Association Analysis
Part 1

Dr. Sanjay Ranka
Professor
Computer and Information Science and Engineering
University of Florida

Mining Associations

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

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:

TID	Bread	Milk	Diaper	Beer	Eggs	Coke
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Definition of Association Rule

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

Association Rule: $X \stackrel{s,c}{\Longrightarrow} y$

Support: $s = \dfrac{\sigma(X \cup y)}{|T|} \quad (s = P(X, y))$

Confidence: $c = \dfrac{\sigma(X \cup y)}{\sigma(X)} \quad (c = P(y \mid X))$

Goal:

Discover all rules having support $\geq$ *minsup* and confidence $\geq$ *minconf* thresholds.

Example: $\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$

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

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

How to Mine Association Rules?

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:

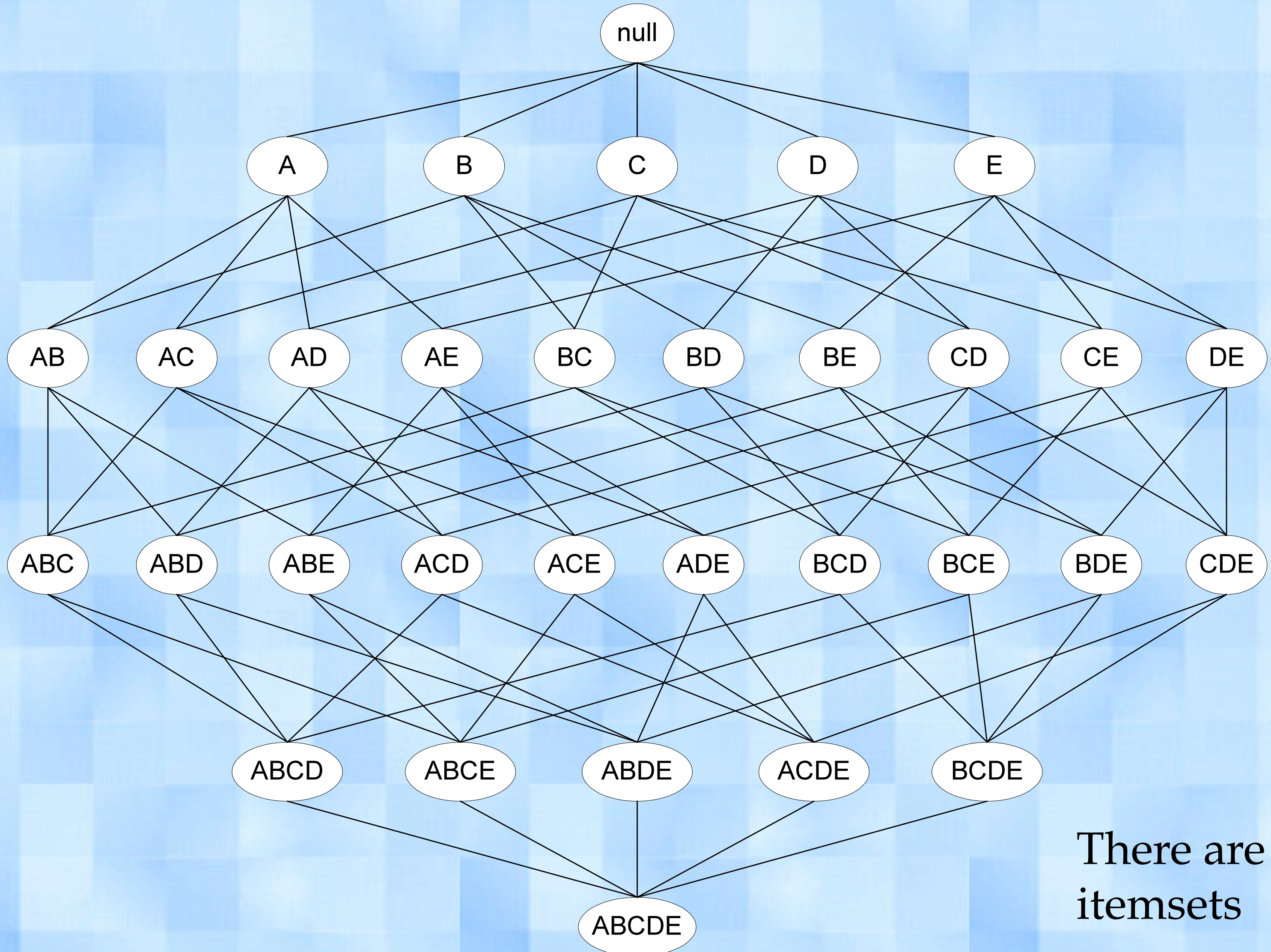
- All the rules above correspond to the same itemset: {Milk, Diaper, Beer}
- Rules obtained from the same itemset have identical support but can have different confidence

How to Mine Association Rules?

- Two step approach:
  1. Generate all frequent itemsets (sets of items whose support > *minsup*)
  2. Generate high confidence association rules from each frequent itemset
     - Each rule is a binary partition of a frequent itemset

- Frequent itemset generation is more expensive operation

Itemset Lattice



There are 2^d possible itemsets

Generating Frequent Itemsets

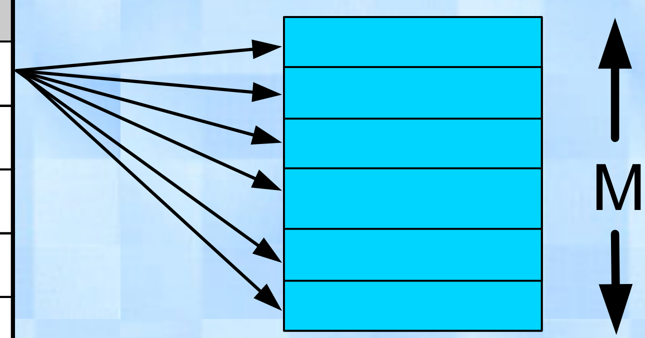
- Naive approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database

Transactions

N

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

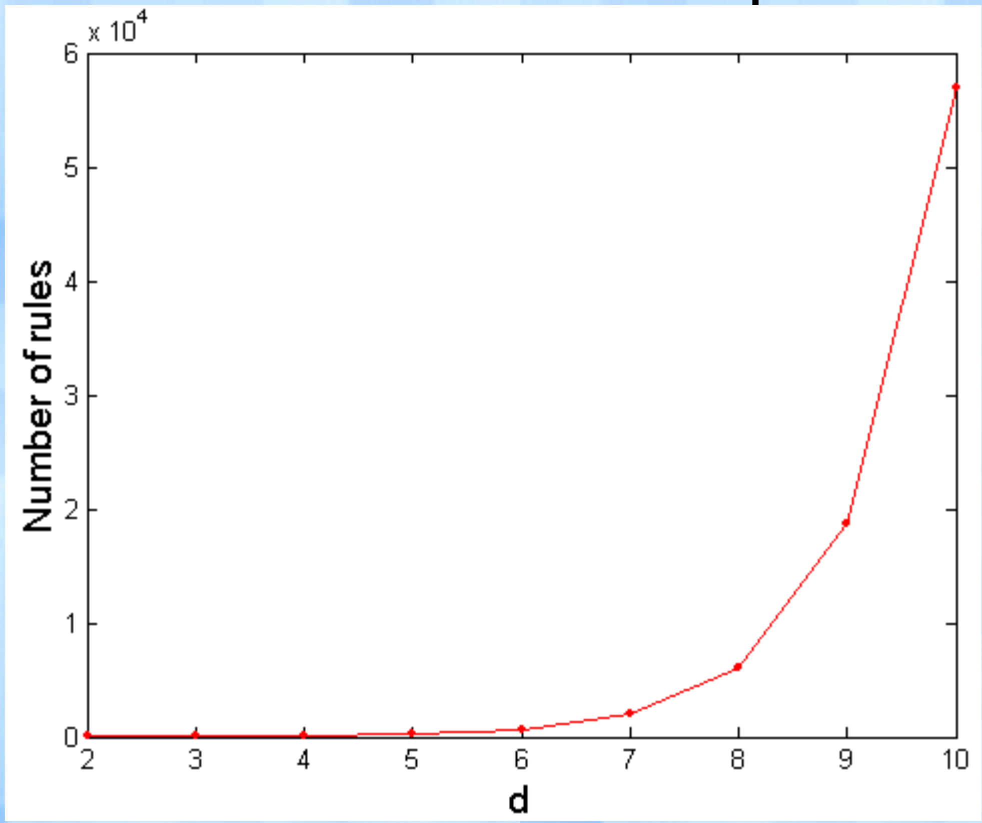
Candidates



  - Complexity ~ O(NM) => Expensive since $M = 2^d$!!!

Computational Complexity

- Given d unique items:
  - Total number of itemsets = 2^d
  - Total number of possible association rules:



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$
$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

Approach for Mining Frequent Itemsets

- Reduce the number of candidates (M)
  - Complete search: $M=2^d$
  - Use Apriori heuristic to reduce M
- Reduce the number of transactions (N)
  - Reduce size of N as the size of itemset increases
  - Used by DHP and vertical-based mining algorithms
- 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

Reducing Number of Candidates

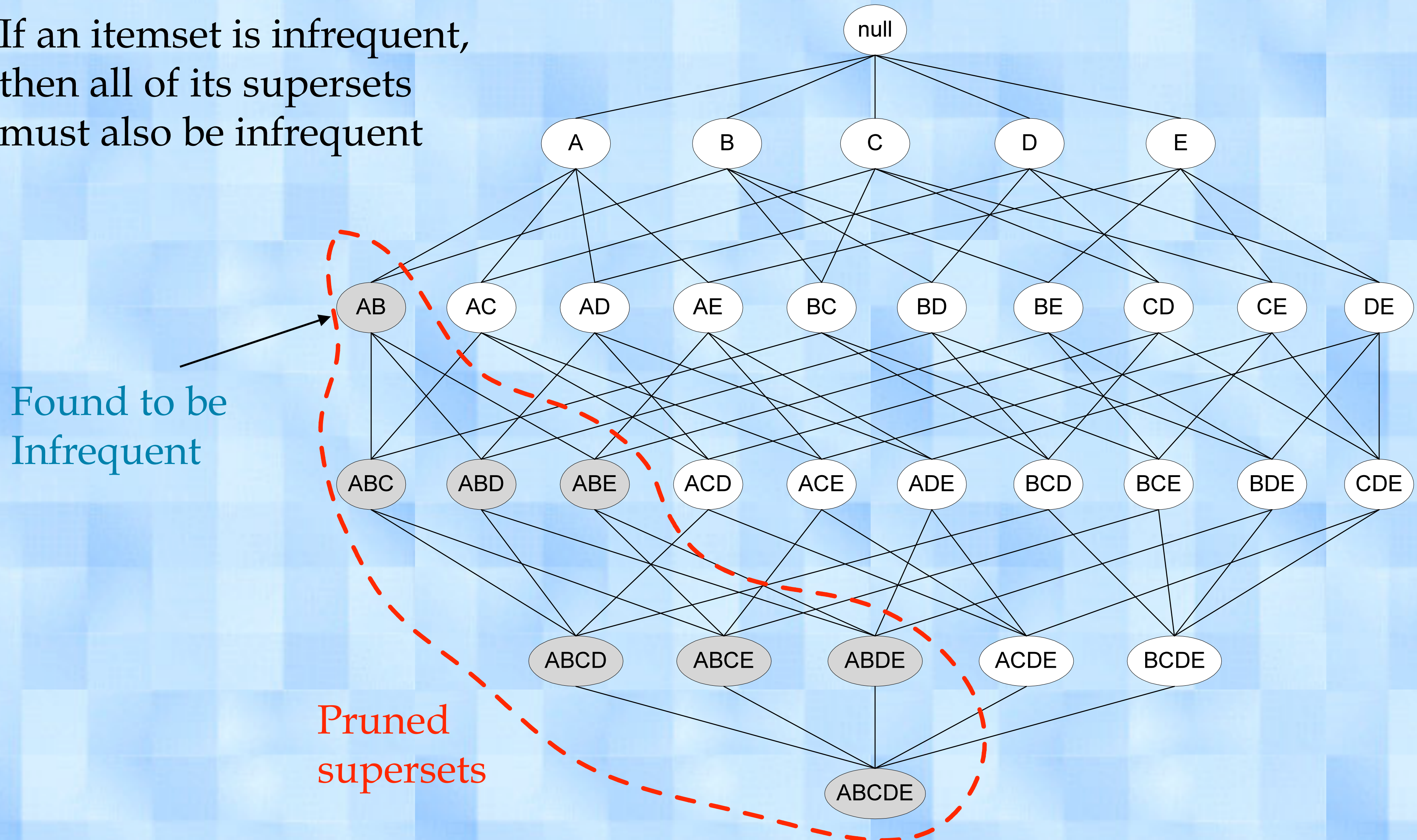
- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent
- Apriori principle holds due to the following property of the support measure:

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

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

Using Apriori Principle for Pruning Candidates

If an itemset is infrequent, then all of its supersets must also be infrequent



Illustrating Apriori Principle

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

Items (1-itemsets)

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

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3

If every subset is considered,

$^6C_1 + {}^6C_2 + {}^6C_3 = 41$

With support-based pruning,

$6 + 6 + 1 = 13$

Reducing Number of Comparisons

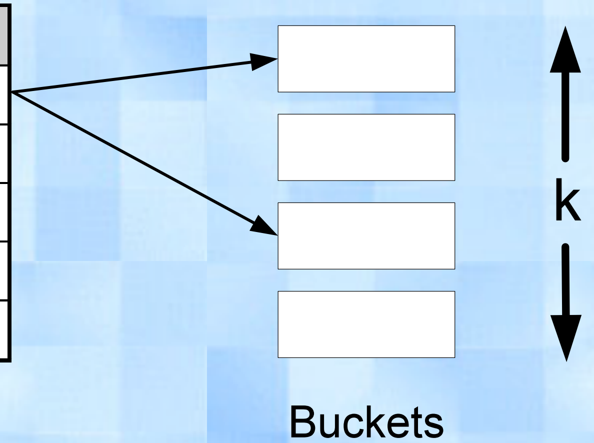
- Candidate counting:
  - Scan the database of transactions to determine the support of candidate itemsets
  - To reduce number of comparisons, store the candidates using a hash structure

Transactions

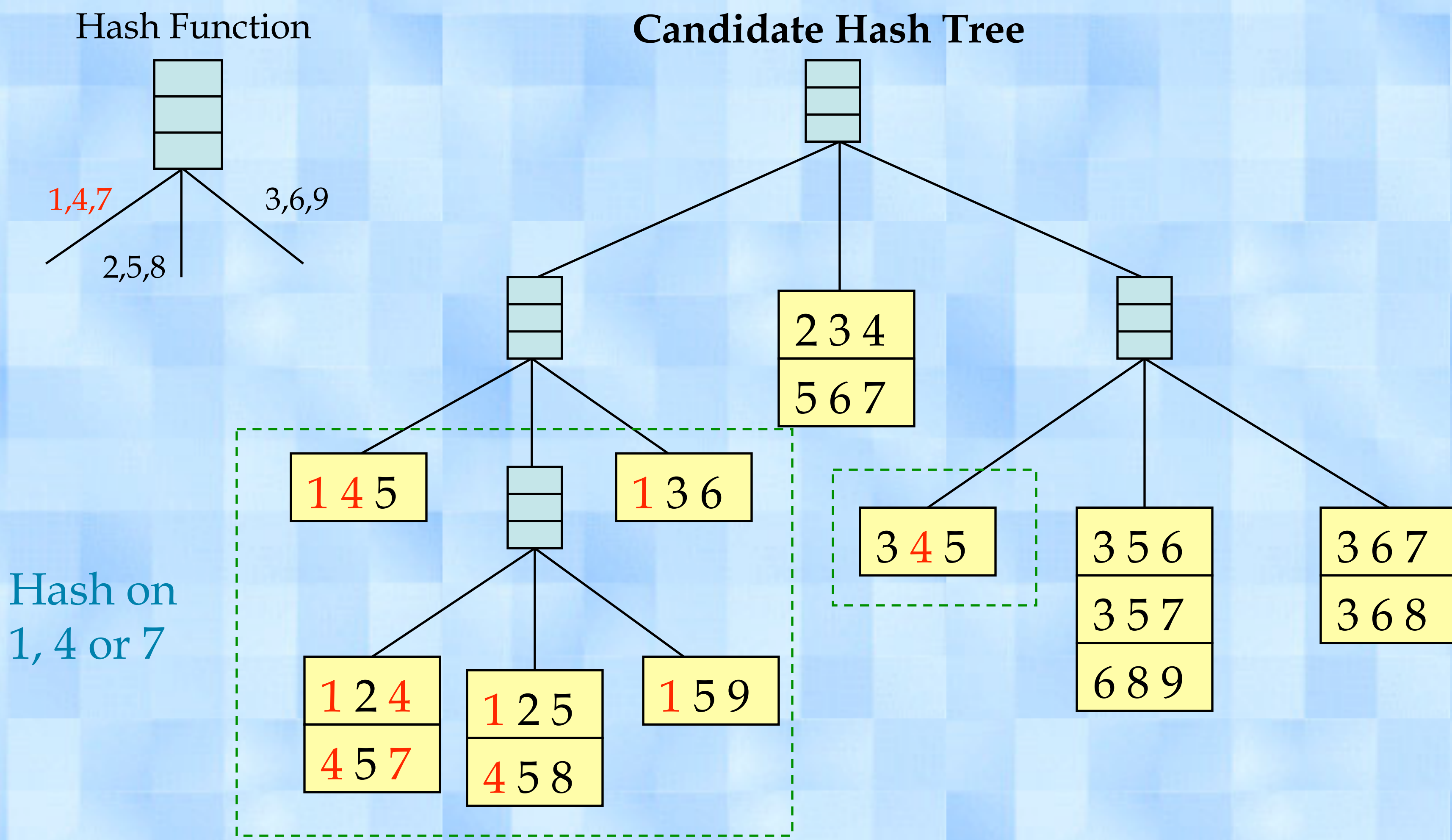
N

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**

Hash Structure

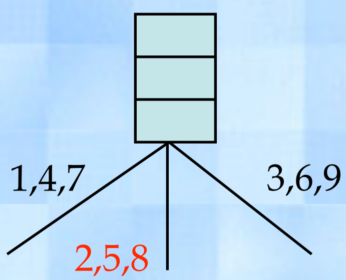


Association Rule Discovery: Hash Tree for Fast Access

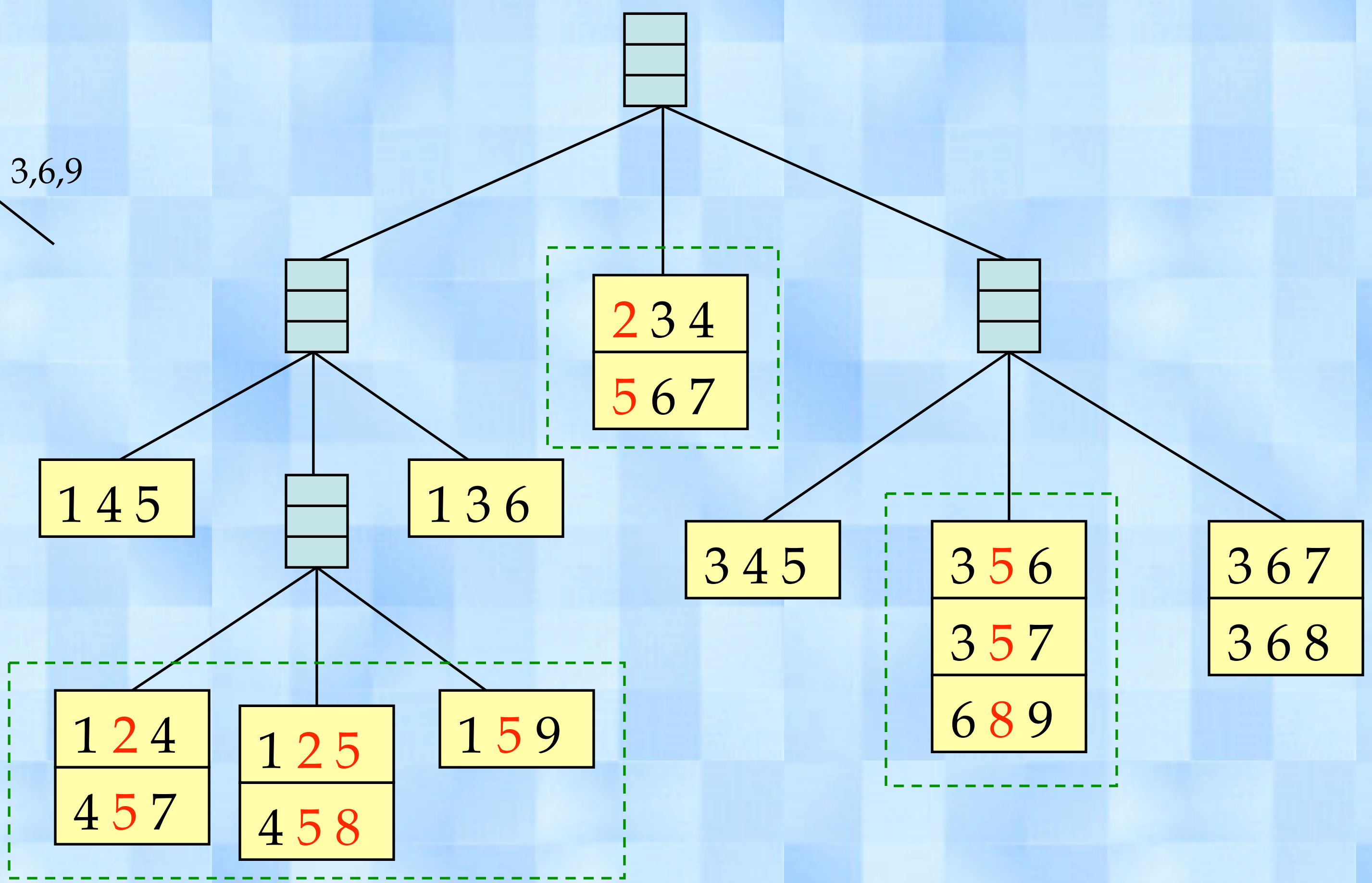


Association Rule Discovery: Hash Tree for Fast Access

Hash Function

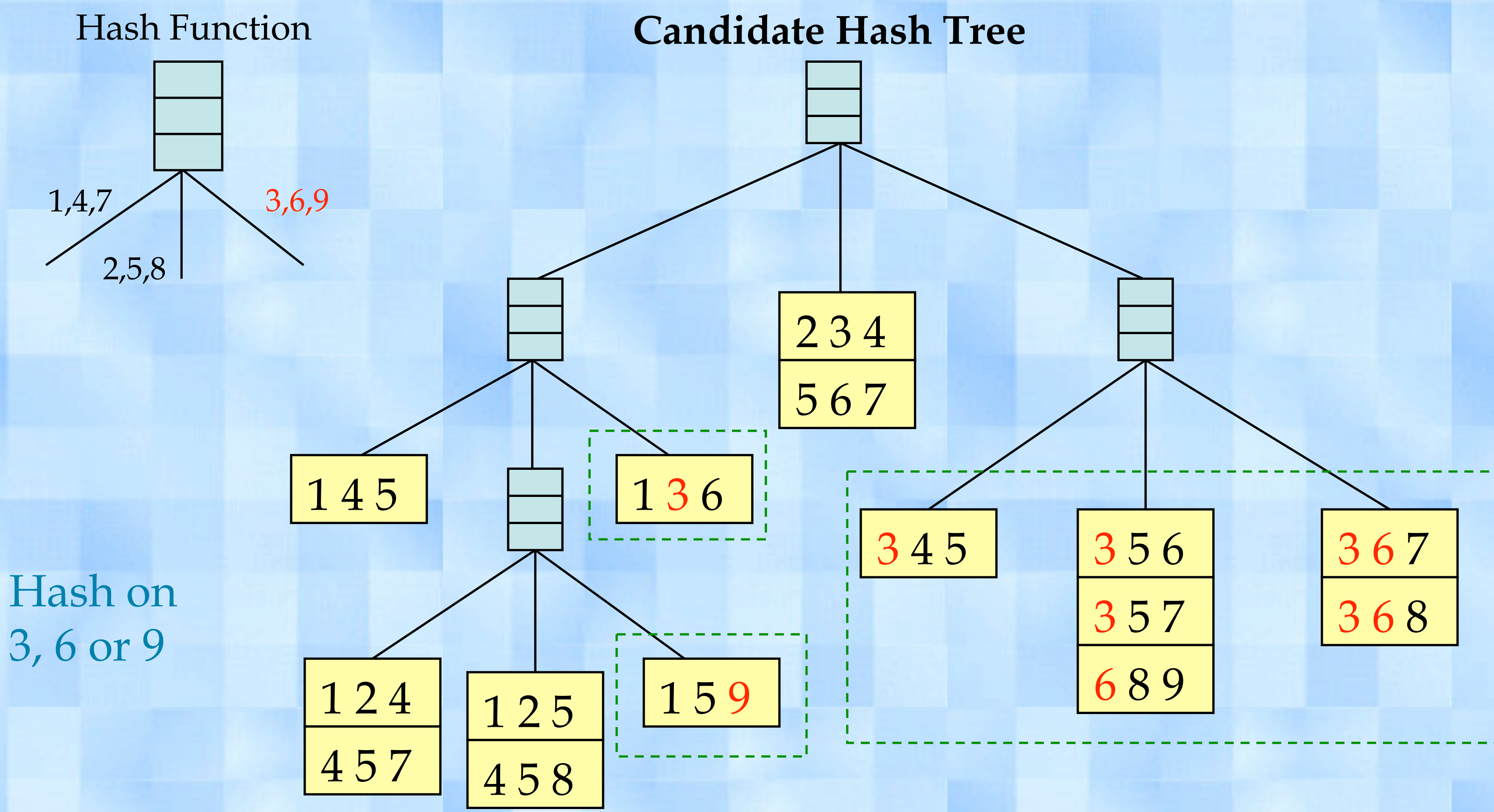


Candidate Hash Tree



Hash on
2, 5 or 8

Association Rule Discovery: Hash Tree for Fast Access



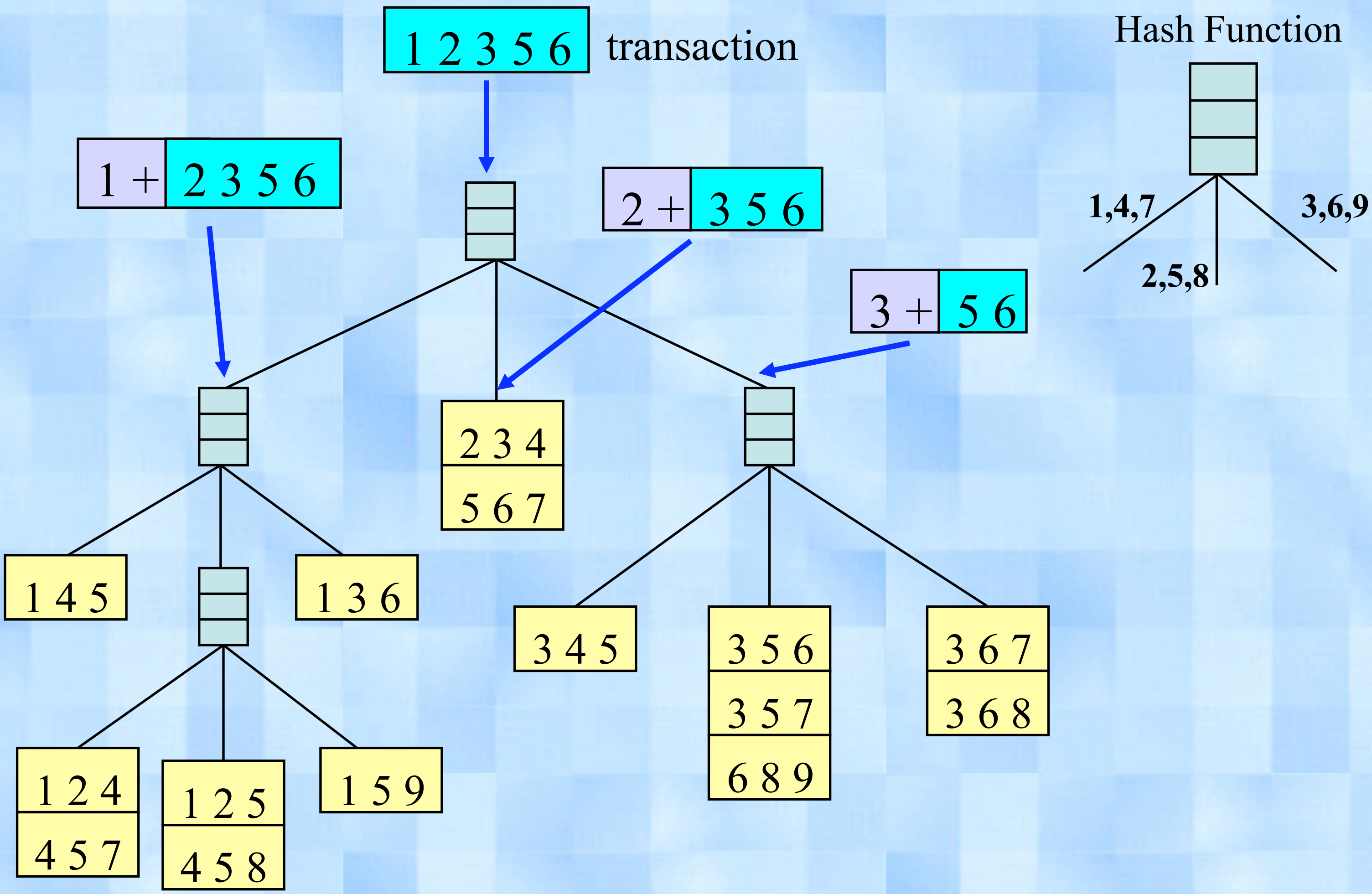
Candidate Counting

- Given a transaction L = {1,2,3,5,6}
- Possible subsets of size 3:

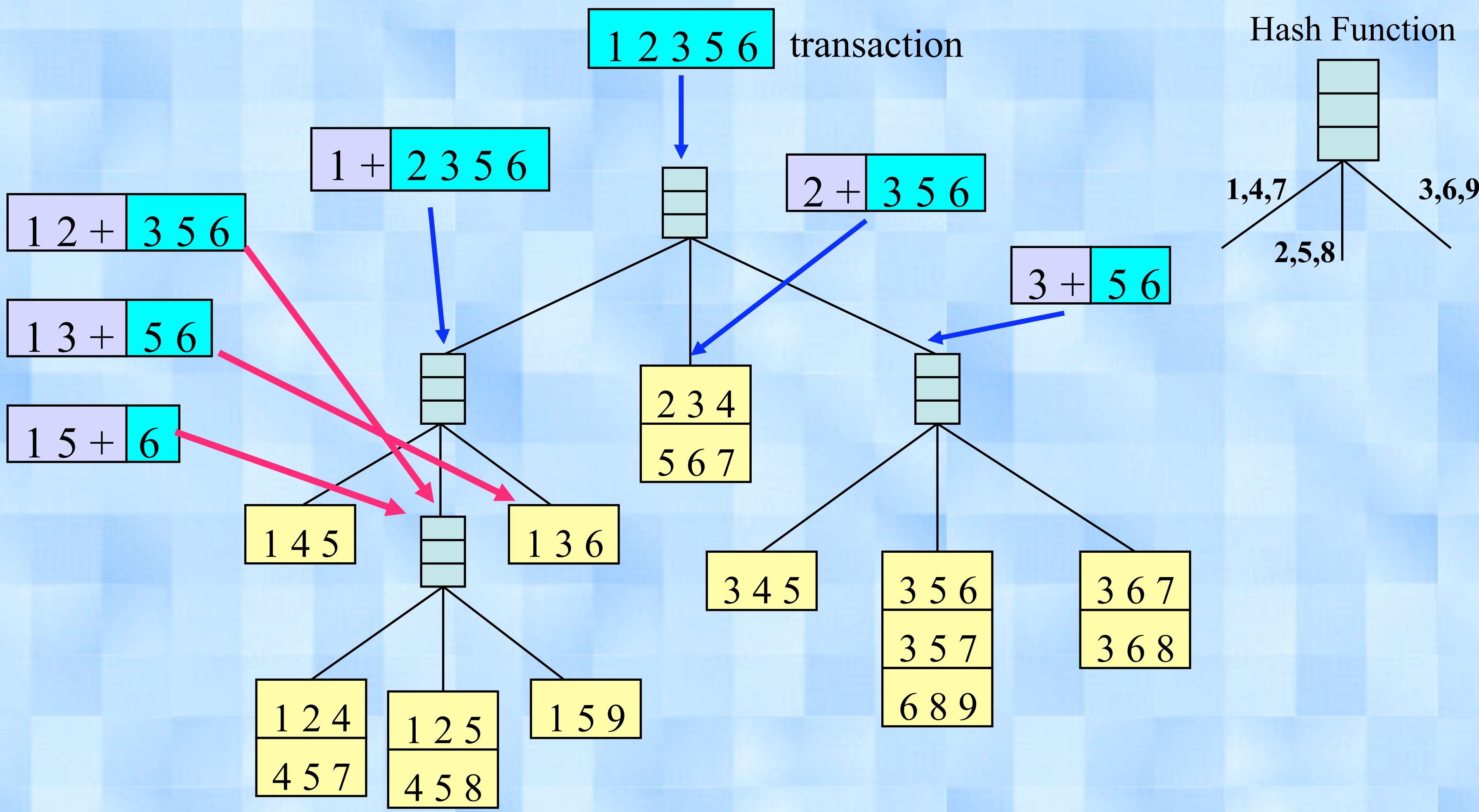
{1,2,3}	{2,3,5}	{3,5,6}
{1,2,5}	{2,3,6}	
{1,2,6}	{2,5,6}	
{1,3,5}		
{1,3,6}		
{1,5,6}		

- If width of transaction is w, there are 2^w-1 possible non-empty subsets

Association Rule Discovery: Subset Operation



Association Rule Discovery: Subset Operation …



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 →D,	ABD →C,	ACD →B,	BCD →A,
A →BCD,	B →ACD,	C →ABD,	D →ABC
AB →CD,	AC → BD,	AD → BC,	BC →AD,
BD →AC,	CD →AB,		

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

Rule Generation

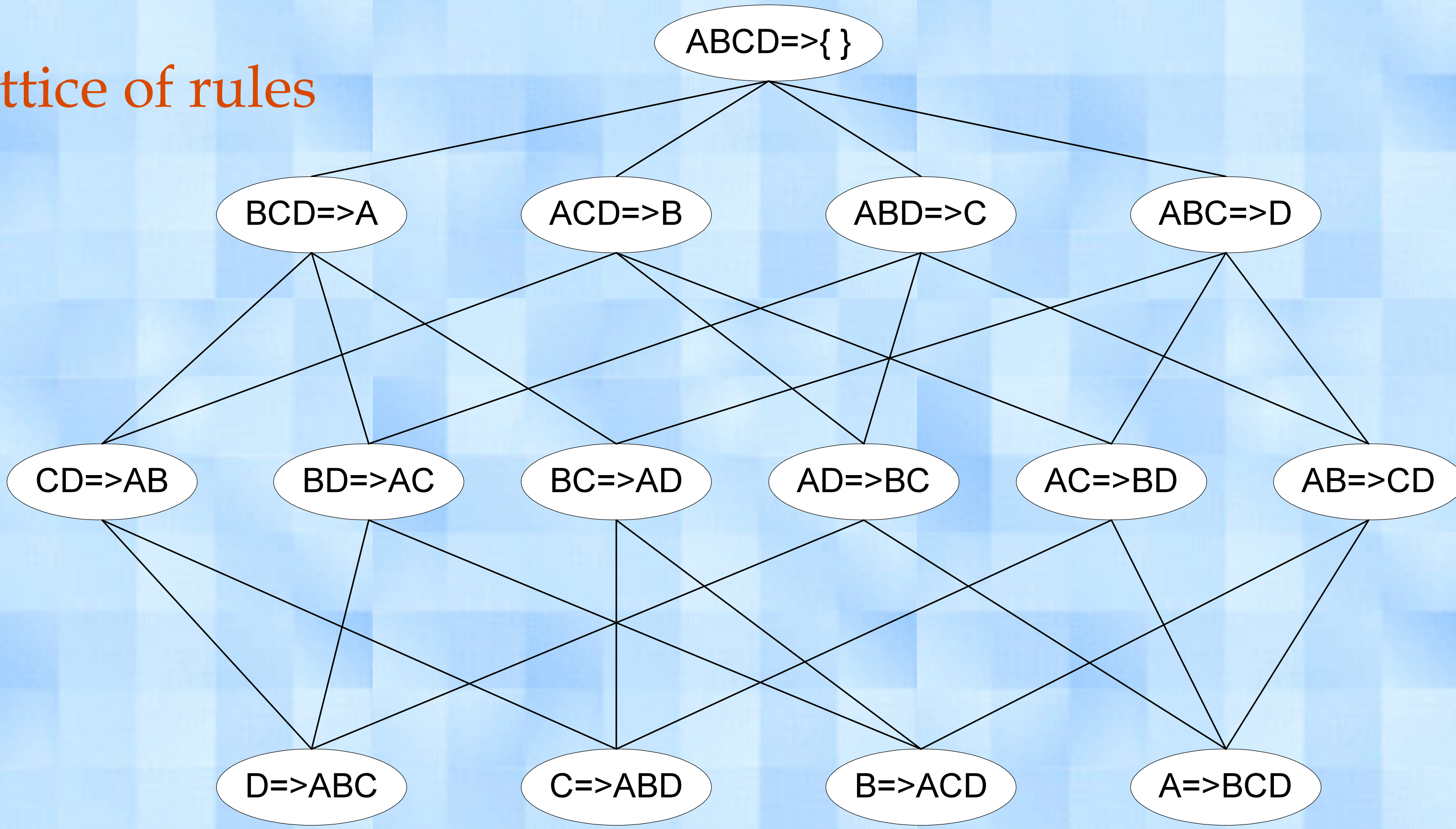
- How to efficiently generate rules from frequent itemsets?
  - In general, confidence does not have an anti-monotone property
  - But confidence of rules generated from the same itemset has an anti-monotone property
  - L = {A,B,C,D}:

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

  - Confidence is non-increasing as number of items in rule consequent increases

Rule Generation for Apriori Algorithm

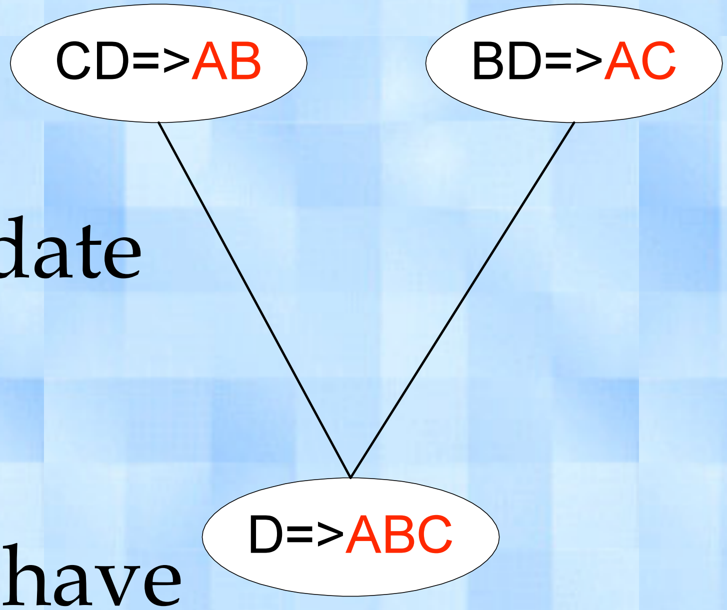
Lattice of rules



- Lattice corresponds to partial order of items in the rule consequent

Rule Generation for Apriori Algorithm …

- Candidate rule is generated by merging two rules that share the same prefix in the rule consequent
- join(CD=>AB,BD=>AC) would produce the candidate rule D => ABC
- Prune rule D=>ABC if its subset AD=>BC does not have high confidence



Other Frequent Itemset Algorithms

- Traversal of Itemset Lattice
  - Apriori uses breadth-first (level-wise) traversal
- Representation of Database
  - Apriori uses horizontal data layout
- Generate-and-count paradigm