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, Butter, Coffee, Eggs**
3	**Milk, Butter, Coffee, Coke**
4	**Bread, Milk, Butter, Coffee**
5	**Bread, Milk, Butter, Coke**

Example of Association Rules

{Butter} $\rightarrow$ {Coffee},
{Milk, Bread} $\rightarrow$ {Eggs,Coke},
{Coffee, Bread} $\rightarrow$ {Milk},

Implication means co-occurrence, not causality!

Definition: Frequent Itemset

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

TID	*Items*
1	Bread, Milk
2	Bread, Butter, Coffee, Eggs
3	Milk, Butter, Coffee, Coke
4	Bread, Milk, Butter, Coffee
5	Bread, Milk, Butter, Coke

Definition: Association Rule

- Association Rule
  - An implication expression of the form X → Y, where X and Y are itemsets
  - Example:
    {Milk, Butter} → {Coffee}

- Rule Evaluation Metrics
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

TID	*Items*
1	Bread, Milk
2	Bread, Butter, Coffee, Eggs
3	Milk, Butter, Coffee, Coke
4	Bread, Milk, Butter, Coffee
5	Bread, Milk, Butter, Coke

Example:

$$\{\text{Milk}, \text{Butter}\} \Rightarrow \textit{Coffee}$$

$$s = \frac{\sigma(\text{Milk}, \textit{Butter}, \text{Coffee})}{|\,\text{T}\,|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Butter}, \text{Coffee})}{\sigma(\text{Milk}, \textit{Butter})} = \frac{2}{3} = 0.67$$

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
- 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!

Mining Association Rules

Example:

$$\{\text{Milk}, \text{Butter}\} \Rightarrow \textit{Coffee}$$

$$s = \frac{\sigma(\text{Milk}, \textit{Butter}, \text{Coffee})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Butter}, \text{Coffee})}{\sigma(\text{Milk}, \textit{Butter})} = \frac{2}{3} = 0.67$$

TID	*Items*
1	**Bread, Milk**
2	**Bread, Butter, Coffee, Eggs**
3	**Milk, Butter, Coffee, Coke**
4	**Bread, Milk, Butter, Coffee**
5	**Bread, Milk, Butter, Coke**

Example of Rules:

{Milk,Bread} $\rightarrow$ {Coffee} (s=1/5=0.2, c=1/3=0.33)
{Milk, Coffee} $\rightarrow$ {Bread} (s=1/5=0.2, c=1/2=0.5)
{Bread, Coffee} $\rightarrow$ {Milk} (s=1/5=0.2, c=1/2=0.5)
{Coffee} $\rightarrow$ {Milk,Bread} (s=1/5=0.2, c=1/3=0.33)
{Bread} $\rightarrow$ {Milk,Coffee} (s=1/5=0.2, c=1/4=0.25)
{Milk} $\rightarrow$ {Bread,Coffee} (s=1/5=0.2, c=1/4=0.25)

Example of Rules:

$\{Milk, Bread\} \rightarrow \{Coffee\}$ (s=1/5=0.2, c=1/3=0.33)
$\{Milk, Coffee\} \rightarrow \{Bread\}$ (s=1/5=0.2, c=1/2=0.5)
$\{Bread, Coffee\} \rightarrow \{Milk\}$ (s=1/5=0.2, c=1/2=0.5)
$\{Coffee\} \rightarrow \{Milk, Bread\}$ (s=1/5=0.2, c=1/3=0.33)
$\{Bread\} \rightarrow \{Milk, Coffee\}$ (s=1/5=0.2, c=1/4=0.25)
$\{Milk\} \rightarrow \{Bread, Coffee\}$ (s=1/5=0.2, c=1/4=0.25)

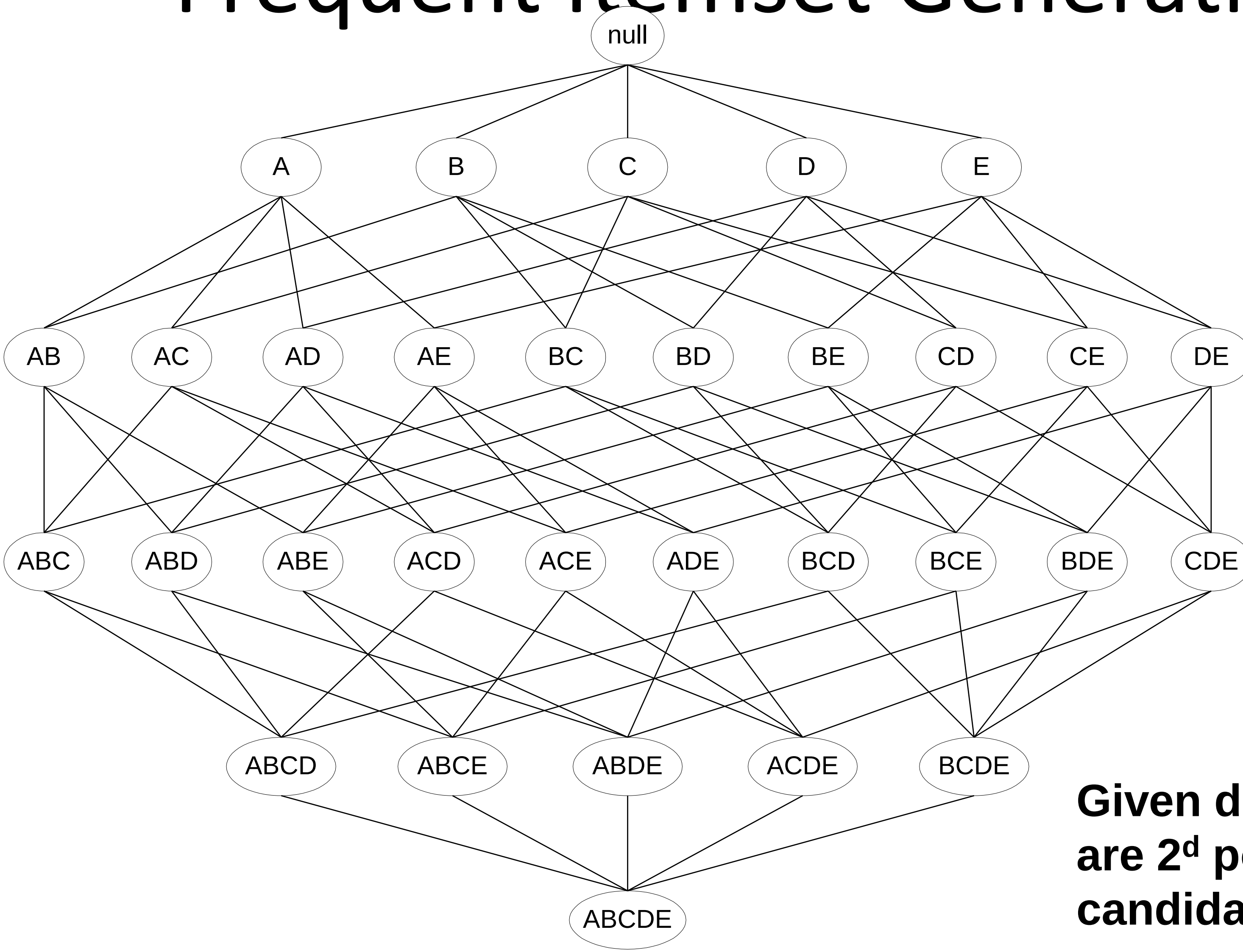
Observations:

- All the above rules are binary partitions of the same itemset:
  {Milk, Bread, Coffee}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may 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 is still computationally expensive

Frequent Itemset Generation



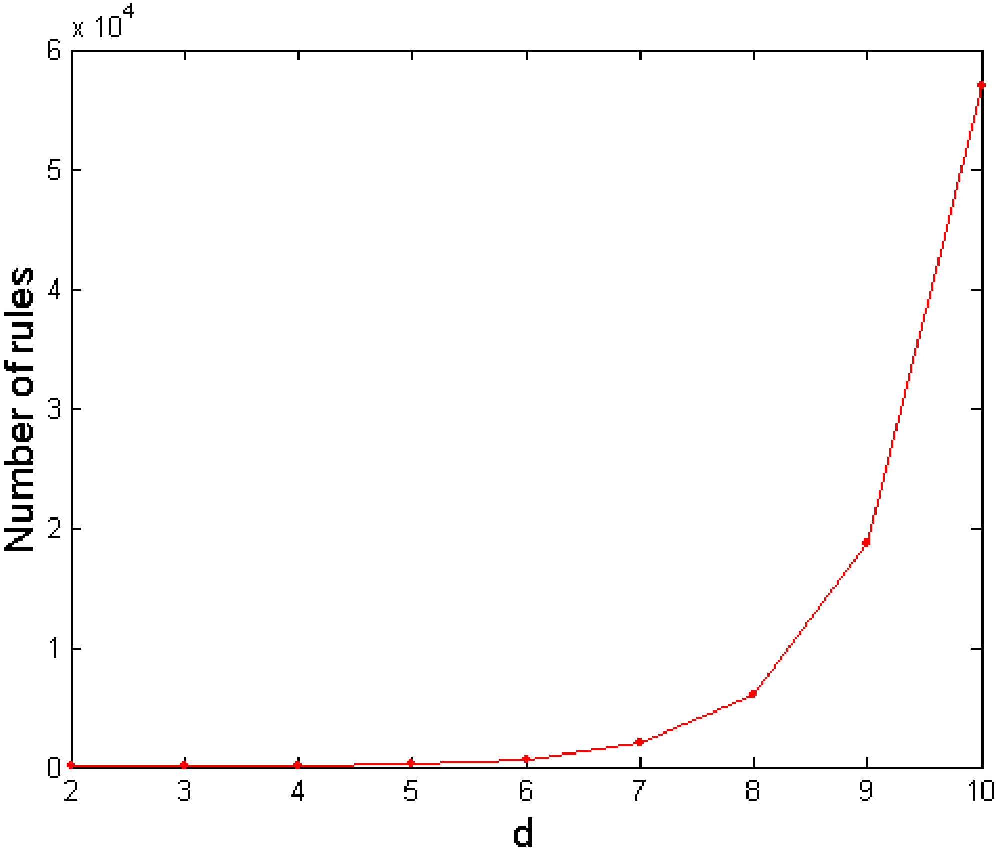
Given d items, there are 2^d possible candidate itemsets

Frequent Itemset Generation

- Brute-force approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database
  - N The number of transactions
  - M the list of candidates
  - W – The number of items (the width of the transaction)
  - Match each transaction against every candidate
  - Complexity ~ O(NMw) => 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

Frequent Itemset Generation Strategies

- Reduce the number of candidates (M)
  - Complete search: $M=2^d$
  - Use pruning techniques to reduce M
- Reduce the number of transactions (N)
  - Reduce size of N as the size of itemset increases
  - Used by vertical-dataset 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:

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

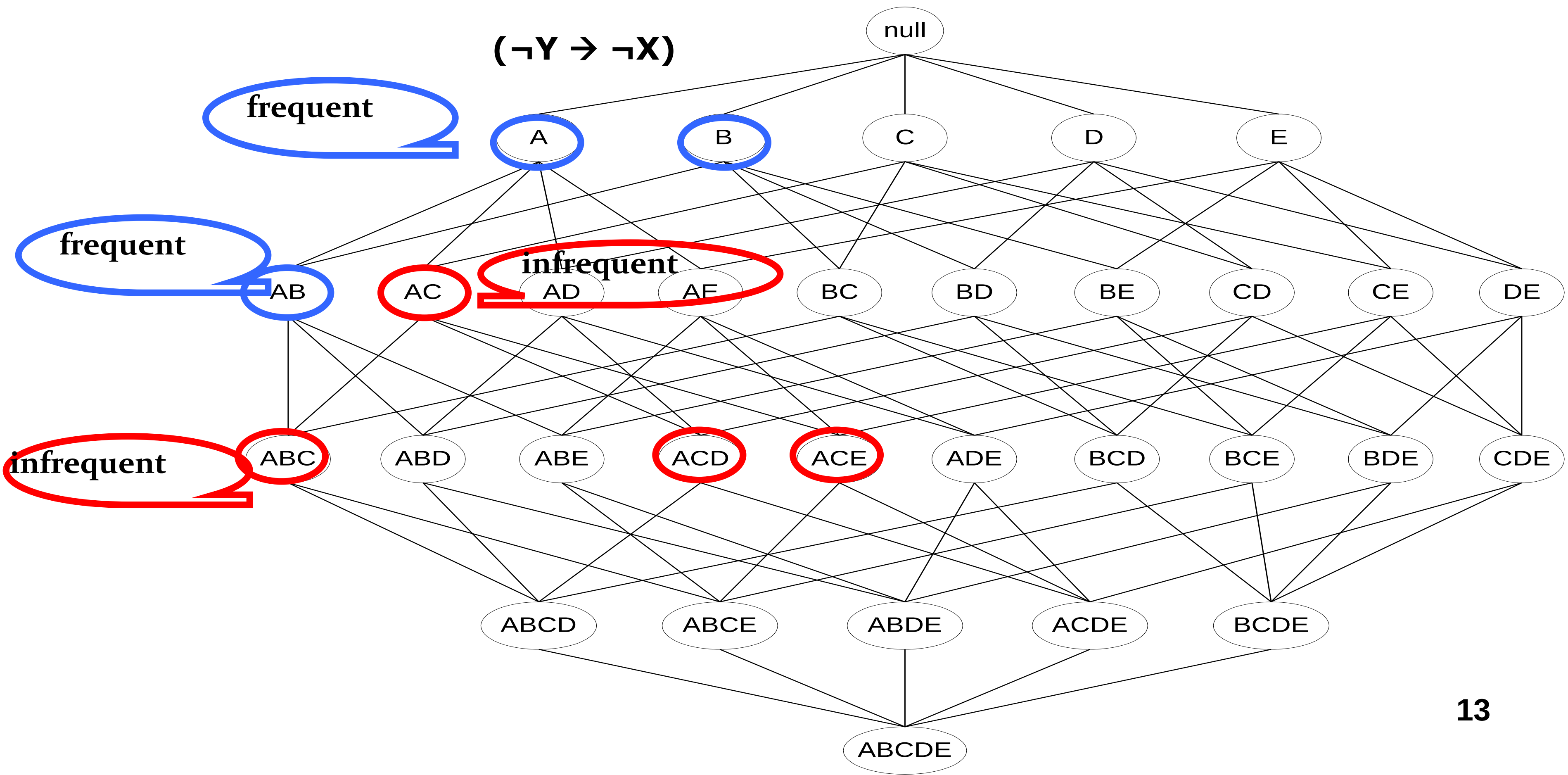
- For all X, Y in the dataset, support (X) >= Suport(Y)
  - Support of an itemset never exceeds the support of its subsets
  - This is known as the anti-monotone property of support

Apriori Principle

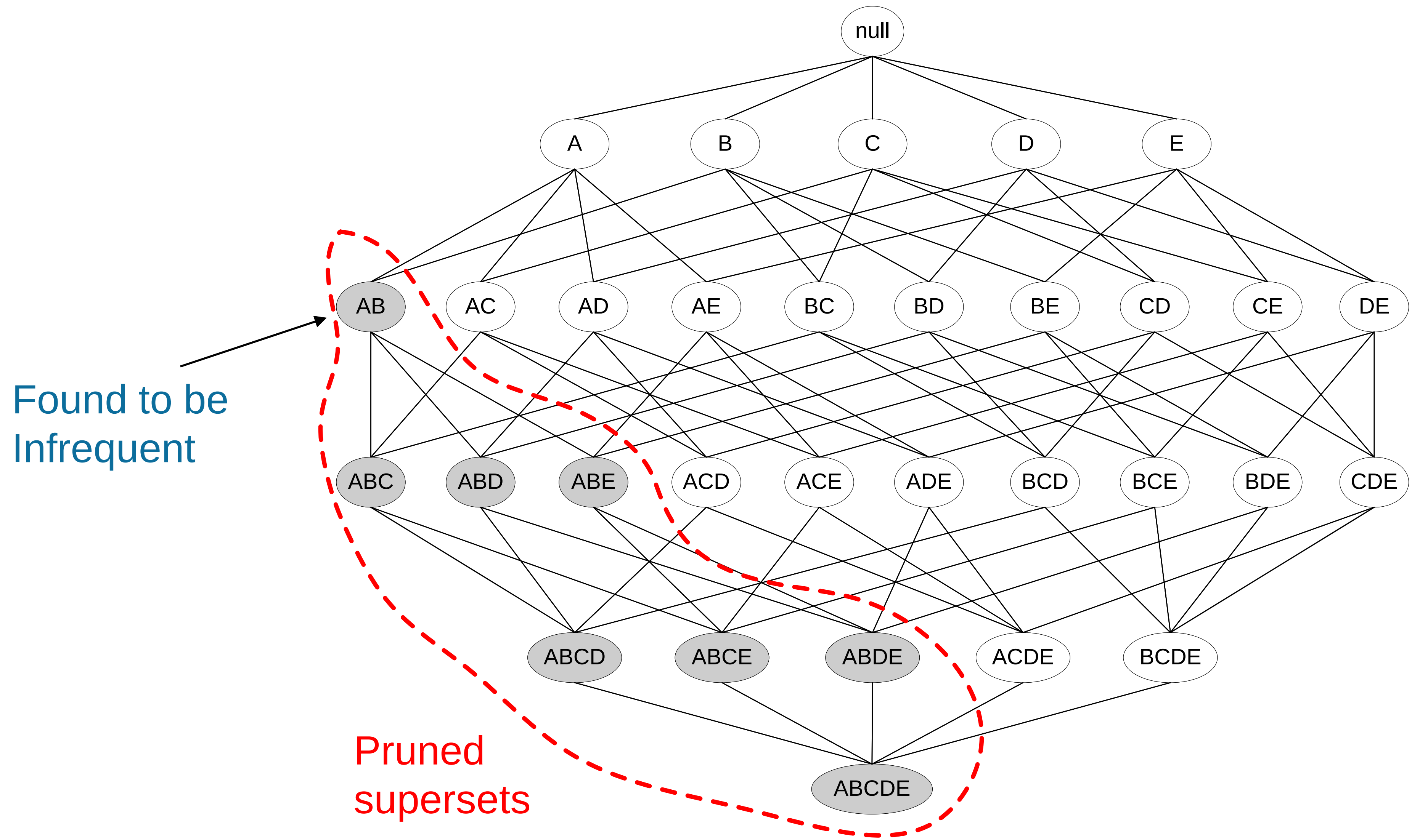
- If an itemset is frequent, then all of its subsets must also be frequent
- If an itemset is infrequent, then all of its supersets must be infrequent too

(X → Y)

(¬Y → ¬X)



Illustrating Apriori Principle



Illustrating Apriori Principle

Items (1-itemsets)

Item	Count
Bread	**4**
Coke	**2**
Milk	**4**
Coffee	**3**
Cookies	**4**
Eggs	**1**

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Itemset	Count
{Bread,Milk}	**3**
{Bread,Coffee}	**2**
{Bread,Cookies}	**3**
{Milk,Coffee}	**2**
{Milk, Cookies }	**3**
{Coffee, Cookies }	**3**

Minimum Support = 3

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Cookies}	**3**

If every subset is considered,

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

With support-based pruning,

$6 + 6 + 1 = 13$

$$n\,C\,r = n!/(r!)(n-r)!$$