October 24, 2025

Inter-University Research Institute Corporation /
Research Organization of Information and Systems
National Institute of Informatics

Pattern Mining

Data Mining 02 (データマイニング)

Mahito Sugiyama (杉山麿人)

Today's Outline

- Pattern mining
  - Partial order structure + monotonicity
  - The Apriori principle to avoid combinatorial explosion of search space
- Compression of patterns
  - Maximal, closed, ZDD
- Formal concept analysis

Market Basket Analysis

Transaction database

	Purchased items
ID 1:	Milk, Egg, Potato
ID 2:	Milk, Bread
ID 3:	Milk
ID 4:	Bread, Potato
ID 5:	Milk, Egg, Potato
ID 6:	Milk
ID 7:	Milk, Potato
ID 8:	Milk, Egg, Bread, Potato

Binary Representation

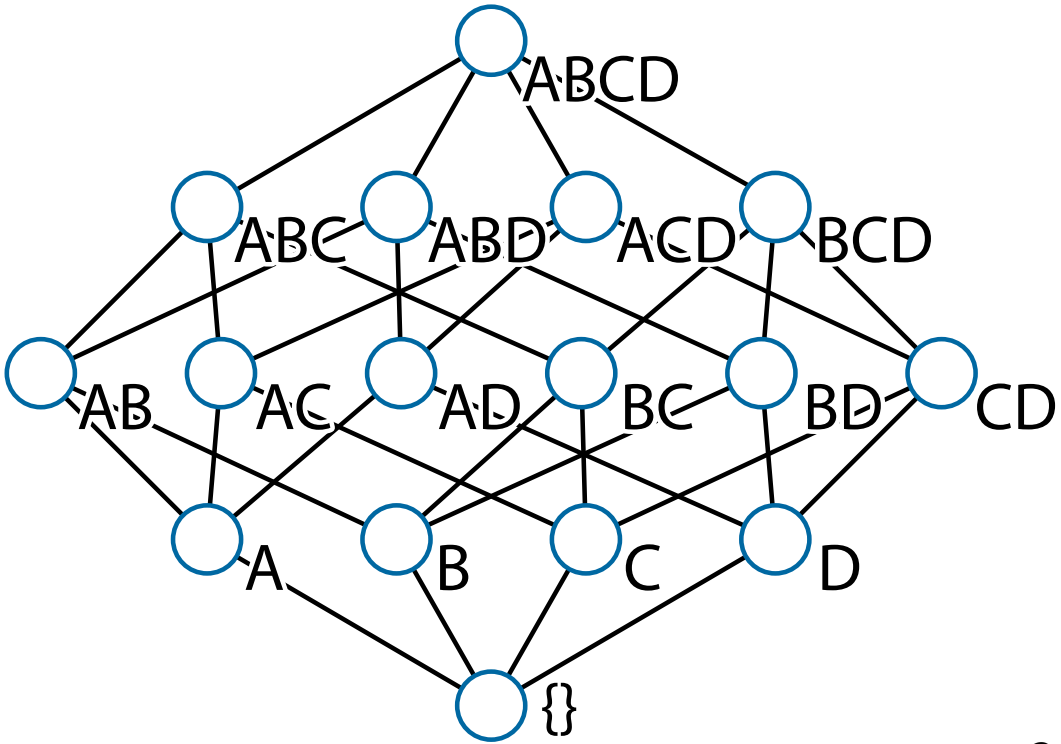
Transaction database
(Binary vectors)

	Milk	Egg	Bread	Potato
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1

Itemset Lattice

Transaction database
(Binary vectors)

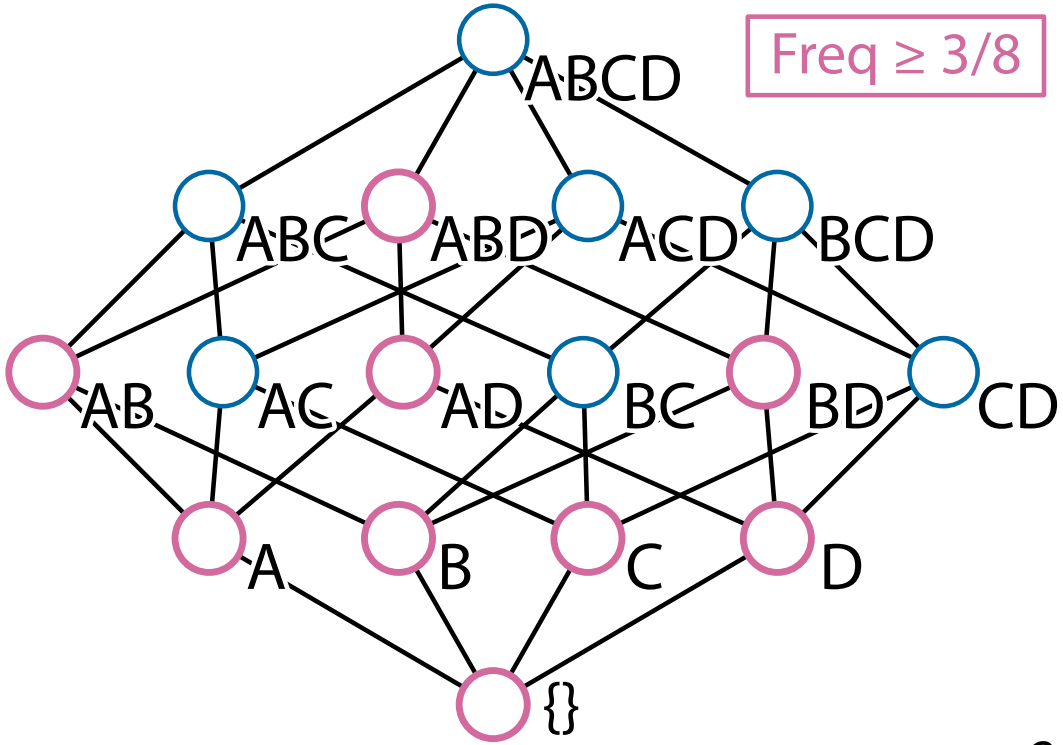
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Find Frequently Copurchased Itemsets

Transaction database
(Binary vectors)

	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Problem Definition of Pattern Mining

- S: the set of patterns, each $x \in S$: pattern
- A dataset is a multiset $D \subseteq S$ with a multiplicity function $\mathbf{1}_D : S \to \mathbb{N}$
  - $\mathbf{1}_D(x)$ is the number of x in D
  - $|D| = \sum_{x \in S} \mathbf{1}_D(x)$
- Let $\xi : S \to \mathbb{R}$ be a function measuring the importance of patterns
  - Assume that $\xi(x)$ can be computed from a dataset D
- **The pattern mining problem:**
  Given a threshold σ, enumerate the set $F = \{x \in S \mid \xi(x) \geq \sigma\}$

How About Generate-And-Test?

- Let the set of items $V = \{1, 2, \dots, n\}$
- $S = 2^V$ in itemset mining, each pattern $x \subseteq V$ is called an itemset
- Generate-and-test strategy:
  - (i) Pick up an itemset $x \subseteq V$
  - (ii) Compute its importance $\xi(x)$
  - (iii) Output x if $\xi(x) \geq \sigma$
  - (iv) Repeat the above for all itemsets

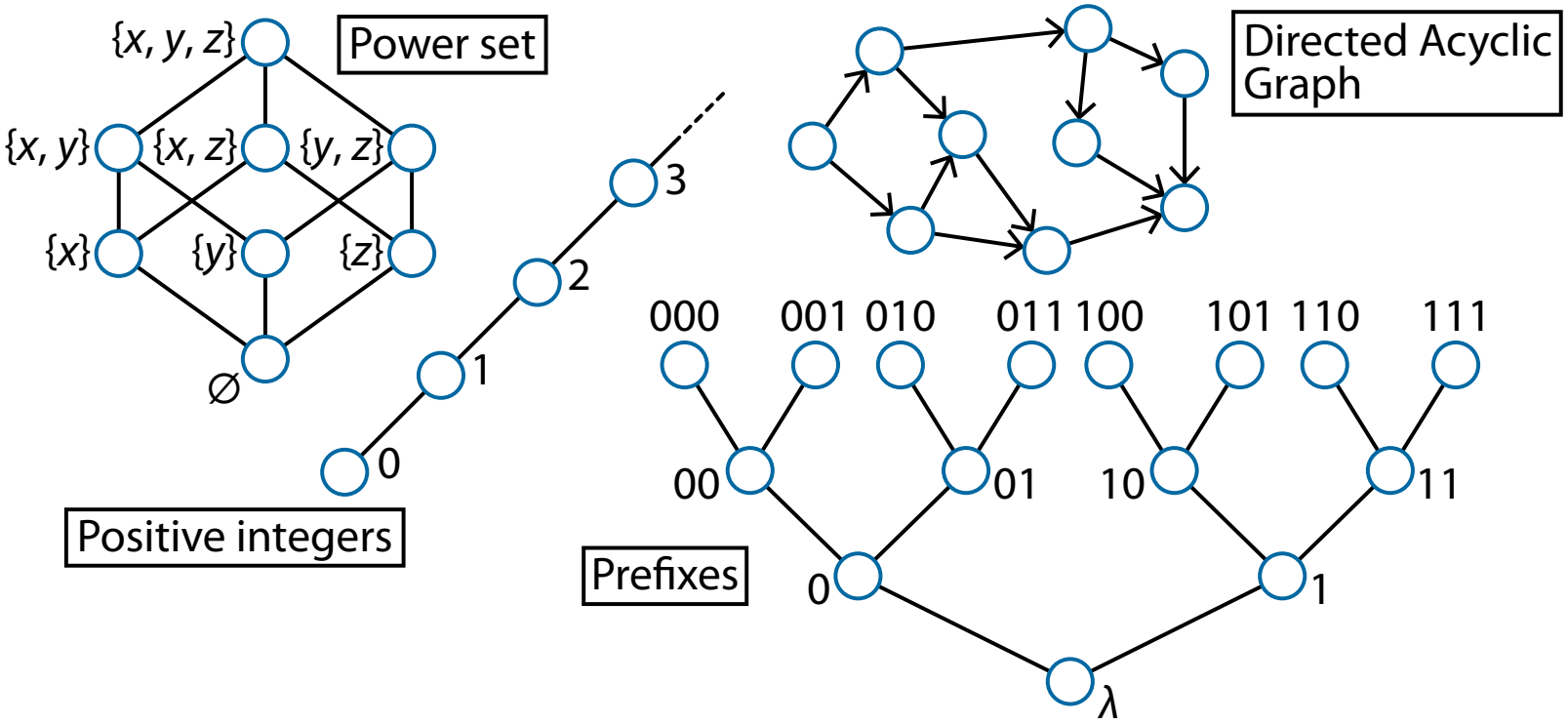
Combinatorial Explosion!!

$n = \|V\|$	# Patterns	Approximate time required
10	2^{10}	0.00000057 sec.
20	2^{20}	0.00059 sec.
30	2^{30}	0.6 sec.
40	2^{40}	10.2 min.
50	2^{50}	174 hours.
70	2^{70}	7 million days
100	2^{100}	8 thousand billion days

Partial Order Structures

- Let us assume that S is a poset (partially ordered set) $(S, \preceq)$
- "$\preceq$" is a partial order if
  - (i) $x \preceq x$ (reflexivity)
  - (ii) $x \preceq y, y \preceq x \Rightarrow x = y$ (antisymmetry)
  - (iii) $x \preceq y, y \preceq z \Rightarrow x \preceq z$ (transitivity)
- If $x \preceq y$, a pattern y is more "precise" and x is more "general"
  - Deriving y from x is refinement, x from y is generalization
  - The upper set $\uparrow x = \{s \in S \mid s \succeq x\}$
- In itemset mining, $x \subseteq y \iff x \preceq y$

Various Posets (Partially Ordered Sets)



Order Between Syntax And Semantics

- To use the partial order structure of $(S, \preceq)$, ξ should be order homomorphism, i.e.,

  $x \preceq y \Rightarrow \xi(x) \leq \xi(y)$

- The structure "$\preceq$" in the syntax world and that of "$\leq$" in the semantics world matches $\rightarrow$ efficient search!
- In reality, we need patterns with high values of ξ, so we require

  $x \preceq y \Rightarrow \xi(x) \geq \xi(y)$ (e.g. $\xi'(x) = 1/\xi(x)$)
  - ξ is anti-monotonic with respect to $\preceq$

Frequency

- The most popular instance of ξ is the frequency η define as

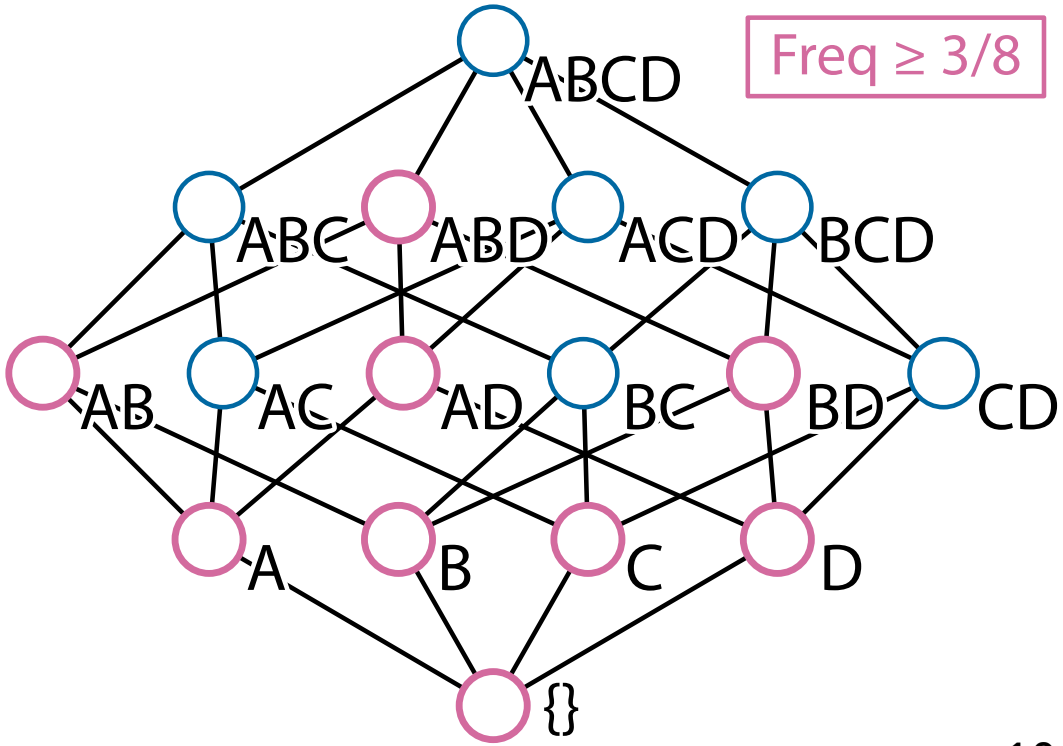
$$\eta(x) = \frac{1}{|D|} \sum_{s \succeq x} \mathbf{1}_D(s)$$

  - $\eta'(x) = |D|\eta(x)$ is called the support
    - It denotes how many times x appears in the dataset D
  - η (and η') is always anti-monotonic
- A pattern $x \in S$ is called a frequent pattern if $\eta(x) \geq \sigma$
- The supporting set of x is $D \cap \uparrow x$
  - $\eta'(x) = |D \cap \uparrow x|$

Frequent Patterns

Transaction database
(Binary vectors)

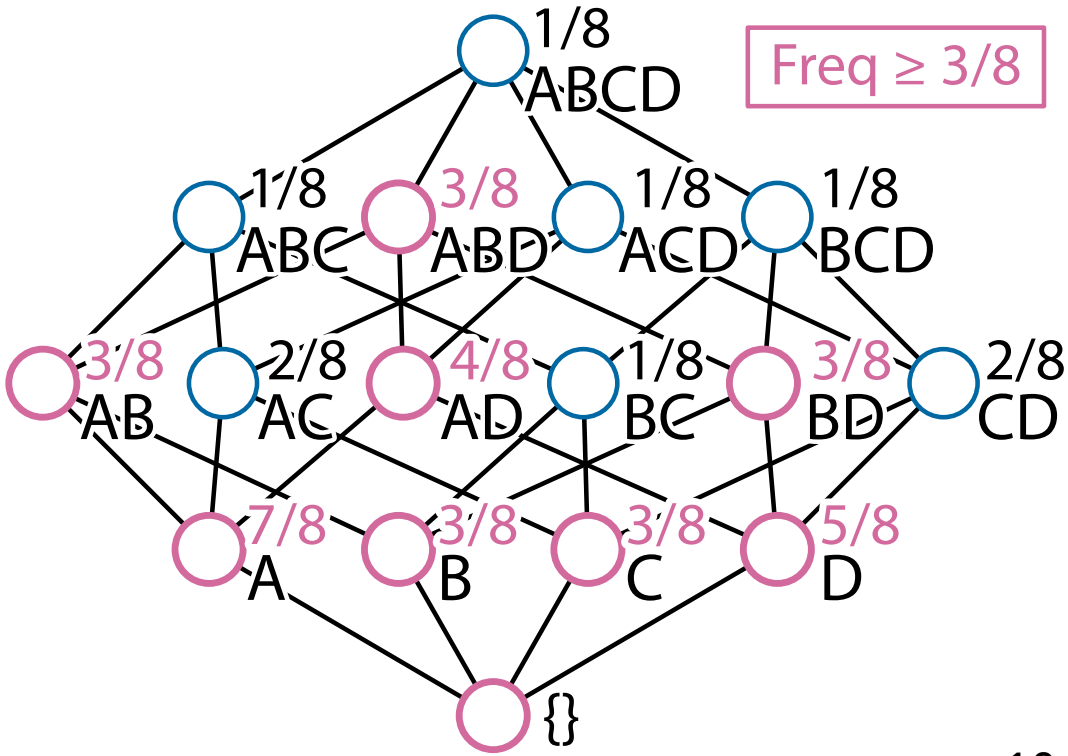
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Anti-Monotonicity of Frequency

Transaction database
(Binary vectors)

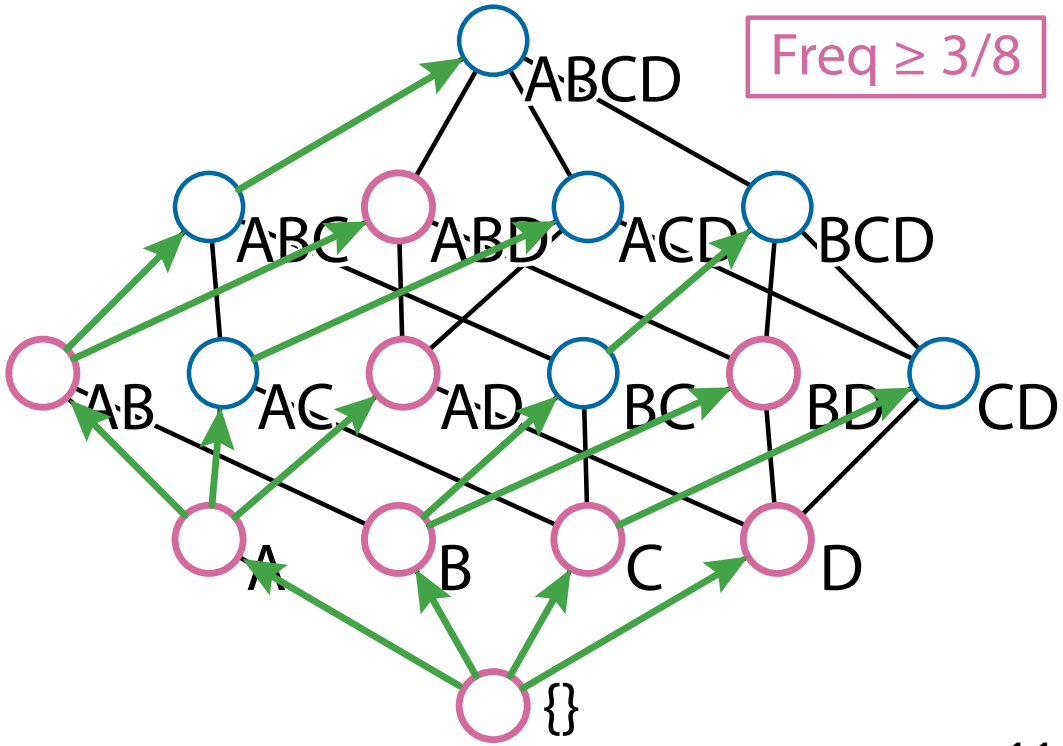
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Prefix-Based Search Tree

Transaction database
(Binary vectors)

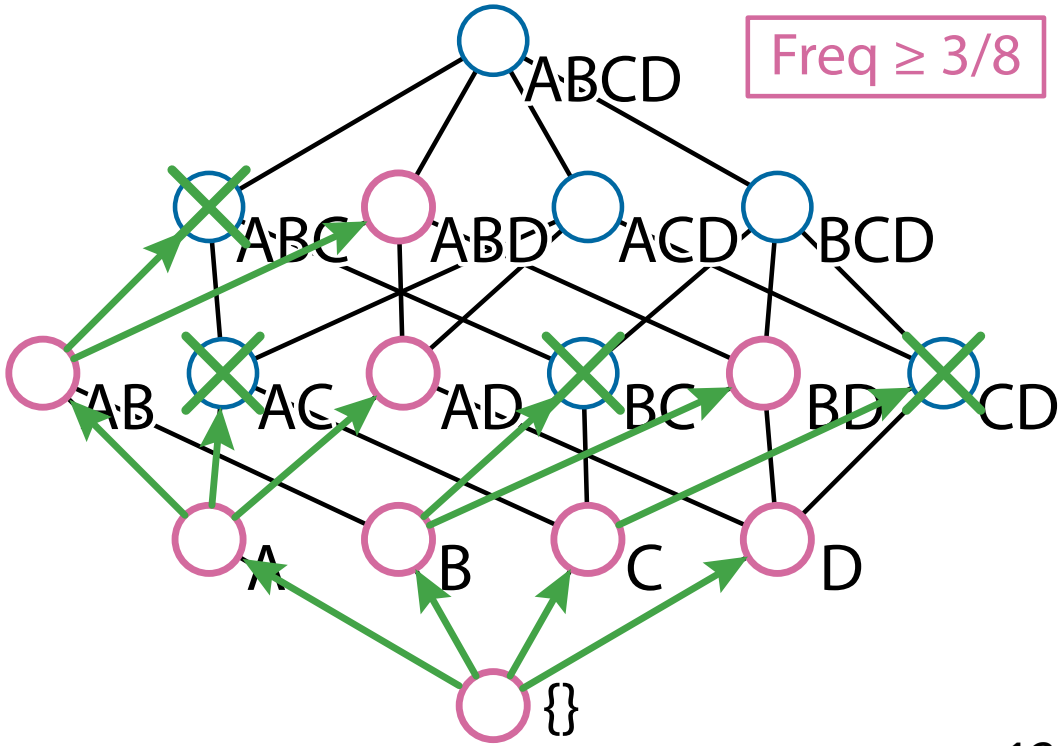
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Apriori Principle

Transaction database
(Binary vectors)

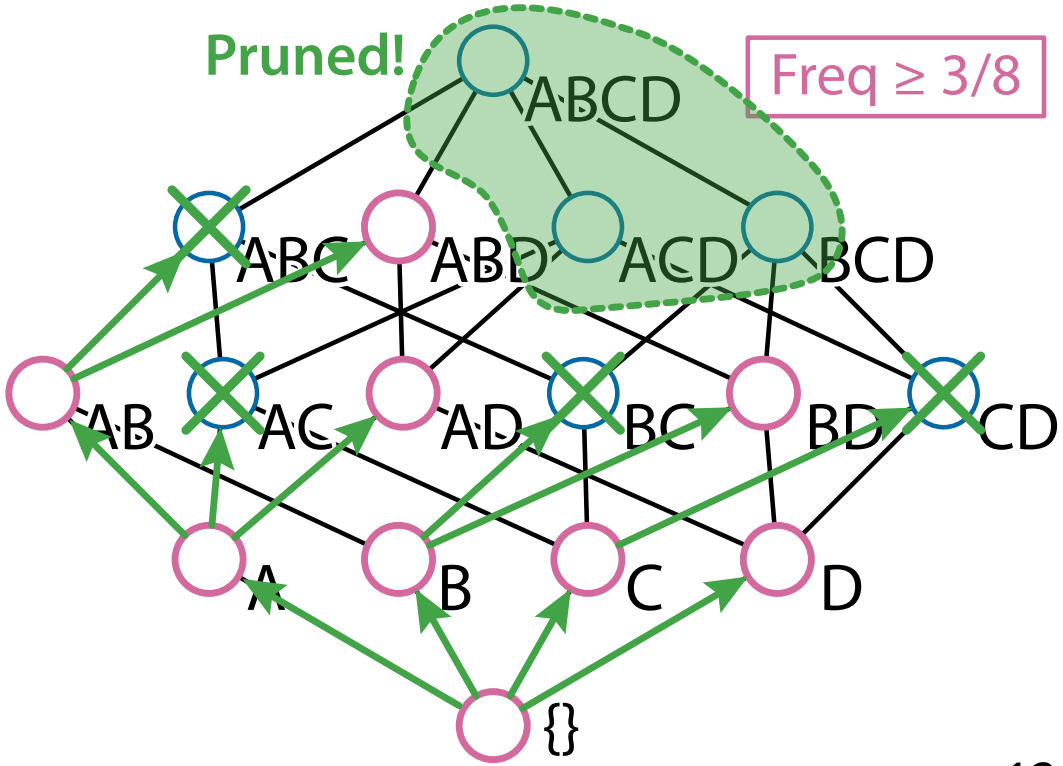
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Apriori Principle

Transaction database
(Binary vectors)

	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



The Apriori Algorithm

Apriori algorithm

```
PatternMining(σ)
    PatternEnumeration(⊥, σ)
PatternEnumeration(x, σ)
    for each s ⋗ x
        if ξ(s) ≥ σ
            Output s
            PatternEnumeration(s, σ)
```

- $x \lessdot s \iff (x \prec s \text{ and } x \preceq y \prec s) \Rightarrow x = y$

VLDB 1994

(KDD started in 1995)

Fast Algorithms for Mining Association Rules

Rakesh Agrawal Ramakrishnan Srikant*

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We consider the problem of discovering association rules between items in a large database of sales transactions. We present two new algorithms for solving this problem that are fundamentally different from the known algorithms. Empirical evaluation shows that these algorithms outperform the known algorithms by factors ranging from three for small problems to more than an order of magnitude for large problems. We also show how the best features of the two proposed algorithms can be combined into a hybrid algorithm, called AprioriHybrid. Scale-up experiments show that AprioriHybrid scales linearly with

tires and auto accessories also get automotive services done. Finding all such rules is valuable for cross-marketing and attached mailing applications. Other applications include catalog design, add-on sales, store layout, and customer segmentation based on buying patterns. The databases involved in these applications are very large. It is imperative, therefore, to have fast algorithms for this task.

The following is a formal statement of the problem [4]: Let $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. Let $\mathcal{D}$ be a set of transactions, where each transaction T is a set of items such that $T \subset$

Association Rules

- An association rule is $x \to y$, where $x, y \in S$ and $x \cap y = \emptyset$
- The confidence of $x \to y$ is the conditional probability of the occurrence of $x \cup y$ given x, i.e.,

  $$\text{conf}(x \to y) = \frac{\eta(x \cup y)}{\eta(x)}$$

  - A rule is strong if the confidence is larger than a threshold
  - In market basket analysis, which item will be bought if x is bought
- Association rule finding is a post processing of FPM

Implementations and Datasets

- The fastest implementation is LCM (by Uno sensei at NII)
  - http://research.nii.ac.jp/~uno/code/lcm.html
  - It won the FIMI04 competition
- Other algorithms: FP-growth, Eclat
- FIMI data repository
  - A famous repository of benchmark dataset for itemset mining
  - http://fimi.ua.ac.be/data/

Compressing Patterns

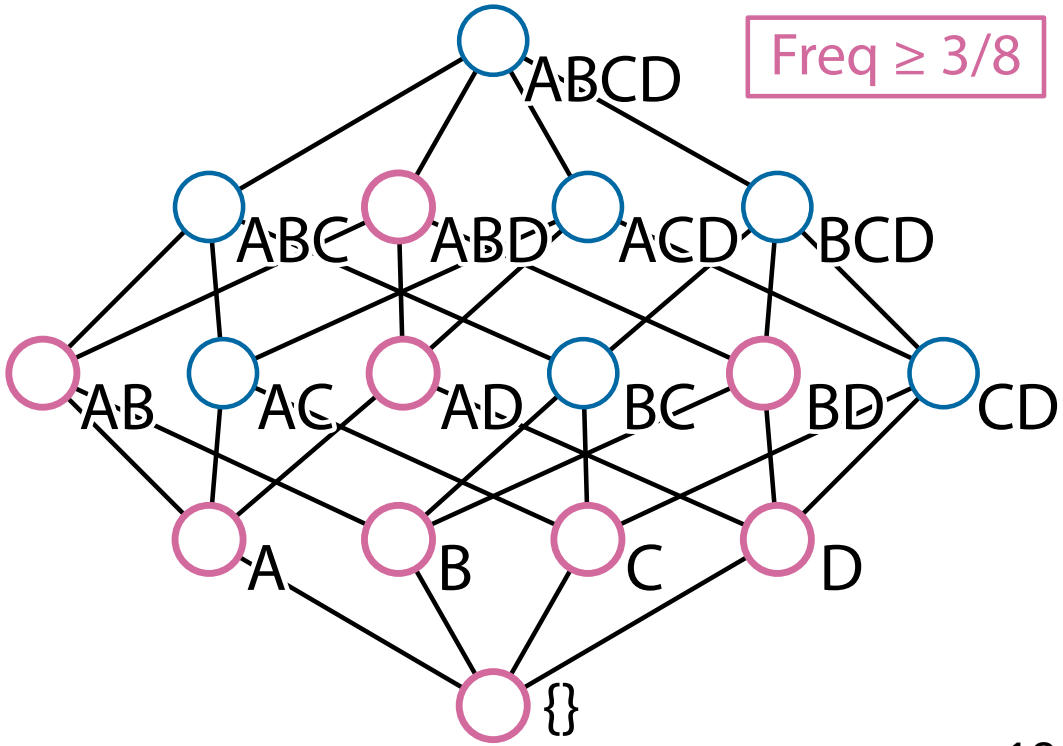
- Too many patterns will be generated
  - Many of them are redundant
- How to compress/summarize patterns?

1. Maximal patterns
   - All frequent patterns can be recovered but their frequencies are lost
2. Closed patterns
   - All frequent patterns with their frequencies can be recovered
3. ZDD (Zero-suppressed binary Decision Diagram)

Maximal Patterns (no superset is frequent)

Transaction database
(Binary vectors)

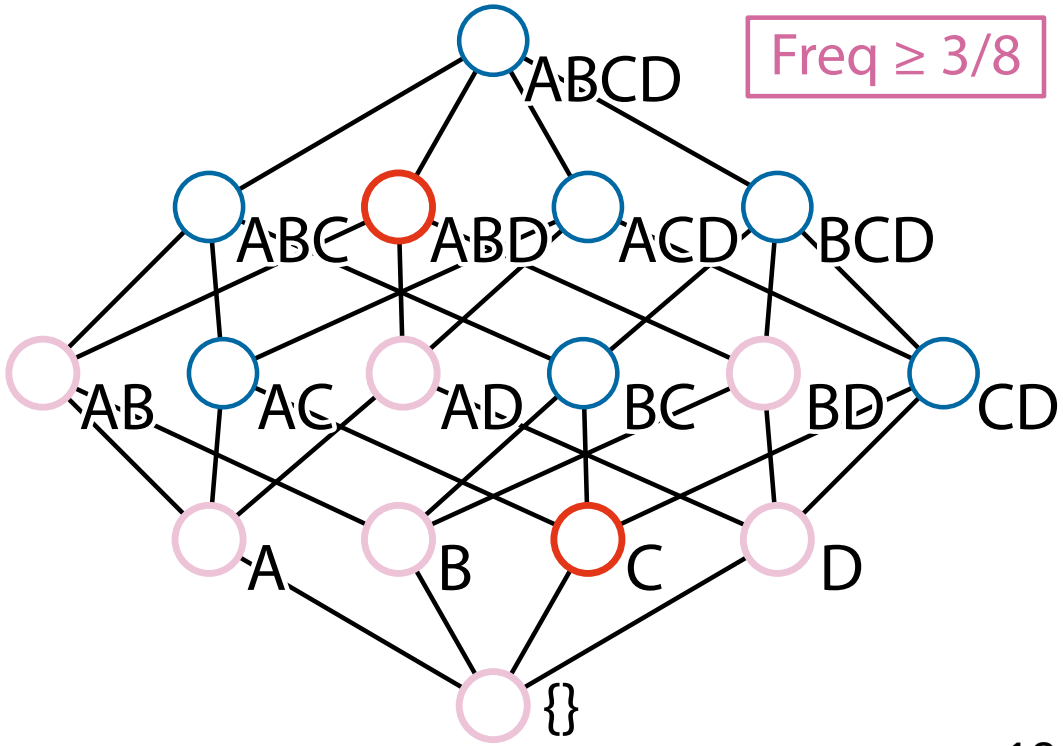
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Maximal Patterns (no superset is frequent)

Transaction database
(Binary vectors)

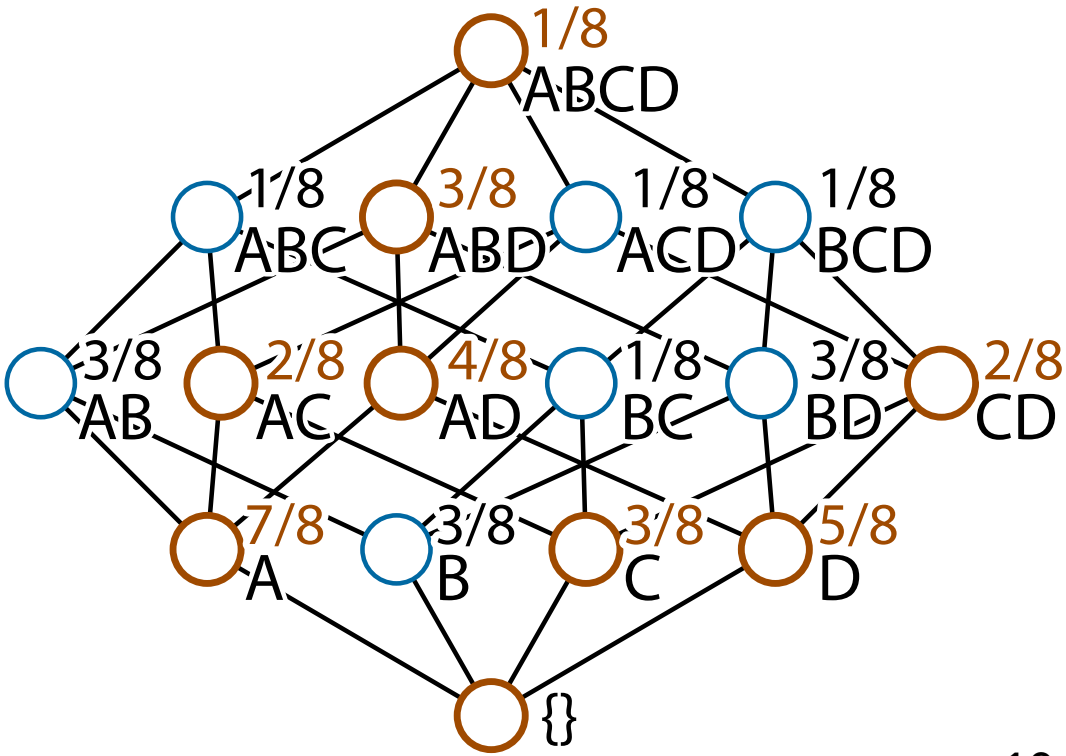
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Closed Patterns (no superset with the same freq.)

Transaction database
(Binary vectors)

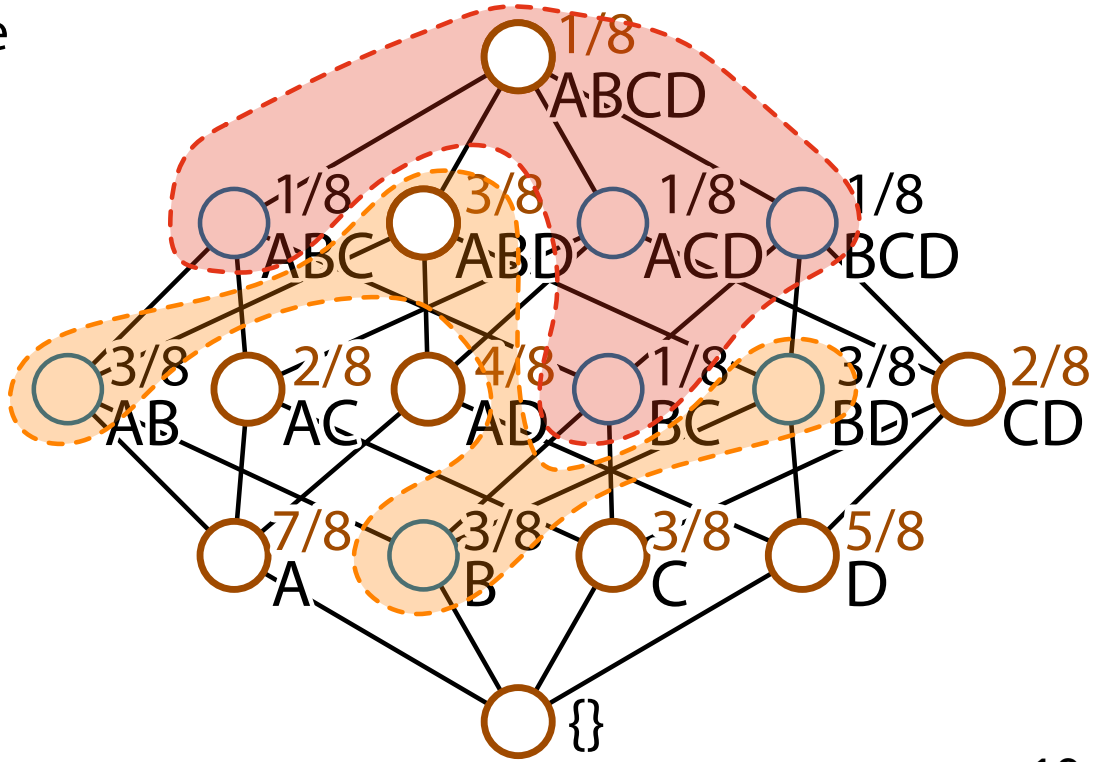
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Closed Patterns (no superset with the same freq.)

Transaction database
(Binary vectors)

	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Pergamon

Information Systems Vol. 24, No. 1, pp. 25–46, 1999
© 1999 Elsevier Science Ltd. All rights reserved
Printed in Great Britain
0306-4379/99 $20.00 + 0.00

PII: S0306-4379(99)00003-4

EFFICIENT MINING OF ASSOCIATION RULES USING CLOSED ITEMSET LATTICES†

NICOLAS PASQUIER, YVES BASTIDE, RAFIK TAOUIL and LOTFI LAKHAL

Laboratoire d'Informatique (LIMOS), Université Blaise Pascal - Clermont-Ferrand II, Complexe Scientifique des Cézeaux, 63177 Aubière Cedex France

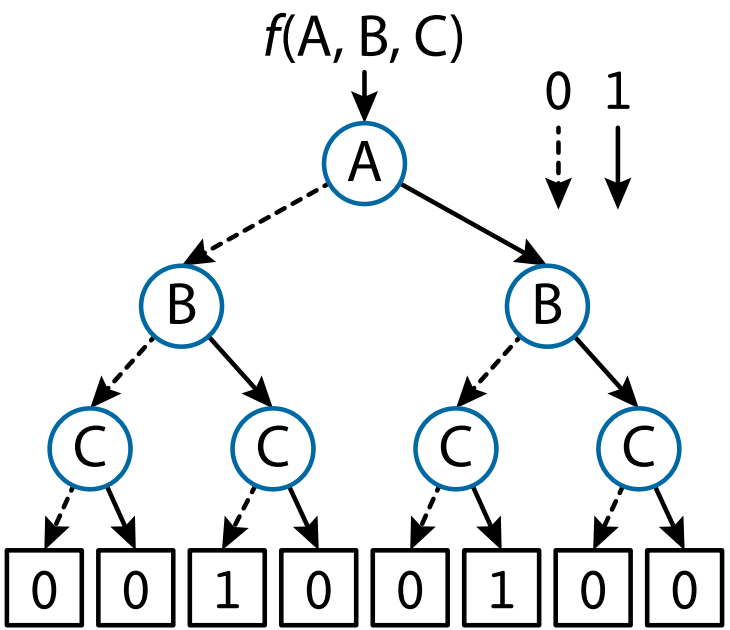
(Received 13 June 1998; in final revised form 16 October 1998)

Abstract — Discovering association rules is one of the most important task in data mining. Many efficient algorithms have been proposed in the literature. The most noticeable are Apriori, Mannila's algorithm, Partition, Sampling and DIC, that are all based on the Apriori mining method: pruning the subset lattice (itemset lattice). In this paper we propose an efficient algorithm, called Close, based on a new mining method: pruning the closed set lattice (closed itemset lattice). This lattice, which is a sub-order of the subset lattice, is closely related to Wille's concept lattice in formal concept analysis. Experiments comparing Close to an optimized version of Apriori showed that Close is very efficient for mining dense and/or correlated data such as census style data, and performs reasonably well for market basket style data. ©1999 Elsevier Science Ltd. All rights reserved

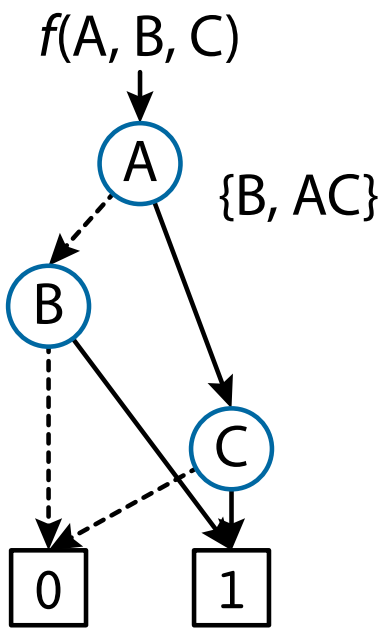
Represent Combinations by ZDD

A	B	C	*f*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

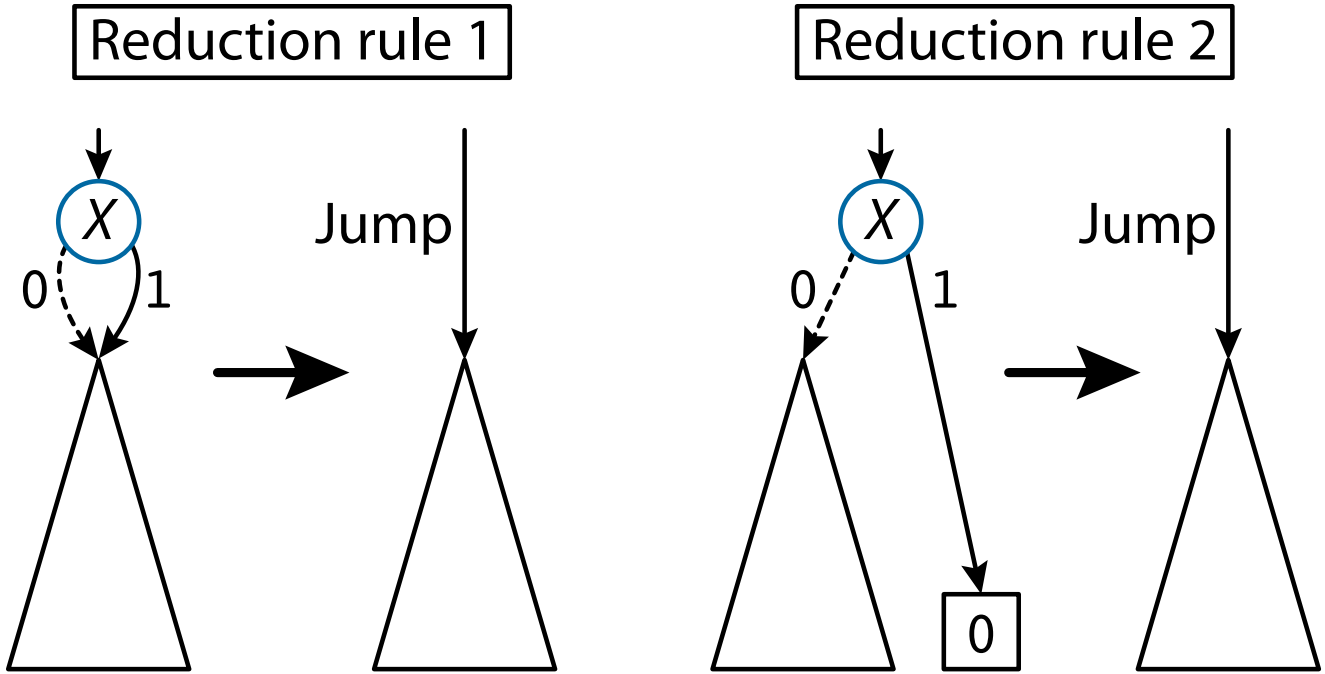
Binary decision tree



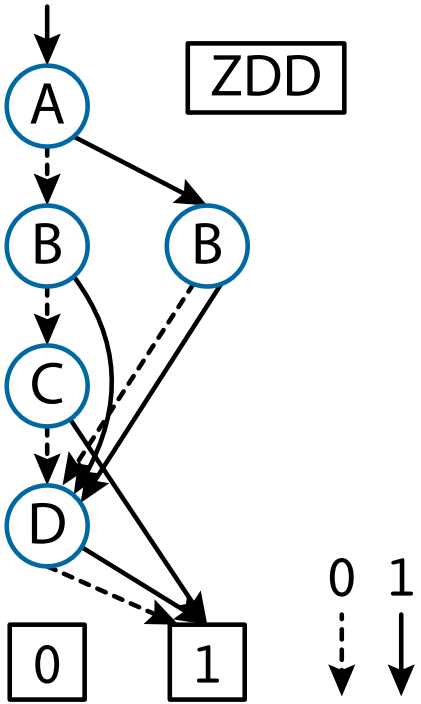
ZDD

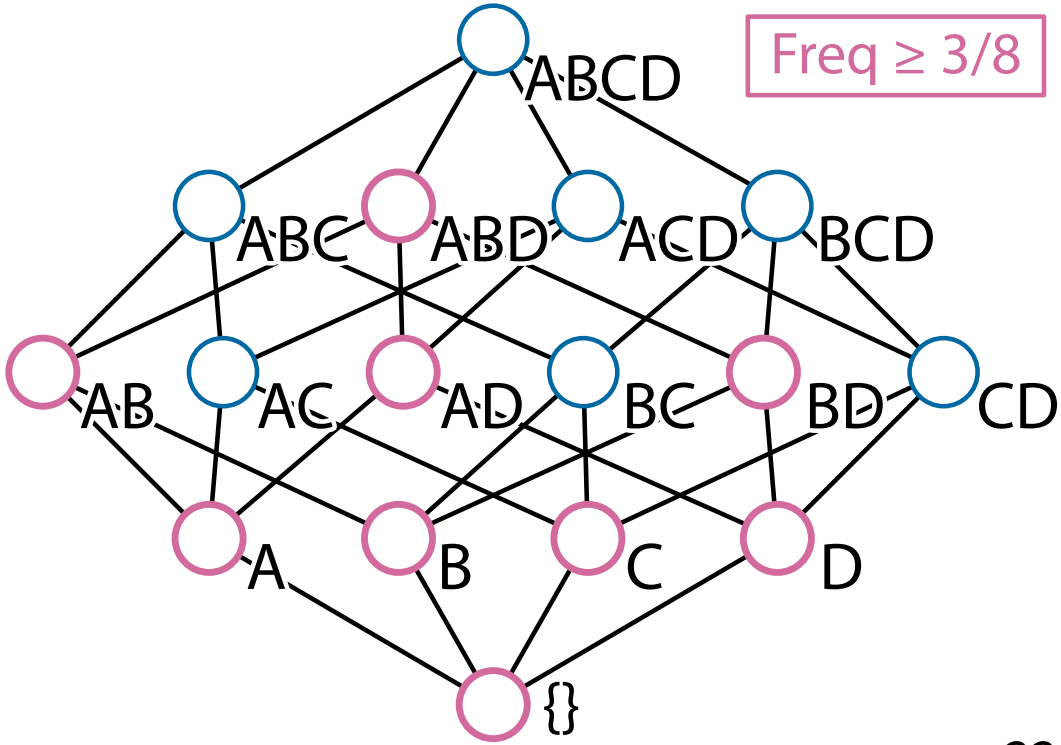


Two Reduction Rules of ZDD



Compress Frequent Patterns by ZDD





PAKDD 2008

LCM over ZBDDs: Fast Generation of Very Large-Scale Frequent Itemsets Using a Compact Graph-Based Representation

Shin-ichi Minato[1], Takeaki Uno[2], and Hiroki Arimura[1]

[1] Graduate School of Information Science and Technology,
Hokkaido University, Sapporo, 060-0814 Japan
{minato,arim}@ist.hokudai.ac.jp
[2] National Institute of Informatics, Tokyo 101–8430, Japan
uno@nii.ac.jp

Abstract. Frequent itemset mining is one of the fundamental techniques for data mining and knowledge discovery. In the last decade, a number of efficient algorithms have been presented for frequent itemset mining, but most of them focused on only enumerating the itemsets that

Formal Concept Analysis

- Formal concept analysis builds a concept hierarchy
  - A concept coincides with a closed pattern in pattern mining
- A context is a triple (G, M, I)
  - G, M: a set of objects and attributes, $I \subseteq G \times M$: a binary relation
  - G corresponds to the set of individuals, M the set of items
- For subsets $A \subseteq G$ and $B \subseteq M$, define the mapping $'$ as

  $A' = \{m \in M \mid (g, m) \in I \text{ for all } g \in A\}$

  $B' = \{g \in G \mid (g, m) \in I \text{ for all } m \in B\}$

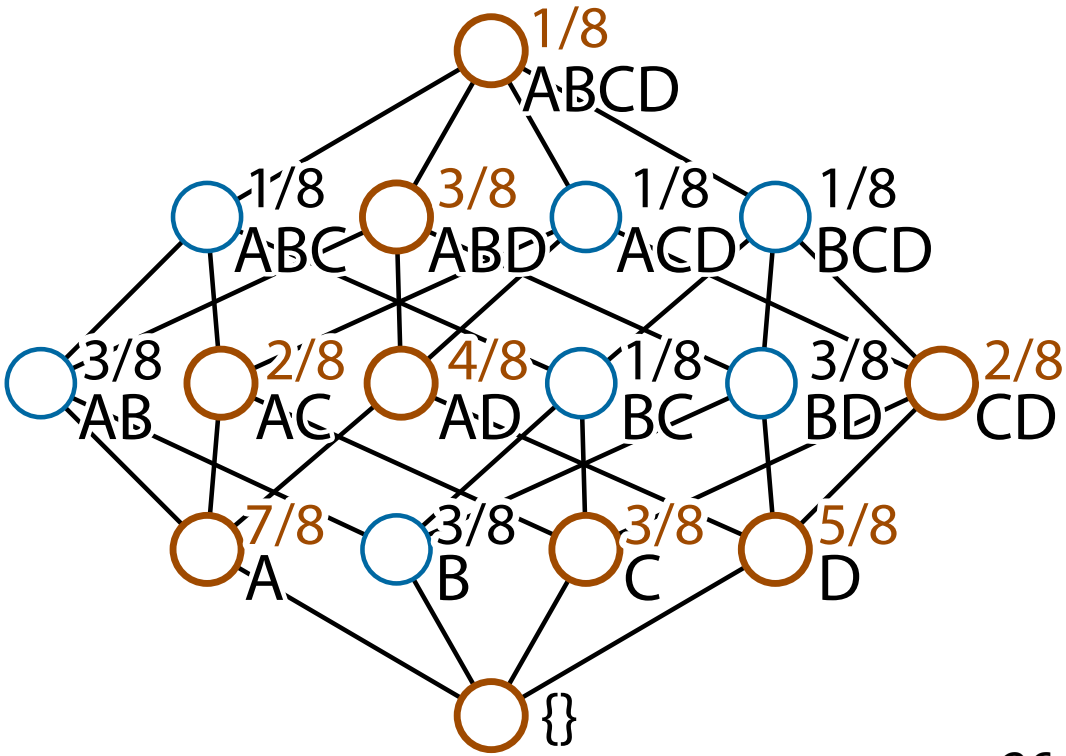
Concept Lattice And Closure Operator

- A pair (A, B) is called a concept if $A' = B$ and $A = B'$
  - (A, B) is a concept $\iff$ B is a closed pattern
- The set of all concepts is called a concept lattice
- The operator $'$ is a Galois connection between 2^G and 2^M
  - $A' \subseteq B \iff A \subseteq B'$
- The mapping $''$ (applying $'$ twice) is a closure operator on (G, M, I)
  - A subset $A \subseteq G$ is a concept $\iff$ $A'' = A$
  - Intuitively, A'' expands A as much as possible without changing their frequencies

Concept Lattice

Transaction database
(Binary vectors)

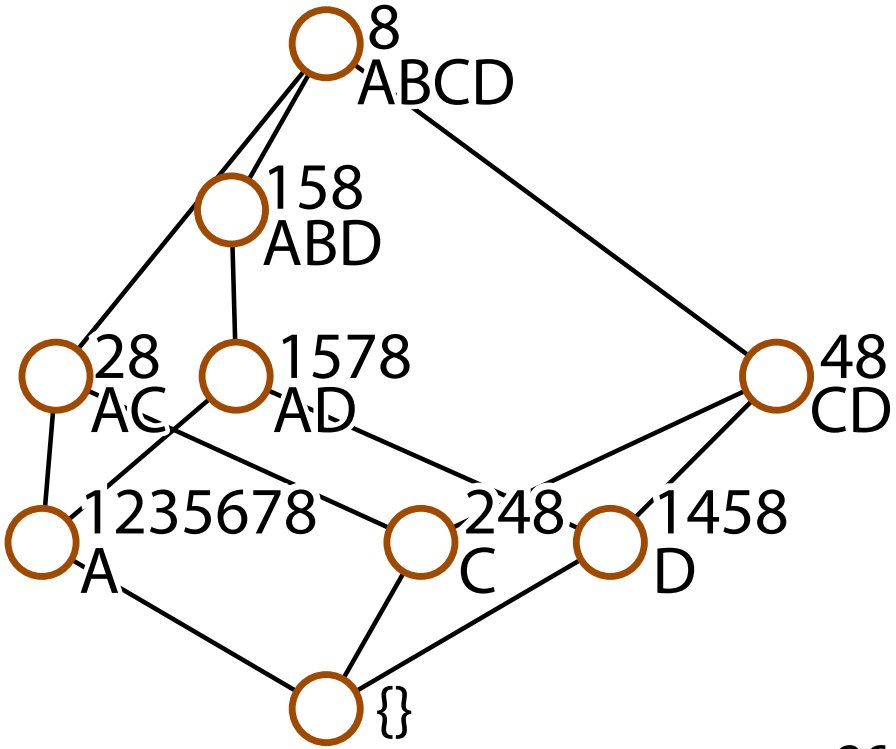
	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



Concept Lattice

Transaction database
(Binary vectors)

	A	B	C	D
ID 1:	1	1	0	1
ID 2:	1	0	1	0
ID 3:	1	0	0	0
ID 4:	0	0	1	1
ID 5:	1	1	0	1
ID 6:	1	0	0	0
ID 7:	1	0	0	1
ID 8:	1	1	1	1



LCM and Reverse Search

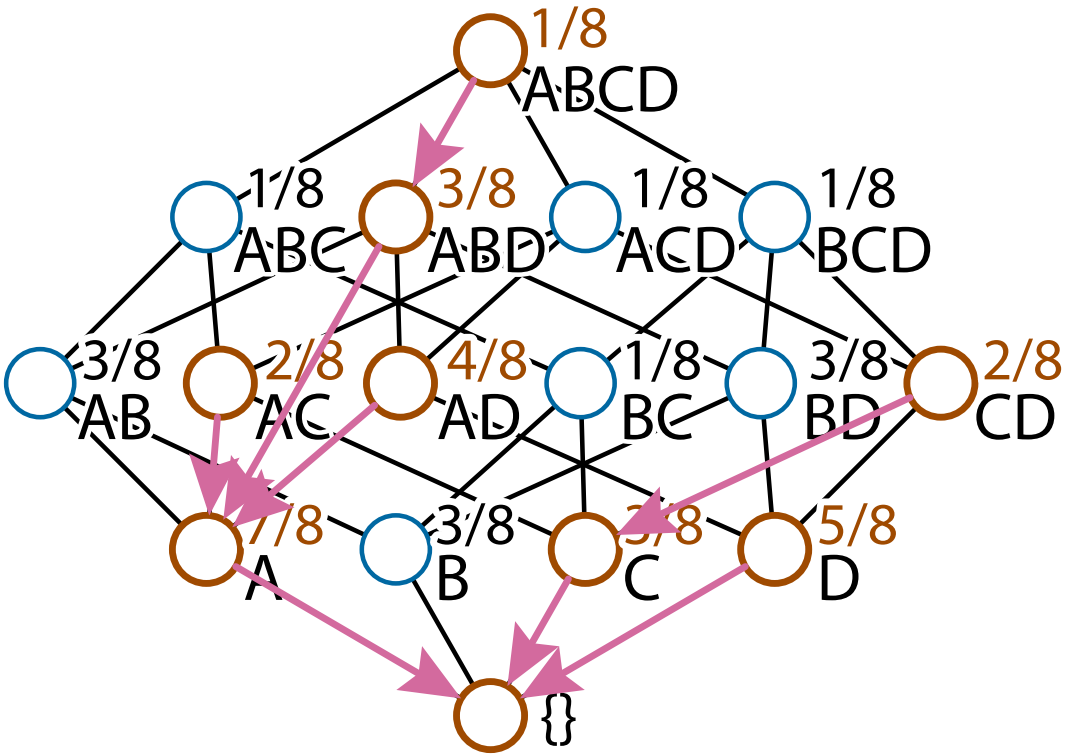
- LCM (Linear time Closed sets Miner) jumps from a closed set to the next closed set in linear time (linear delay) using ″
- Its essential is reverse search (Avis & Fukuda, 1996)
- Define a reduction map (RM), which satisfies:
  - (i) RM is unique for all solution
  - (ii) By iteratively applying RM, we can arrive at the root of the tree from any solution
  - (iii) Children generation is the inverse of RM

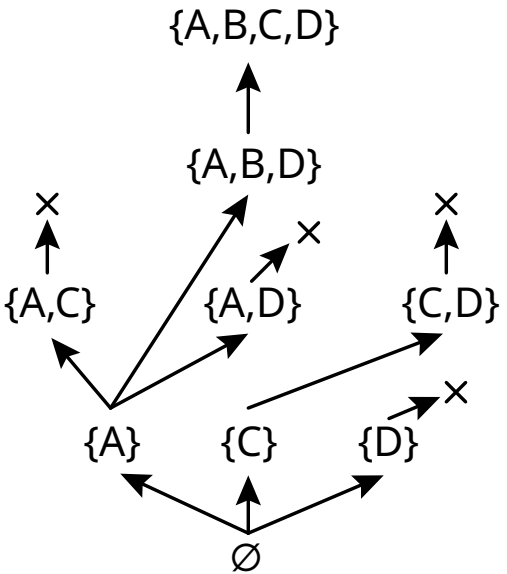
  then it is guaranteed that reverse search enumerates all solutions

Reduction Map and Search Tree for LCM

- Reduction map for LCM:
  (i) Remove items one by one until the frequency changes
  (ii) Apply closure operator to it
- We identify items and their indices $1, 2, \ldots, n$
- Inverse of the reduction map is called prefix preserving closure extension (PPC extension)
  - Define closure tail cl_tail(x) as the minimum item j such that $(x \cap \{1, 2, \ldots, j\})'' = x$
  - An itemset $y = (x \cup \{i\})''$ is PPC extension of x if $i \notin x$ and $i >$ cl_tail(x) and $y \cap \{1, \ldots, i-1\} = x \cap \{1, \ldots, i-1\}$

Reduction Map and Search Tree





Enumeration by PPC Extension

- One can enumerate all closed itemsets using PPC extension
  - (i) For all y with $\eta(y) \neq 1$, there exists a closed itemset x such that y is a PPC extension of x
  - (ii) For all y with $\eta(y) \neq 1$, there exists exactly one closed itemset x that has y as a PPC extension
  - (iii) For all closed set x and its PPC extension y, $\eta(x) > \eta(y)$ always holds

Theorem. LCM outputs all closed itemsets in a given dataset D. The time complexity of generating a closed itemset x is $O(|x| \cdot |D| \cdot \max_{y \in D} |y|)$.