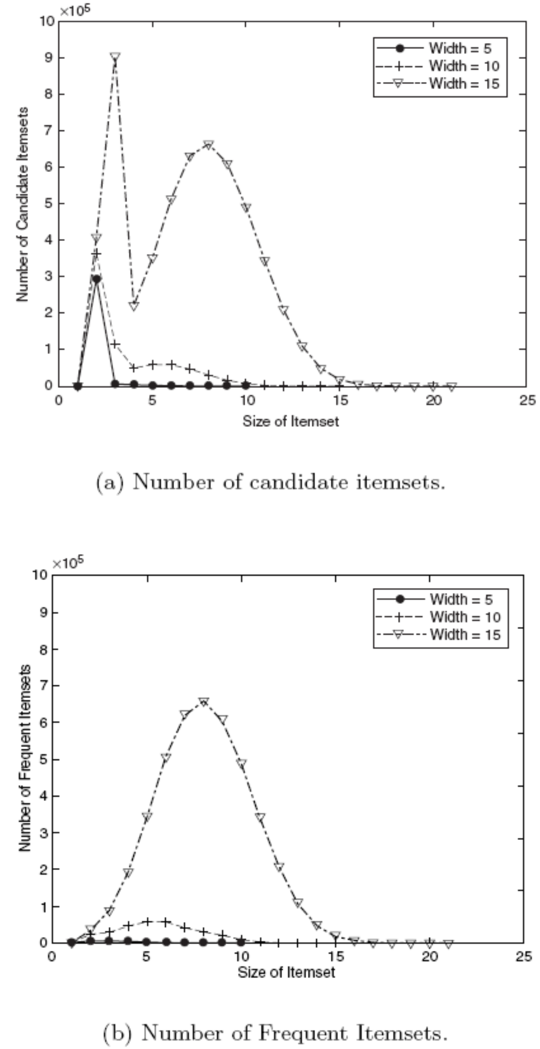
Factors Affecting Complexity of Apriori



(a) Number of candidate itemsets.

(b) Number of frequent itemsets.

Figure 6.13. Effect of support threshold on the number of candidate and frequent itemsets.



(a) Number of candidate itemsets.

(b) Number of Frequent Itemsets.

Figure 6.14. Effect of average transaction width on the number of candidate and frequent itemsets.

Compact Representation of Frequent Itemsets

- Some itemsets are redundant because they have identical support as their supersets

TID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

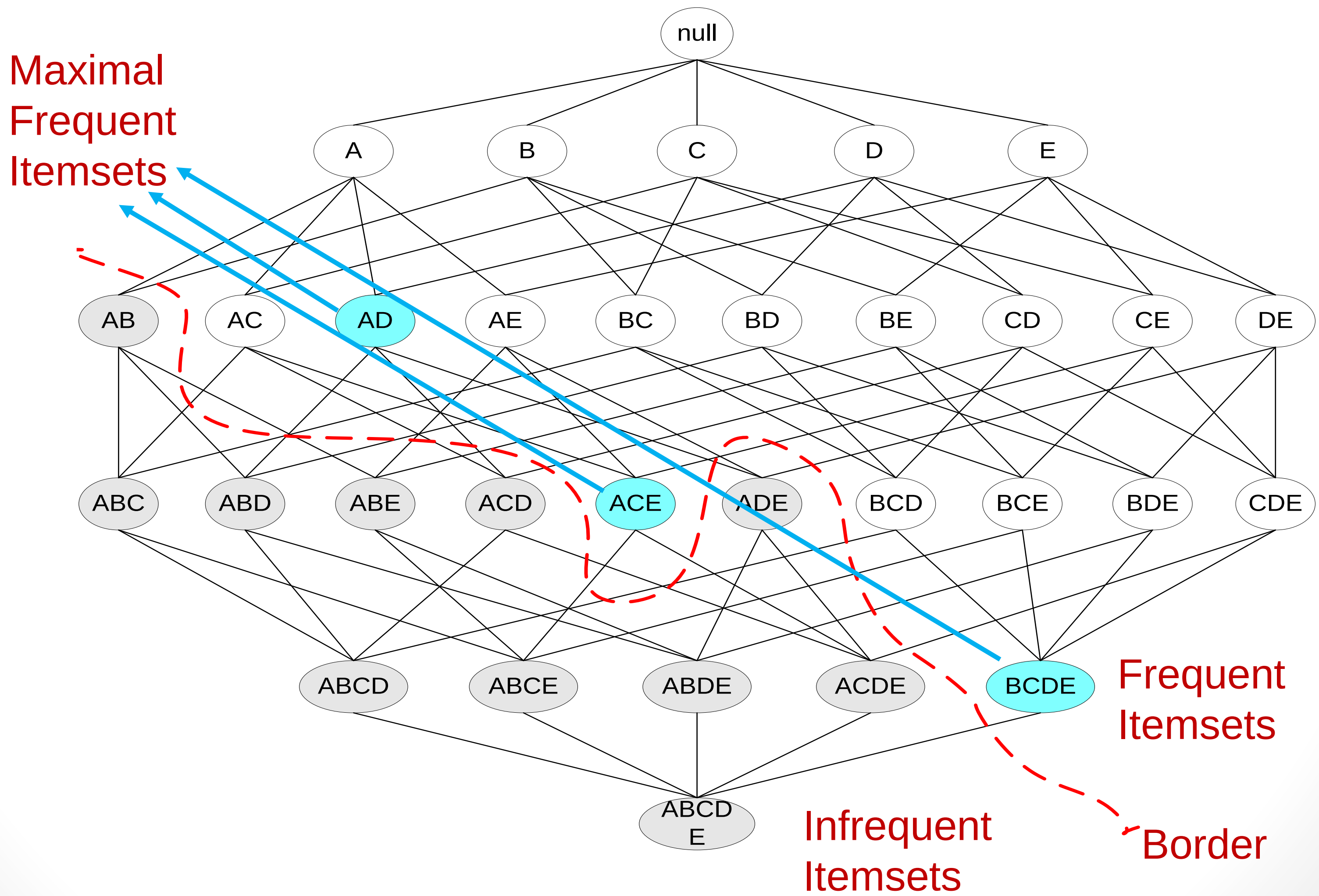
- Number of frequent itemsets $= 3 \times \sum_{k=1}^{10} \binom{10}{k}$ $= 3 \times 2^n$
- Need a compact representation

Ref: https://en.wikipedia.org/wiki/Combination#Example_of_counting_combinations

Maximal Frequent Itemset

An itemset is **maximal frequent** if none of its immediate supersets is frequent.

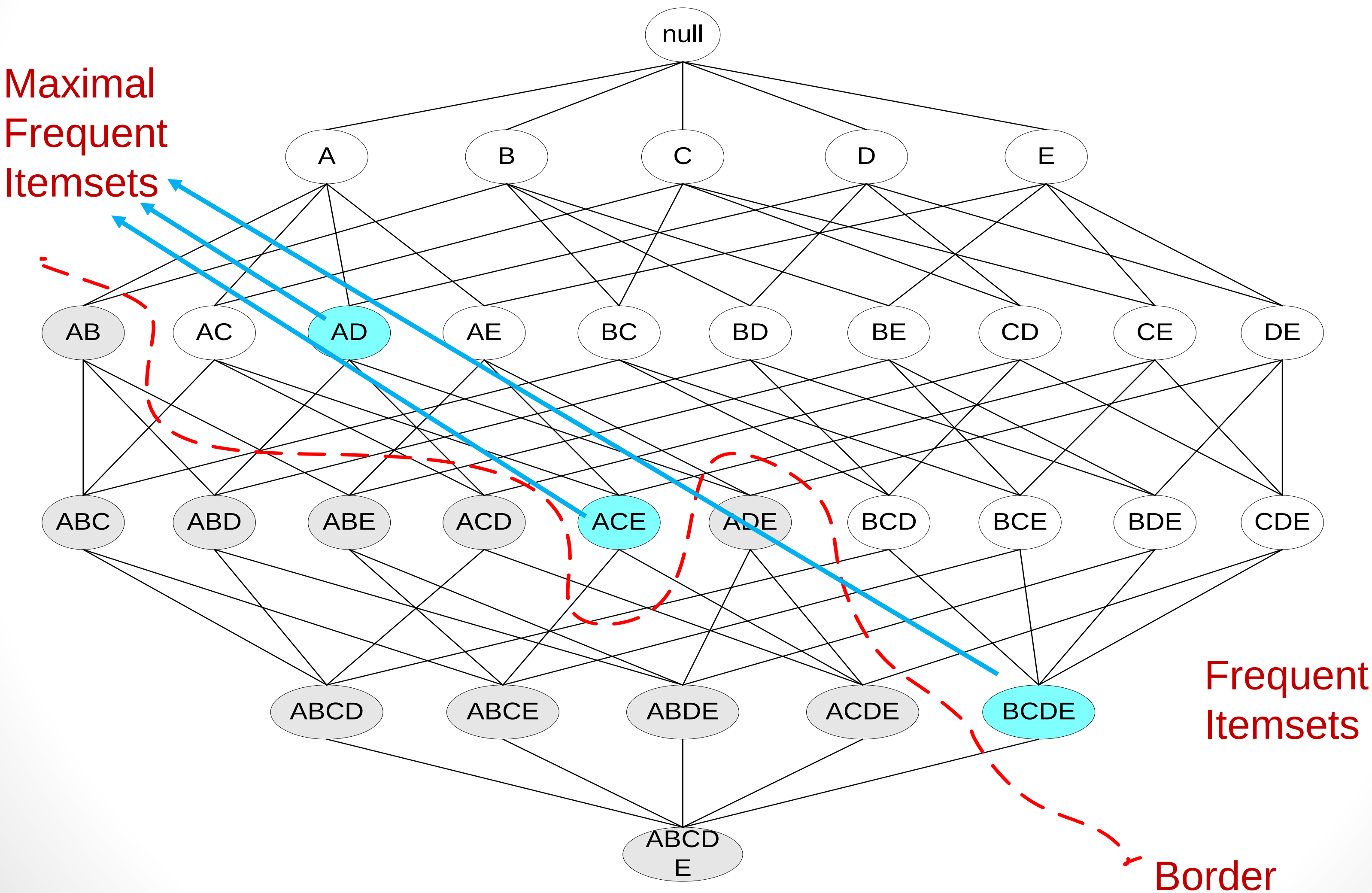
(= all immediate supersets are infrequent)



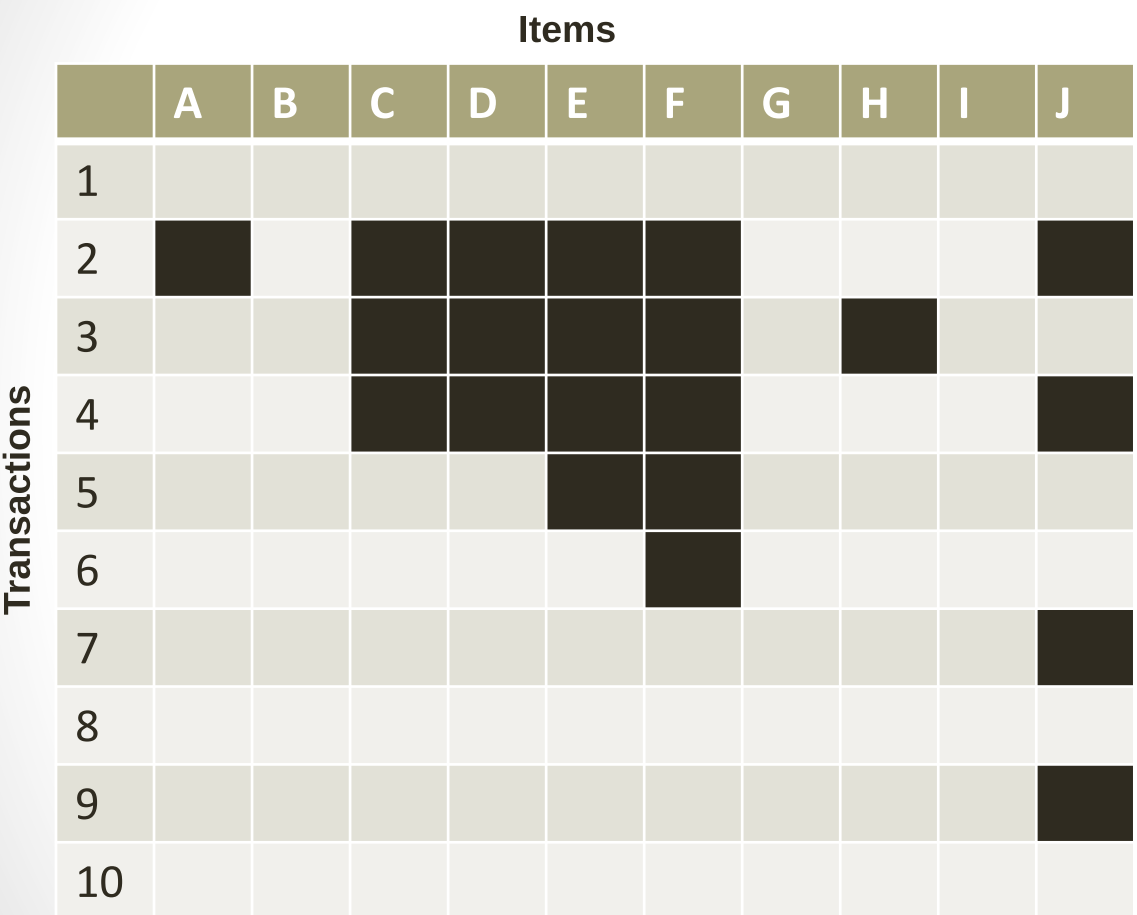
Asst. Prof. Dr. Rachsuda Setthawong

Why Maximal Frequent Itemset?

Use this compact representation to derive all frequent itemsets.



An illustrative example



Support threshold (by count) : 5

Frequent itemsets: {F}

Maximal itemsets: {F}

Support threshold (by count): 4

Frequent itemsets: {E}, {F}, {E,F}, {J}

Maximal itemsets: {E,F}, {J}

Support threshold (by count): 3

Frequent itemsets:

All subsets of {C,D,E,F} + {J}

Maximal itemsets:

{C,D,E,F}, {J}

Asst. Prof. Dr. Rachsuda Setthawong

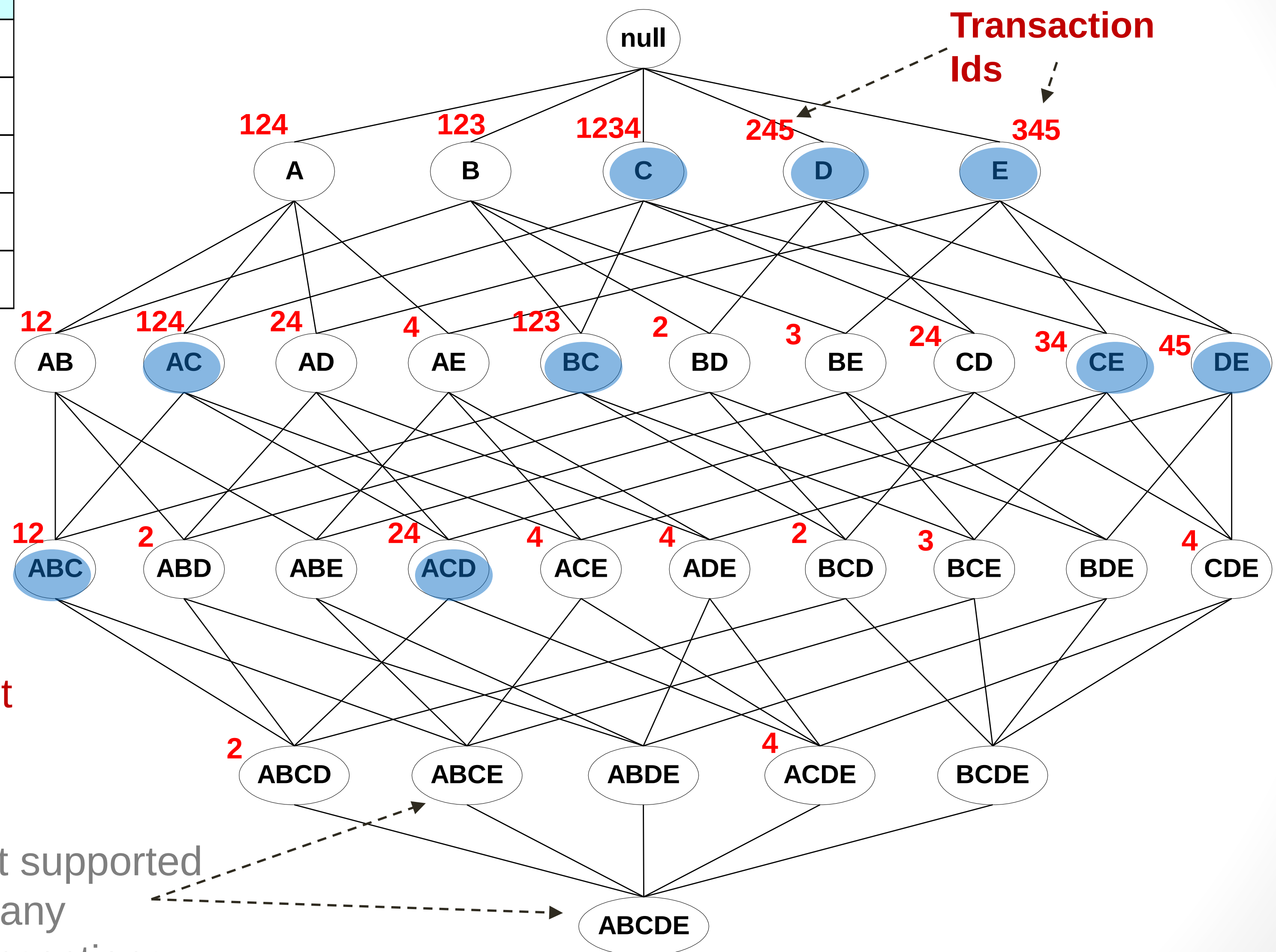
Closed Frequent Itemset

- An itemset *X* is closed if **none** of its immediate supersets has the **same support count** as *X*.

  (= **all** immediate supersets has **lower** support count)

Closed Frequent Itemsets

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE

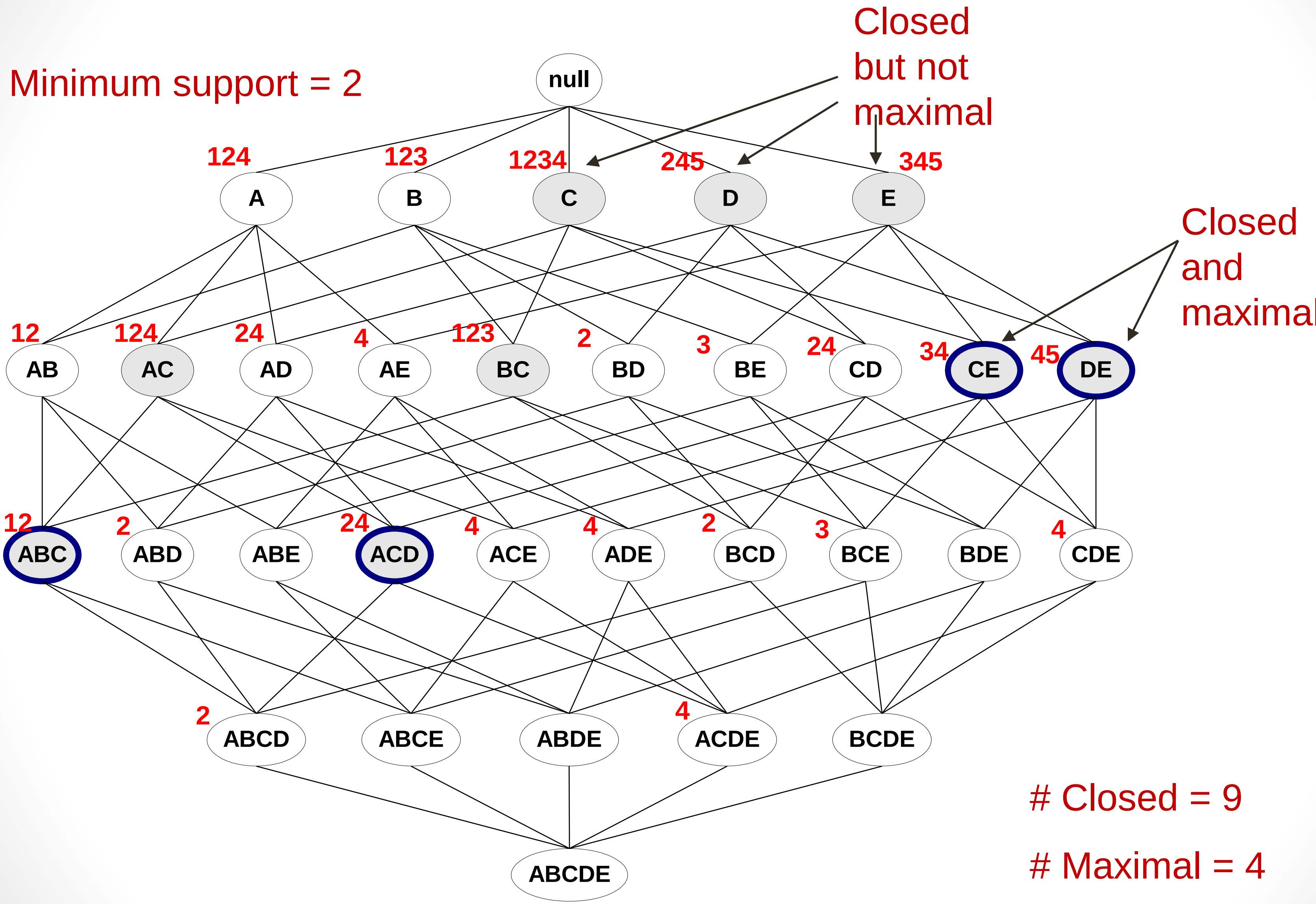


: Closed Frequent Itemset

Not supported by any transactions

Asst. Prof. Dr. Rachsuda Setthawong

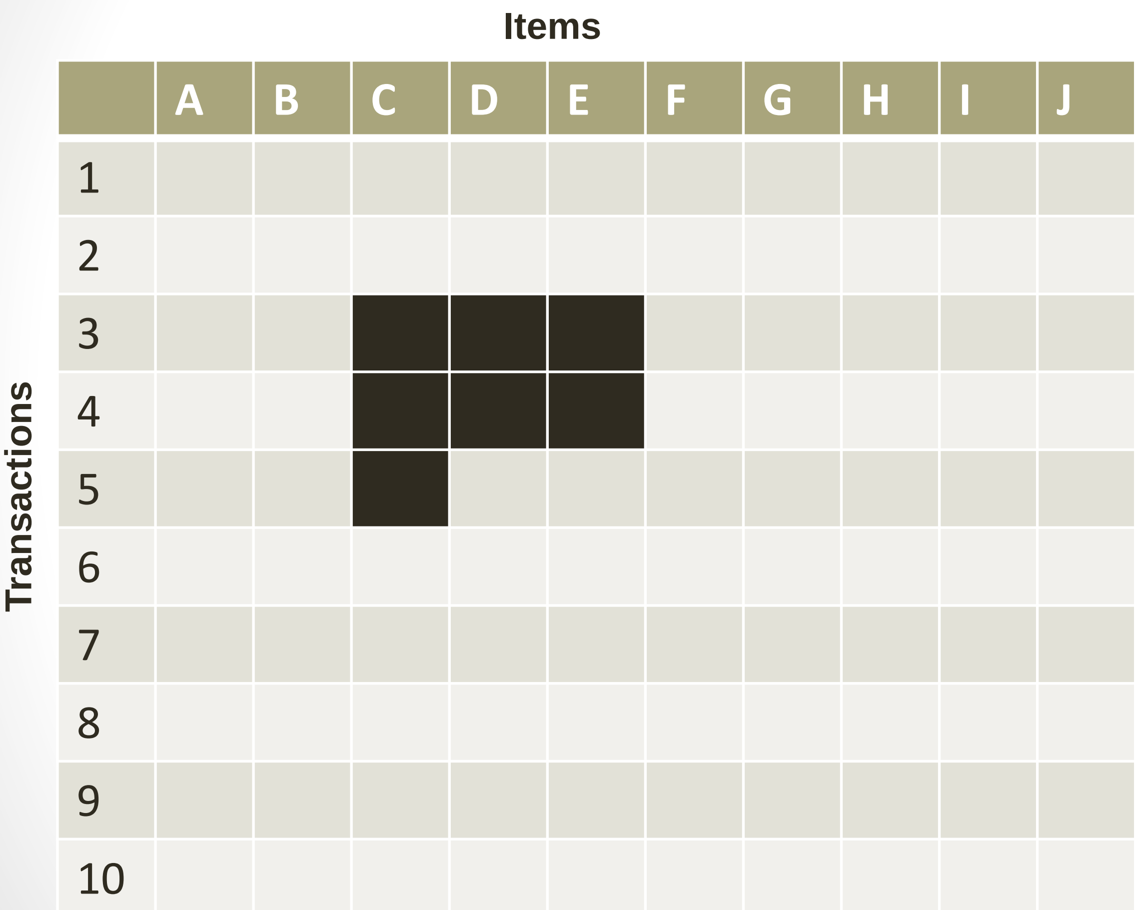
Maximal vs Closed Frequent Itemsets



Asst. Prof. Dr. Rachsuda Setthawong

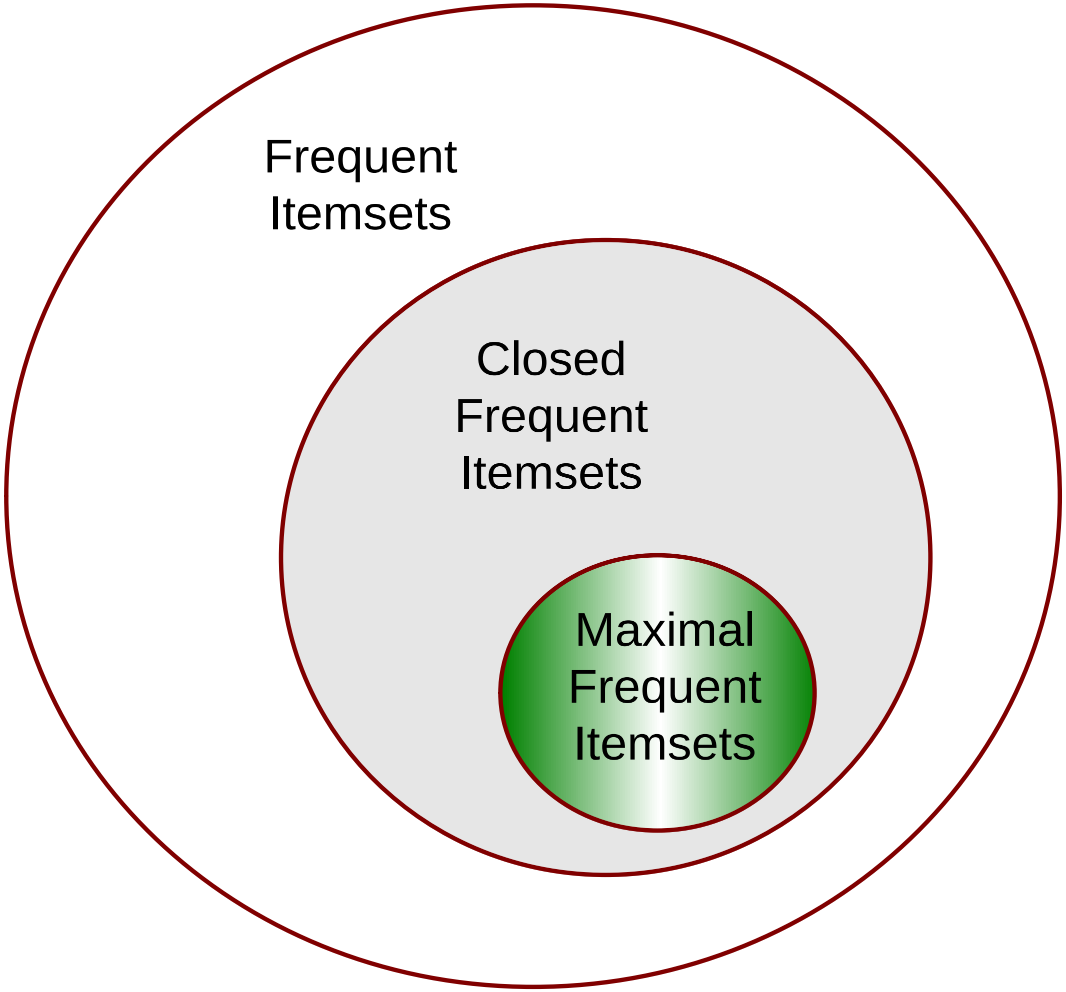
An Example

(all immediate supersets has **lower** support count)



Itemsets	Support (counts)	Closed itemsets
{C}	**3**	✓
{D}	2	
{E}	2	
{C,D}	2	
{C,E}	2	
{D,E}	2	
{C,D,E}	**2**	✓

Maximal vs Closed Itemsets



FP-growth Algorithm

- Is an alternative approach to generate frequent itemsets.
- Use FP-tree as a compressed representation of the input data
- Once an FP-tree has been constructed, it uses a recursive divide-and-conquer approach to mine the frequent itemsets

FP-tree construction - 1

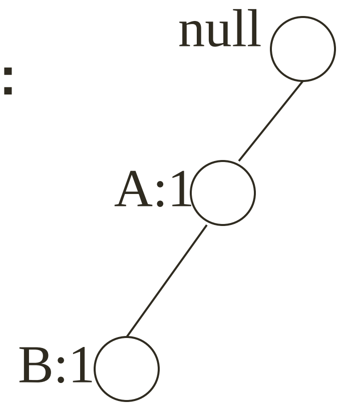
TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

- Sort items in each transaction based on the number of occurrences
  - A: 8
  - B: 7
  - C: 6
  - D: 5
  - E: 3

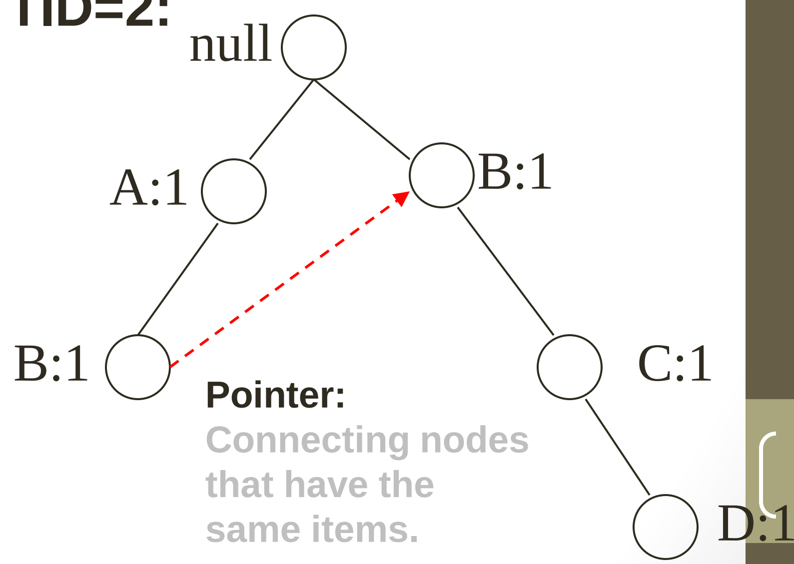
FP-tree construction - 2

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=1:



After reading TID=2:

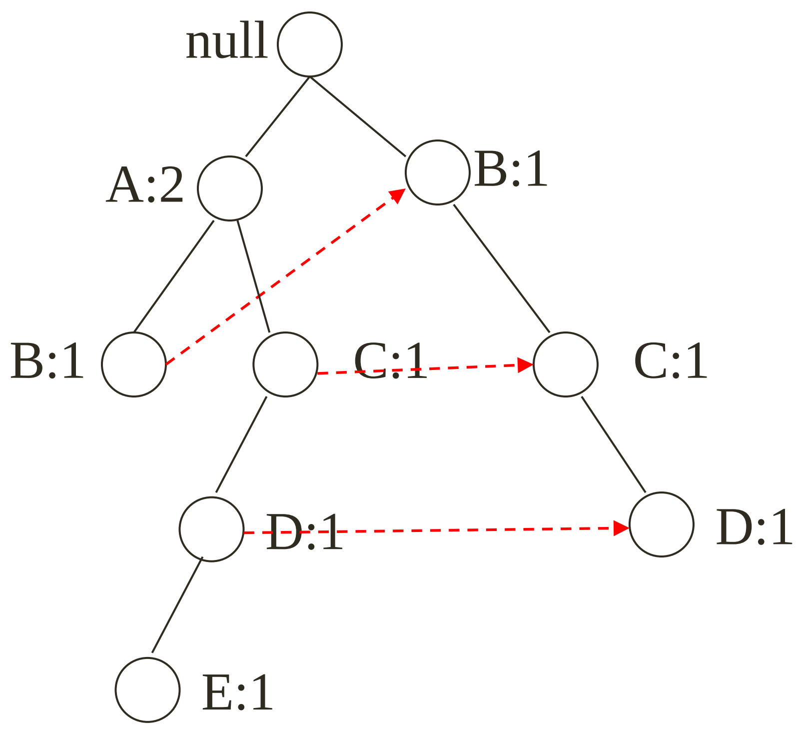


Pointer:
Connecting nodes that have the same items.

FP-tree construction - 3

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=3:

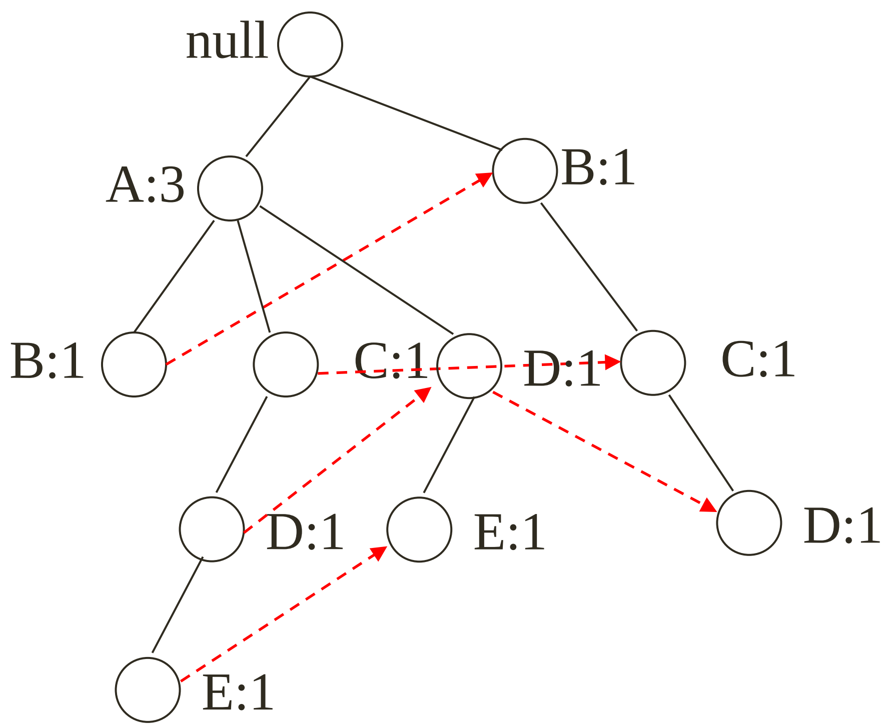


Asst. Prof. Dr. Rachsuda Setthawong

FP-tree construction - 4

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=4:

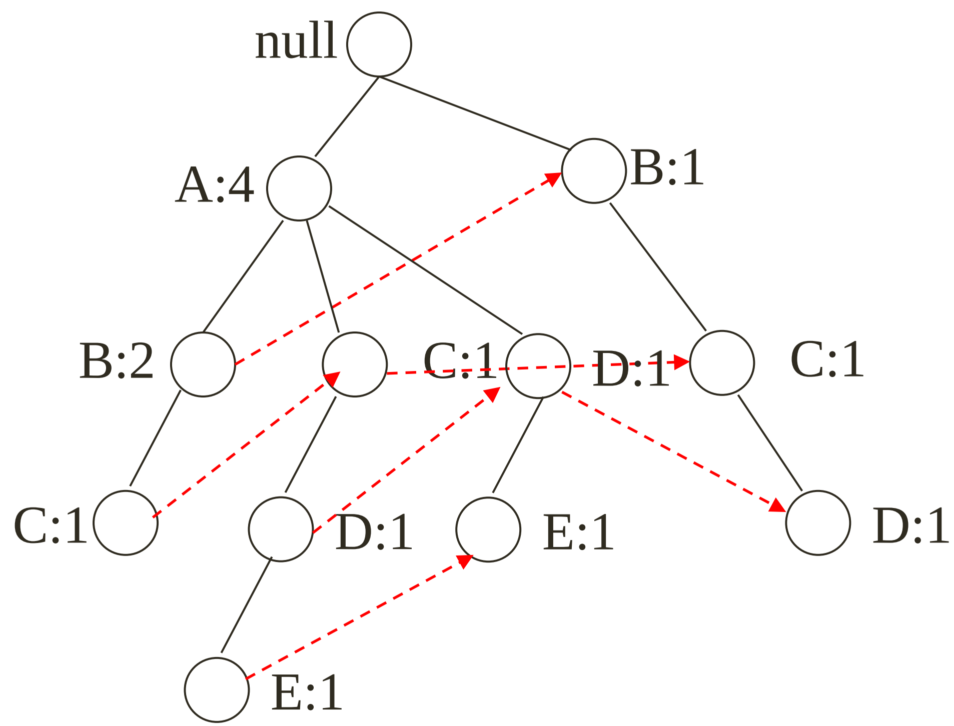


Asst. Prof. Dr. Rachsuda Setthawong

FP-tree construction - 5

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

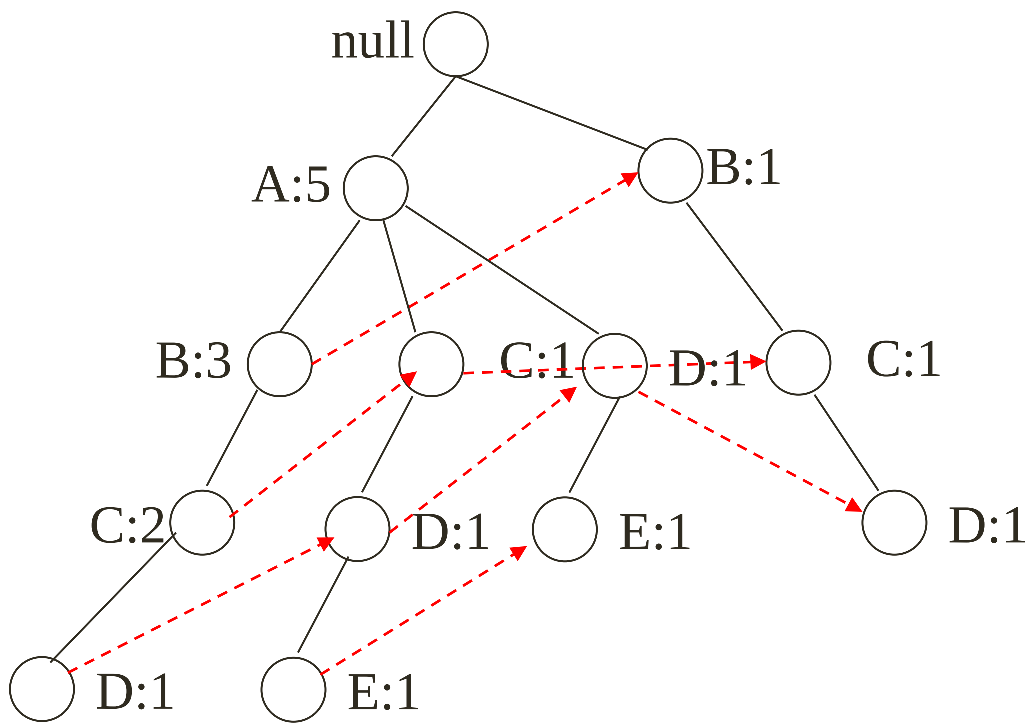
After reading TID=5:



FP-tree construction - 6

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=6:

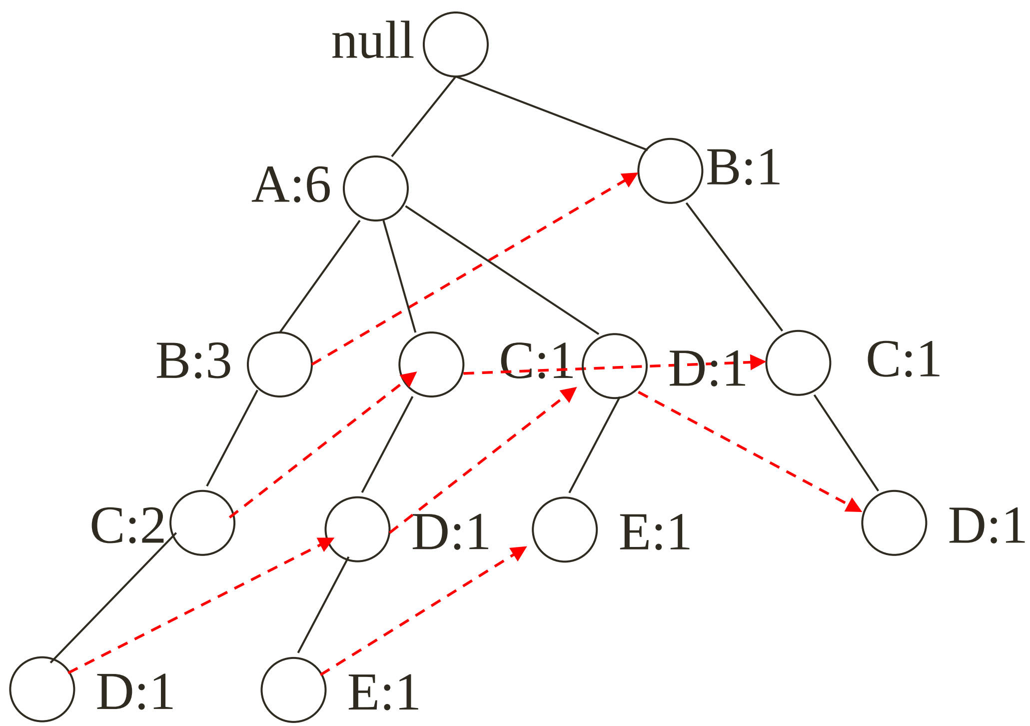


Asst. Prof. Dr. Rachsuda Setthawong

FP-tree construction - 7

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

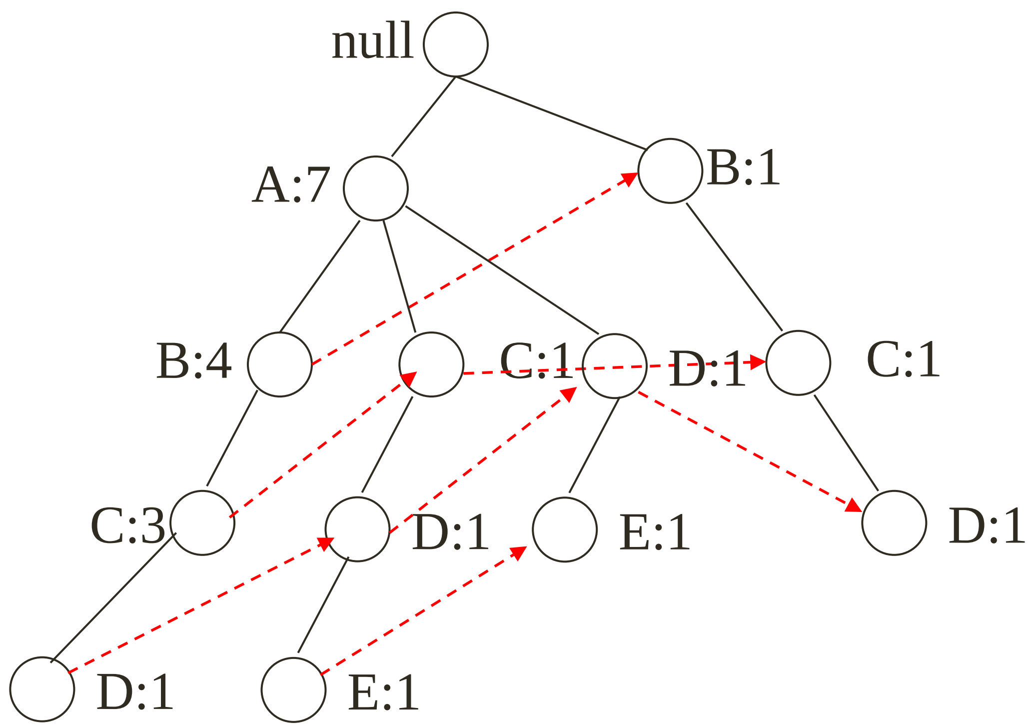
After reading TID=7:



FP-tree construction - 8

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

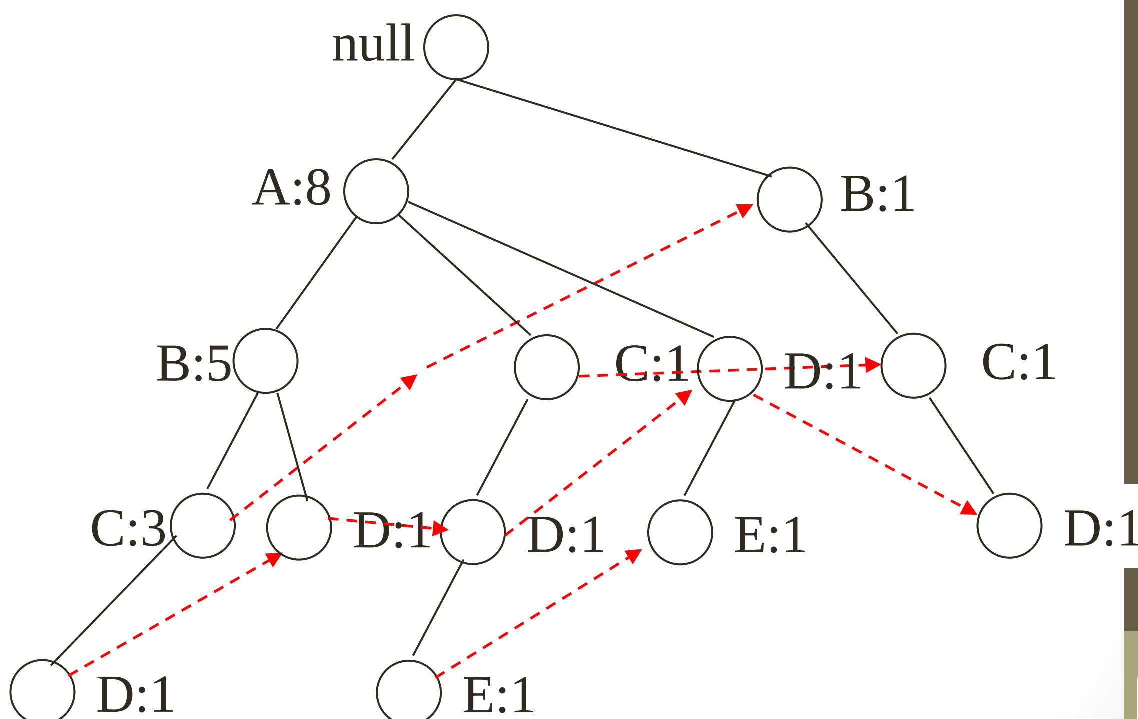
After reading TID=8:



FP-tree construction - 9

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=9:



FP-Tree Construction - 10

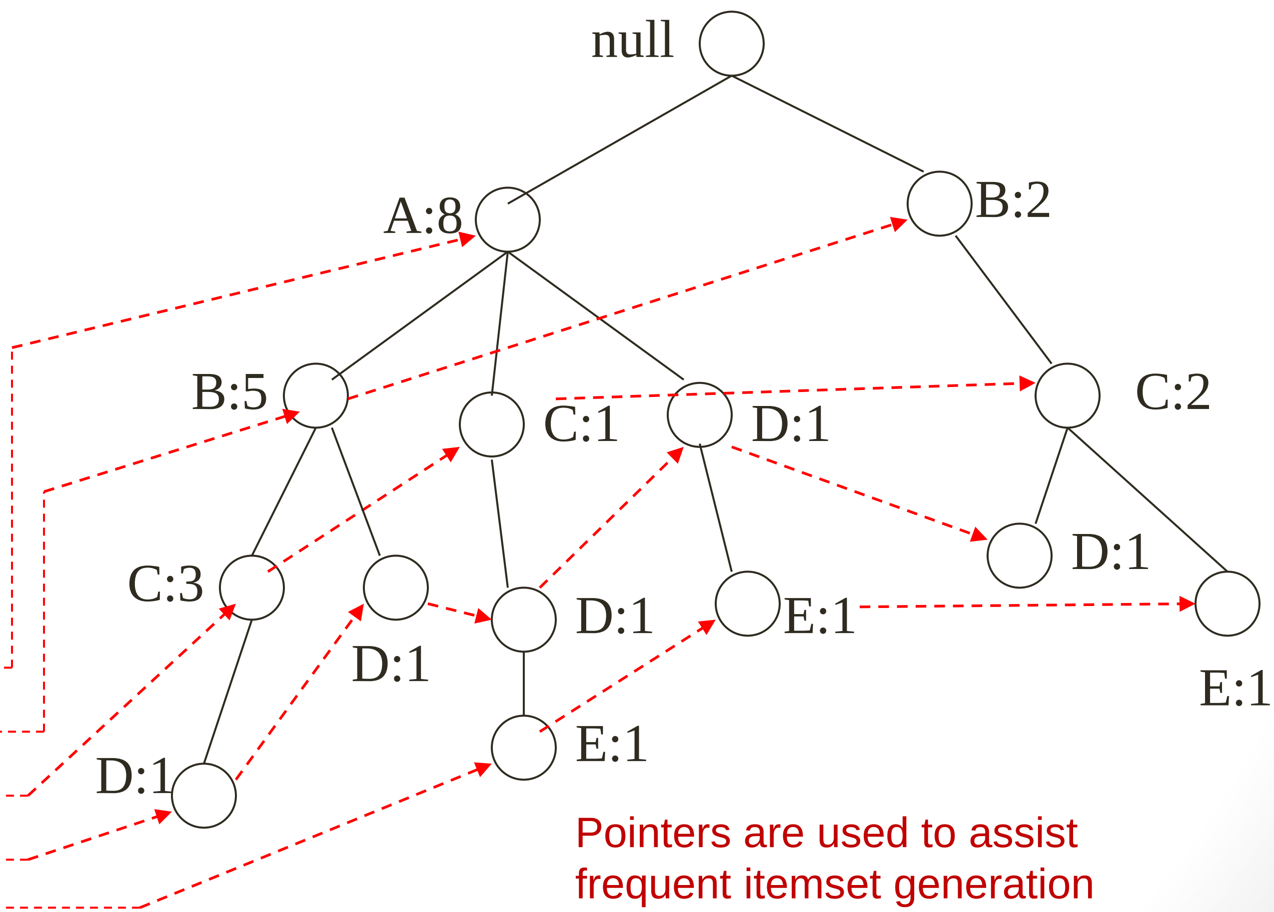
TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Transaction Database

After reading TID=10:

Header table

Item	Pointer
A	
B	
C	
D	
E	



Pointers are used to assist frequent itemset generation

Asst. Prof. Dr. Rachsuda Setthawong

Finding the frequent itemsets ending in a particular item

- Since every transaction is mapped onto a path in the FP-tree, we can derive the frequent itemsets ending with a particular items, e.g., E, by examining only the paths containing node e.
  - Constructing conditional FP-tree for E.

- The same process continues for other suffices until all the paths associated with nodes D, C, B and A are processed.

Rule Generation

- Given a frequent itemset L, find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement
  - If {A,B,C,D} is a frequent itemset, candidate rules:

    $ABC \rightarrow D$, $ABD \rightarrow C$, $ACD \rightarrow B$, $BCD \rightarrow A$,
    $A \rightarrow BCD$, $B \rightarrow ACD$, $C \rightarrow ABD$, $D \rightarrow ABC$
    $AB \rightarrow CD$, $AC \rightarrow BD$, $AD \rightarrow BC$, $BC \rightarrow AD$,
    $BD \rightarrow AC$, $CD \rightarrow AB$

- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \varnothing$ and $\varnothing \rightarrow L$)

Asst. Prof. Dr. Rachsuda Setthawong

Rule Generation

- How to efficiently generate rules from frequent itemsets?
  - Confidence does **not** have an anti-monotone property

    $c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$

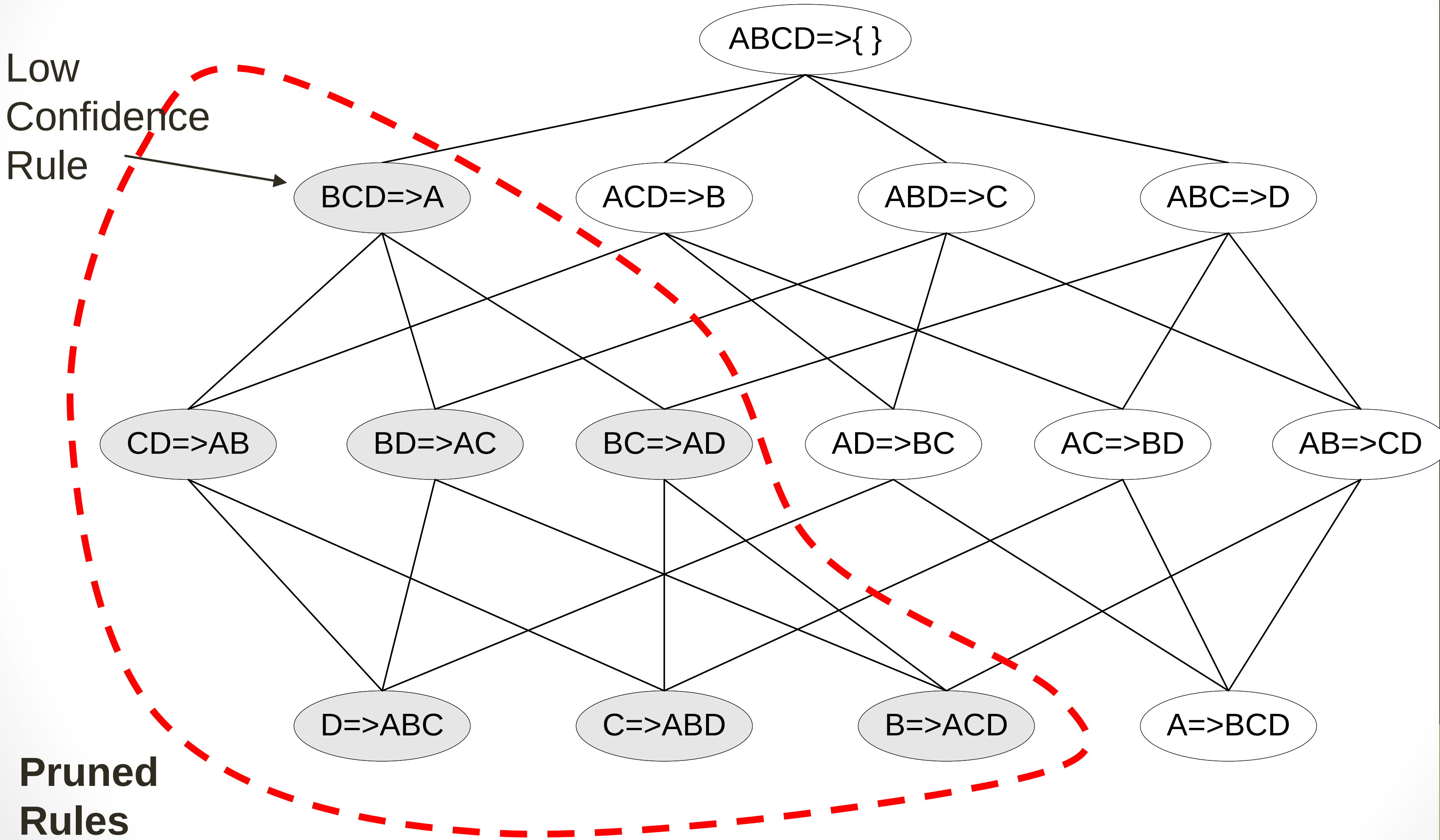
  - But **confidence of rules** generated from the **same** itemset has an **anti-monotone** property
  - e.g., L = {A,B,C,D}:

$$c(ABC \rightarrow \mathbf{D}) \geq c(AB \rightarrow \mathbf{CD}) \geq c(A \rightarrow \mathbf{BCD})$$

    - Confidence is anti-monotone w.r.t. **number of items on the RHS** of the rule
    - In other words,
      - if a set is infrequent then all of its superset are also infrequent.
      - if a set is frequent, then all of its subset are frequent.

Rule Generation for Apriori Algorithm

Lattice of rules



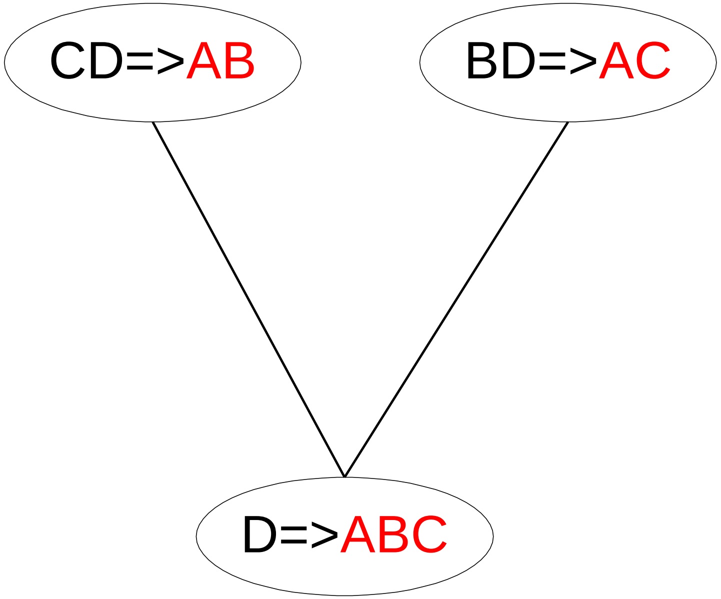
Asst. Prof. Dr. Rachsuda Setthawong

Rule Generation for Apriori Algorithm

- Candidate rule is generated by **merging two rules** that **share the same prefix in the rule consequent**.

- join(CD=>**A**B,BD=>**A**C)
  would produce the candidate
  rule D => **A**BC

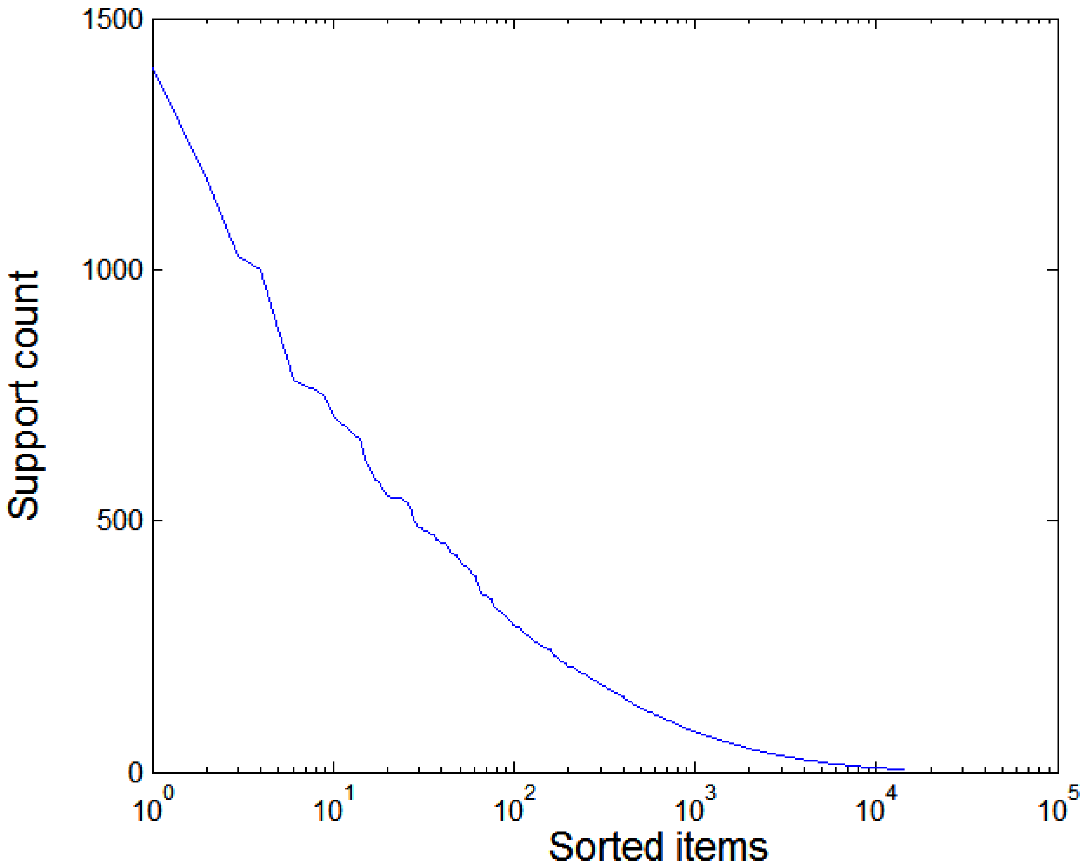
- PRUNE rule D=>**A**BC if its
  subset AD=>**BC*** does NOT have
  HIGH confidence.
  - *: in previous slide



Effect of Support Distribution

- Many real data sets have skewed support distribution

Support distribution of a retail data set



Asst. Prof. Dr. Rachsuda Setthawong

Effect of Support Distribution

- How to set the appropriate *minsup* threshold?
  - If ***minsup*** is set **too high**, we could miss itemsets involving interesting rare items (e.g., expensive products).
  - If ***minsup*** is set **too low**, it is computationally expensive and the number of itemsets is very large.
- Using a single minimum support threshold may not be effective.

Multiple Minimum Support

- How to apply multiple minimum supports?
  - MS(i): minimum support for item i
  - e.g.: MS(Milk)=5%, MS(Coke) = 3%,
    MS(Broccoli)=0.1%, MS(Salmon)=0.5%
  - MS({Milk, Broccoli}) = min (MS(Milk), MS(Broccoli))
    = 0.1%
  - Challenge: Support is no longer anti-monotone
    - Suppose: Support(Milk, Coke) = 1.5% and
      Support(Milk, Coke, Broccoli) = 0.5%
    - {Milk,Coke} is infrequent but {Milk,Coke,Broccoli} is frequent

Pattern Evaluation

- Association rule algorithms tend to produce too many rules.
  - many of them are uninteresting or redundant.
  - Redundant IF {A,B,C} → {D} and {A,B} → {D} have SAME support & confidence.

- Interestingness measures can be used to **prune/rank** the derived patterns.

- In the original formulation of association rules, support & confidence are the only measures used.

Types of Interestingness Measures

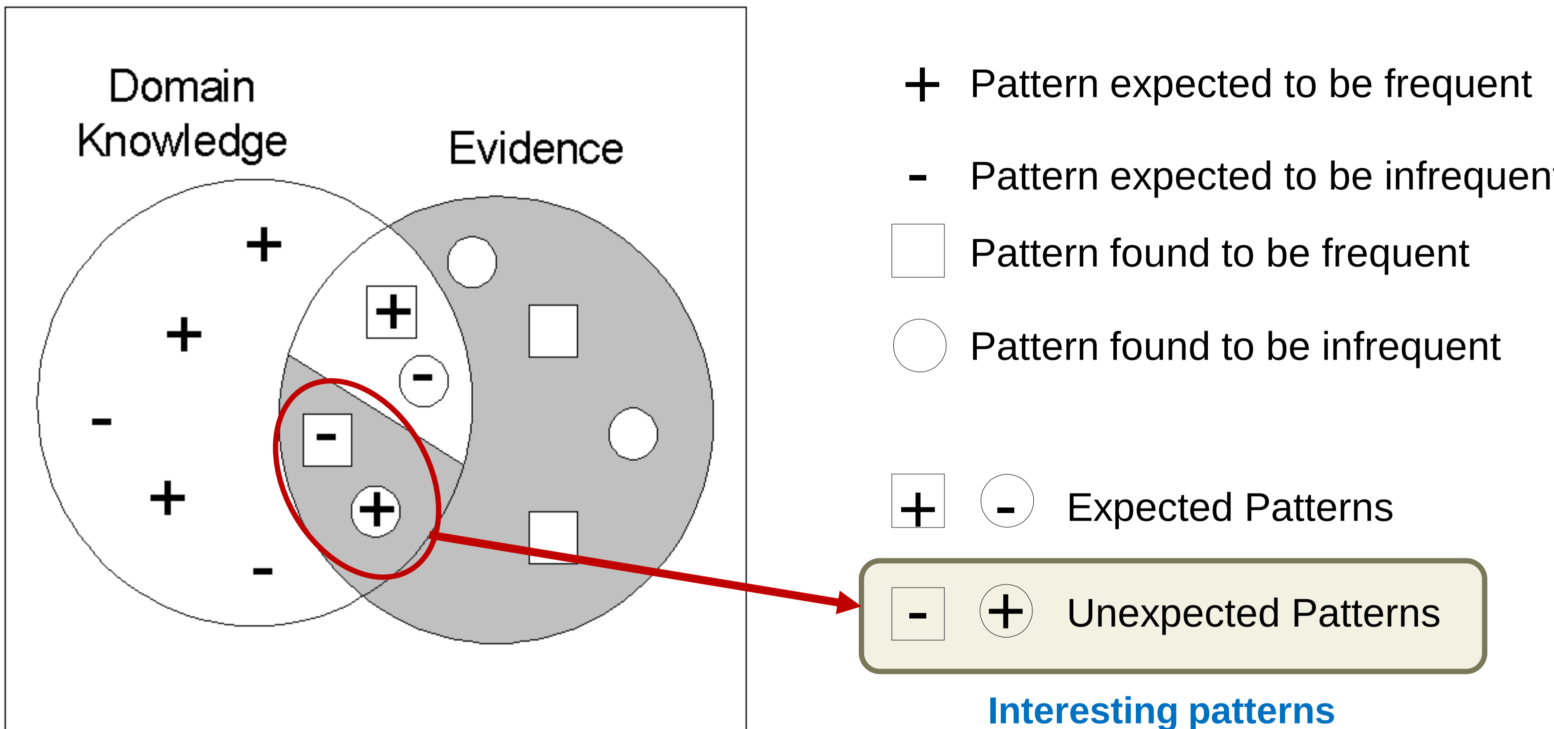
- **Objective measure:**
  - Rank patterns based on **statistics** computed from data
  - e.g., 21 measures of association (support, confidence, Laplace, Gini, mutual information, Jaccard, etc).

- **Subjective measure:**
  - Rank patterns according to **user's interpretation**
    - A pattern is subjectively interesting IF it CONTRADICTS the EXPECTATION of a user (Silberschatz & Tuzhilin).
    - A pattern is subjectively interesting IF it is ACTIONABLE (Silberschatz & Tuzhilin).

Asst. Prof. Dr. Rachsuda Setthawong

Interestingness via Unexpectedness

- Need to model expectation of users (domain knowledge)



Computing Interestingness Measure

- Given a rule $X \rightarrow Y$, information needed to compute **rule interestingness** can be obtained from a contingency table

Contingency table for $X \rightarrow Y$

	Y	$\bar{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\bar{X}$	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	N

f_{11}: frequency of X and Y
f_{10}: **frequency** of X and $\bar{Y}$
f_{01}: **frequency** of $\bar{X}$ and Y
f_{00}: **frequency** of $\bar{X}$ and $\bar{Y}$

Used to define various measures

- support, confidence, lift, Gini, J-measure, etc.

Drawback of Confidence

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	65	15	80
	80	20	100

Association Rule: Tea $\rightarrow$ Coffee

Confidence= P(Coffee|Tea) = 15/20 = 0.75

but P(Coffee) = 80/100 = 0.8

$\Rightarrow$ Although confidence is high, rule is misleading

$\Rightarrow$ P(Coffee|$\overline{\text{Tea}}$) =65/80 = 0.8125

Measure for Association Rules

- So, what kind of rules do we really want?
  - Confidence($X \rightarrow Y$) should be sufficiently high
    - To ensure that people who buy X will more likely buy Y than not buy Y
  - Confidence($X \rightarrow Y$) > support(Y)
    - Otherwise, rule will be misleading because having item X actually reduces the chance of having item Y in the same transaction
    - Is there any measure that capture this constraint?
      - Answer: Yes. There are many of them.

Lift and Interest Factor

$$Lift = \frac{c(A \rightarrow B)}{s(B)}$$

For binary variables,

$$Interest\ Factor = I(A,B) = \frac{s(A,B)}{s(A) \times s(B)} = \frac{N f_{11}}{f_{1+} f_{+1}}$$

	Y	Y	
X	f_{11}	f_{10}	f_{1+}
X	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	N

$$I(A,B) \begin{cases} = 1, & if\ A\ and\ B\ are\ independent; \\ > 1, & if\ A\ and\ B\ are\ positively\ correlated; \\ < 1, & if\ A\ and\ B\ are\ negatively\ correlated. \end{cases}$$

Remark: Lift is used for rules.
Interest factor is used for itemsets.

For the tea-coffee example,

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	65	15	80
	80	20	100

	Y	Y	
X	f_{11}	f_{10}	f_{1+}
X	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	N

Association Rule: Tea → Coffee

Confidence= P(Coffee|Tea) = 15/20 = 0.75

but P(Coffee) = 80/100 = 0.8

$$I = \frac{0.15}{0.2 \times 0.8} = 0.9375$$

Lift = 0.75 / 0.8 = 0.9375 **(negative correlated)**

The rule Tea → Coffee is pruned if considering lift result.

Correlation Analysis

- For continuous variables,

  Pearson's correlation coefficient (Ch. 2, pg. 77)

- For binary variables,

$$\phi = \frac{f_{11}f_{00} - f_{01}f_{10}}{\sqrt{f_{1+}f_{+1}f_{0+}f_{+0}}}$$

	Y	Y	
X	f_{11}	f_{10}	f_{1+}
X	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	N

$$\phi(A,B)\begin{cases} = 0, & \textit{if A and B are statistically independent;} \\ > 0, & \textit{if A and B are positively correlated;} \\ < 0, & \textit{if A and B are negatively correlated.} \end{cases}$$

For the tea-coffee example,

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	65	15	80
	80	20	100

	Y	Y	
X	f_{11}	f_{10}	f_{1+}
X	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	N

$$\phi = \frac{f_{11}f_{00} - f_{01}f_{10}}{\sqrt{f_{1+}f_{+1}f_{0+}f_{+0}}}$$

$$\phi(A,B) = \frac{15 \times 15 - 65 \times 5}{\sqrt{20 \times 80 \times 80 \times 20}} = -0.0625$$

(slightly negative correlation)

Asst. Prof. Dr. Rachsuda Setthawong

IS Measure
(for asymmetric binary variables)

$$IS(A,B) = \sqrt{I(A,B) \times s(A,B)} = \frac{s(A,B)}{\sqrt{s(A)s(B)}}$$

No. of documents having words co-occurences ({p, q} or {r, s})

	p	$\overline{p}$	
q	88	5	93
$\overline{q}$	5	2	7
	93	7	100

	r	$\overline{r}$	
s	2	5	7
$\overline{s}$	5	88	93
	7	93	100

IS(p, q) = .88 / sqrt(.93 x .93) = 0.946

IS(r, s) = .02 / sqrt(.07 x .07) = 0.286

Association of {p, q} is stronger than {r, s} (more interesting).

Asst. Prof. Dr. Rachsuda Setthawong

Alternative Objective Interestingness Measures—symmetric objective measures for the itemset {A, B}

Measure (Symbol)	Definition
Correlation	$\phi = \frac{Nf_{11} - f_{1+}f_{+1}}{\sqrt{f_{1+}f_{+1}f_{0+}f_{+0}}}$
Odds ratio	$\alpha = \frac{f_{11}f_{00}}{f_{10}f_{01}}$
Kappa	$\kappa = \frac{Nf_{11} + Nf_{00} - f_{1+}f_{+1} - f_{0+}f_{+0}}{N^2 - f_{1+}f_{+1} - f_{0+}f_{+0}}$
Interest	$I = \frac{Nf_{11}}{f_{1+}f_{+1}}$
Cosine	$IS = \frac{f_{11}}{\sqrt{f_{1+}f_{+1}}}$
Piatetsky-Shapiro	$PS = \frac{f_{11}}{N} - \frac{f_{1+}f_{+1}}{N^2}$
Collective strength	$S = \frac{f_{11} + f_{00}}{f_{1+}f_{+1} + f_{0+}f_{+0}} \times \frac{N - f_{1+}f_{+1} - f_{0+}f_{+0}}{N - f_{11} - f_{00}}$
Jaccard	$\zeta = \frac{f_{11}}{f_{1+} + f_{+1} - f_{11}}$
All-confidence	$h = min\left[\frac{f_{11}}{f_{1+}}, \frac{f_{11}}{f_{+1}}\right]$

Asst. Prof. Dr. Rachsuda Setthawong

Alternative Objective Interestingness Measures—asymmetric objective measures for the itemset {A, B}

Measure (Symbol)	Definition
Goodman-Kruskal	$\lambda = \frac{\sum_j max_k f_{jk} - max_k f_{+k}}{N - max_k f_{+k}}$
Mutual Information	$M = \frac{\sum_i \sum_j \frac{f_{ij}}{N} log \frac{N f_{ij}}{f_{i+} f_{+j}}}{\sum_i \frac{f_{i+}}{N} log \frac{f_{i+}}{N}}$
J-Measure	$J = \frac{f_{11}}{N} log \frac{N f_{11}}{f_{1+} f_{+1}} + \frac{f_{10}}{N} log \frac{N f_{10}}{f_{1+} f_{+0}}$
Gini index	$G = \frac{f_{1+}}{N} \times \left[(\frac{f_{11}}{f_{1+}})^2 + (\frac{f_{10}}{f_{1+}})^2 \right] - (\frac{f_{+1}}{N})^2 + \frac{f_{0+}}{N} \times \left[(\frac{f_{01}}{f_{0+}})^2 + (\frac{f_{00}}{f_{0+}})^2 \right] - (\frac{f_{+0}}{N})^2$
Laplace	$L = \frac{f_{11} + 1}{f_{1+} + 2}$
Coviction	$V = \frac{f_{1+} f_{+0}}{N f_{10}}$
Certainty factor	$F = (\frac{f_{11}}{f_{1+}} - \frac{f_{+1}}{N}) / (1 - \frac{f_{+1}}{N})$
Added Value	$AV = (\frac{f_{11}}{f_{1+}} - \frac{f_{+1}}{N})$

References

- Lecture Notes for Chapter 9, Introduction to Data Mining, by, Tan, Steinbach, Kumar
- Measuring quality of association rules
  - http://michael.hahsler.net/research/association_rules/measures.html