Data Warehouse and Data Mining

Dhruv Gupta

dhruv.gupta@ntnu.no

14-February-2023

NTNU | Norwegian University of Science and Technology

1. Announcements and References
   - Administrative
   - References for Today's Lecture

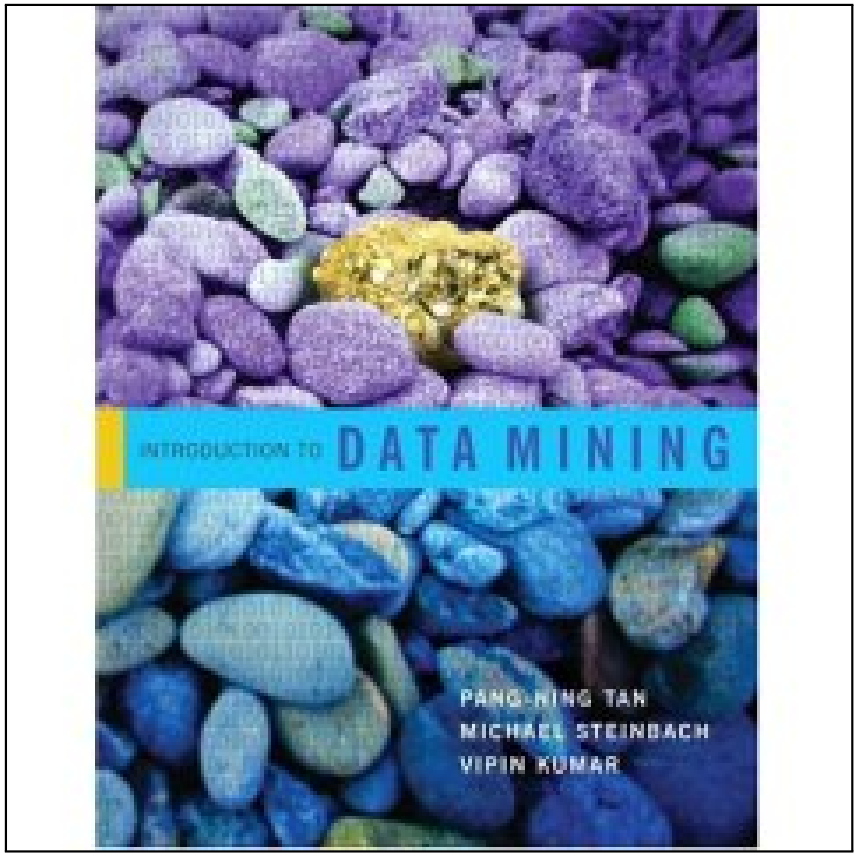
2. Association Rules
   - Association Rule Mining
   - Definitions
   - Problem Definition
   - Frequent Itemset Generation
   - The Apriori Algorithm
     - Efficient Candidate Generation
     - Complexity Factors
   - Rule Generation
   - Maximal and Closed Itemsets
   - Alternative Methods for Itemset Generation
     - FP-Growth Algorithm
     - ECLAT Algorithm
   - Interestingness Measures

Administrative

1. Second Assignment
   - Available and due by 23.February.2023.
2. Volunteers for feedback regarding course
   - Interested? Please contact me by email!

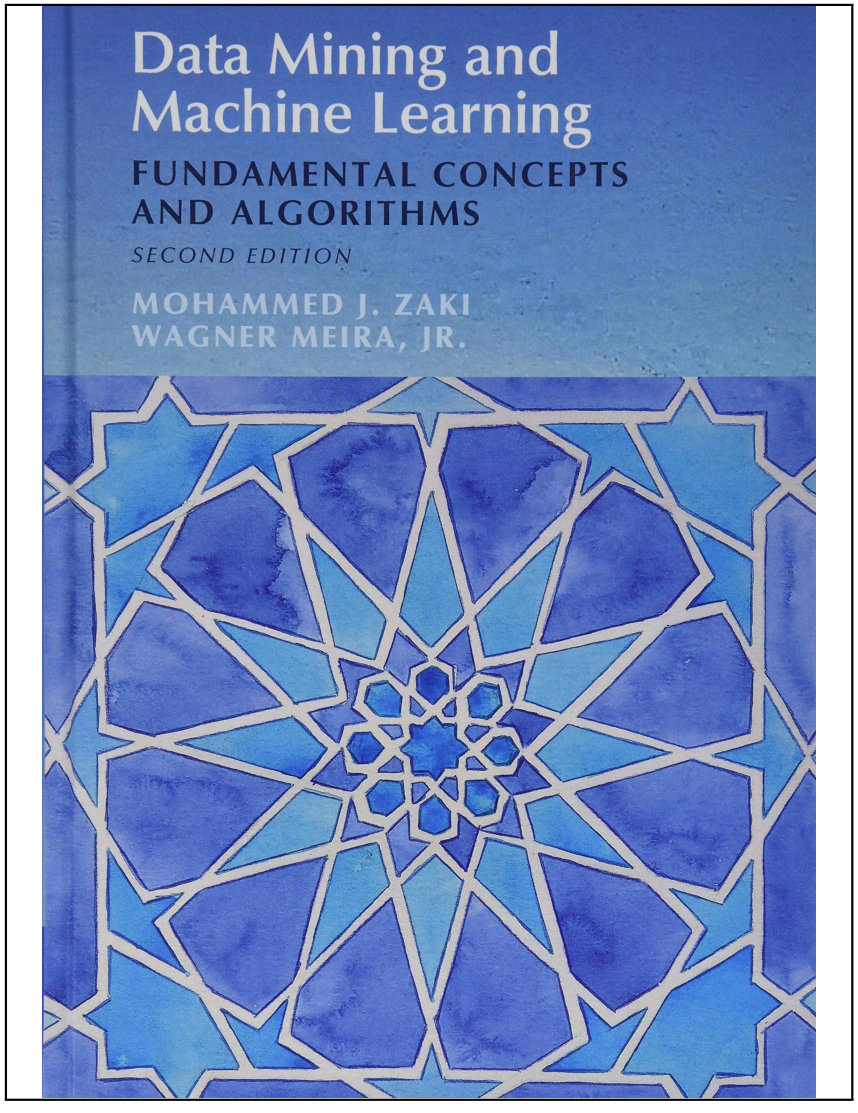
References for "Association Rules"

1. Book: Tan et al. *"Introduction to Data Mining"*, 1st Edition, 2006, Pearson Education Inc.
2. Text and images for majority of slides in "Association Rules" subsection are based on the book by Tan et al.



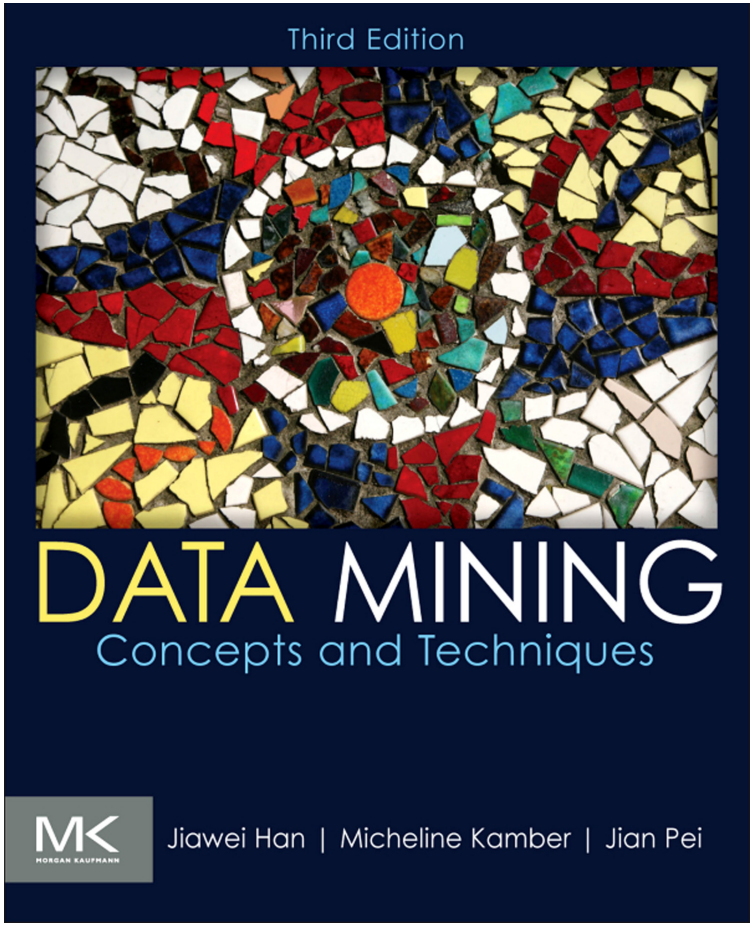
References for "FP-Growth Algorithm"

1. Book: Zaki and Meira. *"Data Mining and Machine Learning: Fundamental Concepts and Algorithms"*, 2nd Edition, 2020, Cambridge University Press.
2. Text and images for some slides in "FP-Growth Algorithm" subsection are based on the book by Zaki and Meira.



References for "FP-Growth Algorithm"

1. Book: Han et al. *"Data Mining Concepts and Techniques"*, 3rd Edition, 2012, Morgan Kaufmann Publishers.
2. All text and images for some slides in "FP-Growth Algorithm" are based on the book by Han et al.



1. Announcements and References
   - Administrative
   - References for Today's Lecture

2. Association Rules
   - Association Rule Mining
   - Definitions
   - Problem Definition
   - Frequent Itemset Generation
   - The Apriori Algorithm
     - Efficient Candidate Generation
     - Complexity Factors
   - Rule Generation
   - Maximal and Closed Itemsets
   - Alternative Methods for Itemset Generation
     - FP-Growth Algorithm
     - ECLAT Algorithm
   - Interestingness Measures

Association Rule Mining

1. Association Analysis: discovering interesting relationships hidden in large data sets.
2. Uncovered relationships can be represented in the form of association rules or sets of frequent items.
3. When applied to retail transactions, commonly known as **Market Basket Analysis**.
4. Other application areas: medical diagnosis, Web mining, and scientific data analysis.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Association Rule Mining

1. 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.
2. Example of association rules:
   - $\{Diaper\} \rightarrow \{Beer\}$
   - $\{Milk, Bread\} \rightarrow \{Eggs, Cola\}$
   - $\{Beer, Bread\} \rightarrow \{Milk\}$
3. Note that implication means co-occurence, not causality.

T-ID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Cola
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Cola

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Definitions

- Binary Representation: presence of an item in a transaction is often considered more important than its absence, therefore an item is represented as an asymmetric binary variable.

T-ID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Cola
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Cola

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Definitions

- Binary Representation: presence of an item in a transaction is often considered more important than its absence, therefore an item is represented as an asymmetric binary variable.

T-ID	Bread	Milk	Diapers	Beers	Eggs	Cola
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

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Definitions

- Set of all transactions: $T = \{t_1, t_2, \ldots, t_N\}$.
- Set of all Items: $I = \{i_1, i_2, \ldots, i_d\}$.
- Each transaction t_i contains a subset of items chosen from I.
- Itemset: a collection of zero or more items.
- k–Itemset: itemset containing k items.
  Example: $\{Beer, Diapers, Milk\}$ is a 3-itemset.

T-ID	Bread	Milk	Diapers	Beers	Eggs	Cola
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

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Definitions

- Support Count ($\sigma(X)$): number of transactions that contain a particular itemset.

$$\sigma(X) = |\{t_i | X \subseteq t_i, t_i \in T\}| \tag{1}$$

- Example: $\sigma(\{Beer, Diapers, Milk\}) = 2$.

T-ID	Bread	Milk	Diapers	Beers	Eggs	Cola
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

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Definitions

- Association Rule: is an implication expression of the form $X \rightarrow Y$, where, $X \cap Y = \emptyset$.
- Support $s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$.
- Example: $s(\{Milk, Diaper\} \rightarrow \{Beer\}) = \frac{\sigma(\{Milk, Diaper, Beer\})}{|T|} = 2/5 = 0.4$
- Confidence $c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$.
- Example: $c(\{Milk, Diaper\} \rightarrow \{Beer\}) = \frac{\sigma(\{Milk, Diaper, Beer\})}{\sigma(\{Milk, Diaper\})} = 2/3 = 0.\overline{6}$

T-ID	Bread	Milk	Diapers	Beers	Eggs	Cola
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

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Association Rule Mining Problem

- Given a set of transactions T, the goal of association rule mining is to find all rules having:
  - support $\geq$ minimum support threshold,
  - confidence $\geq$ minimum confidence threshold.
- Minimum support (*minsup*) and confidence (*minconf*) represent thresholds on interesting association rules.
- Brute-Force Approach:
  1. List all possible association rules.
  2. Compute the support and confidence for each rule.
  3. Prune rules that do not satisfy the minsup and minconf thresholds.
- Brute-force is computationally prohibitive!

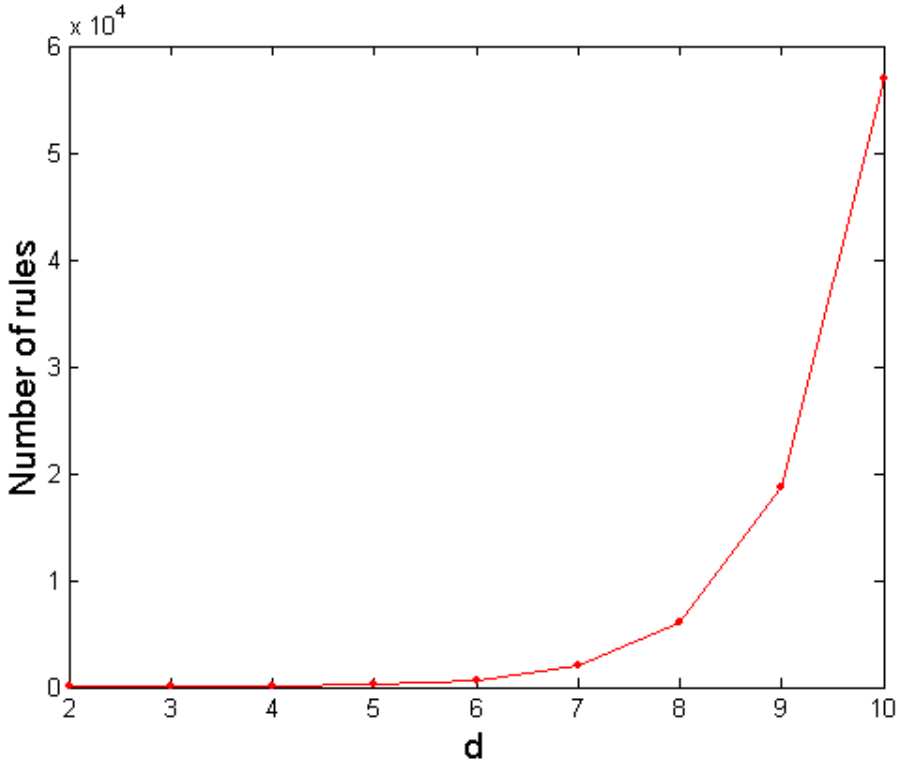
All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Brute-Force Association Rule Mining

- Consider, we have d unique items.
- Total number of itemsets = 2^d.
- Total number of possible association rules

$$R = \sum_{k=1}^{d} \left[{}^{d}C_k \cdot \sum_{j=1}^{d-k} {}^{d-k}C_j \right],$$
$$= 3^d - 2^d + 1.$$

- Example: for $d = 6$, then $R = 602$ rules to analyze.



All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Mining Association Rules

- Consider, all rules that can be generated from the itemset $\{Milk, Diaper, Beer\}$:
  1. $\{Milk, Diaper\} \rightarrow \{Beer\}$ $(s = 0.4, c = 0.\overline{6})$.
  2. $\{Milk, Beer\} \rightarrow \{Diaper\}$ $(s = 0.4, c = 1.0)$.
  3. $\{Diaper, Beer\} \rightarrow \{Milk\}$ $(s = 0.4, c = 0.\overline{6})$.
  4. $\{Beer\} \rightarrow \{Milk, Diaper\}$ $(s = 0.4, c = 0.\overline{6})$.
  5. $\{Diaper\} \rightarrow \{Milk, Beer\}$ $(s = 0.4, c = 0.5)$.
  6. $\{Milk\} \rightarrow \{Diaper, Beer\}$ $(s = 0.4, c = 0.5)$.

T-ID	Bread	Milk	Diapers	Beers	Eggs	Cola
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

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Mining Association Rules

- Consider, all rules that can be generated from the itemset $\{Milk, Diaper, Beer\}$:
  1. $\{Milk, Diaper\} \rightarrow \{Beer\}$ $(s = 0.4, c = 0.\bar{6})$.
  2. $\{Milk, Beer\} \rightarrow \{Diaper\}$ $(s = 0.4, c = 1.0)$.
  3. $\{Diaper, Beer\} \rightarrow \{Milk\}$ $(s = 0.4, c = 0.\bar{6})$.
  4. $\{Beer\} \rightarrow \{Milk, Diaper\}$ $(s = 0.4, c = 0.\bar{6})$.
  5. $\{Diaper\} \rightarrow \{Milk, Beer\}$ $(s = 0.4, c = 0.5)$.
  6. $\{Milk\} \rightarrow \{Diaper, Beer\}$ $(s = 0.4, c = 0.5)$.
- Rules originating from the same itemset have identical support but can have different confidence.
- Thus, we may decouple the support and confidence requirements.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Mining Association Rules

- Two-step approach:
  1. Frequent Itemset Generation: Generate all itemsets whose support $\geq$ minimum support.
  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 a computationally expensive process.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Frequent Itemset Generation — Brute-Force Approach

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

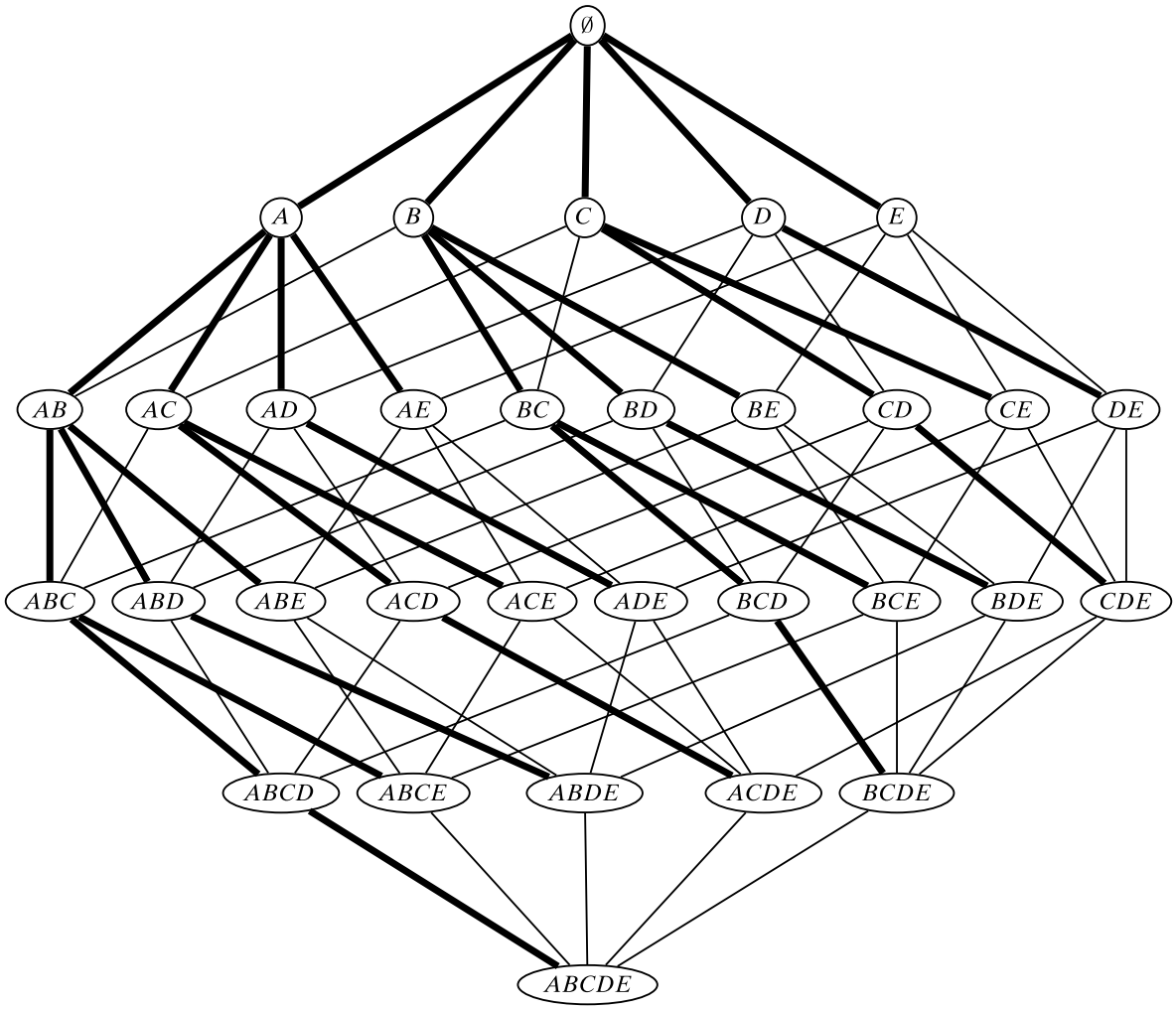
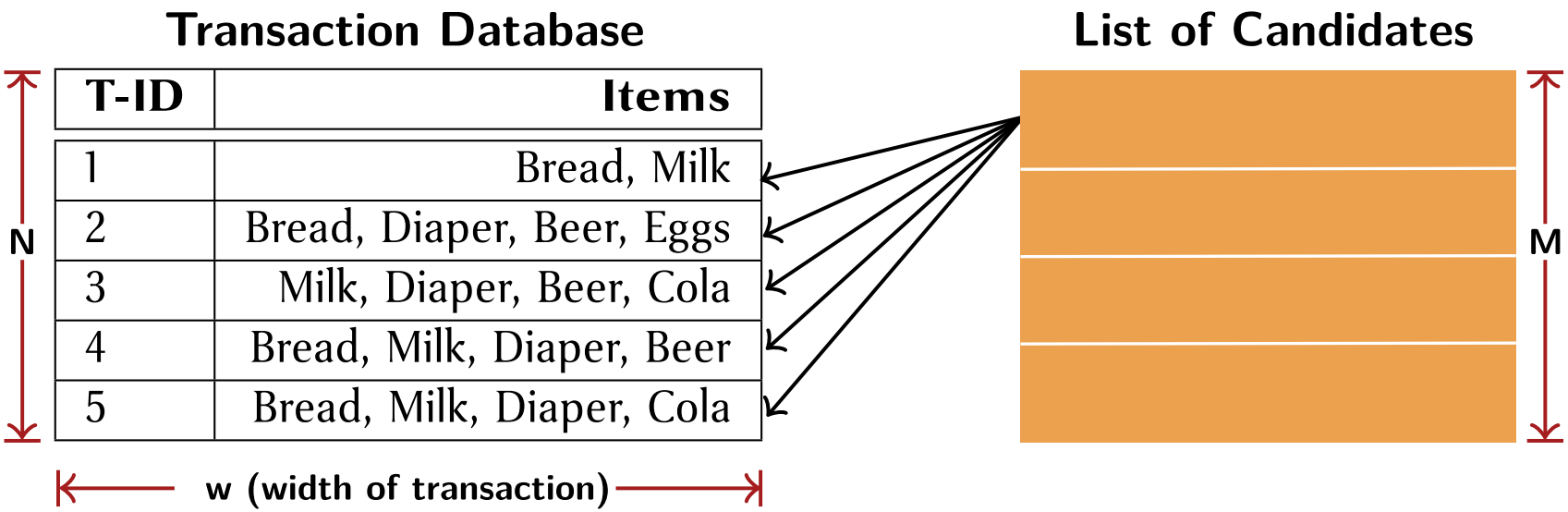


Figure 8.2. Itemset lattice and prefix-based search tree (in bold).

Brute-Force Frequent Itemset Generation

- Each itemset in the lattice is a candidate frequent itemset.
- Count the support of each candidate by scanning the database.
- Match each transaction against every candidate.
- Complexity — $\mathcal{O}(NMw)$ — very expensive since $M = 2^d$.



All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Improving Brute-Force Frequent Itemset Generation

- Reduce the number of candidates (M)
  - Complete search: $M = 2^d$.
  - Use pruning techniques to reduce M.
- 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.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Reducing Number of Candidates — The Apriori Principle

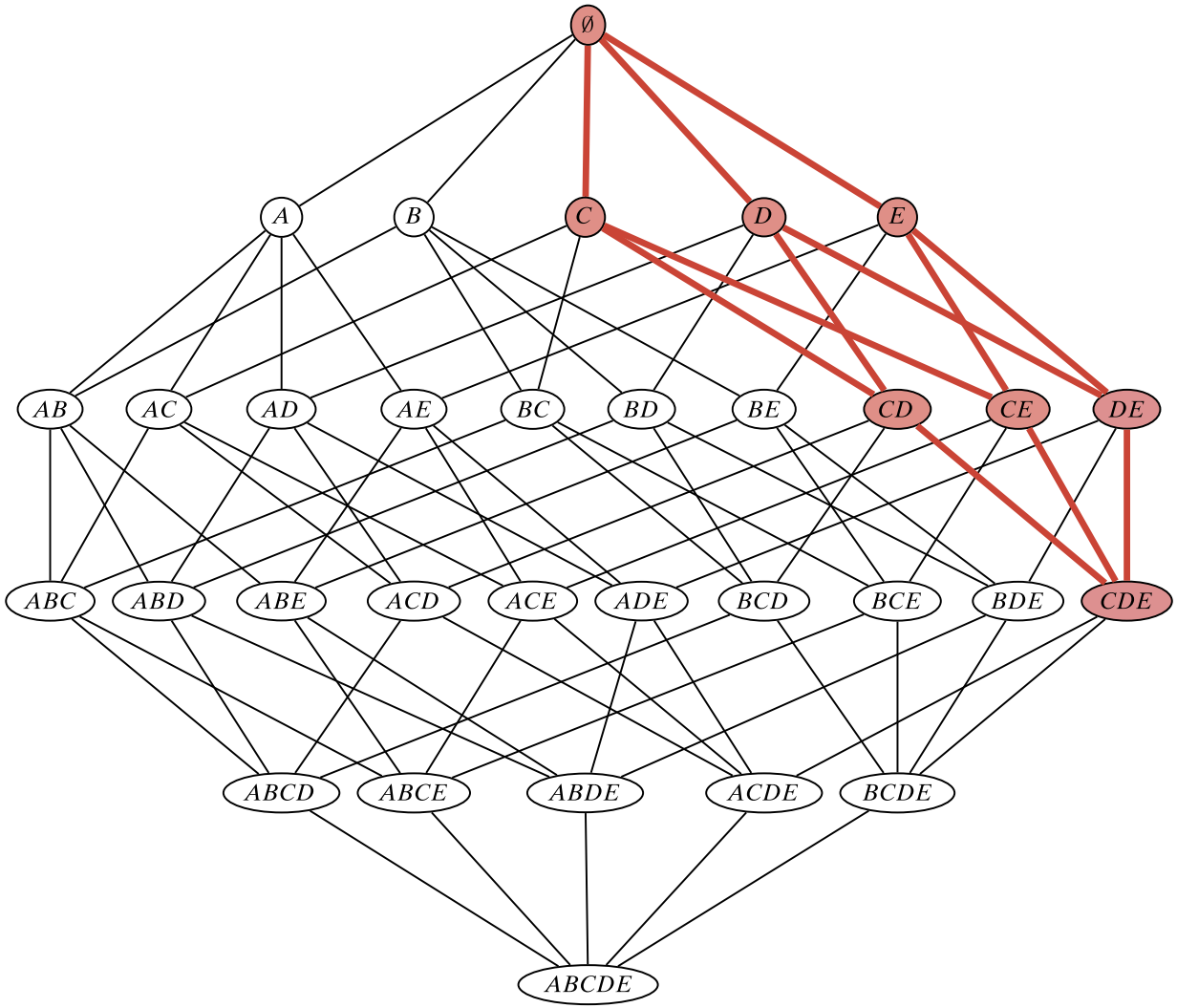
- Apriori Principle: If an itemset is frequent, then all of its subsets must also be frequent.

$$\forall X, Y : (X \subseteq Y) \implies s(X) \geq s(Y) \tag{2}$$

- Support of an itemset never exceeds the support of its subsets.
- This is known as the anti-monotone property of support.
- Conversely, if an itemset is infrequent then all of its supersets must be infrequent too.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

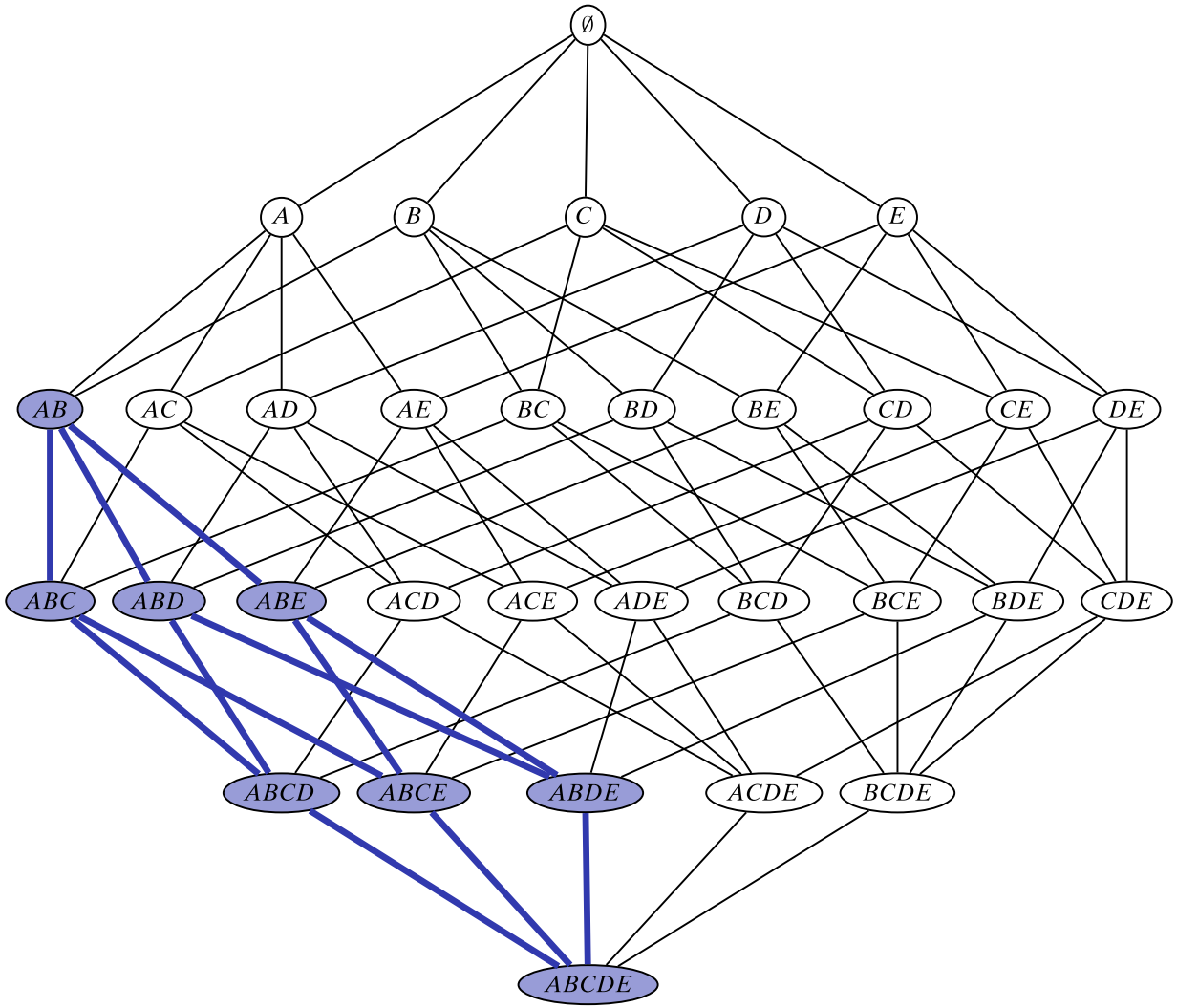
Reducing Number of Candidates — The Apriori Principle



An illustration of the Apriori principle. If $\{c, d, e\}$ is frequent, then all subsets of this itemset are frequent.

All from: 1. Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc. 2. Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

Reducing Number of Candidates — The Apriori Principle



An illustration of support-based pruning. If $\{a, b\}$ is infrequent, then all supersets of $\{a, b\}$ are infrequent.

All from: 1. Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc. 2. Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

Apriori Algorithm — Example

Example 8.6. Consider the example dataset in Figure 8.1; let $minsup = 3$. Figure 8.3 shows the itemset search space for the Apriori method, organized as a prefix tree where two itemsets are connected if one is a prefix and immediate subset of the

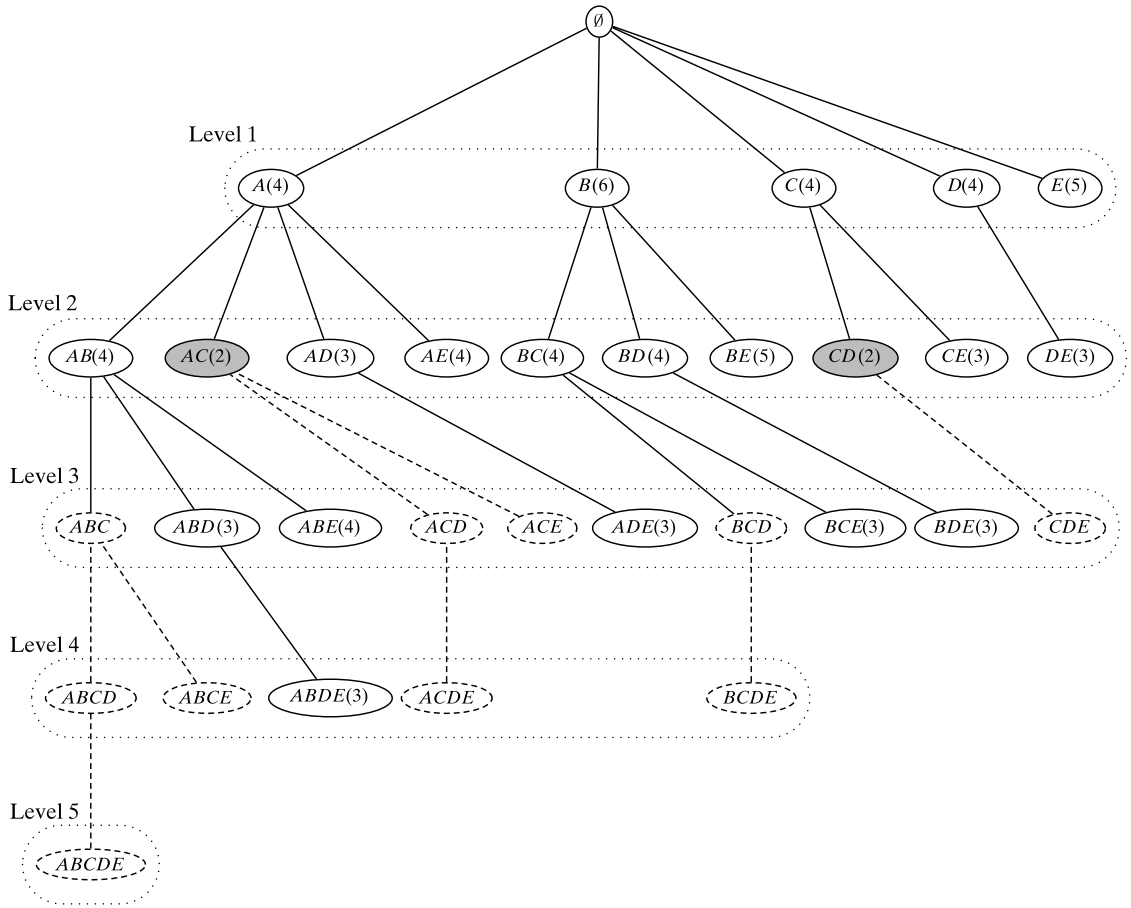


Figure 8.3. Apriori: prefix search tree and effect of pruning. Shaded nodes indicate infrequent itemsets; dashed nodes and lines indicate all of the pruned nodes and branches; solid lines indicate frequent itemsets.

All from: Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

Apriori Algorithm

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

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Illustrating Apriori Principle

TID	*Items*
1	Bread, 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	4
Eggs	1

Minimum Support = 3

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

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

If every subset is considered,

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

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 1 = 13$$

Triplets (3-itemsets)

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

Candidate Generation Step — Brute-Force Method

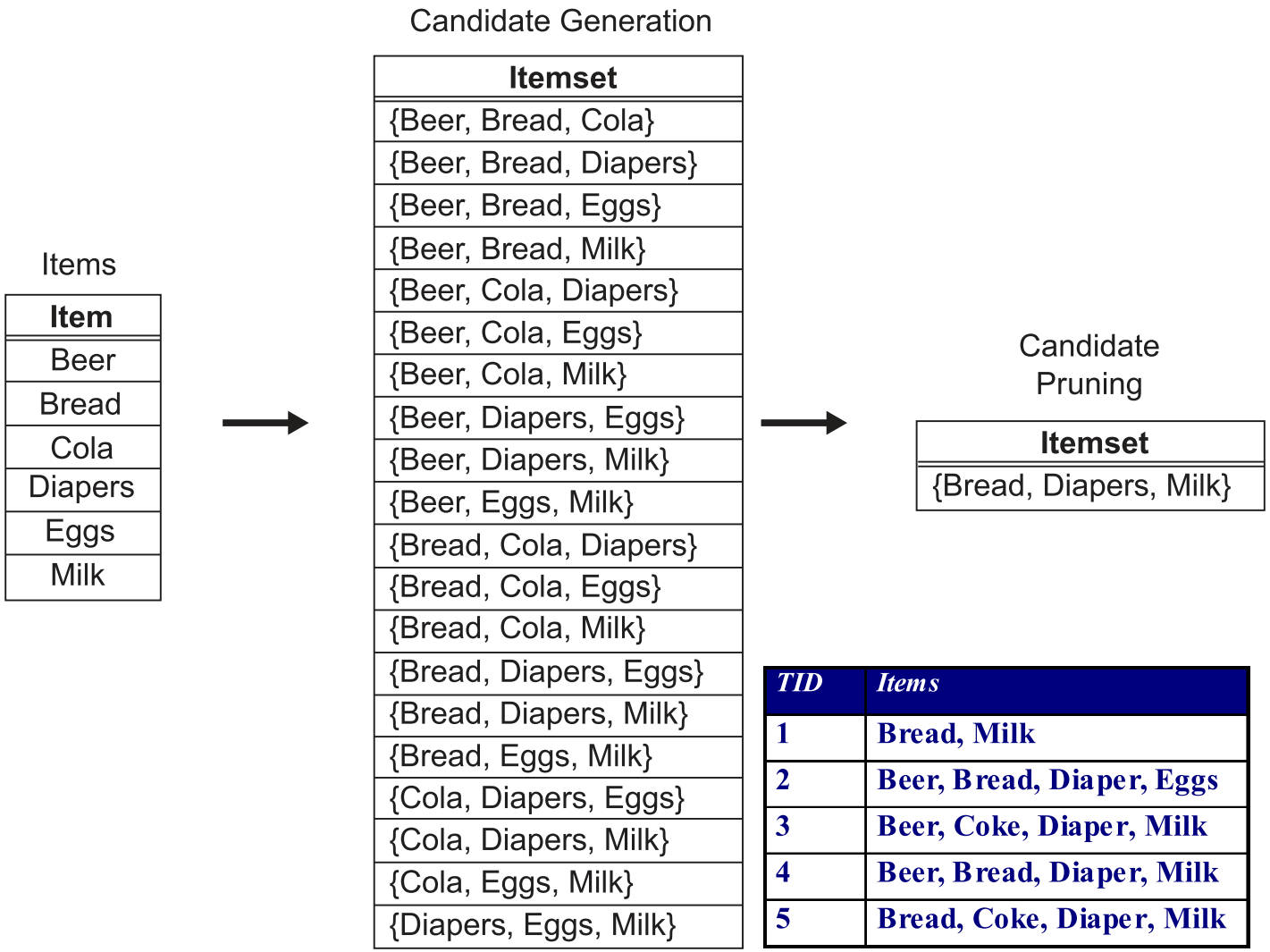


Figure 5.6. A brute-force method for generating candidate 3-itemsets.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Candidate Generation Step — Merge $F_{k-1} \times F_1$ Itemsets

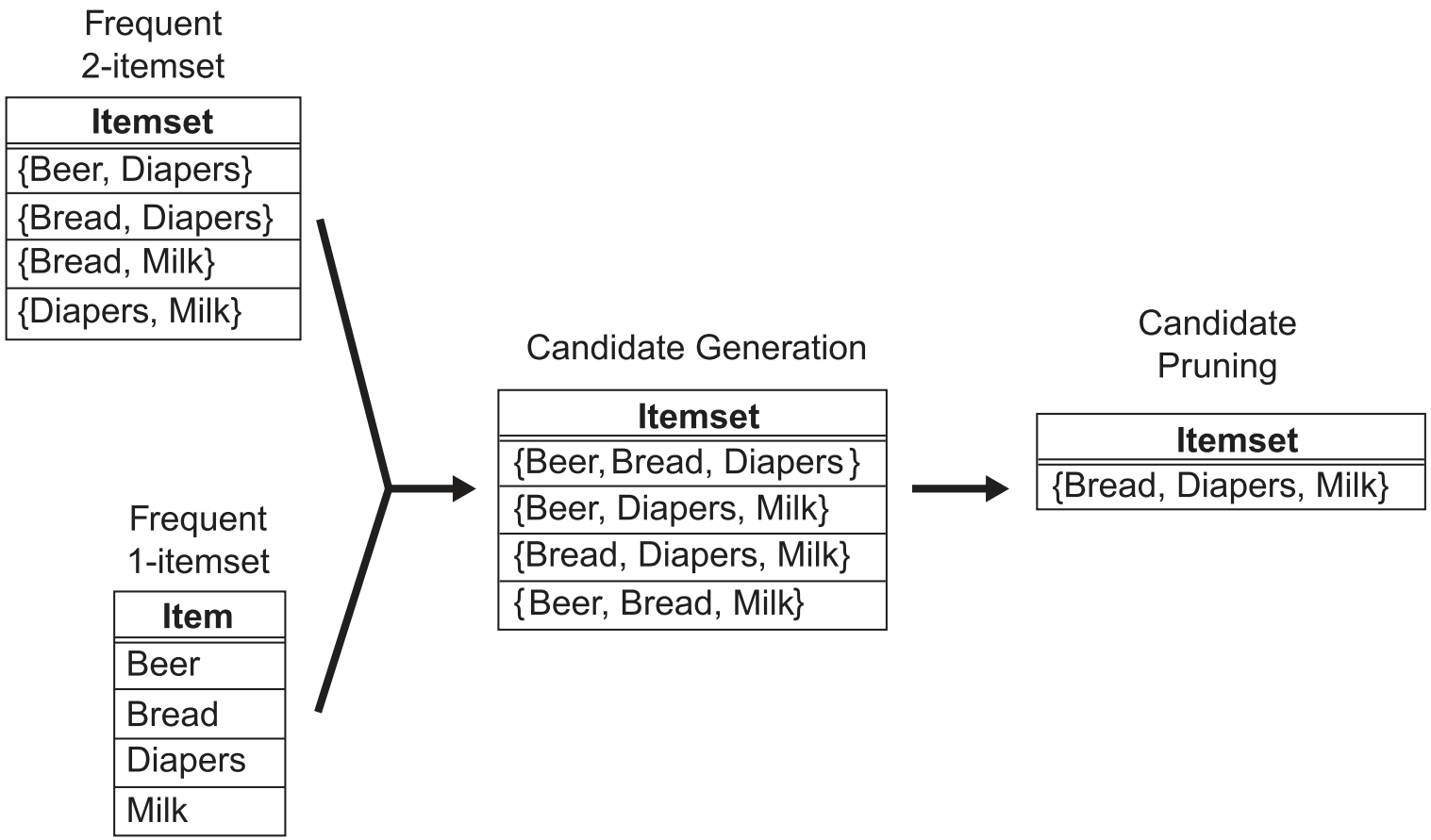


Figure 5.7. Generating and pruning candidate k-itemsets by merging a frequent $(k-1)$-itemset with a frequent item. Note that some of the candidates are unnecessary because their subsets are infrequent.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Candidate Generation Step — Merge $F_{k-1} \times F_{k-1}$ Itemsets

- Merge two frequent $(k-1)$-itemsets if their first $(k-2)$ items are identical.
- $F_3 = \{ABC, ABD, ABE, ACD, BCD, BDE, CDE\}$.
  - $Merge(\underline{AB}C, \underline{AB}D) = \underline{AB}CD$.
  - $Merge(\underline{AB}C, \underline{AB}E) = \underline{AB}CE$.
  - $Merge(\underline{AB}D, \underline{AB}E) = \underline{AB}DE$.
- Do not merge ($\underline{A}BD, \underline{A}CD$ because they share only prefix of length 1 instead of length 2.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Candidate Generation Step — Merge $F_{k-1} \times F_{k-1}$ Itemsets

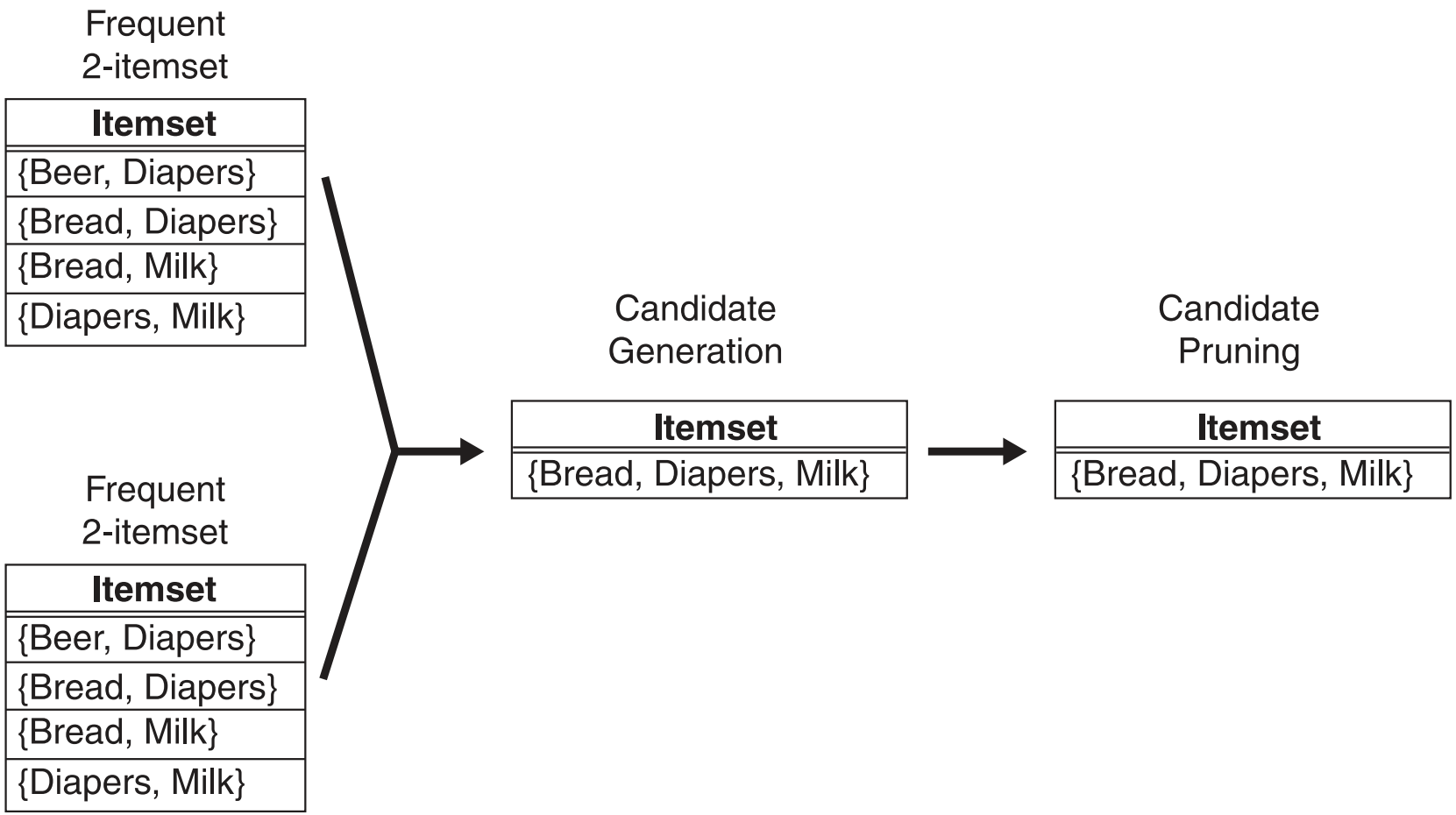


Figure 5.8. Generating and pruning candidate k-itemsets by merging pairs of frequent $(k-1)$-itemsets.

Factors Affecting Complexity of Apriori Algorithm

- 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 itemsets.
  - If number of frequent itemsets also increases, both computation and I/O costs may also increase.
- Size of database.
  - Run time of algorithm increases with number of transactions.
- Average transaction width.
  - Transaction width increases the max length of frequent itemsets.
  - Number of subsets in a transaction increases with its width, increasing computation time for support counting.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

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.
- Example: if $\{A, B, C, D\}$ is a frequent itemset, candidate rules:

$$
\begin{array}{llll}
ABC \rightarrow D & ABD \rightarrow C & ACD \rightarrow B & BCD \rightarrow A \\
A \rightarrow BCD & B \rightarrow ACD & C \rightarrow ABD & D \rightarrow ABC \\
AB \rightarrow CD & AC \rightarrow BD & AD \rightarrow BC & BC \rightarrow AD \\
BD \rightarrow AC & CD \rightarrow AB &
\end{array}
$$

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

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Rule Generation

- In general, confidence does not have an anti-monotone property:

$$c(ABC \rightarrow D) \text{ can be larger or smaller than } c(AB \rightarrow D).$$

- But confidence of rules generated from the same itemset has an anti-monotone property.
- Example: suppose $\{A, B, C, D\}$ is a frequent 4−itemset:

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

- Confidence is anti-monotone w.r.t. number of items on the RHS of the rule.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Rule Generation

- The Apriori algorithm uses a level-wise approach for generating association rules.
- Each level corresponds to the number of items that belong to the rule consequent.
- Initially, extract all high confidence rules that have one item in rule consequent.
- These rules are then joined to generate new candidate rules.
- Example: if $\{ACD\} \rightarrow \{B\}$ and $\{ABD\} \rightarrow \{C\}$ are high confidence rules, then the candidate rule $\{AD\} \rightarrow \{BC\}$ is generated by merging the consequents of both rules.
- If any node in the lattice has low confidence, entire subgraph spanned by the node can be pruned immediately.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Rule Generation

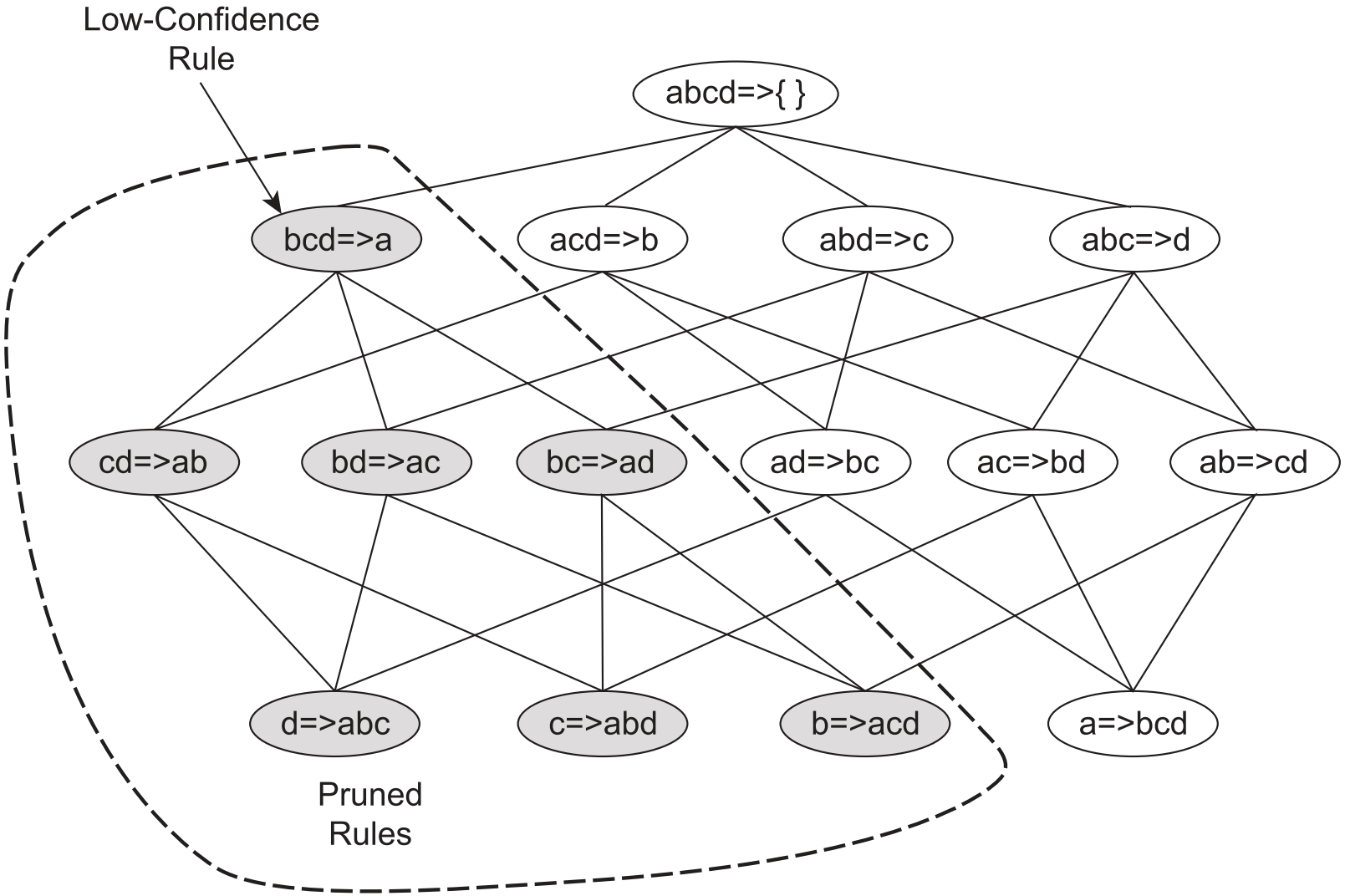


Figure 5.15. Pruning of association rules using the confidence measure.

Compact Representation of Frequent Itemsets

- Some itemsets are redundant because they have identical support as their supersets.
- Number of frequent itemsets = $3 \cdot \sum_{k=1}^{10} {}^{10}C_k$.
- Need a more compact representation.

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

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Maximal Frequent Itemsets

- An itemset is maximal frequent if it is frequent and none of its immediate supersets is frequent.

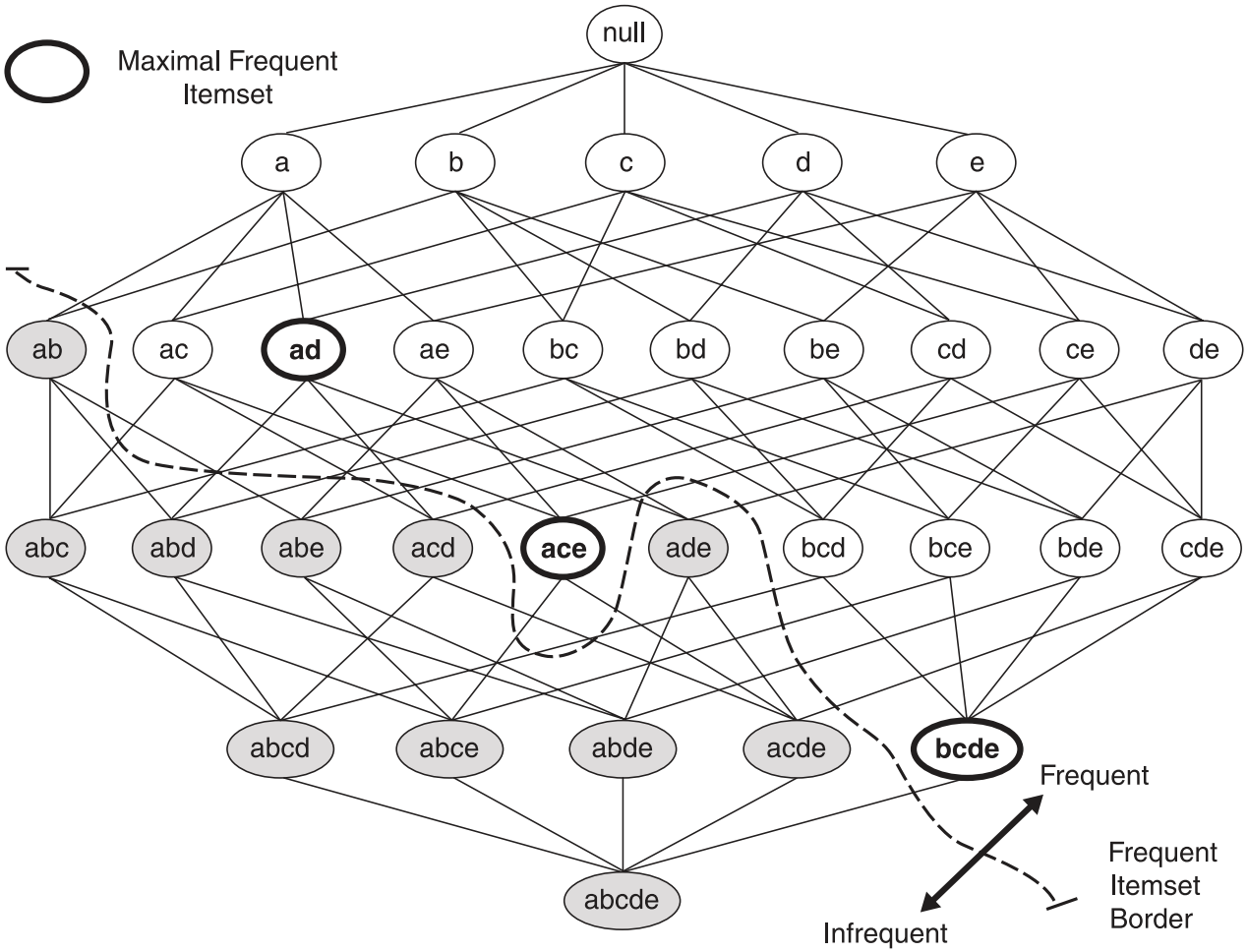


Figure 5.16. Maximal frequent itemset.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

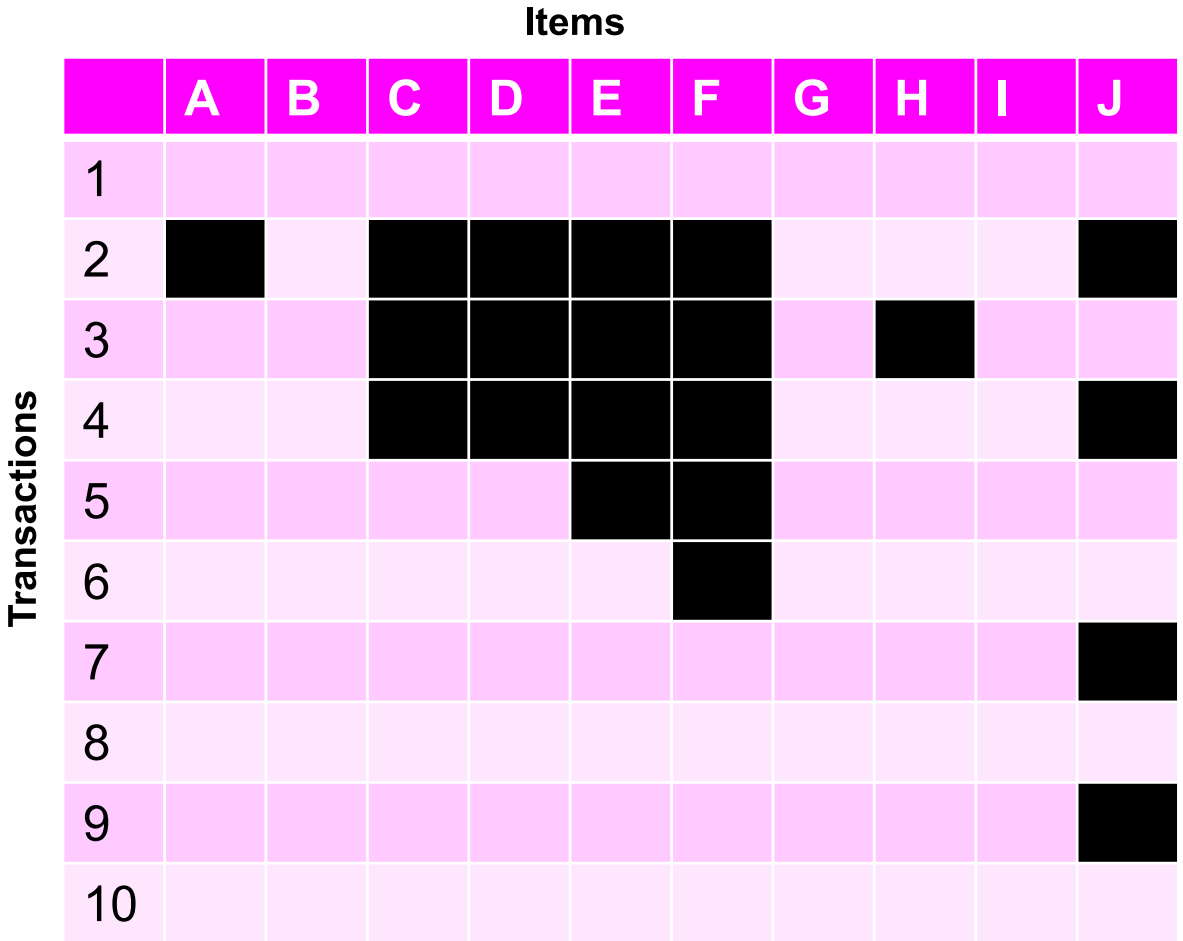
Maximal Frequent Itemsets

- What are the maximal frequent itemsets in this data.
- Minimum support threshold = 5.
- $(A1, A2, \ldots, A10)$, $(B1, B2, \ldots, B10)$, and $(C1, C2, \ldots, C10)$.

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

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Maximal Frequent Itemsets



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}

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Closed Itemset and Closed Frequent Itemset

- An itemset X is closed if none of its immediate supersets has the same support as the itemset X.
- X is not closed if at least one of its immediate supersets has support count as X.
- An itemset is a closed frequent itemset if it is closed and its support is greater than or equal to minsup.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Closed Itemset and Closet Frequent Itemset

TID	Items
1	abc
2	abcd
3	bce
4	acde
5	de

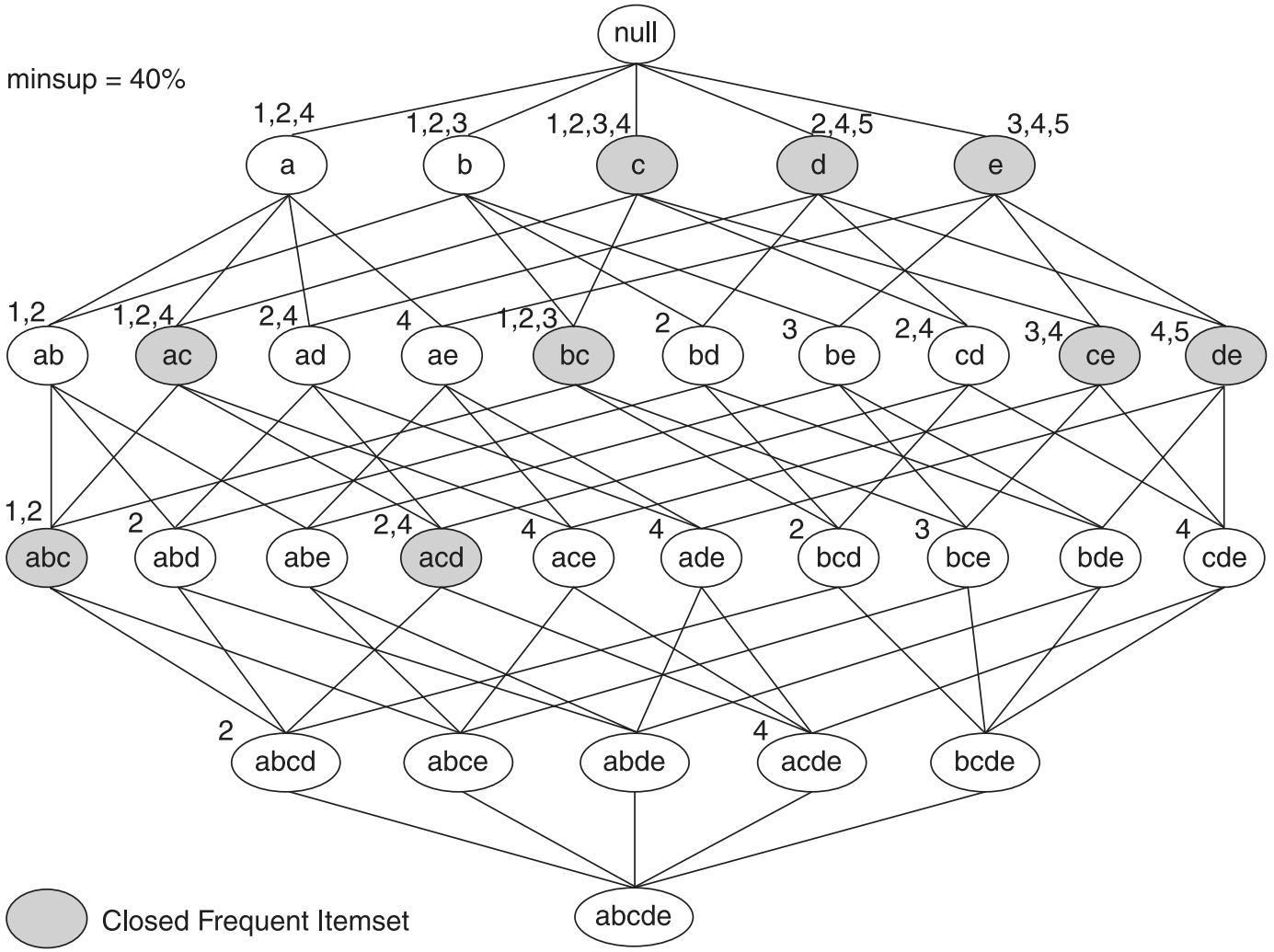
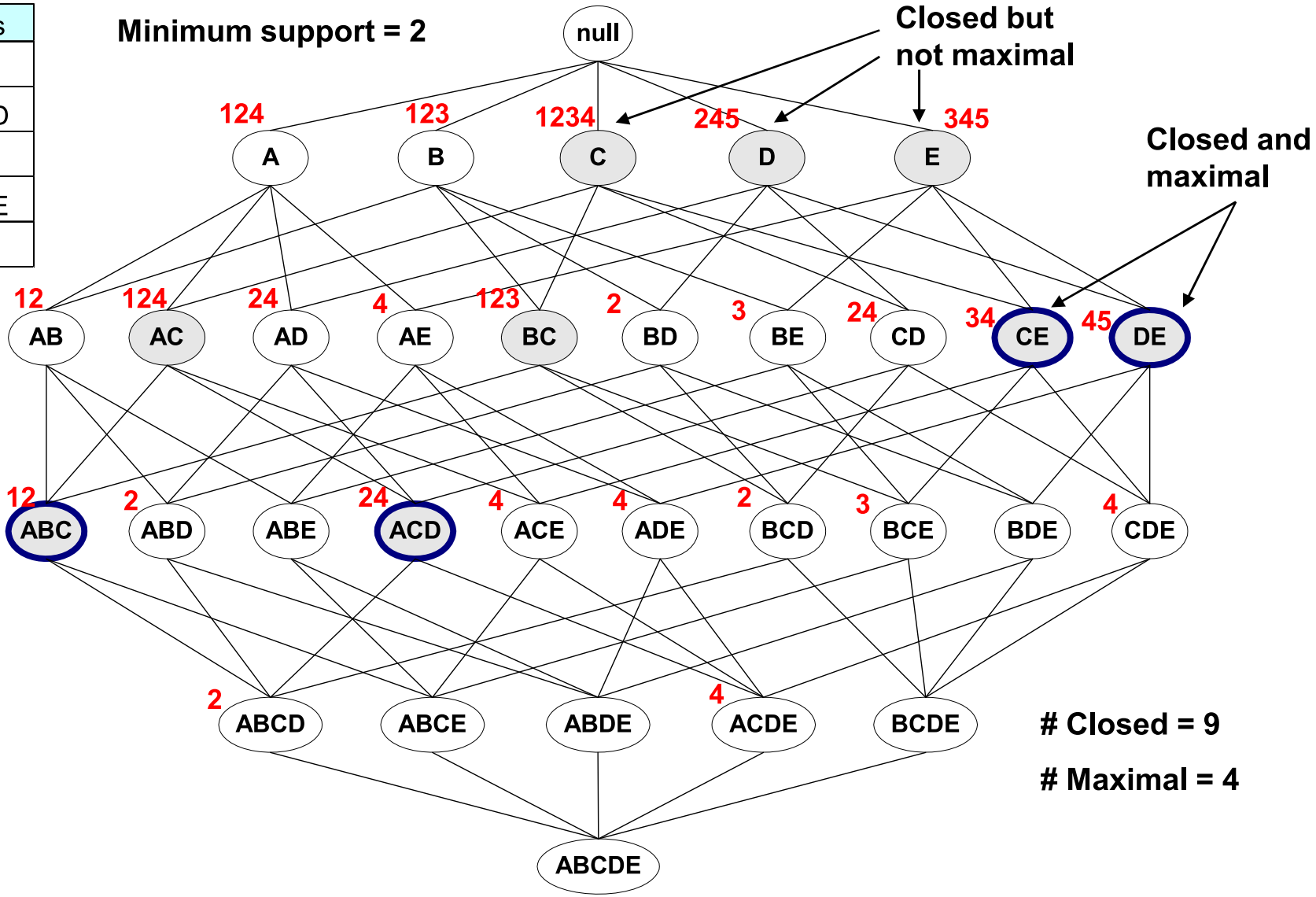


Figure 5.17. An example of the closed frequent itemsets (with minimum support equal to 40%).

Maximal Frequent and Closed Frequent Itemsets

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



Frequent, Maximal Frequent, Closed Frequent Itemsets

- All maximal frequent itemsets are closed because none of the maximal frequent itemsets can have the same support count as their immediate supersets.

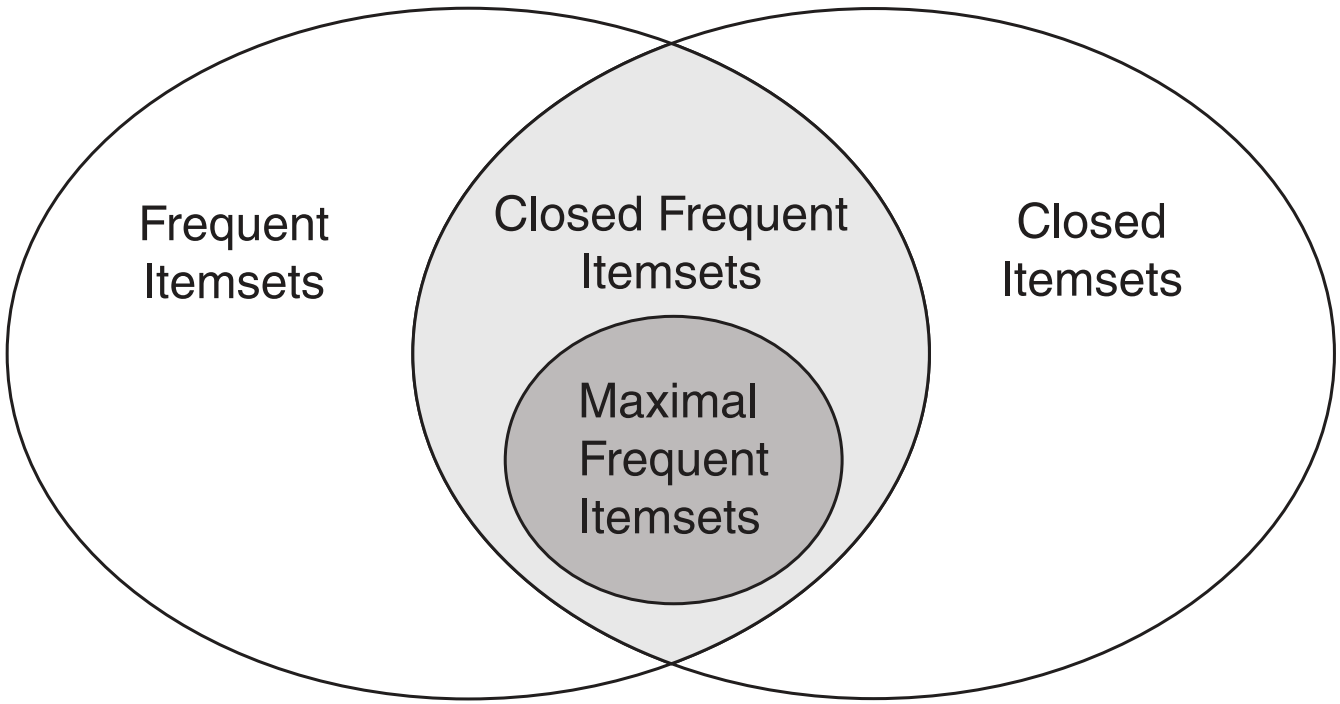


Figure 5.18. Relationships among frequent, closed, closed frequent, and maximal frequent itemsets.

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice: general-to-specific or specific-to-general.

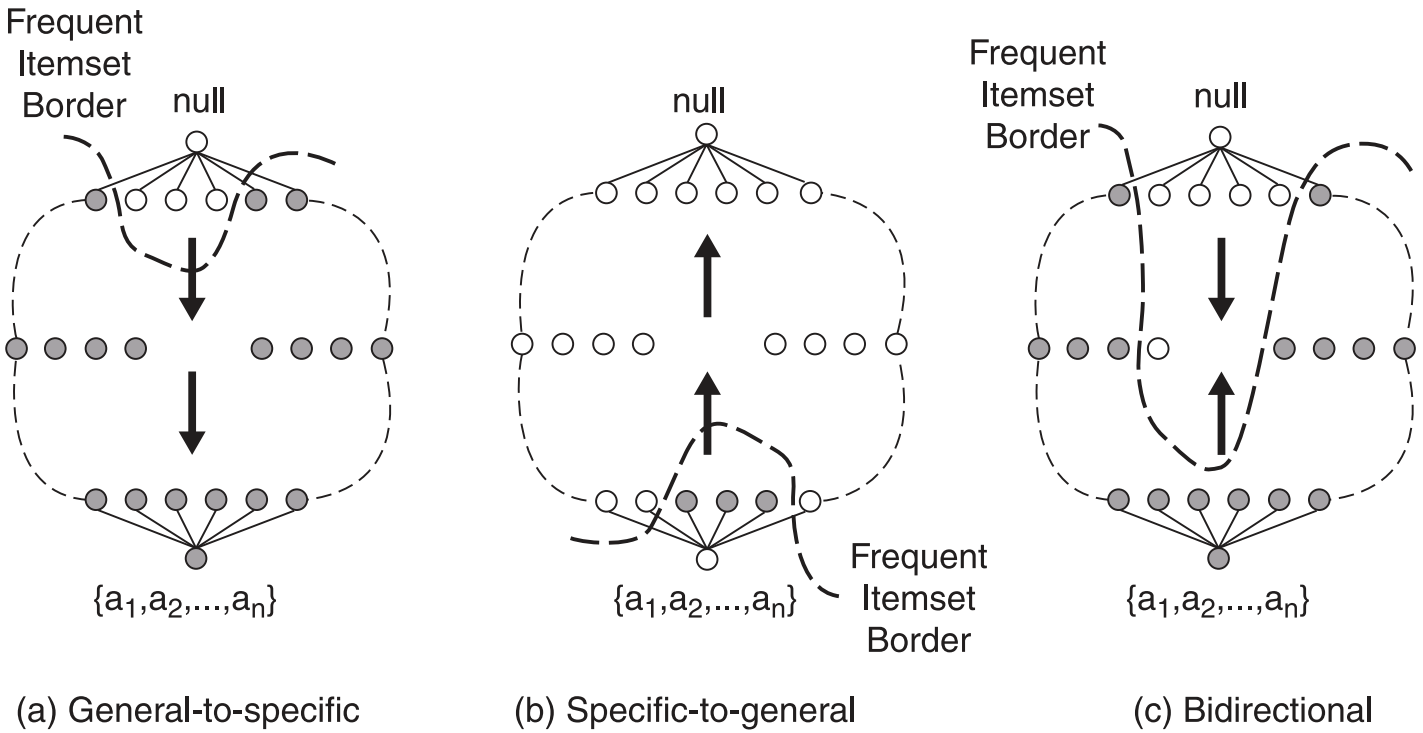
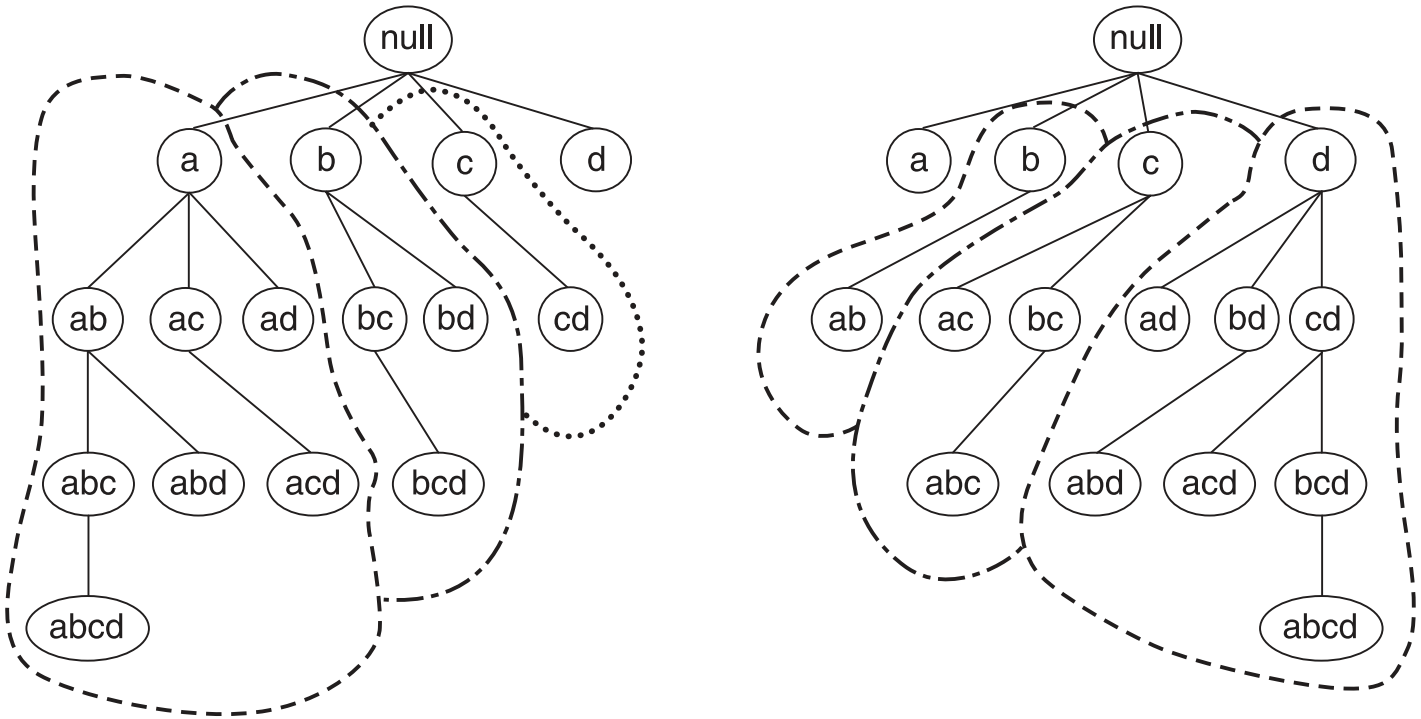


Figure 5.19. General-to-specific, specific-to-general, and bidirectional search.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice: equivalent classes.



(a) Prefix tree.

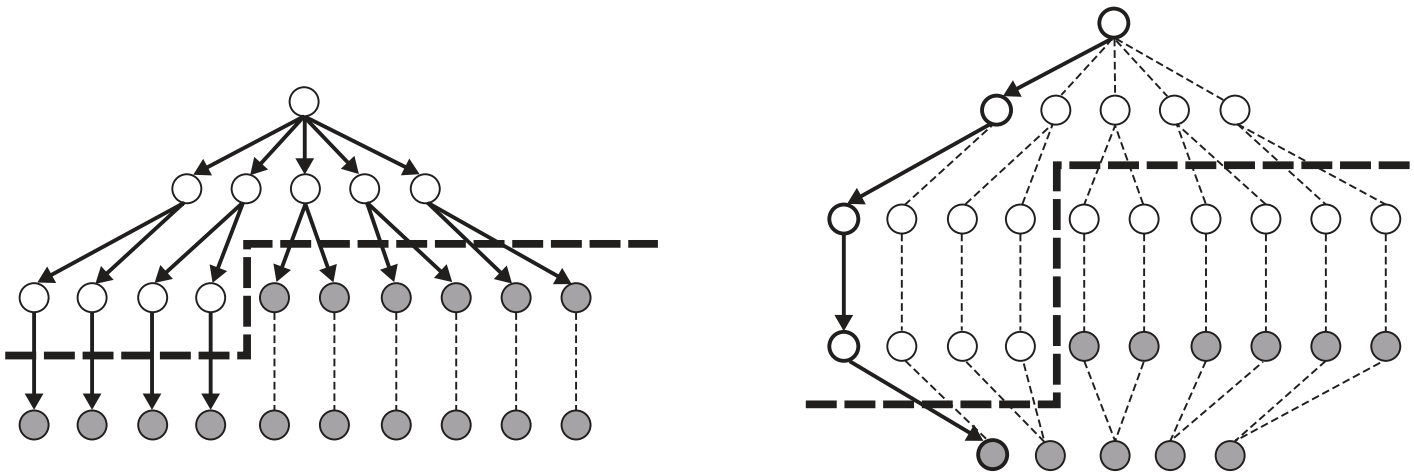
(b) Suffix tree.

Figure 5.20. Equivalence classes based on the prefix and suffix labels of itemsets.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice: breadth-first-search or depth-first-search.



(a) Breadth first

(b) Depth first

Figure 5.21. Breadth-first and depth-first traversals.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice: general-to-specific or specific-to-general.

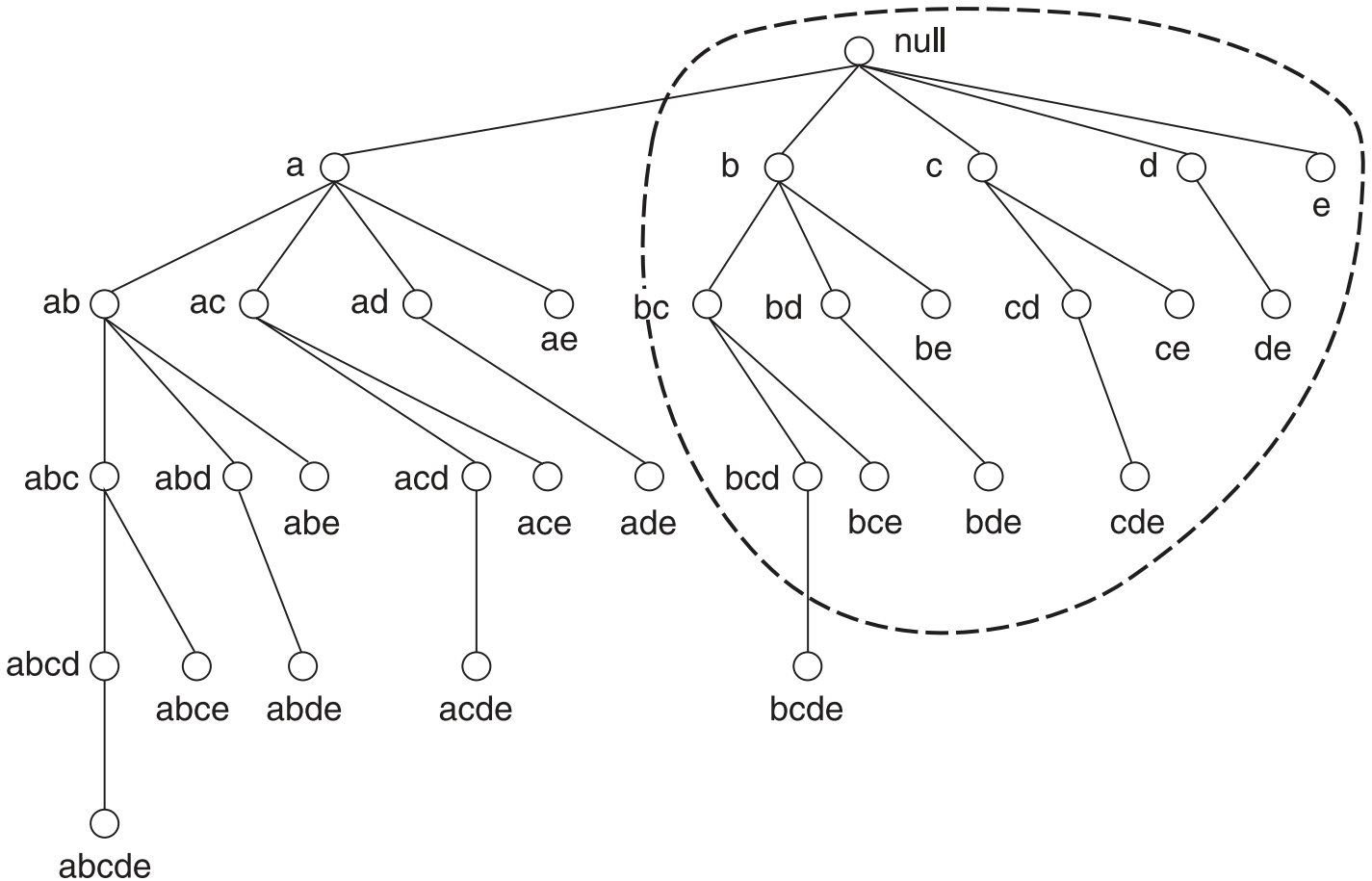


Figure 5.22. Generating candidate itemsets using the depth-first approach.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice: general-to-specific or specific-to-general.

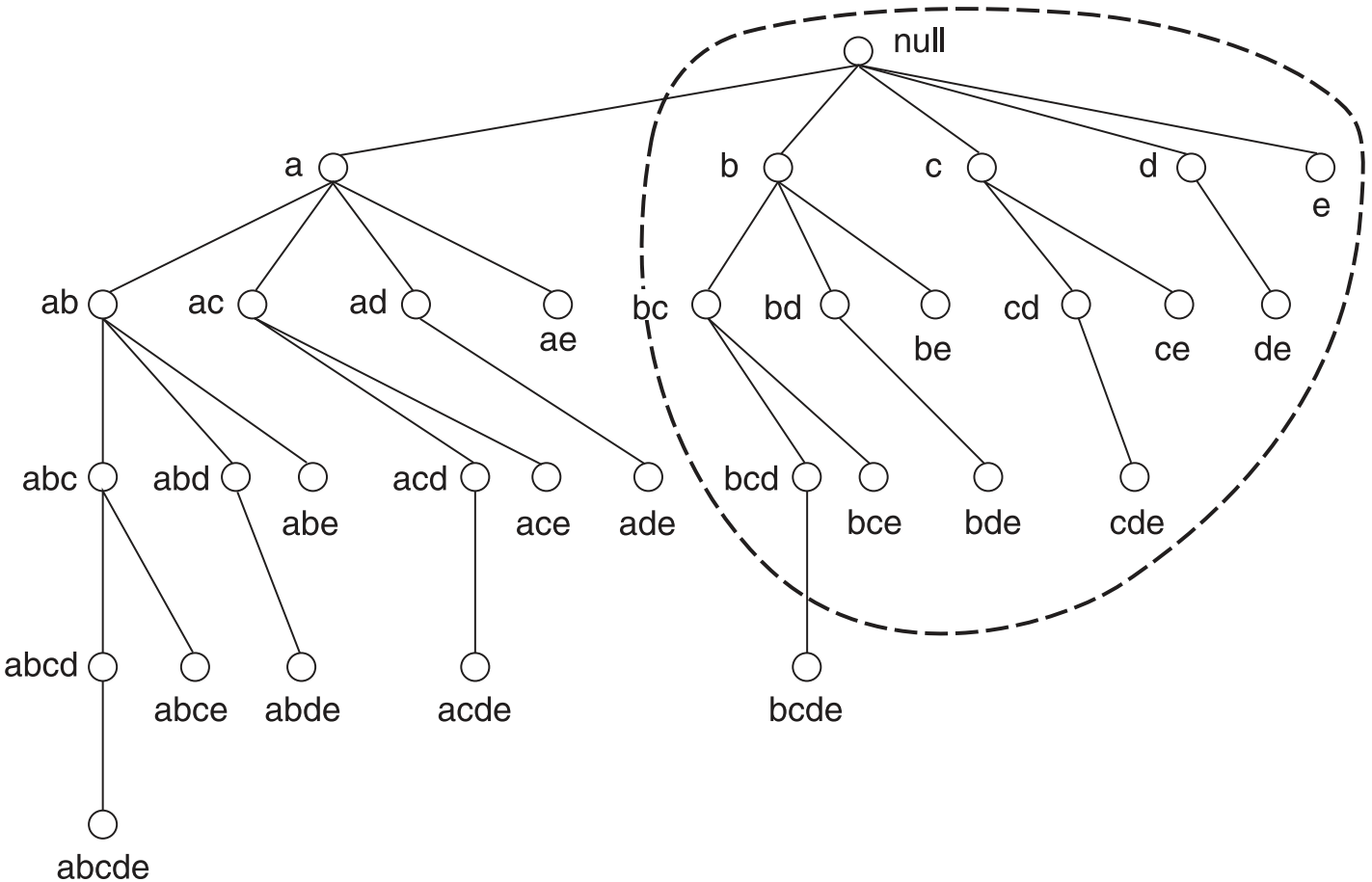


Figure 5.22. Generating candidate itemsets using the depth-first approach.

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Alternative Methods for Frequent Itemset Generation

- Database Representation: horizontal or vertical data layout.

Horizontal Data Layout

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

Vertical Data Layout

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

Figure 5.23. Horizontal and vertical data format.

Drawbacks of Apriori Algorithm

- Apriori candidate generate-and-test method significantly reduces the size of candidate sets, leading to good performance gain.
- However, it can suffer from two nontrivial costs:
  1. Need to generate a huge number of candidate sets.
     - For example, if there are 10^4 frequent 1-itemsets, the Apriori algorithm will need to generate more than 10^7 candidate 2-itemsets.
  2. Need to repeatedly scan the whole database and check a large set of candidates by pattern matching.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

FP-Growth Algorithm

- FP-Growth mines the complete set of frequent itemsets without such a costly candidate generation process.
- FP-growth, which adopts a divide-and-conquer strategy as follows:
  1. Compress the database representing frequent items into a frequent pattern tree, or FP-tree, which retains the itemset association information.
  2. Divides the compressed database into a set of conditional databases (projected trees) each associated with one "pattern fragment," and mines each database separately. For each "pattern fragment," only its associated data sets need to be examined.
- FP-Growth substantially reduce the size of the data sets to be searched, along with the "growth" of patterns being examined.

All from: Han et al. "Data Mining Concepts and Techniques", 3rd Edition, 2012, Morgan Kaufmann Publishers.

FP-Growth Algorithm

1. Obtain the transaction database.

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

(b) Transaction database

x	A	B	C	D	E
$\mathbf{t}(x)$	1	1	2	1	1
	3	2	4	3	2
	4	3	5	5	3
	5	4	6	6	4
		5			5
		6			

(c) Vertical database

Figure 8.1. An example database.

FP-Growth Algorithm

1. Obtain the transaction database.
2. Sort the items in the transaction database by their support.

Item	Support
B	6
E	5
A	4
C	4
D	4

All from: Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

FP-Growth Algorithm

1. Obtain the transaction database.
2. Sort the items in the transaction database by their support.
3. Sort the order of the items in each transaction of the database with descending order of support.

Item	Support
B	6
E	5
A	4
C	4
D	4

t	Transaction
1	BEAD
2	BEC
3	BEAD
4	BEAC
5	BEACD
6	BCD

All from: Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

FP-Growth Algorithm

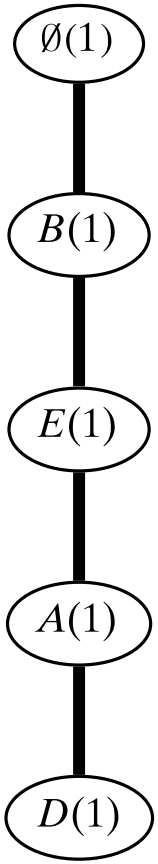
1. Obtain the transaction database.
2. Sort the items in the transaction database by their support.
3. Sort the order of the items in each transaction of the database with descending order of support.
4. Construct the FP-Tree step-by-step by adding each of the transactions (where items in them are reordered with descending support).

Item	Support
B	6
E	5
A	4
C	4
D	4

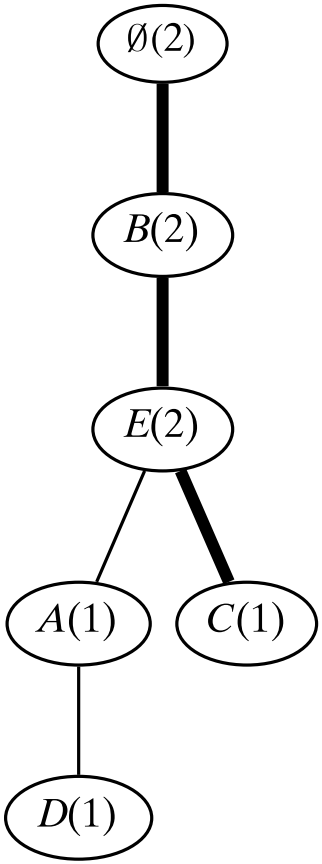
t	Transaction
1	BEAD
2	BEC
3	BEAD
4	BEAC
5	BEACD
6	BCD

All from: Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

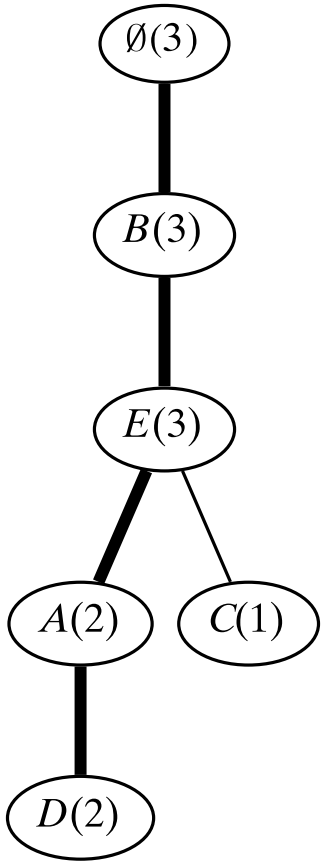
FP-Growth Algorithm



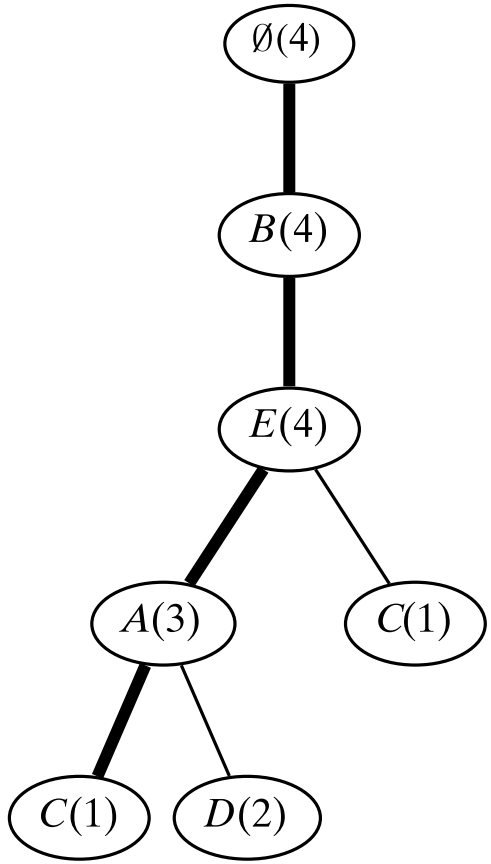
(a) $\langle 1, BEAD \rangle$



(b) $\langle 2, BEC \rangle$

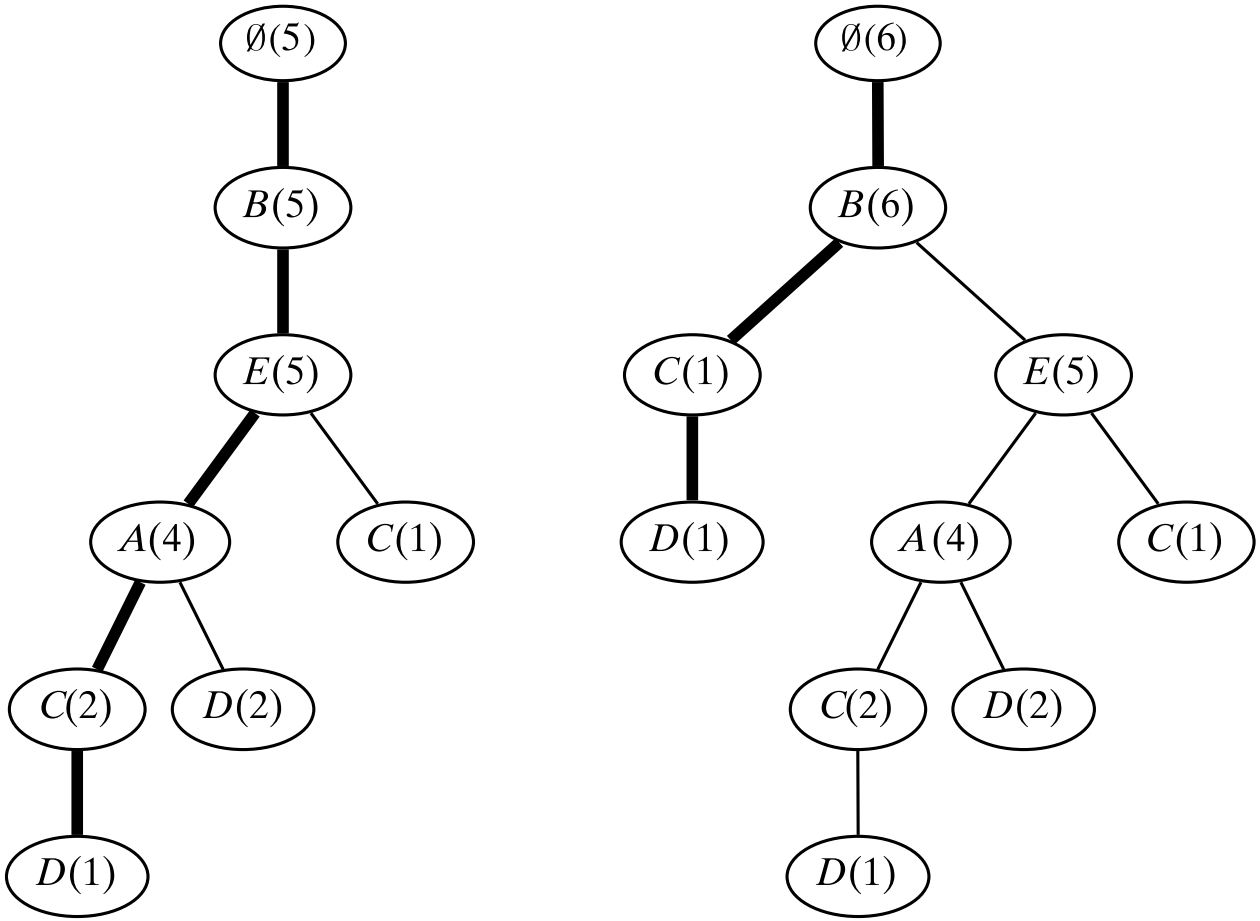


(c) $\langle 3, BEAD \rangle$



(d) $\langle 4, BEAC \rangle$

FP-Growth Algorithm

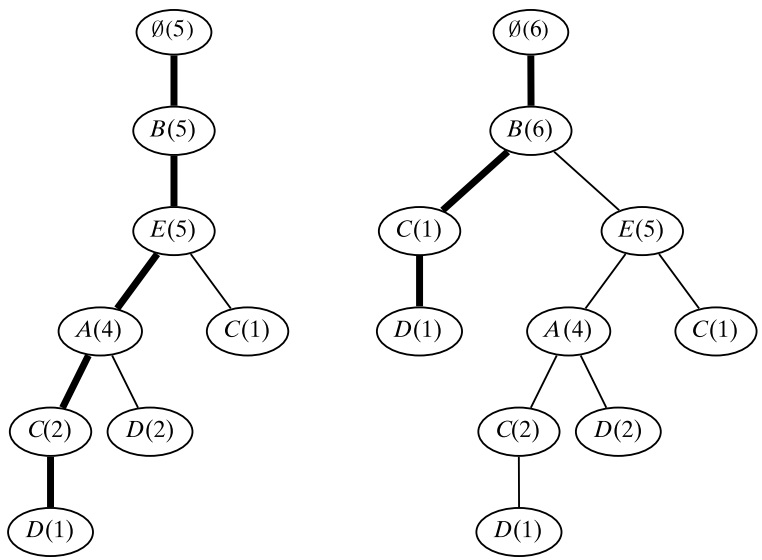


(e) $\langle 5, BEACD \rangle$ (f) $\langle 6, BCD \rangle$

Figure 8.7. Frequent pattern tree: bold edges indicate current transaction.

FP-Growth Algorithm

1. Obtain the transaction database.
2. Sort the items in the transaction database by their support.
3. Sort the order of the items in each transaction of the database with descending order of support.
4. Construct the FP-Tree step-by-step by adding each of the transactions (where items in them are reordered with descending support).
   - The FP-tree serves as an index in lieu of the original database.



(e) $\langle 5, BEACD \rangle$ (f) $\langle 6, BCD \rangle$

Figure 8.7. Frequent pattern tree: bold edges indicate current transaction.

FP-Growth Algorithm

1. Obtain the transaction database.
2. Sort the items in the transaction database by their support.
3. Sort the order of the items in each transaction of the database with descending order of support.
4. Construct the FP-Tree (call it R) step-by-step by adding each of the transactions (where items in them are reordered with descending support).
   - The FP-tree (R) serves as an index in lieu of the original database.
5. Given the FP-Tree (R), projected (conditional) FP-trees (databases) are built for each frequent item i in R in increasing order of support.

All from: Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

FP-Growth Algorithm — Projecting FP-Trees

1. To project R on item i , we find all the occurrences of i in the tree, and for each occurrence, we determine the corresponding path from the root to i (line 13).
2. The count of item i on a given path is recorded in $cnt(i)$ (line 14), and the path is inserted into the new projected tree R_X, where X is the itemset obtained by extending the prefix P with the item i.
3. While inserting the path, the count of each node in R_X along the given path is incremented by the path count $cnt(i)$.
4. We omit the item i from the path, as it is now part of the prefix.
5. The resulting FP-tree is a projection of the itemset X that comprises the current prefix extended with item i (line 9).
6. Call FPGrowth recursively with projected FP-tree R_X and the new prefix itemset X as the parameters (line 16).
7. The base case for the recursion happens when the input FP-tree R is a single path.
8. FP-trees that are paths are handled by enumerating all itemsets that are subsets of the path, with the support of each such itemset being given by the least frequent item in it (lines 2–6).

All from: Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

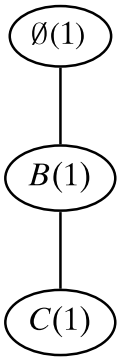
FP-Growth Algorithm — Projecting FP-Trees

Algorithm 8.5: Algorithm FPGROWTH

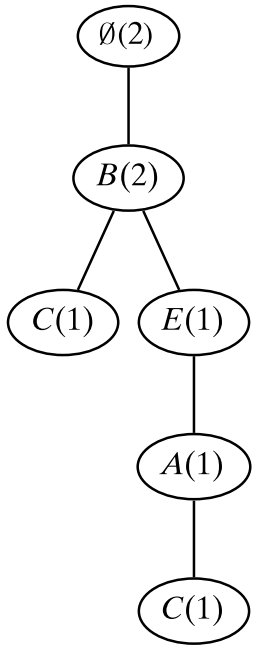
```
// Initial Call: R ← FP-tree(D), P ← ∅, F ← ∅
FPGROWTH (R, P, F, minsup):
1  Remove infrequent items from R
2  if ISPATH(R) then // insert subsets of R into F
3      foreach Y ⊆ R do
4          X ← P ∪ Y
5          sup(X) ← min_{x∈Y}{cnt(x)}
6          F ← F ∪ {(X, sup(X))}
7  else // process projected FP-trees for each frequent item i
8      foreach i ∈ R in increasing order of sup(i) do
9          X ← P ∪ {i}
10         sup(X) ← sup(i) // sum of cnt(i) for all nodes labeled i
11         F ← F ∪ {(X, sup(X))}
12         R_X ← ∅ // projected FP-tree for X
13         foreach path ∈ PATHFROMROOT(i) do
14             cnt(i) ← count of i in path
15             Insert path, excluding i, into FP-tree R_X with count cnt(i)
16         if R_X ≠ ∅ then FPGROWTH (R_X, X, F, minsup)
```

All from: Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

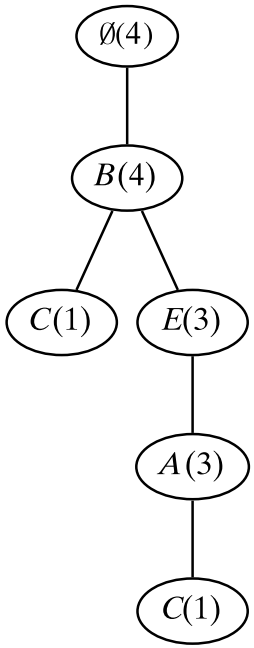
FP-Growth Algorithm — Projecting FP-Trees



(a) Add $BC, cnt = 1$



(b) Add $BEAC, cnt = 1$



(c) Add $BEA, cnt = 2$

Figure 8.8. Projected frequent pattern tree for D.

FP-Growth Algorithm — Projecting FP-Trees

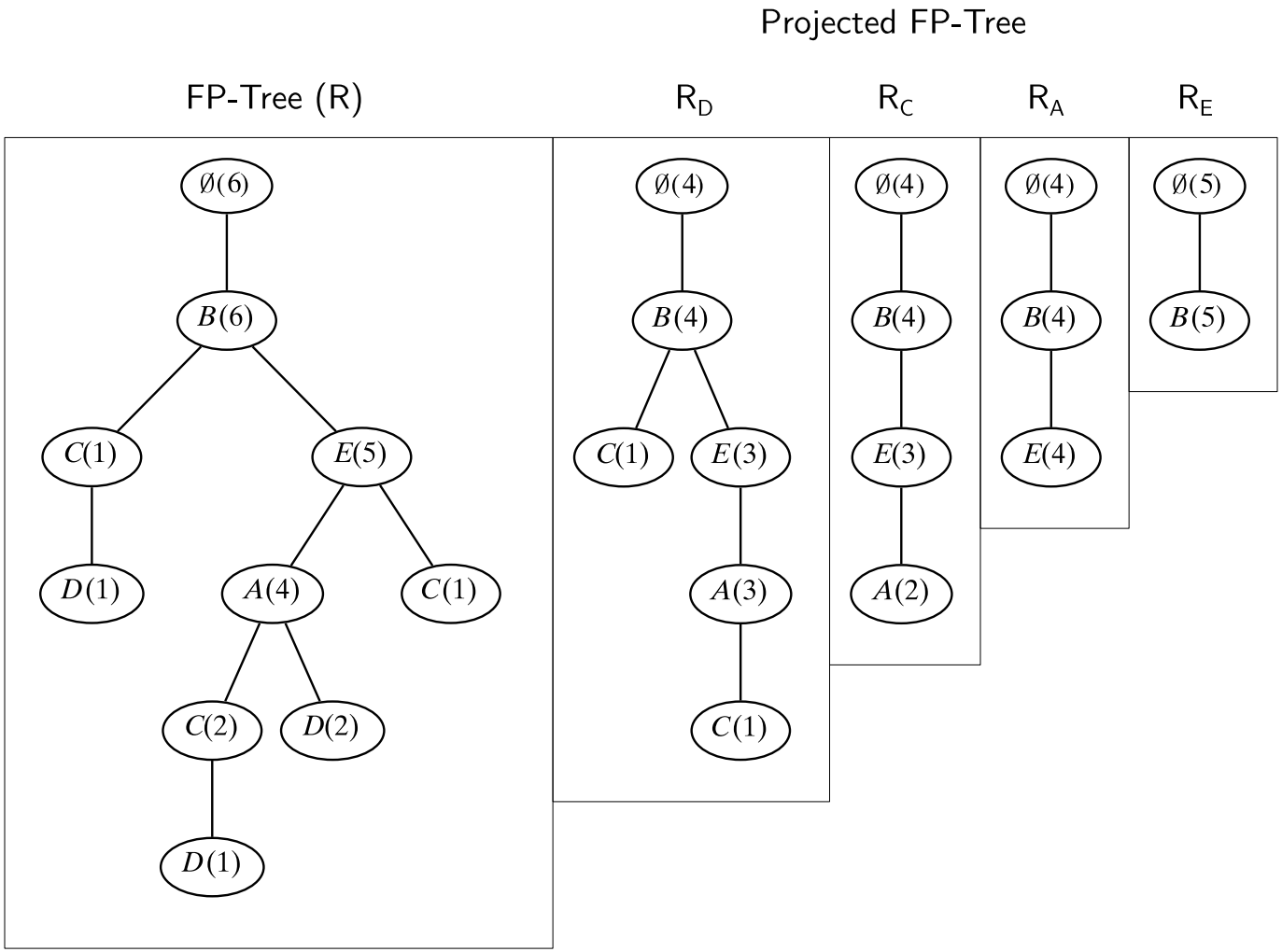


Figure 8.9. FPGrowth algorithm: frequent pattern tree projection.

All from: Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

FP-Growth Algorithm — Mining Frequent Itemsets

- Let, $minsup = 3$.
- $P = D$:
  $\{DB(4), DE(3), DA(3), DBE(3), DBA(3), DEA(3), DBEA(3)\}$
  (after removing C as it is infrequent — support = 2).
- $P = C$:
  $\{CB(3), CE(3), CBE(3)\}$.
- $P = A$:
  $\{AE(4), AB(4), AEB(4)\}$.
- $P = E$: $\{EB(5)\}$.

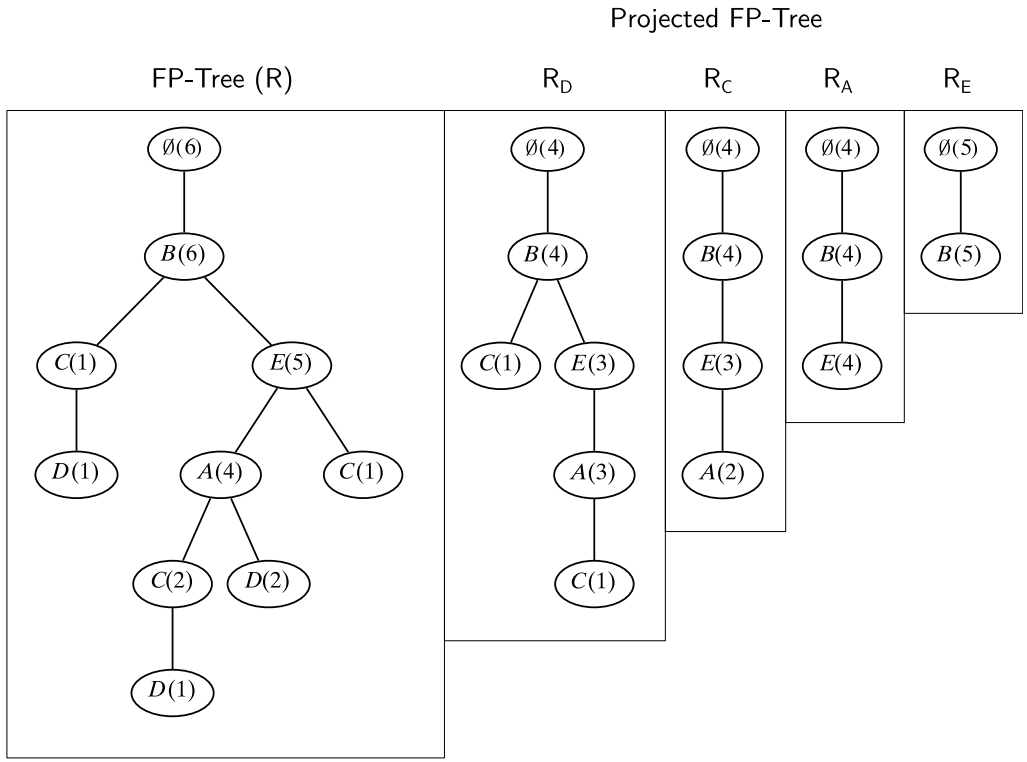


Figure 8.9. FPGrowth algorithm: frequent pattern tree projection.

All from: Zaki and Meira, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms", 2nd Edition, 2020, Cambridge University Press.

ECLAT Algorithm

- For each item, store a list of transaction ids (TIDS).

Horizontal Data Layout

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

Vertical Data Layout

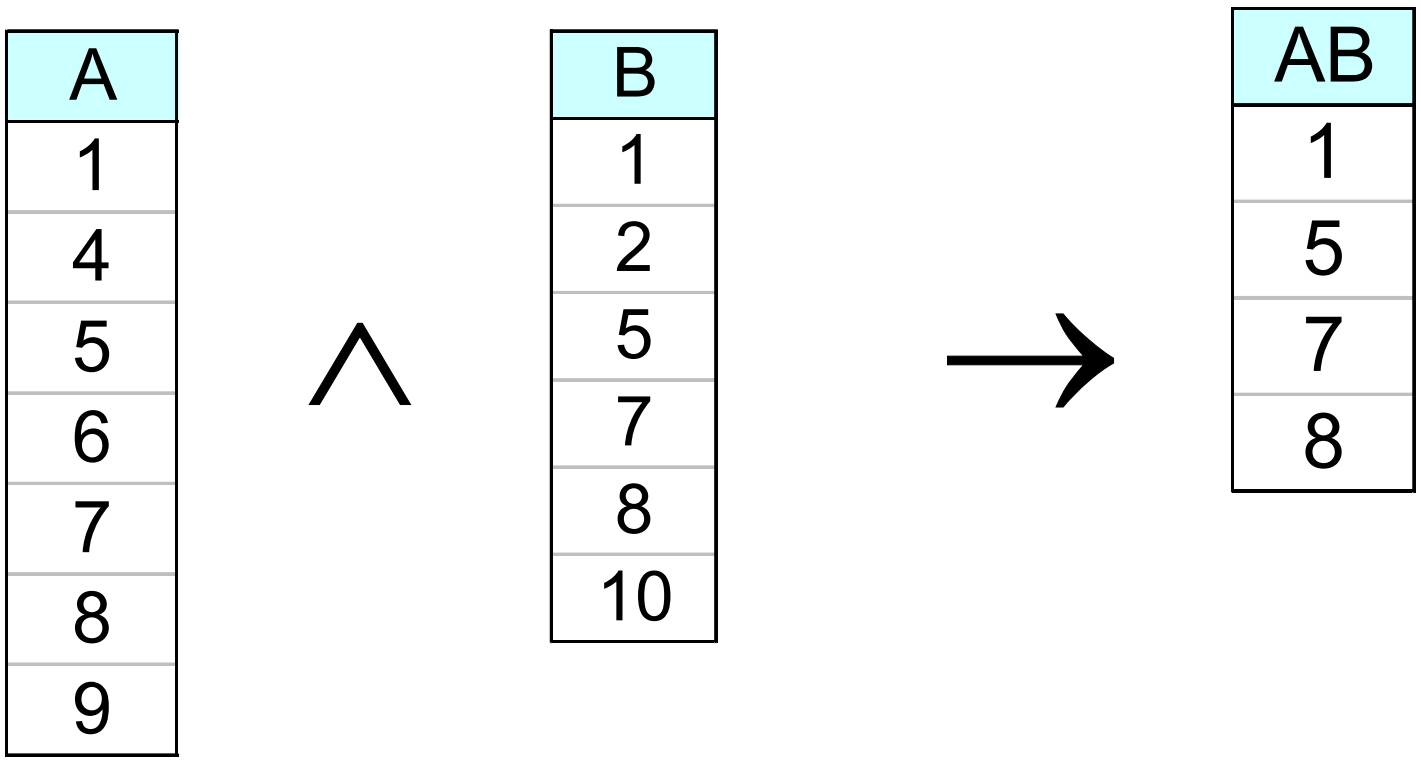
A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

TID-list

All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

ECLAT Algorithm

- Determine support of any k-itemset by intersection tid-lists of two of its $(k-1)$ subsets.
- 3 traversal approaches: top-down, bottom-up, and hybrid.
- Advantage: very fast support counting.
- Disadvantage: intermediate tid-lists may become too large for memory.



All from: Tan et al. "Introduction to Data Mining", 1st Edition, 2006, Pearson Education Inc.

Evaluation of Patterns and Rules

- Association rule algorithms often produce too many rules.
- Many of them are uninteresting or redundant.
- For example: $\{A, B, C\} \rightarrow \{D\}$ and $\{A, B\} \rightarrow \{D\}$ may convey redundant information and also have the same support and confidence
- "Interestingness Measures" can be used to prune and rank the resulting rules.
- We only used support and confidence measures.
- Other examples measures:

$$\text{Lift} = \frac{c(A \rightarrow B)}{s(B)} \tag{3}$$

All from: Kjetil Nørvåg's slides from previous year.

1. Announcements and References
   - Administrative
   - References for Today's Lecture

2. Association Rules
   - Association Rule Mining
   - Definitions
   - Problem Definition
   - Frequent Itemset Generation
   - The Apriori Algorithm
     - Efficient Candidate Generation
     - Complexity Factors
   - Rule Generation
   - Maximal and Closed Itemsets
   - Alternative Methods for Itemset Generation
     - FP-Growth Algorithm
     - ECLAT Algorithm
   - Interestingness Measures