Frequent Itemset Mining

Nadjib LAZAAR

LIRMM- UM
COCONUT Team

(PART III)

IMAGINA 18/19

Webpage: http://www.lirmm.fr/~lazaar/teaching.html
Email: lazaar@lirmm.fr

LCM Algorithm
Linear Closed Item Set Miner

[Uno et al., 03] (version 1) [Uno et al., 04, 05] (versions 2 & 3)

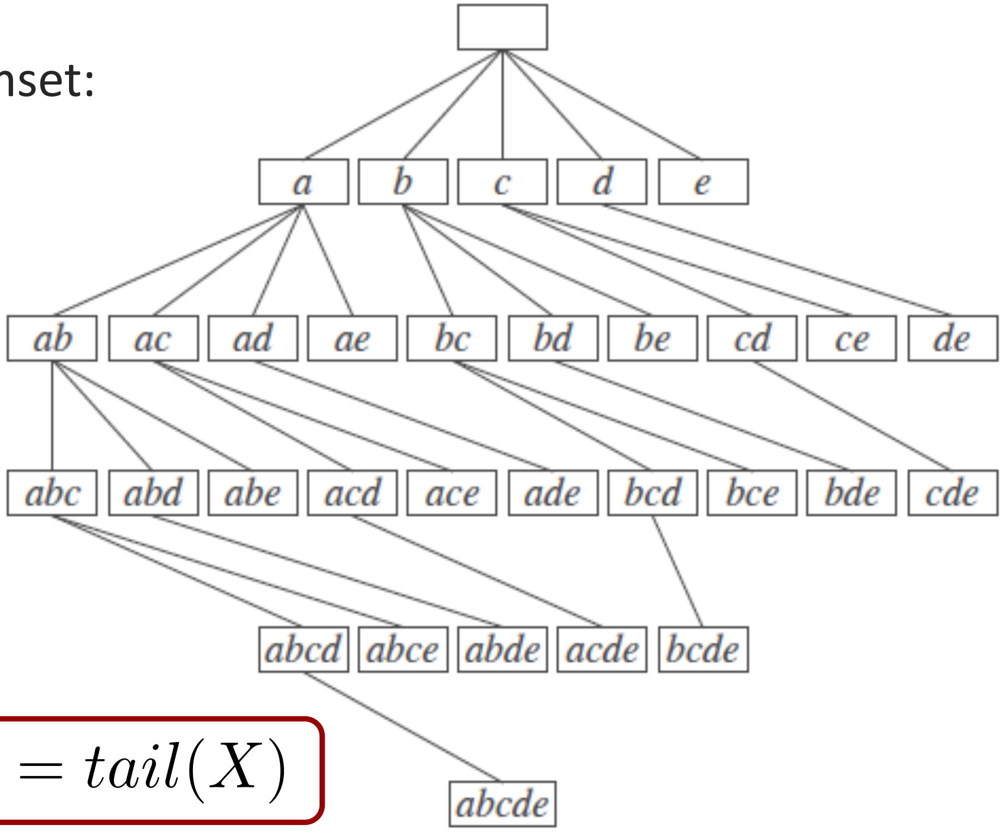
LCM: basic ideas

- The itemset candidates are checked in lexicographic order (depth-first traversal of the prefix tree)
- Step by step elimination of items from the transaction database; recursive processing of the conditional transaction databases
- Maintains both a horizontal and a vertical representation of the transaction database in parallel.
  - Uses the vertical representation to filter the transactions with the chosen split item.
  - Uses the horizontal representation to fill the vertical representation for the next recursion step (no intersection as in Eclat algorithm).

LCM: basic ideas

- The itemset candidates are checked in lexicographic order (depth-first traversal of the prefix tree)
- Let $X = \{x_1, \ldots, x_n\}$ be an itemset:

$$x_1 < \ldots < x_n$$
$$tail(X) = x_n$$
$$X(i) = \{x_1, \ldots, x_i\}$$
$$X[j] = X \cup \{j\}$$



$$\boxed{X \text{ prefix of } Y \Leftrightarrow X = Y(i) \wedge i = tail(X)}$$

LCM: basic ideas

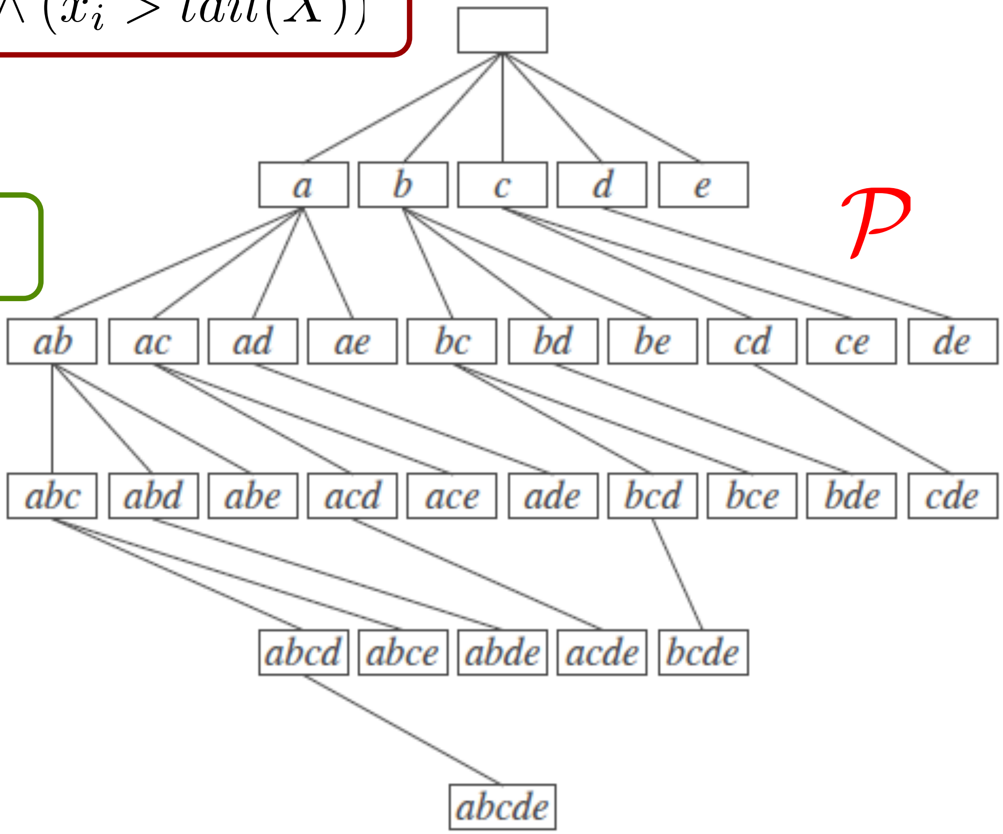
- Parent-child relation $\mathcal{P}$ is defined as:

$$X = \mathcal{P}(Y) \Leftrightarrow (Y = X \cup \{x_i\}) \wedge (x_i > tail(X))$$

- Or equivalently:

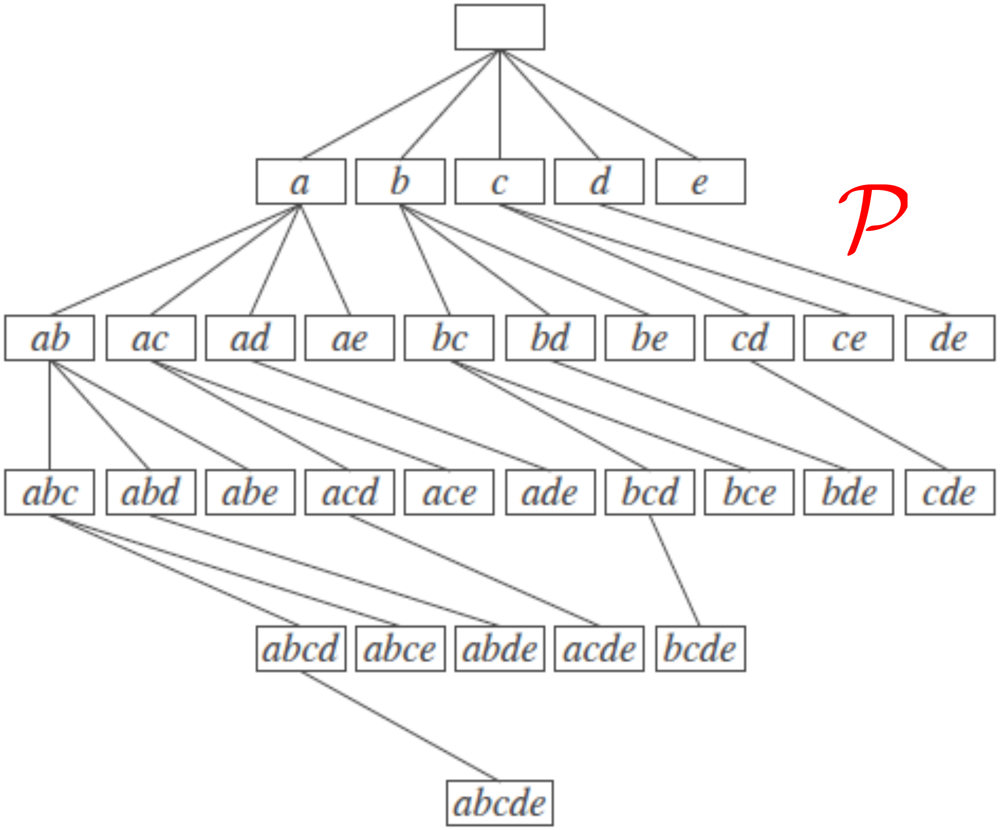
$$X = \mathcal{P}(Y) \Leftrightarrow X = Y \setminus tail(Y)$$

- $\mathcal{P}$ is an acyclic relation



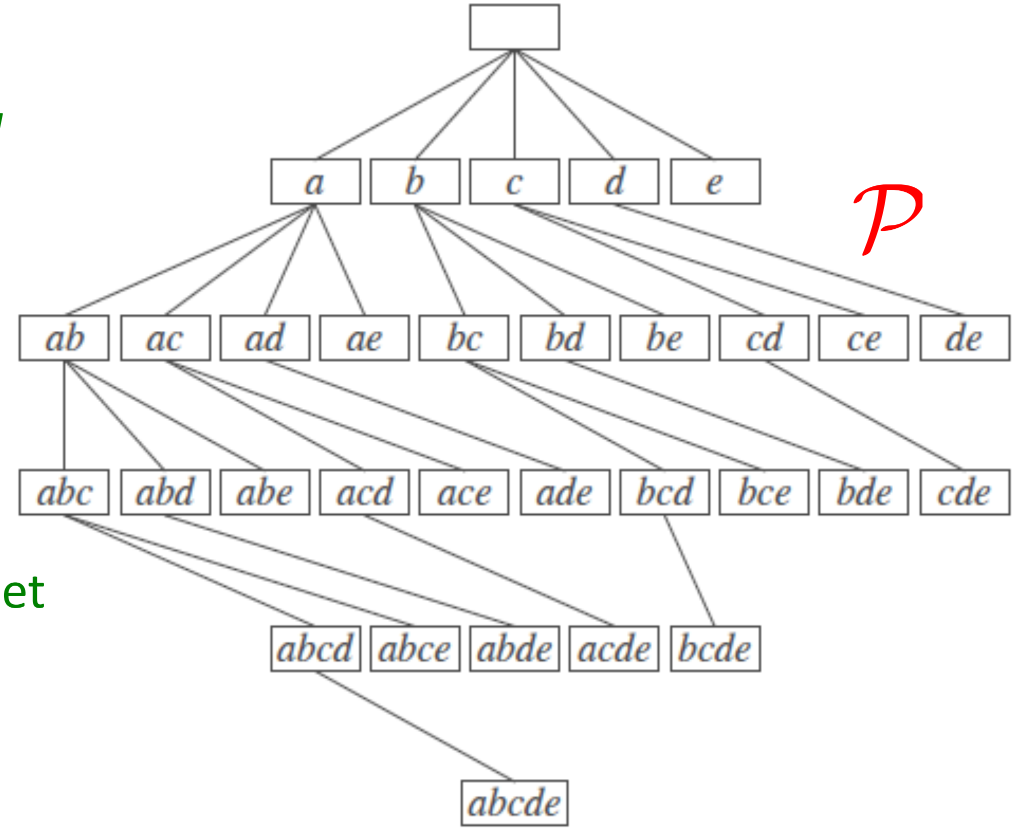
Example (7)

- *bcd and cda are candidates?*
- *tail(abde)=?, tail(a)=?, tail(bd)=?*
- Let X= abde
  - X(1)=?, X(2)=?
  - X(3)=?, X(4)=?
  - X(5)=X(i: i>3)=?
- P(bc)=?, P(ade)=?, P(c)=?



Example (7)

- *bcd* is a candidate, *cda* is not
- *tail(abde)=e, tail(a)=a, tail(bd)=d*
- Let X= abde
  - X(1)=a, X(2)=ab
  - X(3)=abd, X(4)=abde
  - X(5)=X(i: i>3)=X(4)
- P(bc)=b, P(ade)=ad, P(c)=emptyset



Closure (recall)

- A set *S* has a *closure* under an operation *f* iff:

  Forall *x* in *S*, *f(x)* in *S*

  The set *S* is then closed under *f*

- A closure operation is:
  - *Increasing* or *extensive* (the closure of an object contains the object)
  - *Idempotent* (the closure of a closure equals the closure)
  - *Monotone* (X subset of Y then closure of X is subset of closure of Y)

Itemset Closure

- Closure operation on an itemset P is the set of items common to all transactions of cover(P):

$$Clos(P) = \bigcap_{t \in cover(P)} t$$

- An itemset *P* is closed iff it is equals to its closure:

$$Clos(P) = P$$

Example (1)

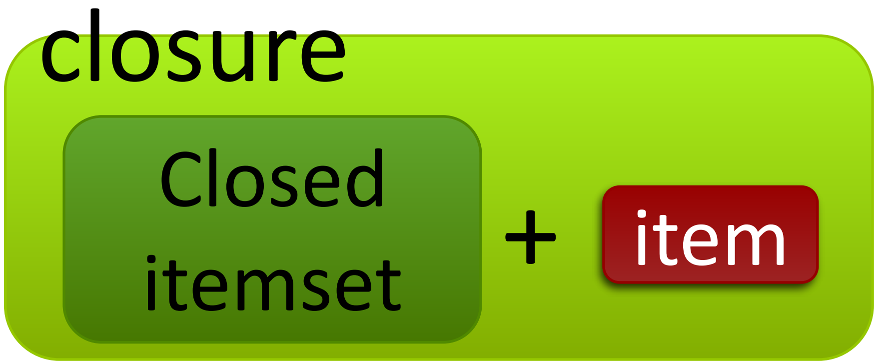
$$Clos(P) = \bigcap_{t \in cover(P)} t$$

t	Items				
t_1	A		C	D	
t_2		B	C		E
t_3	A	B	C		E
t_4		B			E
t_5	A	B	C		E
t_6		B	C		E

Q: Give the closure of: A, AB, AC, D, B

Closure Extension [Pasquier et al., 99]

- Closure extension is a rule for constructing a closed itemset from another one
  - Add an item and take its closure!



LCM: Lemma 1

- Let X and X' two itemsets:

$$X' \text{ is child of } X \Leftrightarrow \begin{cases} freq(X') > 0 & \text{condition 1} \\ X \textit{ is a prefix of } X' & \text{condition 2} \\ X' = \bigcap_{t \in cover(X)} t & \text{condition 3} \end{cases}$$

Example (2)

$$X' \text{ is child of } X \Leftrightarrow \begin{cases} freq(X') > 0 \\ X \text{ } is \text{ } a \text{ } prefix \text{ } of \text{ } X' \\ X' = \bigcap_{t \in cover(X)} t \end{cases}$$

t	Items				
t_1	A		C	D	
t_2		B	C		E
t_3	A	B	C		E
t_4		B			E
t_5	A	B	C		E
t_6		B	C		E

Q: Give the set of closed itemsets and the child relation between them

LCM: Algorithm

Algorithm LCM (X : frequent closed item set)

1. **output** X
2. **For** each $i > i(X)$ **do**
3. **If** $X[i]$ is frequent **and** $X[i] = I(\mathcal{T}(X[i]))$ **then**
   Call LCM($X[i]$)
4. **End for**

Theorem 1 *Let* $0 < \sigma < 1$ *be a minimum support. Algorithm LCM enumerates, given the root closed item set* $\perp = I(\mathcal{T}(\emptyset))$*, all frequent closed item sets in linear time in the number of frequent closed item sets in* $\mathcal{C}$. ∎

[Uno et al., 03]

LCM: Algorithm

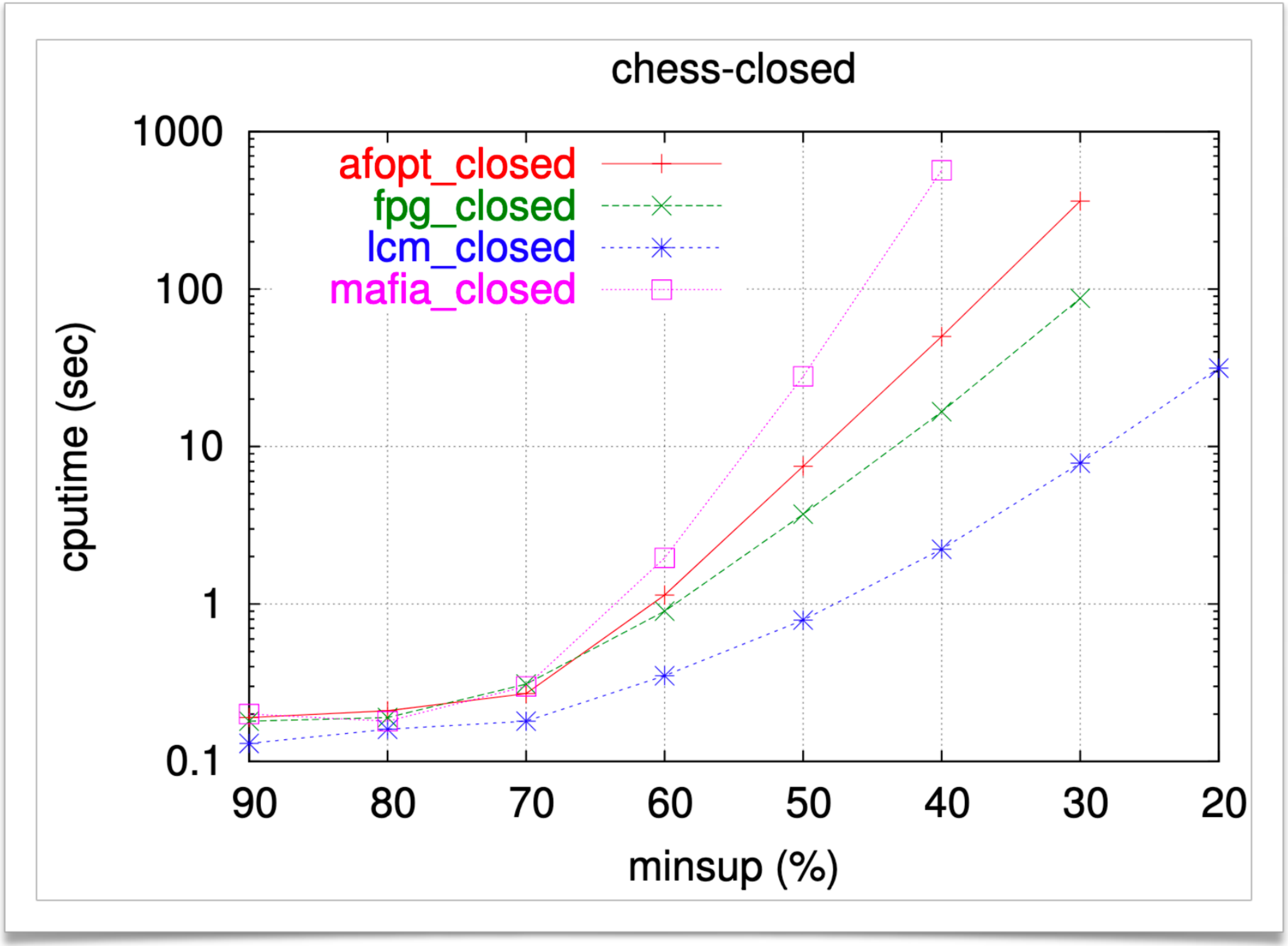
Algorithm 1: LCM

InOut : X : Closed Frequent Itemset ;
In : θ : minsup

$print(X)$
foreach $i > tail(X)$ **do**
  if $freq(X \cup \{i\}) \geq \theta$ **then**
    $Y \leftarrow \bigcap_{t \in cover(X \cup \{i\})} t$
    if $Y = child(X \cup \{i\})$ **then** $LCM(X, \theta)$

[Uno et al., 03]

Some results



Tutorials

link: http://www.lirmm.fr/~lazaar/imagina/TD3.pdf