KHAI THÁC LUẬT KẾT HỢP

PGS.TS. Võ Đình Bảy
Khoa CNTT, Trường đại học Công nghệ TP.HCM
bayvodinh@gmail.com

DẪN NHẬP

- Xét CSDL khảo sát tiện nghi sử dụng ở các hộ gia đình như sau:

Hộ	Tiện nghi sở hữu
1	Tivi, MáyVitính
2	Tủlạnh, Máylạnh
3	Tivi, Máygiặt, Máylạnh
4	Tivi, Tủlạnh, Máylạnh
5	Tivi, Máygiặt, MáyVitính
6	Tivi, Tủlạnh, Máygiặt
7	Tivi, Tủlạnh, MáyVitính
8	Tivi, Tủlạnh, Máygiặt, Máylạnh, MáyVitính

LUẬT KẾT HỢP

- **Luật kết hợp là biểu thức có dạng:**
  - Tivi → Máyvitính [50%, 57%] hay

    sử dụng:Tivi → sử dụng:Máyvitính [50%, 57%]

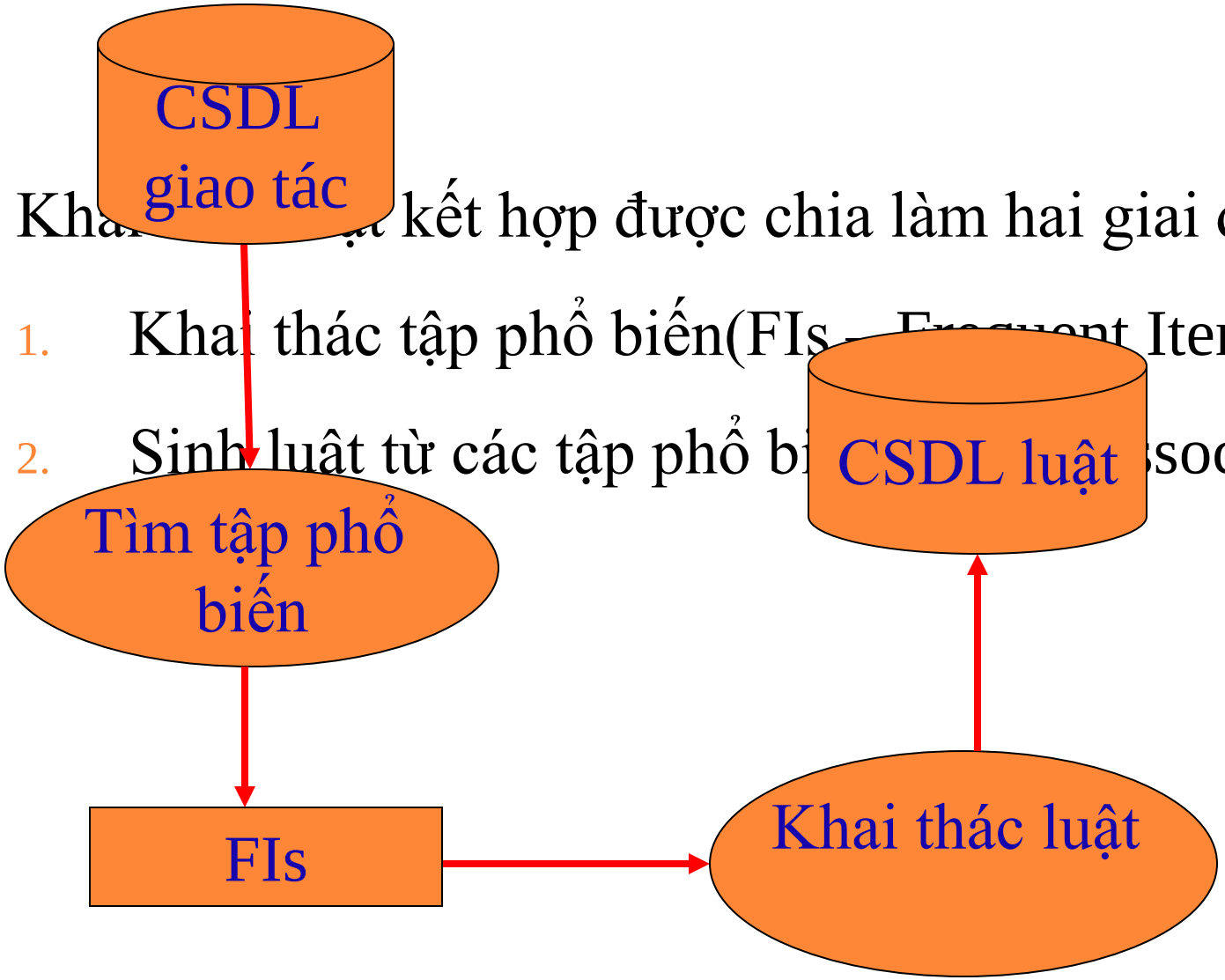
Nghĩa là: “57% hộ gia đình sử dụng Tivi **thì** cũng sử dụng Máyvitính. Tivi và Máyvitính xuất hiện chung trong 50% dòng dữ liệu."

KHAI THÁC LUẬT KẾT HỢP

Khai thác luật kết hợp được chia làm hai giai đoạn:

1. Khai thác tập phổ biến (FIs – Frequent Itemsets).
2. Sinh luật từ các tập phổ biến (ARs – Association Rules).

KHAI THÁC LUẬT KẾT HỢP



Khai thác luật kết hợp được chia làm hai giai đoạn:

1. Khai thác tập phổ biến(FIs – Frequent Itemsets).
2. Sinh luật từ các tập phổ biến(Association

1. Khai thác tập phổ biến
2. Sinh luật kết hợp

KHAI THÁC TẬP PHỔ BIẾN

- Được đề xuất bởi Agrawal năm 1993.
- Mục đích: tìm mối liên hệ giữa các mặt hàng (danh mục) được bán trong siêu thị.
- Đến nay, có nhiều phương pháp được phát triển như:
  - Phương pháp Apriori (Agrawal et al., 1994)
  - Phương pháp IT-tree (Zaki et al., 1997)
  - Phương pháp FP-tree (Han et al., 2000)
  - v.v…

MỘT SỐ PHƯƠNG PHÁP KHAI THÁC TẬP PHỔ BIẾN

1. Apriori do Agrawal et al. đề xuất.
2. Dựa vào IT-tree: Zaki et al.
3. Dựa vào FP-tree: Han et al.
4. Ngoài ra, còn có một số phương pháp được đề xuất như: LCM, DCI, PrePost, v.v…

ĐỊNH NGHĨA

1. Độ phổ biến

Cho CSDL giao dịch D và một itemset $X \subseteq \boldsymbol{I}$, Độ phổ biến của X trong D, kí hiệu $\sigma(X)$, là số giao dịch mà X xuất hiện trong D.

2. Tập phổ biến

Itemset $X \subseteq \boldsymbol{I}$ được gọi là phổ biến nếu $\sigma(X) \geq$ *minSup* (với *minSup* là giá trị do người dùng xác định).

Một số tính chất

1. Mọi tập con của tập phổ biến đều phổ biến, nghĩa là $\forall X \subseteq Y$, nếu $\sigma(Y) \geq minSup$ thì $\sigma(X) \geq minSup$
2. Mọi tập cha của tập không phổ biến đều không phổ biến, nghĩa là $\forall Y \supseteq X$, nếu $\sigma(X) < minSup$ thì $\sigma(Y) < minSup$

Cả hai tính chất trên dễ dàng được chứng minh (xem như bài tập).

THUẬT TOÁN APRIORI

- **Đầu vào**:CSDL giao dịch D và ngưỡng phổ biến $minSup$
- **Đầu ra**: **FIs** chứa tất cả các tập phổ biến của D
- **Mã giả**:

Gọi C_k: Tập các ứng viên có kích thước k
L_k: Các tập phổ biến có kích thước k

$\boldsymbol{L_1} = \{ i \in I: \sigma(i) \geq minSup\}$
for ($k = 2$; L_{k-1} !=∅; k++) do
  $\boldsymbol{C_k}$ = {các ứng viên được tạo từ L_{k-1}}
  for each $t \in D$ do
    for each $c \in C_k$ do
      if $c \subseteq t$ *then* c.count++
  $\boldsymbol{L_k} = \{c \in C_k \mid c.count \geq minSup\}$
FIs = $\cup_k L_k$;

CÁCH TẠO ỨNG VIÊN CỦA APRIORI

- **Tính chất Apriori:**

  Mọi tập con của tập phổ biến cũng phổ biến
- Giả sử ta có $L_3 = \{abc, abd, acd, ace, bcd\}$
- Xét việc kết để tao ra các ứng viên C_4: $L_3 * L_3$
  - *abcd* được tạo từ *abc* và *abd*
  - *acde* được tạo từ *acd* và *ace*
- **Rút gọn:**
  - *acde* bị loại vì *ade* không có trong L_3

$\Rightarrow C_4 = \{abcd\}$

Ví dụ minh họa

Bảng 1: Xét CSDL mẫu

Mã giao dịch	Nội dung giao dịch
1	*A, C, T, W*
2	*C, D, W*
3	*A, C, T, W*
4	*A, C, D, W*
5	*A, C, D, T, W*
6	*C, D, T*

$\sigma(A) = 4$

$\sigma(C) = 6$

$\sigma(D) = 4$

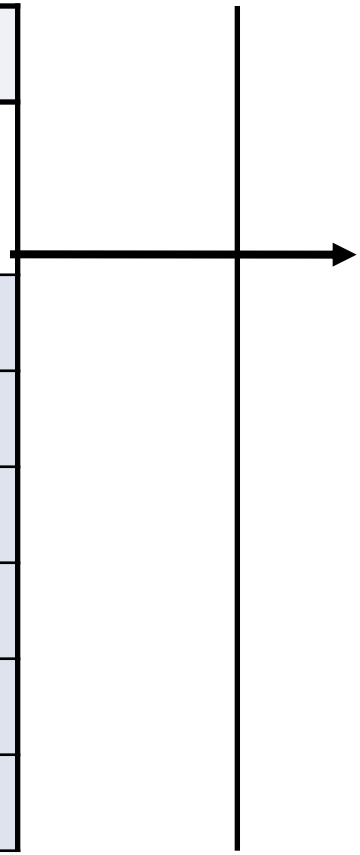
$\sigma(T) = 4$

$\sigma(W) = 5$

Với *minSup* = 3 (hay 50%), ta có

Ví dụ (TT)

Database (D)	
TID	Nội dung
1	*A, C, T, W*
2	*C, D, W*
3	*A, C, T, W*
4	*A, C, D, W*
5	*A, C, D, T, W*
6	*C, D, T*



L_1	
Danh mục	Độ phổ biến
A	**4**
C	**6**
D	**4**
T	**4**
W	**5**

Ví dụ (TT)

C_2	
Danh mục	Độ phổ biến
AC	4
AD	2
AT	3
AW	4
CD	4
CT	4
CW	5
DT	2
DW	3
TW	3

L_2	
Danh mục	Độ phổ biến
AC	4
AT	3
AW	4
CD	4
CT	4
CW	5
DW	3
TW	3

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Ví dụ (TT)

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

C_3	
Danh mục	Độ phổ biến
ACT	3
ACW	4
ATW	3
CDW	3
CTW	3

L_3	
Danh mục	Độ phổ biến
ACT	3
ACW	4
ATW	3
CDW	3
CTW	3

Lưu ý: CDT không có trong C_3 vì DT không có trong L_2!

Ví dụ (TT)

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

C_4

Danh mục	Độ phổ biến
ACTW	3

L_4

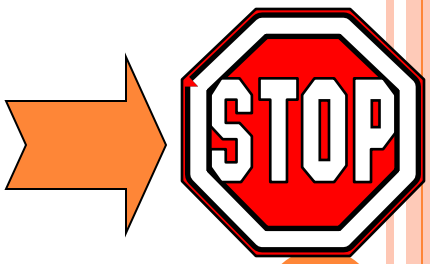
Danh mục	Độ phổ biến
ACTW	3

C_5

Danh mục	Độ phổ biến

L_5

Danh mục	Độ phổ biến



PHƯƠNG PHÁP DỰA TRÊN FP-TREE

- Quét DB lần thứ nhất để tìm tất cả các item đơn phổ biến (single item pattern)
- Sắp xếp các item theo thứ tự giảm của độ phổ biến ⇒ f-list
- Quét DB lần 2, Xây dựng FP-tree

FP- TREE – XÂY DỰNG CÂY

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Item	A	C	D	T	W
σ	4	6	4	4	5

Sắp xếp theo σ

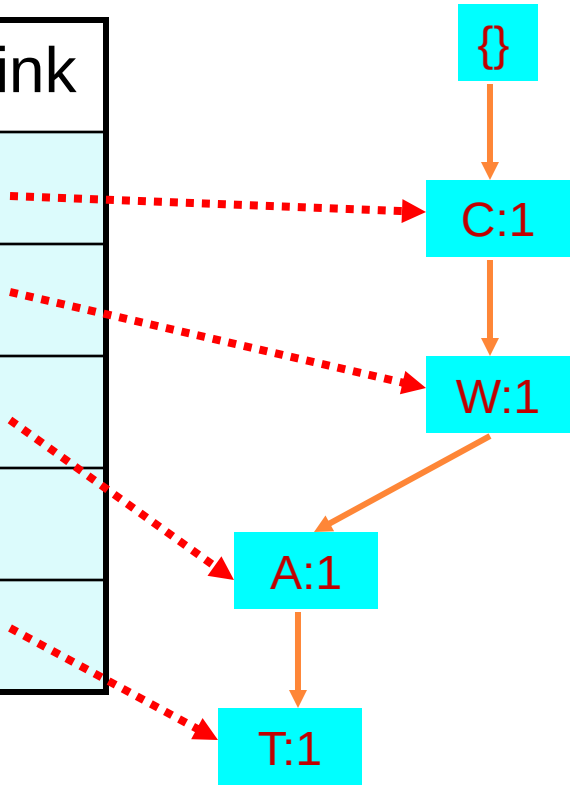
Item	C	W	A	D	T
σ	6	5	4	4	4

FP- TREE – XÂY DỰNG CÂY

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

C, W, A, T

Item	σ	Link
C	6	
W	5	
A	4	
D	4	
T	4	

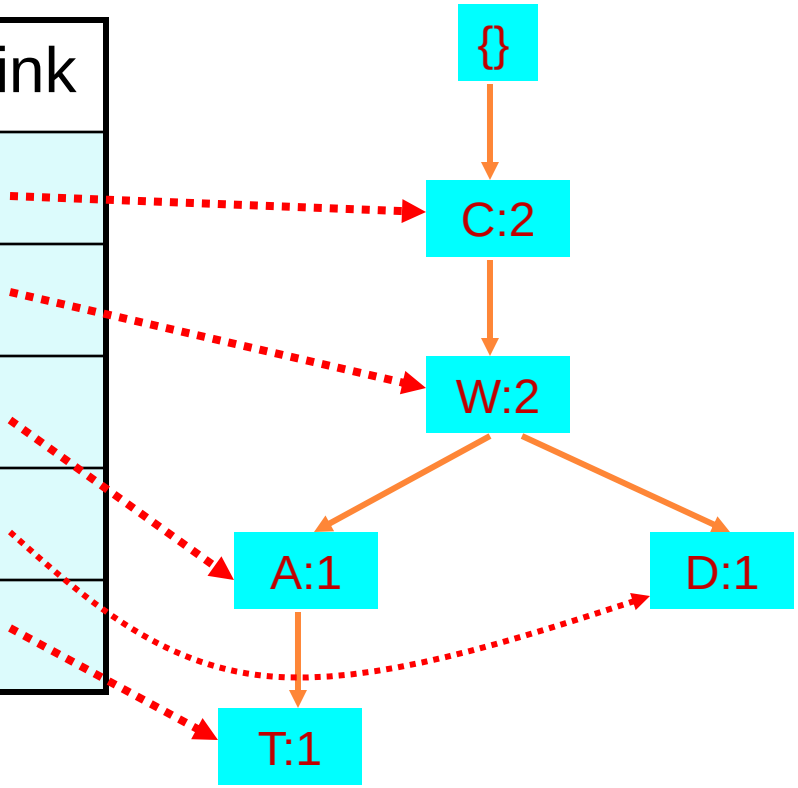


FP-tree với giao dịch 1

FP- TREE – XÂY DỰNG CÂY

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Item	σ	Link
C	6	
W	5	
A	4	
D	4	
T	4	



C, W, A, T

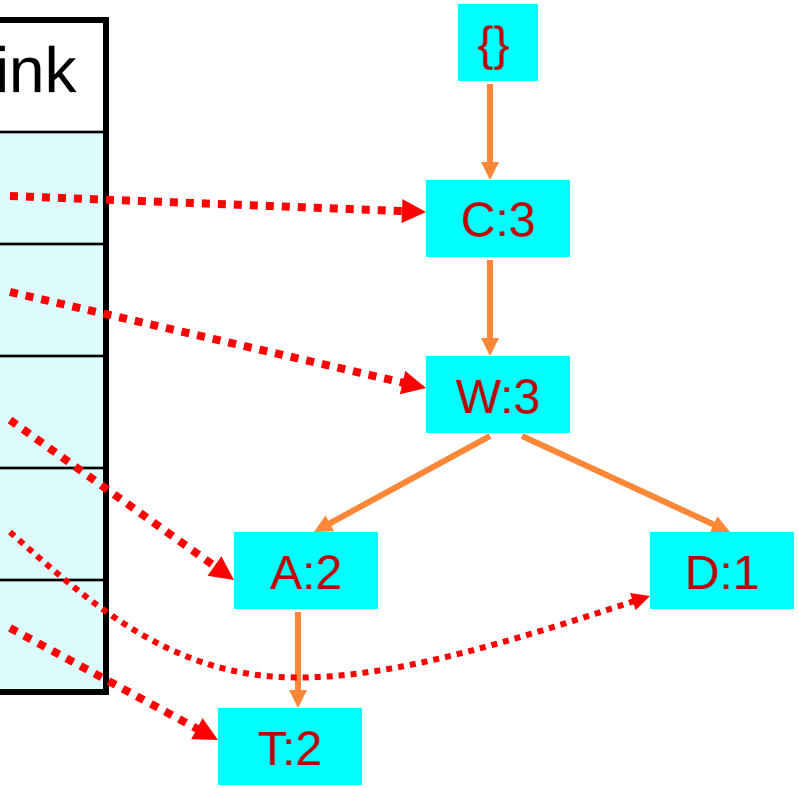
C, W, D

FP-tree với giao dịch 1 và 2

FP- TREE – XÂY DỰNG CÂY

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Item	σ	Link
C	6	
W	5	
A	4	
D	4	
T	4	



C, W, A, T

C, W, D

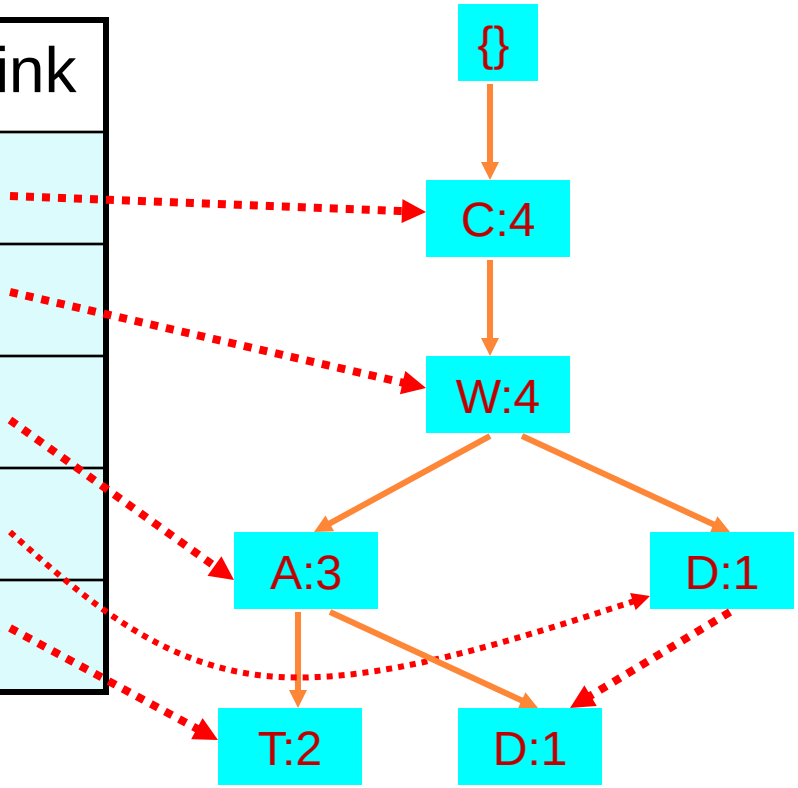
C, W, A, T

FP-tree trên 3 giao dịch đầu

FP- TREE – XÂY DỰNG CÂY

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Item	σ	Link
C	6	
W	5	
A	4	
D	4	
T	4	



C, W, A, T

C, W, D

C, W, A, T

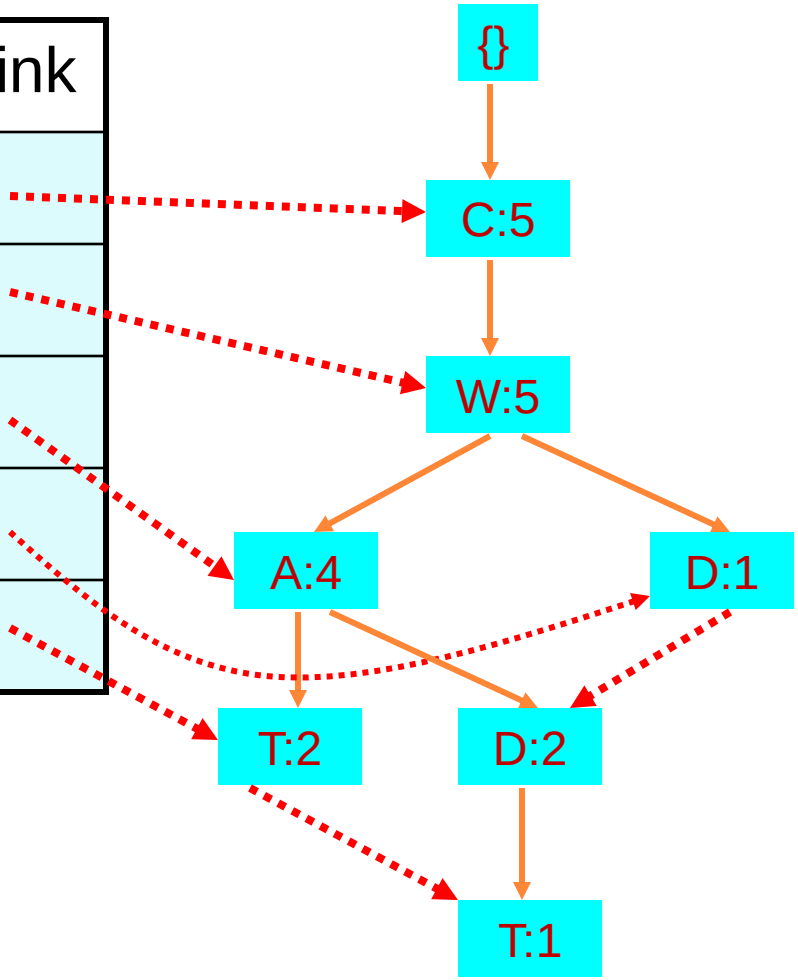
C, W, A, D

FP-tree trên 4 giao dịch đầu

FP- TREE – XÂY DỰNG CÂY

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Item	σ	Link
C	6	
W	5	
A	4	
D	4	
T	4	



C, W, A, T

C, W, D

C, W, A, T

C, W, A, D

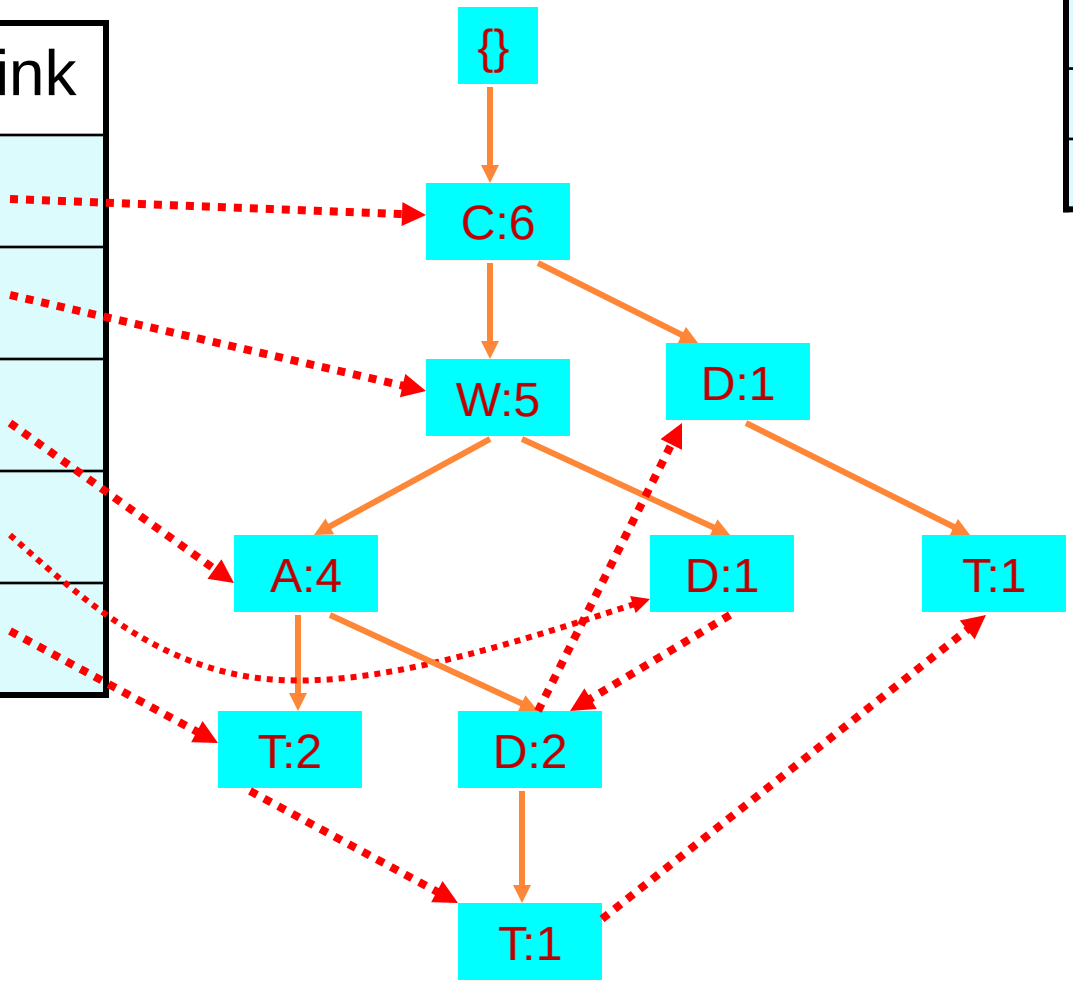
C, W, A, D, T

FP-tree trên 5 giao dịch đầu

FP- TREE – XÂY DỰNG CÂY

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Item	σ	Link
C	6	
W	5	
A	4	
D	4	
T	4	



C, W, A, T

C, W, D

C, W, A, T

C, W, A, D

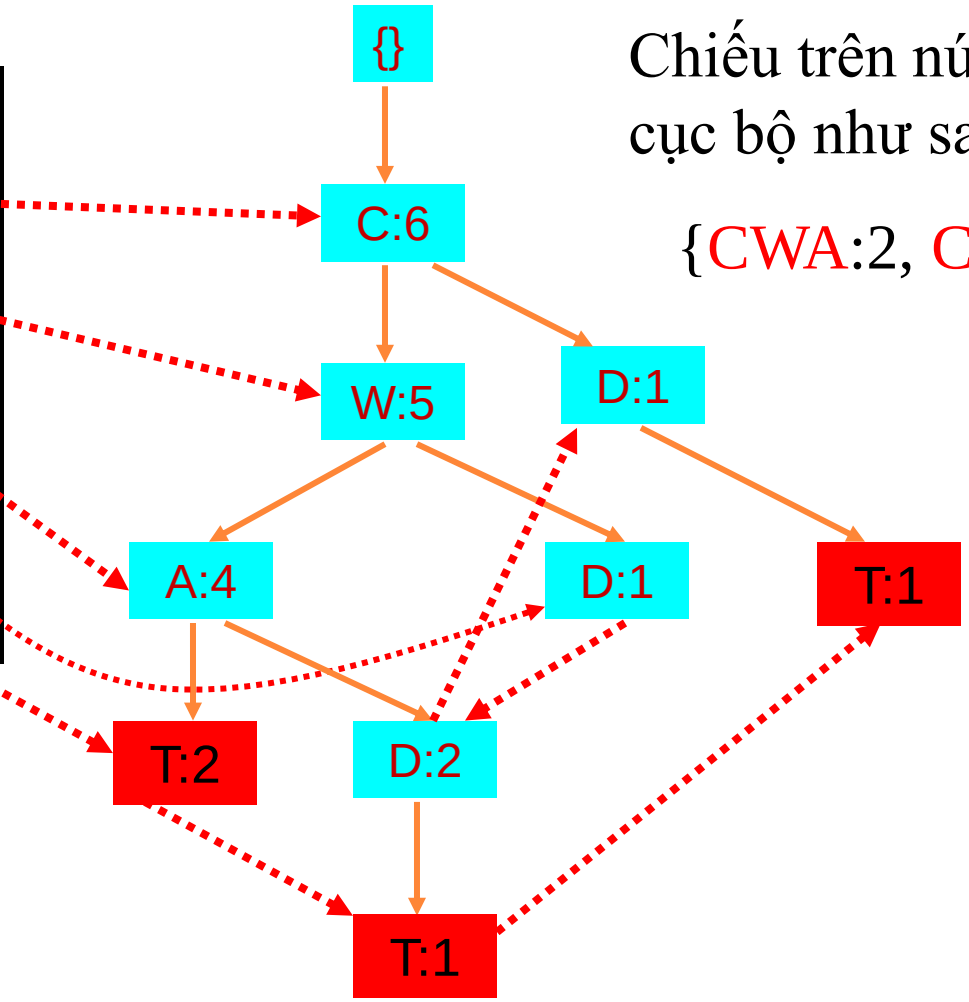
C, W, A, D, T

C, D, T

FP-tree trên cả 6 giao dịch với *minSup* = 3

CHIẾU TRÊN FP-TREE – TT FP-GROWTH

Item	σ	Link
C	6	
W	5	
A	4	
D	4	
T	4	



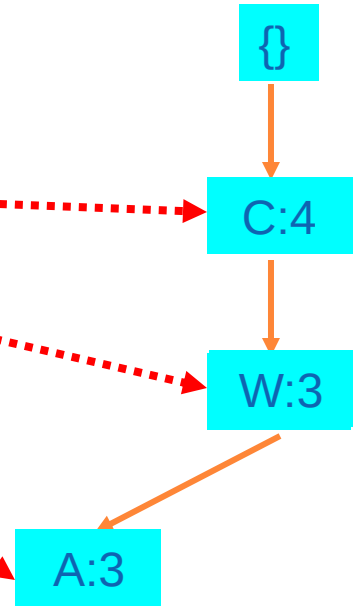
Chiếu trên nút T: ta có CSDL cục bộ như sau:

{CWA:2, CWAD:1, CD:1}

CHIẾU TRÊN T:4

{CWA:2, CWAD:1, CD:1} ⇒Cây cục bộ cho CSDL chiếu trên T như sau:

Item	σ	Link
C	4	
W	3	
A	3	



CWA:2

CWAD:1⇒ CWA:1

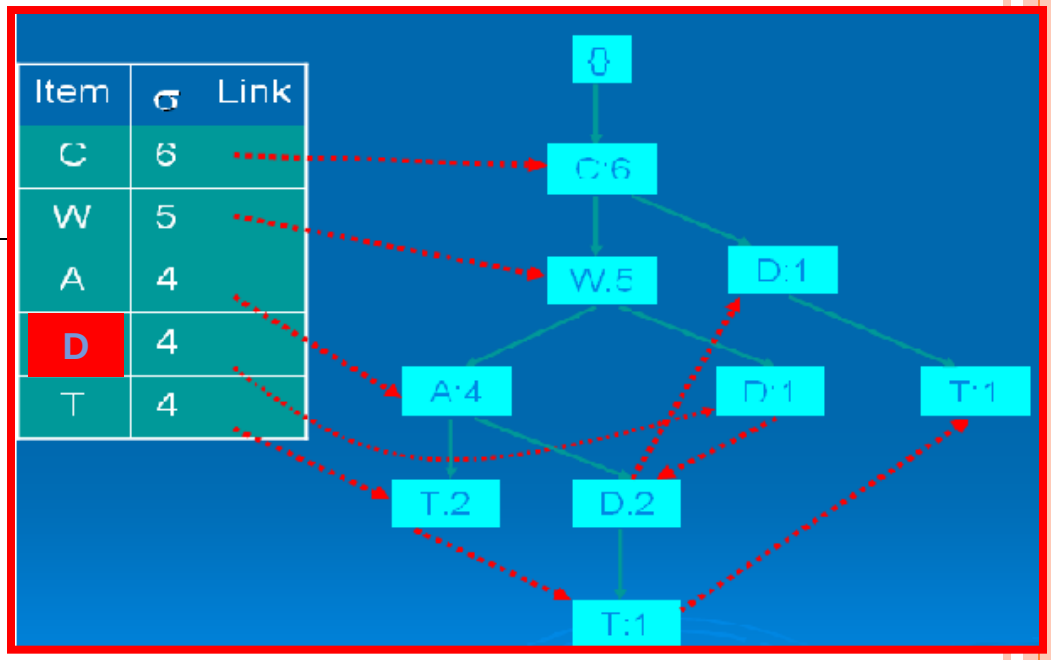
CD:1 ⇒ C:1

Đây là đường đi đơn nên việc tìm các tập phổ biến chỉ đơn giản là tìm các tập con của tập {C, W, A}. Ta có các tập con:

{∅,A:3,W:3,C:4,AW:3,AC:3,WC:3, AWC:3}

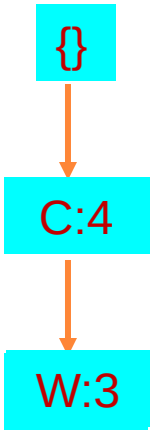
Vì vậy: chiếu trên T sinh ra các tập phổ biến là: {T:4, TA:3, TW:3, TC:4, TAW:3, TAC:3, TWC:3, TAWC:3}.

CHIẾU TRÊN D:4



{CWA:2, CW:1, C:1} ⇒Cây cục bộ như sau:

Item	σ	Link
C	4	
W	3	

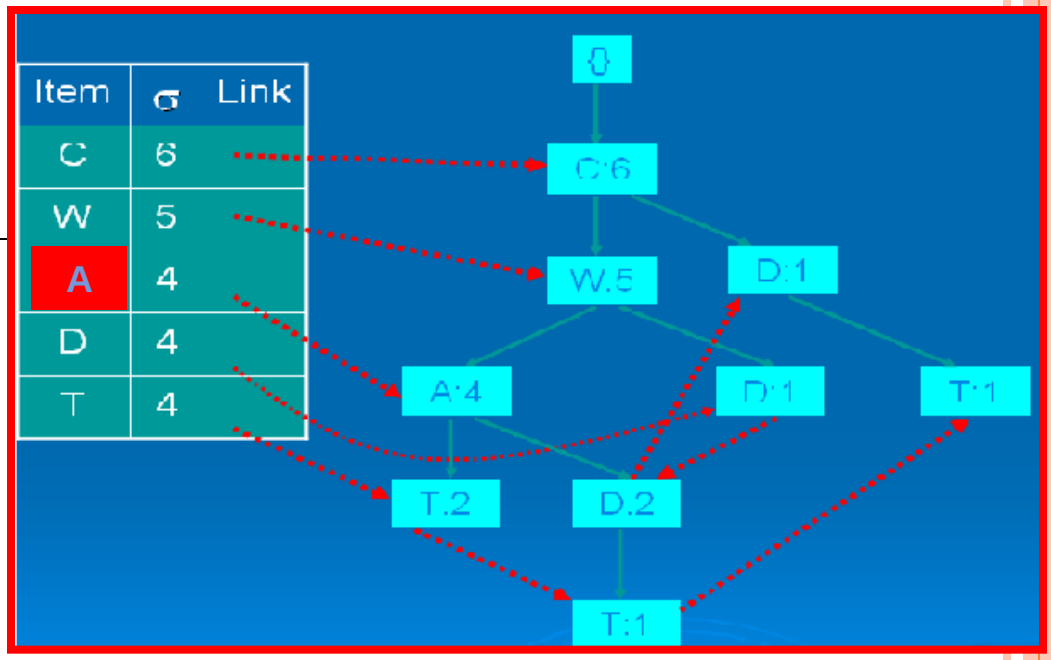


Đường đi đơn ⇒ Các tập con:

{∅, W:3,C:4, WC:3}

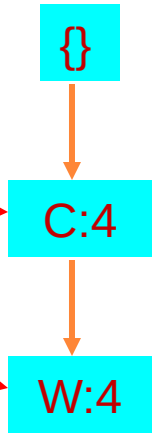
Chiếu trên D sinh ra các tập phổ biến là:{D:4, DW:3, DC:4, DWC:3}.

CHIẾU TRÊN A:4



{CW:4} ⇒Cây cục bộ như sau:

Item	σ	Link
C	4	
W	4	

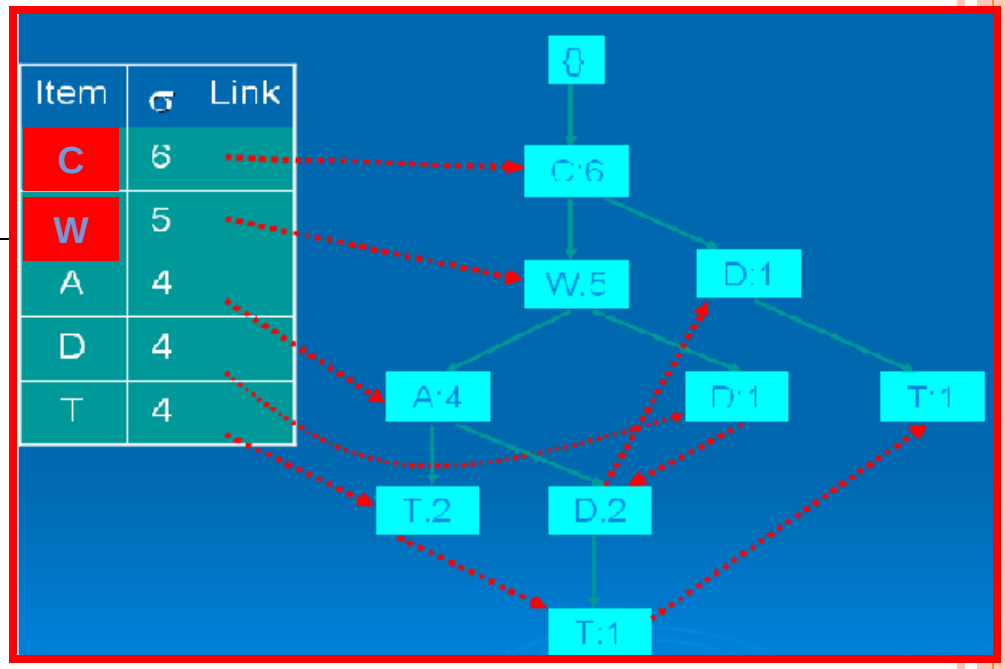


Đường đi đơn ⇒ Các tập con:

{∅, W:4,C:4, WC:4}

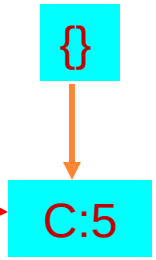
Chiếu trên A sinh ra các tập phổ biến là:{A:4, AW:4, AC:4, AWC:4}.

CHIẾU TRÊN W,C



W:5 {C:5} ⇒Cây cục bộ như sau:

Item	σ	Link
C	5	



Đường đi đơn ⇒ Các tập con:

{∅, C:5}

Chiếu trên W sinh ra các tập phổ biến là:{W:5, WC:5}.

Cuối cùng, chiếu trên C: 6 ta được {∅} ⇒ tập phổ biến:{C:6}.

FP- TREE – NHẬN XÉT

FP-tree duyệt CSDL 2 lần, sau đó dùng phép chiếu để tạo ra CSDL cục bộ của từng item đơn, sau đó tạo cây FP cục bộ và khai thác trên cây cục bộ một cách đệ qui.

Sử dụng phương pháp chia để trị để khai thác tập phổ biến.

Là phương pháp không sinh ứng viên.

Thường rất hiệu quả trên các CSDL có mật độ trùng lặp dữ liệu cao.

PHƯƠNG PHÁP IT- TREE

Kết nối Galois:

Cho quan hệ hai ngôi $\delta \subseteq I \times T$ chứa CSDL cần khai thác. Với: $X \subseteq I$ và $Y \subseteq T$.

Định nghĩa hai ánh xạ giữa $P(I)$ (Tập tất cả các tập con $\neq \emptyset$ của I) và $P(T)$ như sau:

- $t: P(I) \rightarrow P(T)$, $t(X) = \{y \in T \mid \forall x \in X, x\ \delta\ y\}$
- $i: P(T) \rightarrow P(I)$, $i(Y) = \{x \in I \mid \forall y \in Y, x\ \delta\ y\}$

https://vi.wikipedia.org/wiki/%C3%89variste_Galois

PHƯƠNG PHÁP IT- TREE (TT)

Cấu trúc IT-tree và các lớp tương đương:

Cho $X \subseteq I$, ta định nghĩa hàm $p(X,k)=X[1:k]$ gồm k phần tử đầu của X và quan hệ tương đương dựa vào tiền tố như sau:

$$\forall X,Y \subseteq I, X \equiv_{\theta_k} Y \Leftrightarrow p(X,k) = p(Y,k)$$

Mỗi nút trên IT-tree gồm 2 thành phần Itemset-Tidset: $X \times t(X)$ được gọi là IT-pair, thực chất là một lớp tiền tố. Các nút con của X thuộc về lớp tương đương của X vì chúng chia sẻ chung tiền tố X ($t(X)$ là tập các giao dịch có chứa X)

NHẬN XÉT VỀ IT- TREE

1. $\sigma(X) = |t(X)|$
2. Chỉ cần kết hợp các phần tử trên cùng một mức của lớp tương đương là đủ để sinh ra các tập phổ biến.

THUẬT TOÁN ECLAT

```
ECLAT()
      [∅] = {i∈I| σ(i)≥ minSup}
      ENUMERATE_FREQUENT([∅])
ENUMERATE_FREQUENT([P])
   for all l_i ∈ [P] do
      [P_i] = ∅
      for all l_j ∈ [P] with j > i do
         X = l_i ∪ l_j
         T = t(l_i) ∩ t(l_j)
         if |T| ≥ minSup then
            [P_i] = [P_i] ∪ {X×T}
      ENUMERATE_FREQUENT([P_i])
```

Trong đó $t(X) = \{y \in T \mid X$ xuất hiện trong giao dịch y$\}$ được gọi là Tidset của X.

VÍ DỤ MINH HỌA

Xét CSDL mẫu ⇒ định dạng dữ liệu dọc

Mã giao dịch	Nội dung giao dịch
1	*A, C, T, W*
2	*C, D, W*
3	*A, C, T, W*
4	*A, C, D, W*
5	*A, C, D, T, W*
6	*C, D, T*

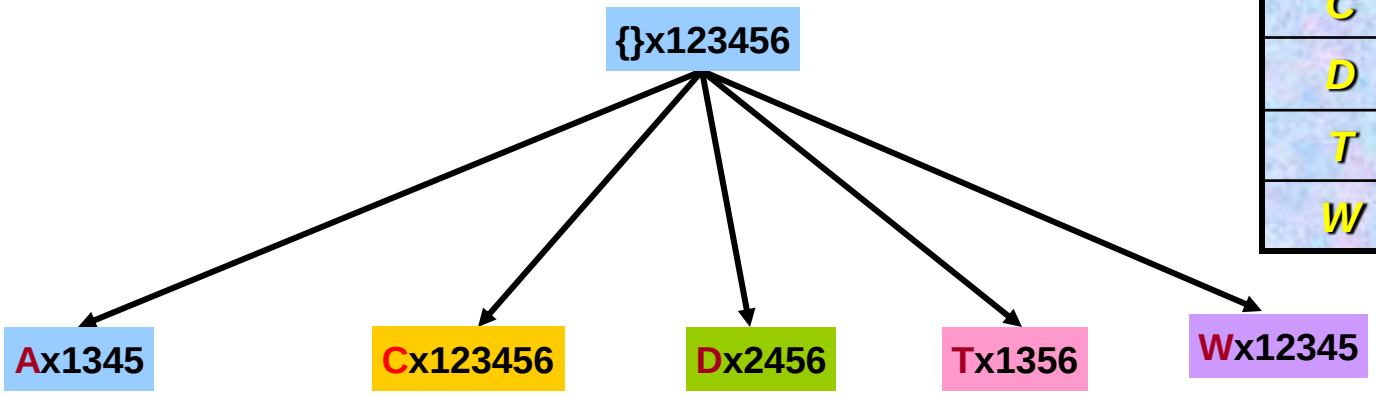
⇒

Mã danh mục	Các giao dịch chứa danh mục
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5

$t(A) = 1345;\ t(AD) = t(A) \cap t(D) = 1345 \cap 2456 = 45$

IT-tree với *minSup*=50%

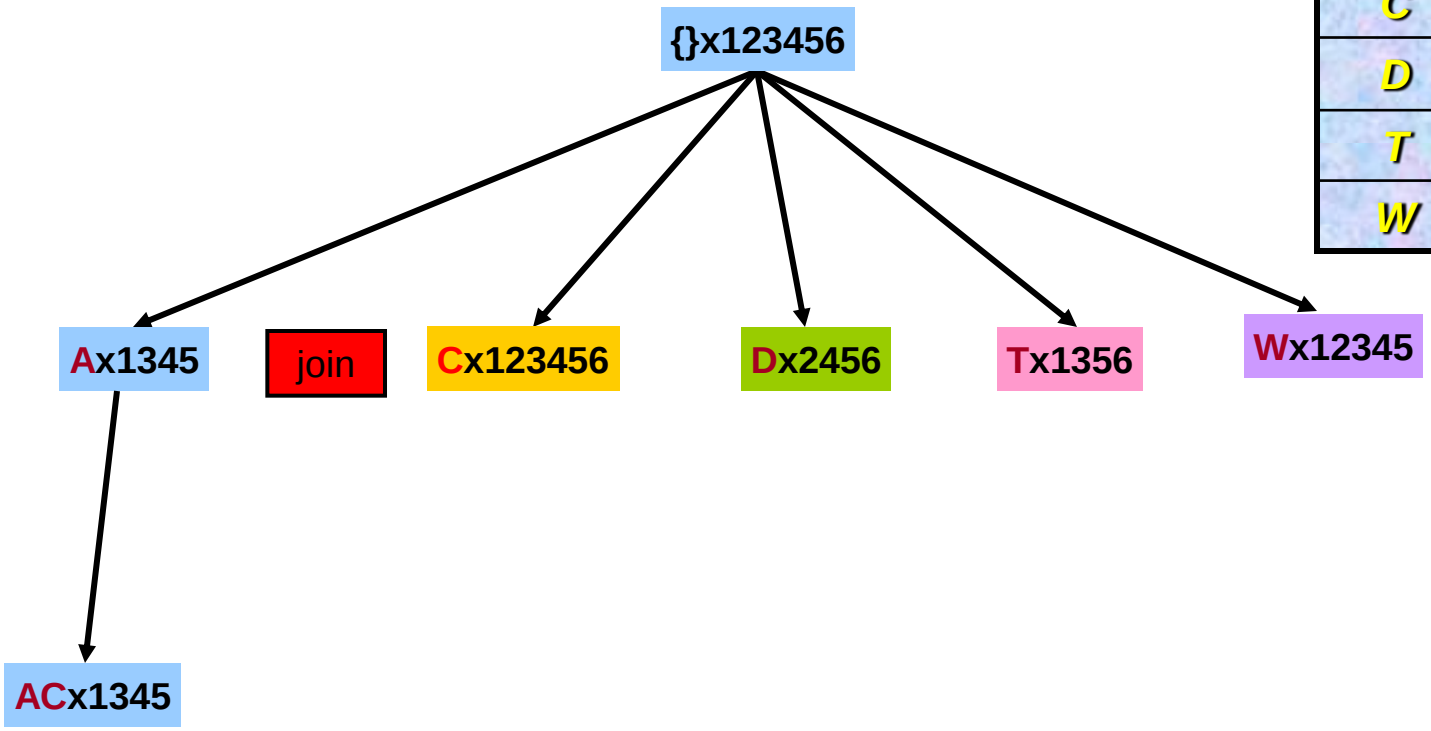
Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



Mức 1 của IT-tree với *minSup* = 3

IT-tree với *minSup*=50%

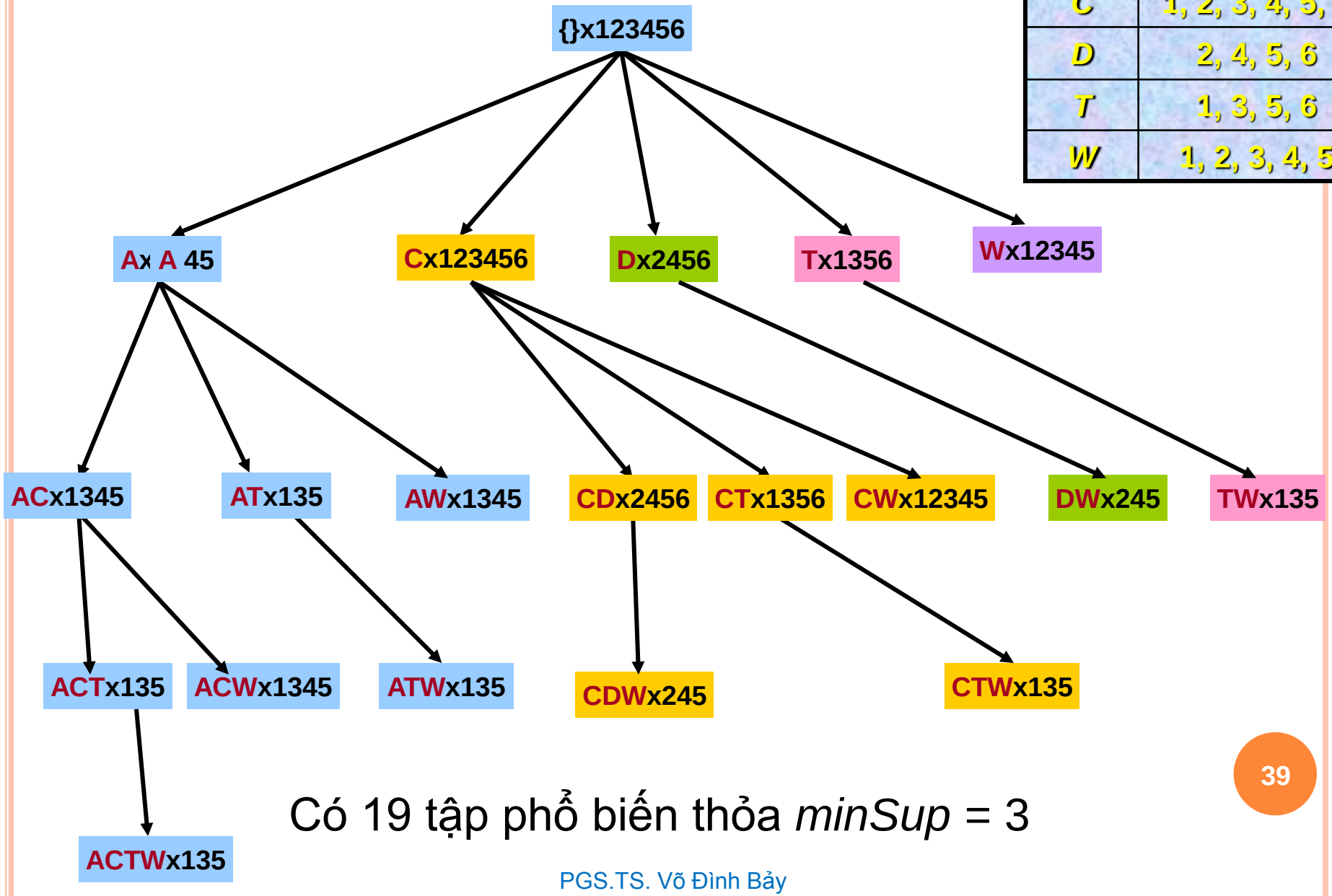
Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



Mức 1 của IT-tree với *minSup* = 3

IT-tree với *minSup*=50%

Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



Có 19 tập phổ biến thỏa *minSup* = 3

NHẬN XÉT

- Thuật toán dựa vào phần giao giữa các Tidset để tính nhanh độ phổ biến nên chỉ quét CSDL 1 lần.
- Có thể sử dụng Diffset để tính nhanh độ phổ biến nhằm làm giảm không gian lưu trữ Tidset.
- Do thuật toán không sinh ứng viên nên hiệu quả khai thác thường cao hơn so với các họ thuật toán sinh ứng viên.
- Khi số tập phổ biến lớn, thời gian khai thác luật lớn $\Rightarrow$ Cần phương pháp khai thác hiệu quả hơn

DIFFSET ĐỂ TÍNH NHANH ĐỘ PHỔ BIẾN

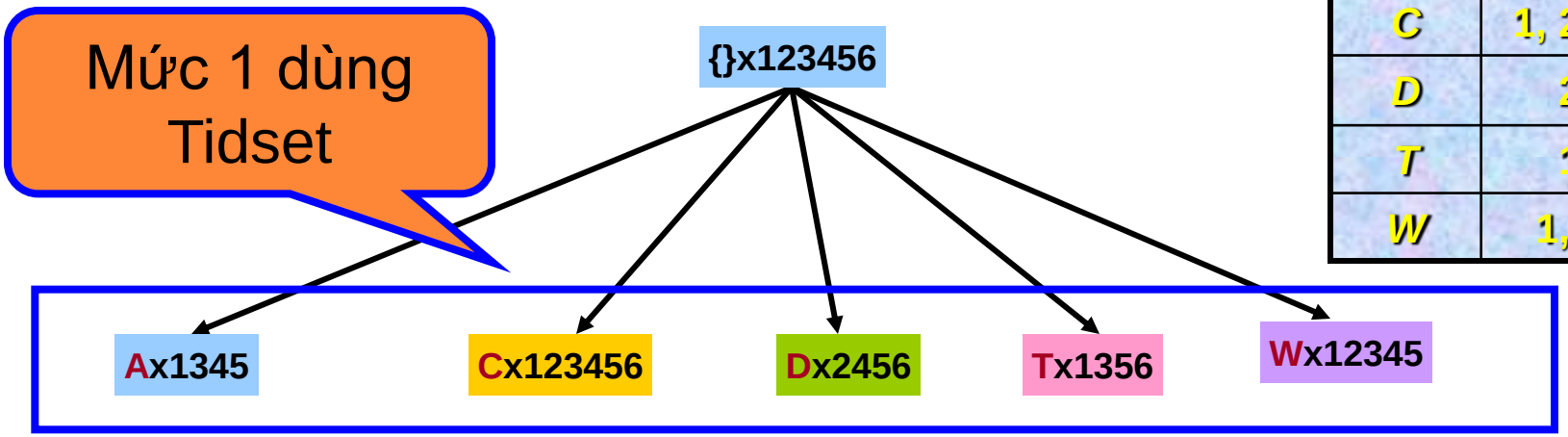
- Diffset của A so với B, kí hiệu $d(AB)$ được định nghĩa như sau:

  $d(AB) = t(A) \setminus t(B)$ trong đó $A, B \in I$

- Gọi PA và PB là 2 nút thuộc lớp tương đương P, ta có: $d(PXY) = d(PY) \setminus d(PX)$ (1)
- $\sigma(PXY) = \sigma(PX) - |d(PXY)|$ (2)
- Diffset thường khá nhỏ so với Tidset (3)
- Từ (1), (2) và (3), chúng ta có thể sử dụng Diffset để thay thế Tidset.

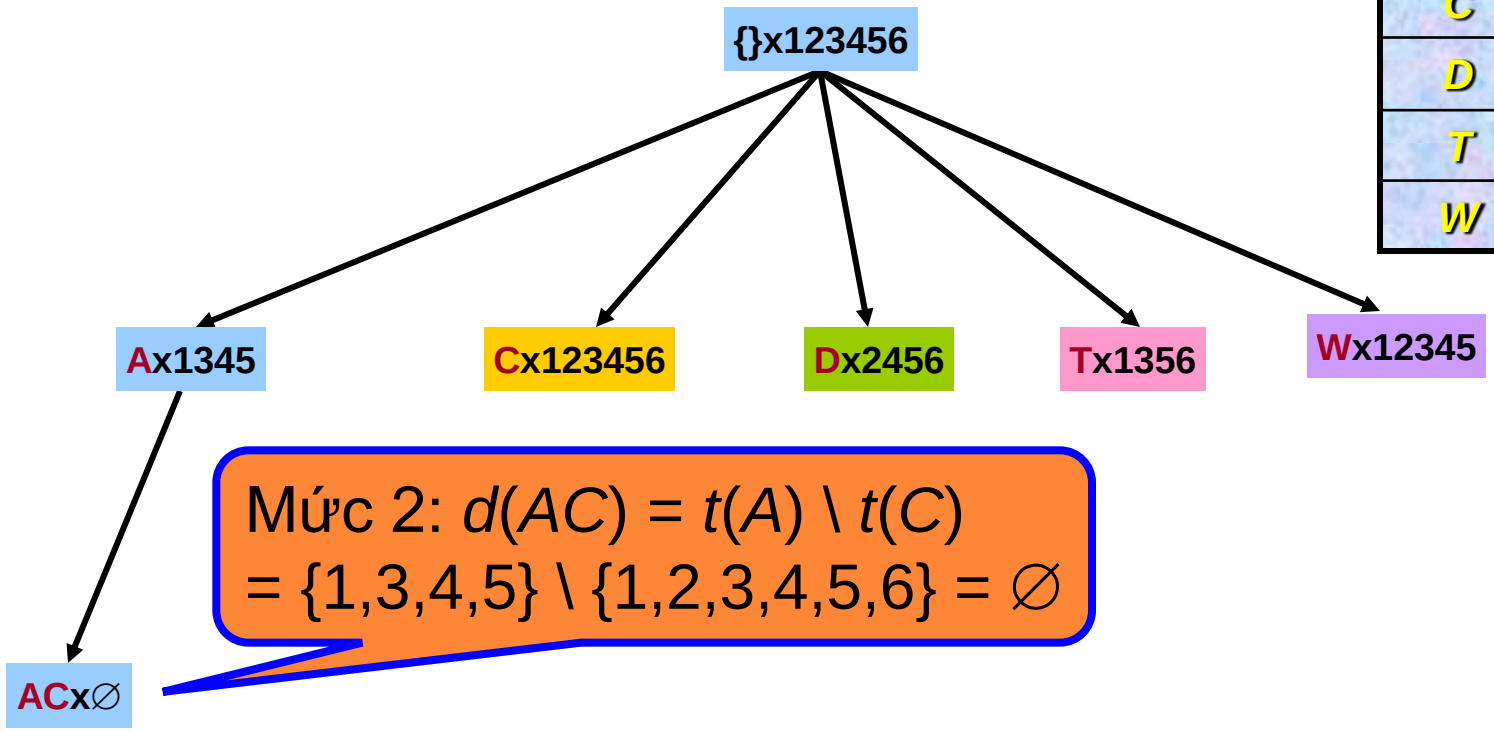
Diffset với *minSup* = 3

Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



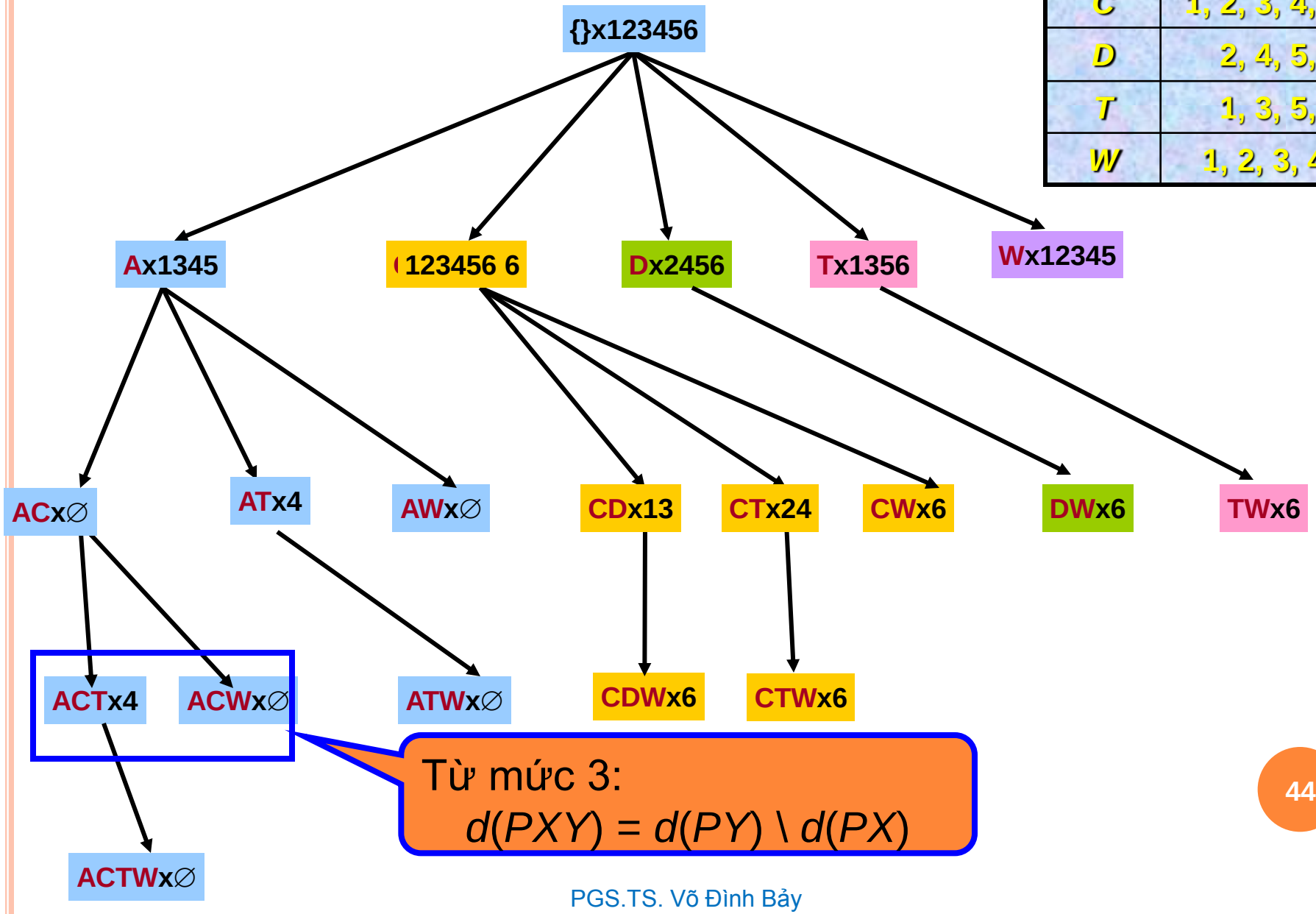
Diffset với *minSup* = 3

Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



Diffset với *minSup* = 3

Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



NHẬN XÉT

- Kích thước Diffset thường khá nhỏ so với Tidset nên tiết kiệm được không gian bộ nhớ và thời gian tính phần "khác nhau".

So sánh độ dài trung bình giữa Tidset và Diffset trên các CSDL chuẩn

CSDL	*MinSup* (%)	Độ dài trung bình Diffset	Độ dài trung bình Tidset	Tỉ lệ Tidset/Diffset
chess	50	26	1820	70
connect	90	143	62204	434.99
mushroom	5	60	622	10.37
pumsb_star	35	301	18977	63.04
pumsb	90	330	45036	136.47
T10I4D100K	0.1	31	230	7.42
T40I10D100K	0.5	96	755	7.86

Tỉ lệ = 1820/26

TÌM TẬP PHỔ BIẾN ĐÓNG
(FREQUENT CLOSED ITEMSETS - FCI)

- ***Toán tử đóng:***

  Cho $X \subseteq I$. c_{it}: $P(I) \rightarrow P(I)$: $c_{it}(X) = i(t(X))$. Ánh xạ c_{it} được gọi là toán tử đóng.

  Ví dụ: $c_{it}(AW) = i(t(AW)) = i(1345) = ACW$

- ***Tập đóng:***

  Cho $X \subseteq I$. X gọi là tập đóng $\Leftrightarrow c_{it}(X) = X$.

TÌM TẬP PHỔ BIẾN ĐÓNG
(FREQUENT CLOSED ITEMSETS - FCI)

$t(AW) = t(A) \cap t(W) = 1345$

Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5

Tid	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

- ***Tập đóng:***

  Cho $X \subseteq I$. X gọi là tập đóng $\Leftrightarrow c_{it}(X) = X$.

  Ví dụ: xét CSDL ở bảng 1 ta có

  - Do $c_{it}(AW) = i(t(AW)) = i(1345) = ACW$

    $\Rightarrow AW$ không phải là tập *đóng*.
  - Do $c_{it}(ACW) = i(t(ACW)) = i(1345) = ACW$

    $\Rightarrow ACW$ là tập *đóng*.

CÁC TÍNH CHẤT CỦA IT-PAIR

Định lý 1:

Cho $X_i \times t(X_i)$ và $X_j \times t(X_j)$ là hai phần tử tùy ý của lớp tương đương $[P]$. Ta có 4 tính chất sau (c là c_{it}):

1. Nếu $t(X_i) = t(X_j)$ thì $c(X_i) = c(X_j) = c(X_i \cup X_j)$
2. Nếu $t(X_i) \subset t(X_j)$ thì $c(X_i) \neq c(X_j)$
   nhưng $c(X_i) = c(X_i \cup X_j)$
3. Nếu $t(X_i) \supset t(X_j)$ thì $c(X_i) \neq c(X_j)$
   nhưng $c(X_j) = c(X_i \cup X_j)$
4. Ngược lại của 1, 2 và 3: $c(X_i) \neq c(X_j) \neq c(X_i \cup X_j)$

NHẬN XÉT VỀ IT-PAIR

1. Tính chất 1 nói rằng, nếu phần giao của hai Tidset bằng nhau thì $|t(X_i)| = |t(X_j)| = |t(X_i \cup X_j)|$ mà $X_i \subset X_i \cup X_j$ và $X_j \subset X_i \cup X_j$ nên X_i, X_j không là tập đóng.
2. Theo tính chất 2, ta có $c(X_i) = c(X_i \cup X_j) \Rightarrow X_i$ không là tập đóng. Bên cạnh đó, do $t(X_i) \neq t(X_j)$ nên X_i và X_j thuộc về 2 tập đóng khác nhau.
3. Tương tự tính chất 2.
4. Theo tính chất 4, X_i, X_j và $X_i \cup X_j$ sẽ thuộc về 3 tập đóng khác nhau.

THUẬT TOÁN TÌM TẬP PHỔ BIẾN ĐÓNG(CHARM)

```
CHARM(D,minSup)
    [∅]={l_i×t(l_i):l_i∈I ∧Sup(l_i)≥minSup}
    CHARM-EXTEND([∅], C = ∅)
    return C

CHARM-EXTEND([P], C)
  for each l_i×t(l_i) in [P] do
    P_i = P_i ∪ l_j and [P_i] = ∅
    for each l_j×t(l_j) with j > i do
      Y =t(l_i) ∩t(l_j)
      CHARM-PROPERTY(X×Y,l_i,l_j,[P_i],[P])
    SUBSUMPTION-CHECK(C, P_i)
    CHARM-EXTEND([P_i], C)
    delete ([P_i]
```

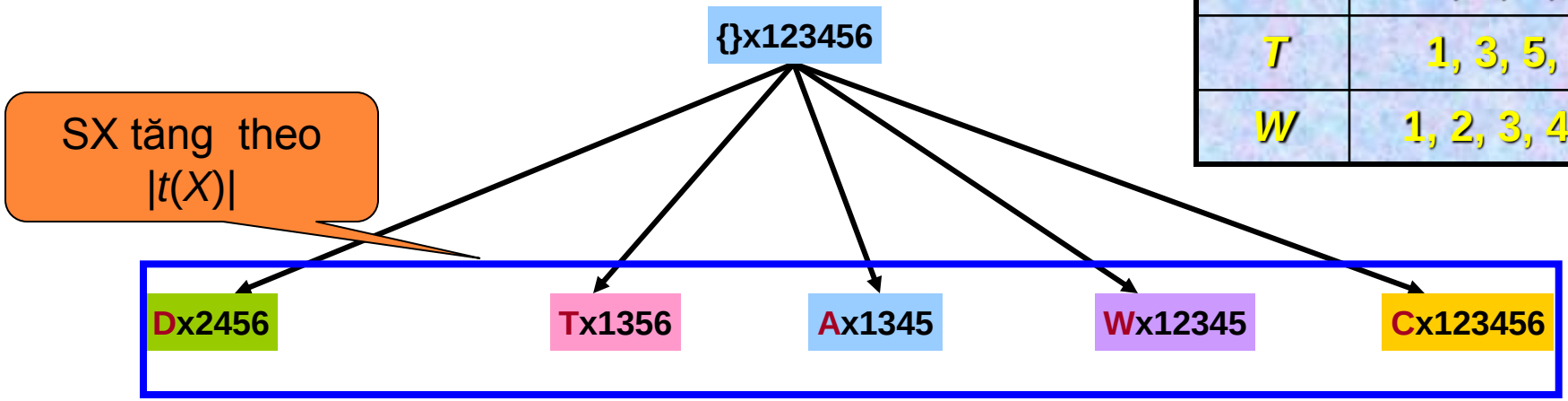
```
CHARM-PROPERTY(X ×Y,l_i,l_j,[P_i],[P])
    if Sup(X) ≥ minSup  then
      if t(l_i)=t(l_j) then
            Remove l_j from [P]
            P_i = P_i ∪ l_j
      elseif  t(l_i) ⊂ t(l_j) then
            P_i = P_i ∪ l_j
      elseif  t(l_i) ⊃ t(l_j) then
            Remove l_j from [P]
            Add X ×Y  to [P_i]
      else
            Add X ×Y to [P_i]
```

Sử dụng bảng băm để kiểm tra tập P có phải là tập đóng hay không?

```
SUBSUMPTION-CHECK(C, P)
if P⊄Y, ∀Y ∈HASHTABLE[|t(P)|] then
        C = C  ∪ P
```

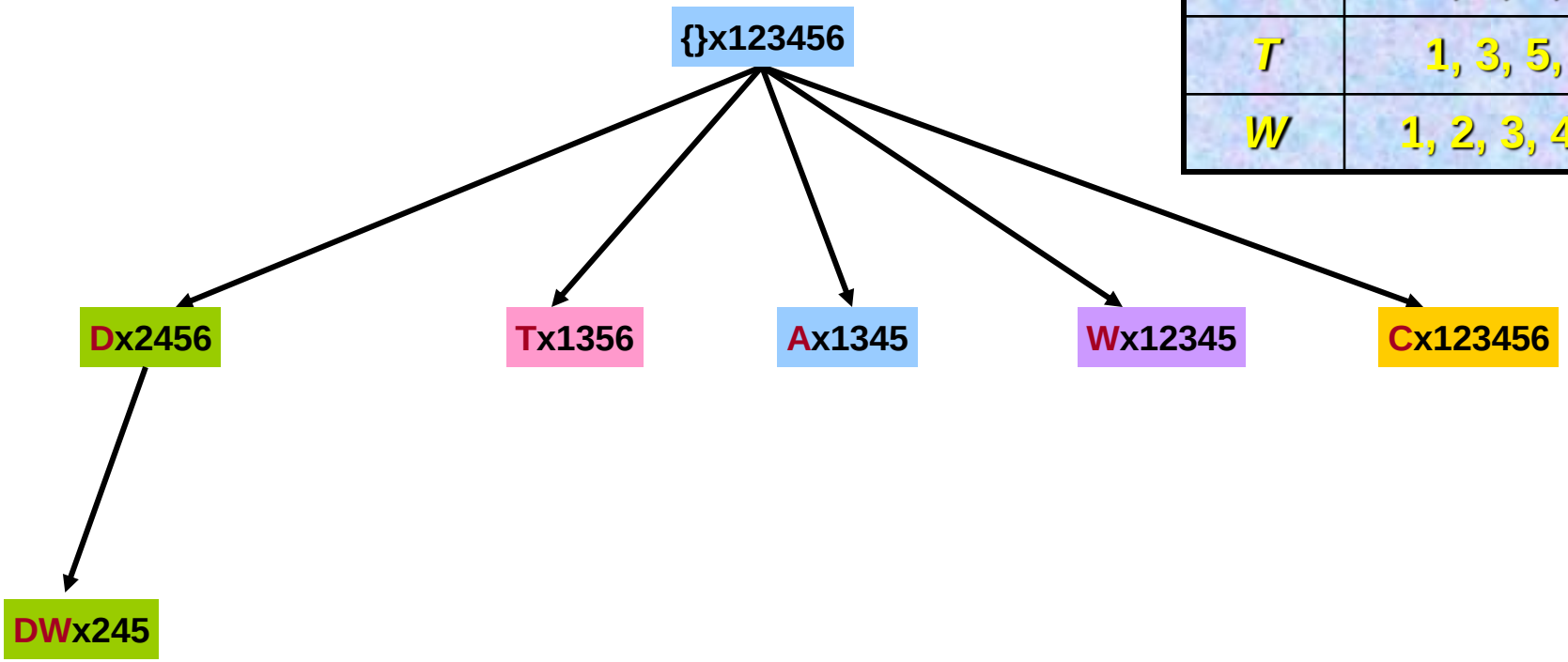
MINH HỌA CHARM (*minSup* = 3)

Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



MINH HỌA CHARM (*minSup* = 3)

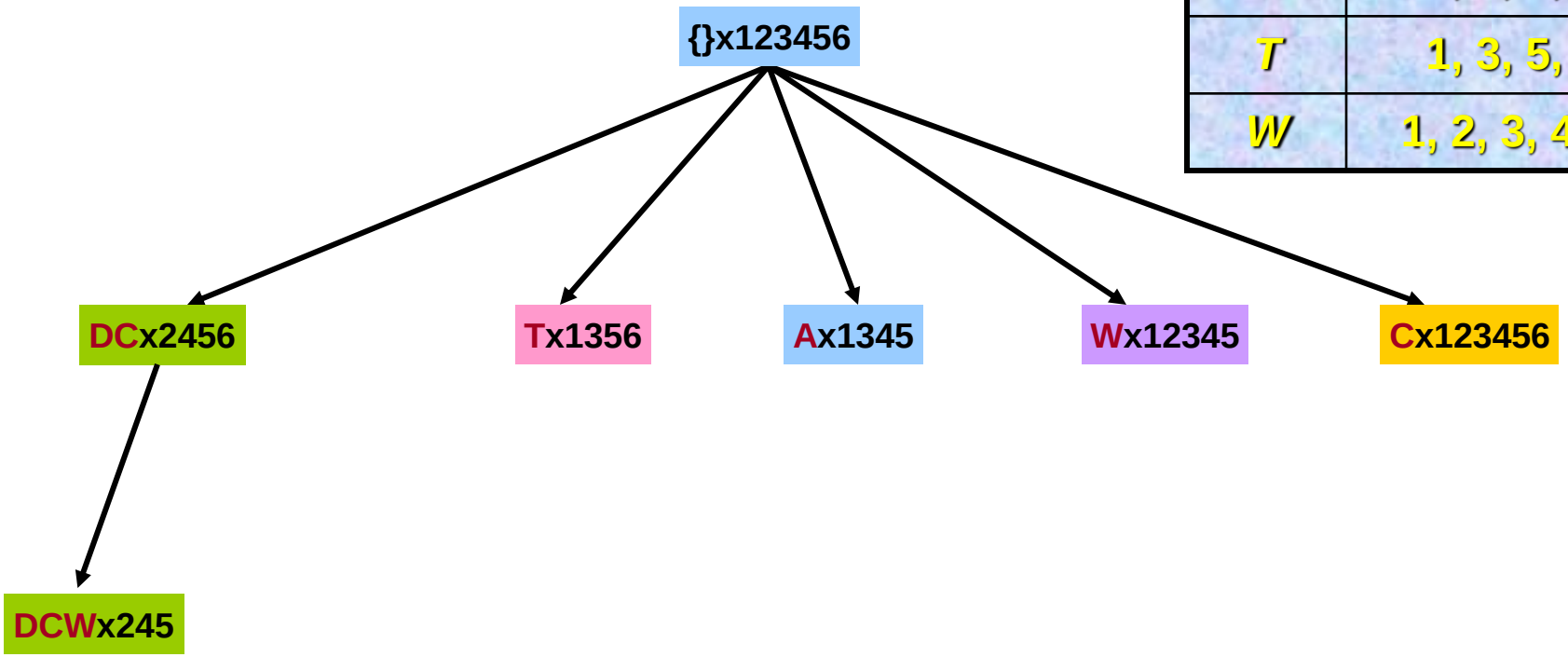
Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



Xét D với T, A, W: Chỉ có σ(DW) thỏa *minSup* nên thêm vào lớp tương đương có tiền tố là D

MINH HỌA CHARM (*minSup* = 3)

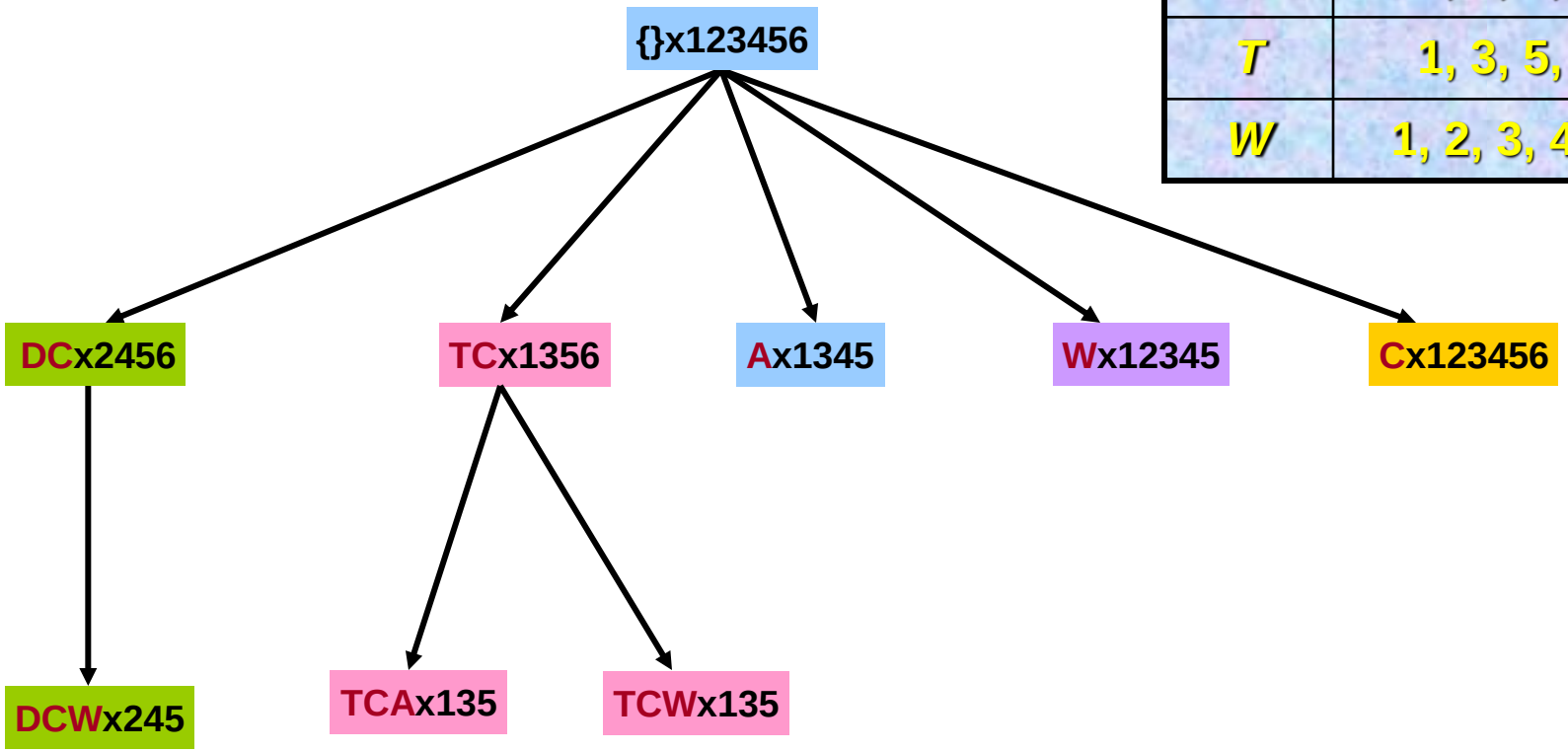
Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



Xét D với C: Do $t(D) \subset t(C) \Rightarrow$ Thỏa tính chất 2 nên D không là tập đóng $\Rightarrow$ Thay D bởi DC và DW bởi DCW

MINH HỌA CHARM (*minSup* = 3)

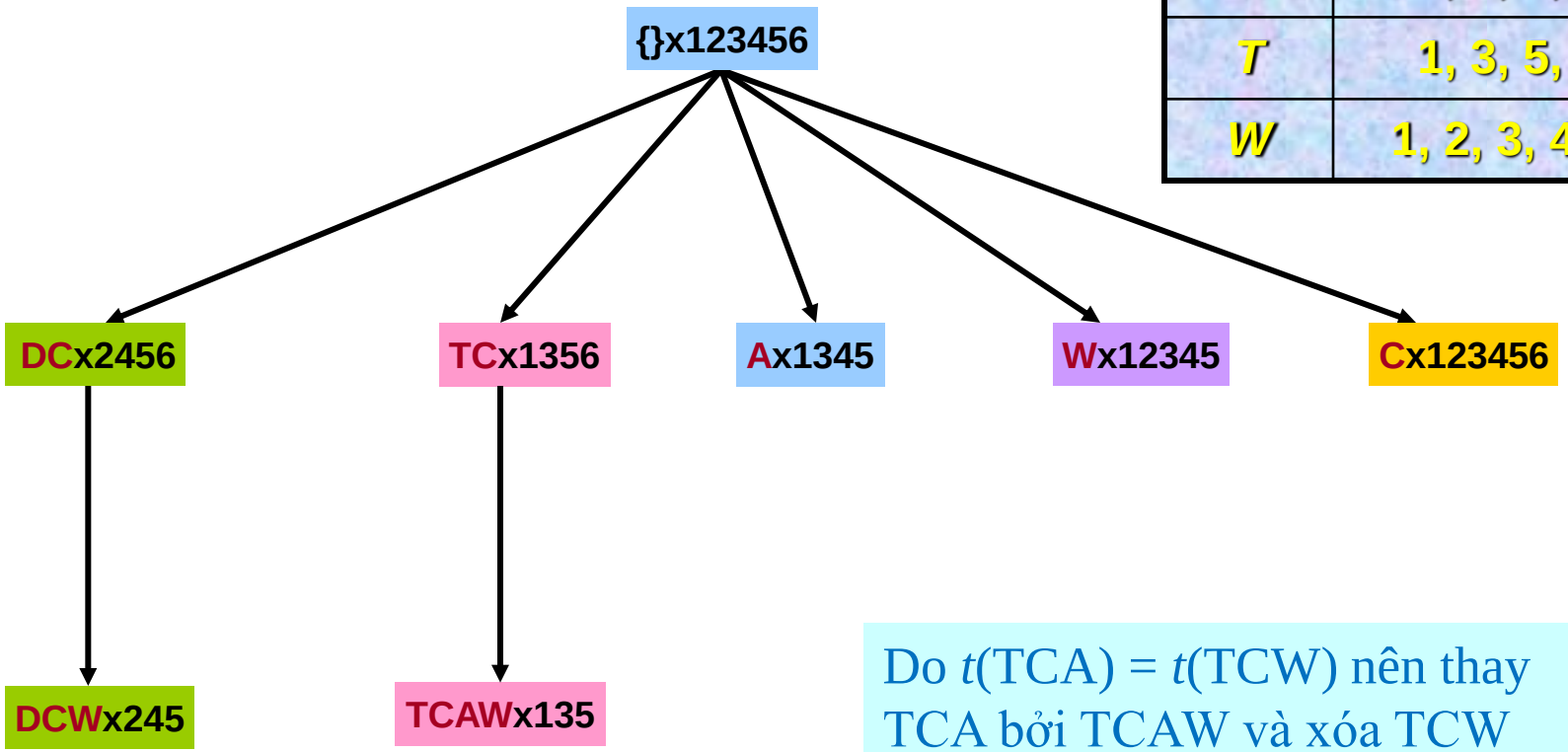
Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



IT-tree sau khi kết hợp T với A, W, C

MINH HỌA CHARM (*minSup* = 3)

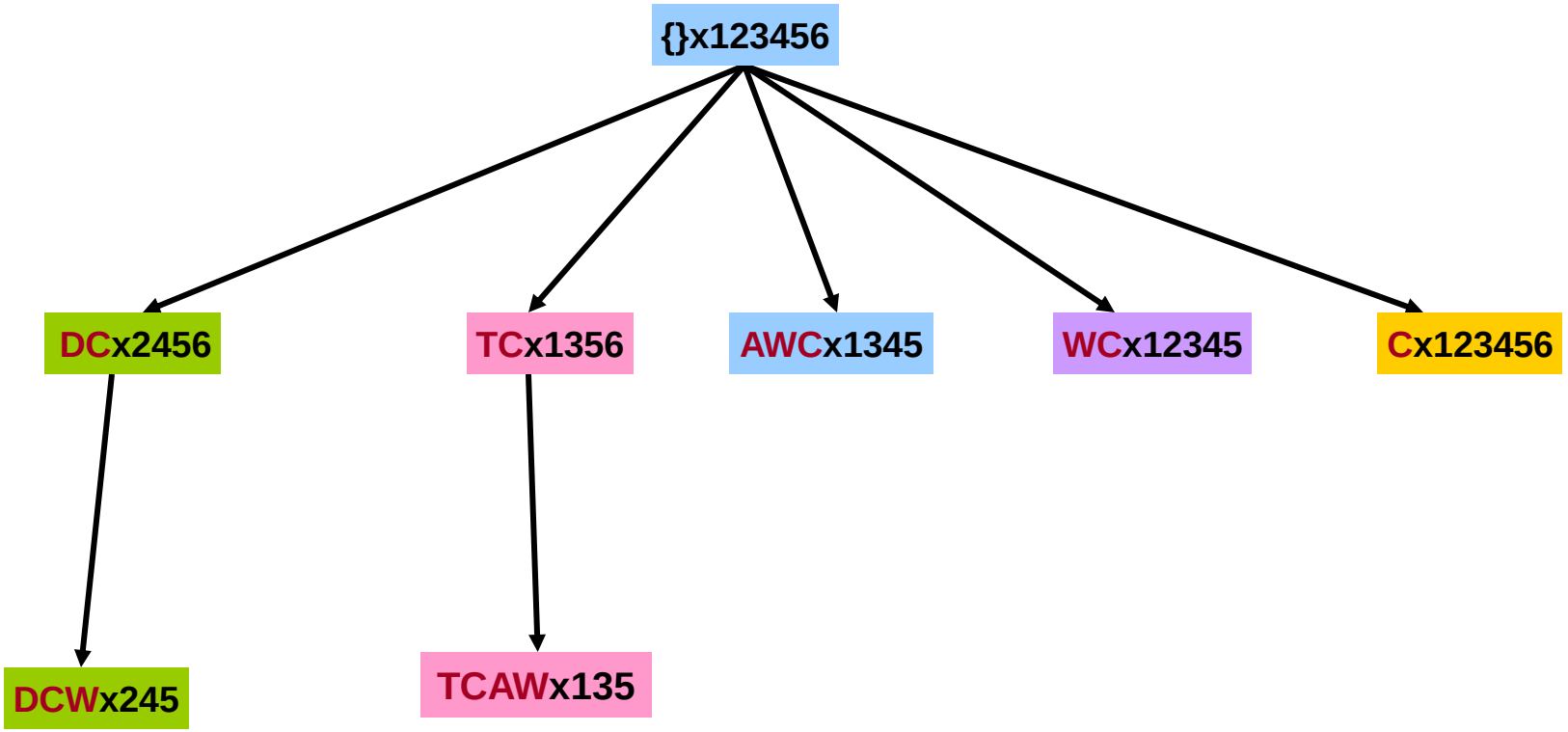
Item	TID
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5



Do *t*(TCA) = *t*(TCW) nên thay TCA bởi TCAW và xóa TCW

Kết hợp TCA với TCW

MINH HỌA CHARM (minSup=50%)



Có tất cả 7 tập phổ biến đóng thỏa *minSup* = 3 gồm: DC, TC, AWC, WC, C, DWC, TAWC

NHẬN XÉT

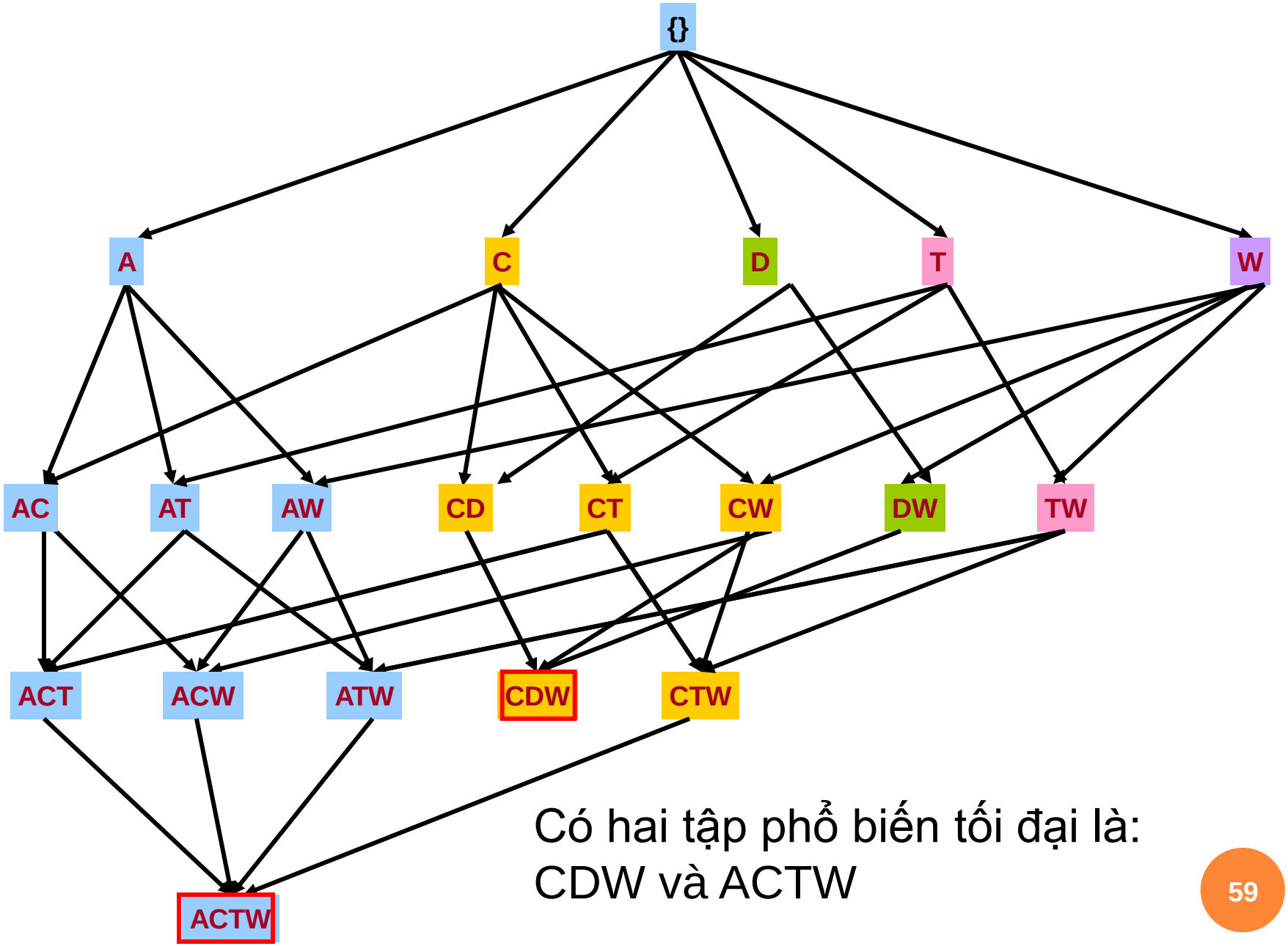
- Số lượng tập phổ biến đóng thường nhỏ hơn nhiều so với số tập phổ biến. Như vậy, việc khai thác luật từ chúng sẽ hiệu quả hơn.

- Mức tìm kiếm trên IT-tree để khai thác FCI thấp hơn so với tìm FI $\Rightarrow$ không gian bộ nhớ yêu cầu cho quá trình gọi đệ qui sẽ nhỏ hơn.

TẬP PHỔ BIẾN TỐI ĐẠI (MAXIMAL FREQUENT ITEMSETS - MFI)

- Định nghĩa:

  Cho tập phổ biến $X \subseteq I$, X được gọi là tập phổ biến tối đại nếu không tồn tại tập phổ biến Y sao cho $X \subseteq Y$.



Có hai tập phổ biến tối đại là:
CDW và ACTW

1. Khai thác tập phổ biến
2. Sinh luật kết hợp

SINH LUẬT TRUYỀN THỐNG (TRADITIONAL ASSOCIATION RULES)

- Định nghĩa:

  Luật kết hợp là biểu thức có dạng $X \rightarrow Y \setminus X$ (q, p) (X, Y là các tập phổ biến) trong đó $X, Y \neq \varnothing$, $X \subset Y$ và $p = \sigma(Y)/\sigma(X) \geq minConf$ gọi là độ tin cậy của luật còn $q = \sigma(Y) \geq minSup$ được gọi là độ phổ biến của luật.

- Như vậy: luật kết hợp là luật sinh ra giữa các tập phổ biến X, $Y \in$ **FIs** trong đó $X \subset Y$.

LUẬT TRUYỀN THỐNG: THUẬT TOÁN*

EXTRACT_AR(FIs, *minConf*)
  SORT (FIs) // Sắp xếp tập FI tăng theo k-itemset
  AR = ∅
  for each $Y \in$ **FIs** do
    for each $X \in$ **FIs** with Y after X do
      if $X \subset Y$ then
        conf = $\sigma(Y)/\sigma(X)$
        if conf $\geq$ *minConf* then
        AR = AR $\cup \{X \rightarrow Y \setminus X\ (\sigma(Y), \text{conf})\}$
  return **AR**

*Trong thực tế: để khai thác nhanh luật kết hợp, chúng ta sử dụng kỹ thuật bảng băm.

MINH HỌA
MINSUP = 3, MINCONF = 80%

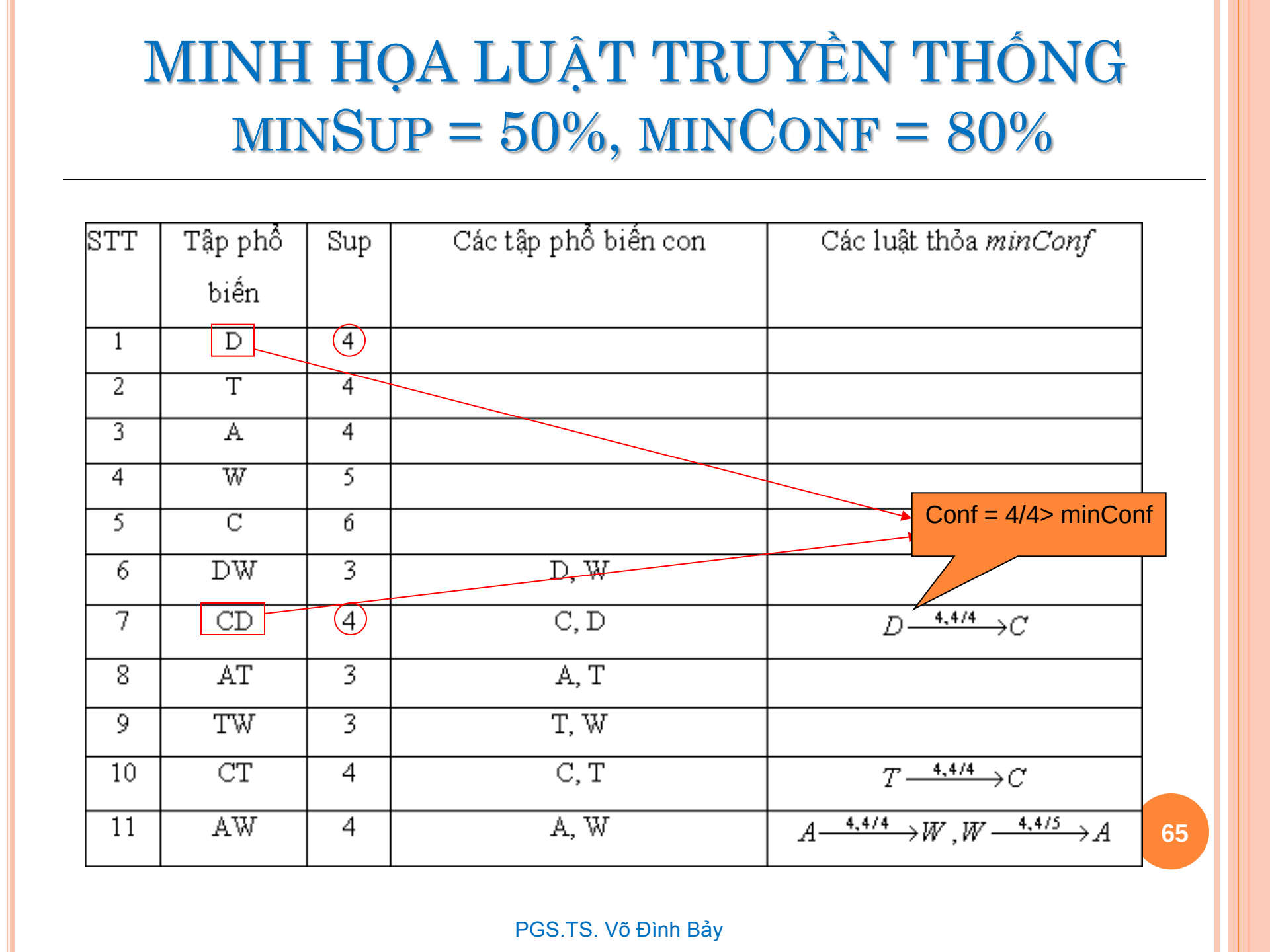
STT	Tập phổ biến	Sup	Các tập phổ biến con	Các luật thỏa *minConf*
1	D	4		
2	T	4		
3	A	4		D →W, conf = ¾<minConf
4	W	5		
5	C	6		
6	DW	3	D, W	
7	CD	4	C, D	$D \xrightarrow{4,4/4} C$
8	AT	3	A, T	
9	TW	3	T, W	
10	CT	4	C, T	$T \xrightarrow{4,4/4} C$
11	AW	4	A, W	$A \xrightarrow{4,4/4} W$, $W \xrightarrow{4,4/5} A$

MINH HỌA LUẬT TRUYỀN THỐNG
MINSUP = 50%, MINCONF = 80%

STT	Tập phổ biến	Sup	Các tập phổ biến con	Các luật thỏa *minConf*
1	D	4		
2	T	4		
3	A	4		
4	W	5		W →D, conf = 3/5<minConf
5	C	6		
6	DW	3	D, W	
7	CD	4	C, D	$D \xrightarrow{4,4/4} C$
8	AT	3	A, T	
9	TW	3	T, W	
10	CT	4	C, T	$T \xrightarrow{4,4/4} C$
11	AW	4	A, W	$A \xrightarrow{4,4/4} W$, $W \xrightarrow{4,4/5} A$

MINH HỌA LUẬT TRUYỀN THỐNG
MINSUP = 50%, MINCONF = 80%

STT	Tập phổ biến	Sup	Các tập phổ biến con	Các luật thỏa *minConf*
1	D	4		
2	T	4		
3	A	4		
4	W	5		
5	C	6		
6	DW	3	D, W	
7	CD	4	C, D	$D \xrightarrow{4,4/4} C$
8	AT	3	A, T	
9	TW	3	T, W	
10	CT	4	C, T	$T \xrightarrow{4,4/4} C$
11	AW	4	A, W	$A \xrightarrow{4,4/4} W$, $W \xrightarrow{4,4/5} A$



MINH HỌA (TT)

12	AC	4	A, C	$A \xrightarrow{4,4/4} C$
13	CW	5	C, W	$C \xrightarrow{5,5/6} W$, $W \xrightarrow{5,5/5} C$
14	CDW	3	C, D, W, CD, CW, DW	$DW \xrightarrow{3,3/3} C$
15	ATW	3	A, T, W, AT, AW, TW	$AT \xrightarrow{3,3/3} W$, $TW \xrightarrow{3,3/3} A$
16	ACT	3	A, C, T, AC, AT, CT	$AT \xrightarrow{3,3/3} C$
17	CTW	3	C, T, W, CT, CW, TW	$TW \xrightarrow{3,3/3} C$
18	ACW	4	A, C, W, AC, AW, CW	$A \xrightarrow{4,4/4} CW$, $W \xrightarrow{4,4/5} AC$ $AC \xrightarrow{4,4/4} W$, $AW \xrightarrow{4,4/4} C$ $CW \xrightarrow{4,4/5} A$
19	ACTW	3	A, C, T, W, AC, AT, AW, CT, CW, TW, ACT, ACW, ATW, CTW	$AT \xrightarrow{3,3/3} CW$, $TW \xrightarrow{3,3/3} AC$ $ACT \xrightarrow{3,3/3} W$, $ATW \xrightarrow{3,3/3} C$ $CTW \xrightarrow{3,3/3} A$

Có tất cả 60 luật trong đó có 22 luật thỏa *minConf* = 80%

CÁC CHỦ ĐỀ NÂNG CAO

PGS.TS. Võ Đình Bảy
Khoa CNTT, Trường đại học Công nghệ TP.HCM
bayvodinh@gmail.com

CONTENTS

- Mining patterns
  - Methods & data formats
  - Dynamic bit vectors
  - Quantitative databases
- Mining association rules
  - Traditional approaches
  - Lattice-based approaches
  - Class association rules
- Future research directions

MINING FREQUENT PATTERNS

- Apriori-based algorithms
  - Apriori (Agrawal & Srikant, 1994)
  - A-Close (Pasquier et al., 1999)
  - BitTableFI (Dong & Han, 2007)

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. VLDB'94, 487-499.

Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. ICDT'99, 398 – 416.

Dong, J., Han, M. (2007). BitTableFI: An efficient mining frequent itemsets algorithm. Knowledge Based Systems 20 (4), 329–335.

MINING FREQUENT PATTERNS

- IT(Itemset Tidset)-tree-based algorithms
  - Eclat (Zaki et al., 1997) & dEclat (Zaki & Hsiao, 2005)
  - CHARM & dCHARM (Zaki & Hsiao, 2005)
  - Index-BitTableFI (Song et al., 2008)
  - DBV-Miner (Vo et al., 2012)

Zaki, M. J., Parthasarathy, S., Ogihara, M., Li, W. (1997). New algorithms for fast discovery of association rules. KDD, 283–286.

Zaki, M. J., Hsiao, C.J. (2005). Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering 17 (4), 462-478

Song, W., Yang, B., Xu, Z. (2008). Index-BitTableFI: An improved algorithm for mining frequent itemsets. Knowledge Based Systems 21, 507–513.

Vo, B., Hong, T.P., Le, B. (2012). DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets. Expert Systems with Applications 39 (8), 7196-7206.

MINING FREQUENT PATTERNS

- FP(Frequent Pattern)-tree-based algorithms
  - FP-Growth (Han et al., 2000)
  - Closet (Pei et al., 2000)
  - FP-Growth* (Grahne & Zhu, 2005)

Han, J., Pei, J., Yin, Y. (2000). Mining frequent patterns without candidate generation. SIGMODKDD'00, 1 – 12.
Pei, J., Han, J., Mao, R. (2000). CLOSET: An efficient algorithm for mining frequent closed itemsets. Proc. of the 5th ACM-SIGMOD Workshop on Research Issues in KDD, 11 – 20 (2000)
Grahne, G., Zhu, J. (2005). Fast algorithms for frequent itemset mining using FP-trees. IEEE Transactions on Knowledge and Data Engineering 17 (10), 1347-1362.

MINING FREQUENT PATTERNS

Data Formats

Horizon Data Format

TID	Items
1	*A, C, T, W*
2	*C, D, W*
3	*A, C, T, W*
4	*A, C, D, W*
5	*A, C, D, T, W*
6	*C, D, T*

Vertical Data Format

Item	TIDs
A	1, 3, 4, 5
C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
T	1, 3, 5, 6
W	1, 2, 3, 4, 5

MINING FREQUENT PATTERNS

Horizon Data Format (at least two database scans)

- Apriori (Agrawal & Srikant, 1994).
- FP-Growth (Han et al., 2000).
- Closet (Pei et al., 2000).
- FP-Growth* (Ghahne & Zhu, 2005).

TID	Items
1	*A, C, T, W*
2	*C, D, W*
3	*A, C, T, W*
4	*A, C, D, W*
5	*A, C, D, T, W*
6	*C, D, T*

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. VLDB'94, 487-499.

Han, J., Pei, J., Yin, Y. (2000). Mining frequent patterns without candidate generation. SIGMODKDD'00, 1 – 12.

Pei, J., Han, J., Mao, R. (2000). CLOSET: An efficient algorithm for mining frequent closed itemsets. Proc. of the 5th ACM-SIGMOD Workshop on Research Issues in KDD, 11 – 20.

Grahne, G., Zhu, J. (2005). Fast algorithms for frequent itemset mining using FP-trees. IEEE Transactions on Knowledge and Data Engineering 17 (10), 1347-1362.

MINING FREQUENT PATTERNS

Vertical Data Format (One scan database)

- Tidset-based & Diffset-based (Zaki & Hsiao, 2005).
- BitTable-based (Dong & Han, 2007; Song et al., 2008; Sahoo et al., 2015)
- Dynamic Bit Vectors (Vo et al., 2012).

Zaki, M. J., Hsiao, C.J. (2005). Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering 17 (4), 462-478.

Dong, J., Han, M. (2007). BitTableFI: An efficient mining frequent itemsets algorithm. Knowledge Based Systems 20 (4), 329–335.

Song, W., Yang, B., Xu, Z. (2008). Index-BitTableFI: An improved algorithm for mining frequent itemsets. Knowledge Based Systems 21, 507–513.

Sahoo ,J., Das, A.K., Goswami, A. (2015). An effective association rule mining scheme using a new generic basis. Knowledge and Information Systems 43 (1), 127–156.

Song, W., Yang, B., Xu, Z. (2008). Index-BitTableFI: An improved algorithm for mining frequent itemsets. Knowledge Based Systems 21, 507–513.

Vo, B., Hong, T.P., Le, B. (2012). DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets. Expert Systems with Applications 39 (8), 7196-7206.

DYNAMIC BIT VECTORs

BitTable: Number of bits in a bit vector is a constant and it is the number of transactions in the database.

Remark: bit string may contain 0-bit at the begin and the end, how to remove them to reduce memory usage?

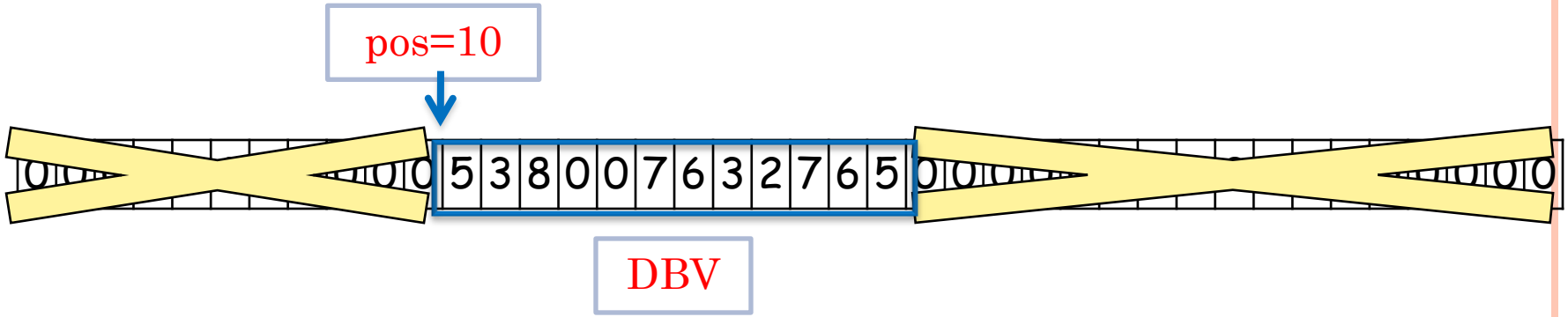
$\Rightarrow$ DBV concept.

Vo, B., Hong, T.P., Le, B. (2012). DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets. Expert Systems with Applications 39 (8), 7196-7206

DYNAMIC BIT VECTORs

Each DBV includes 2 elements

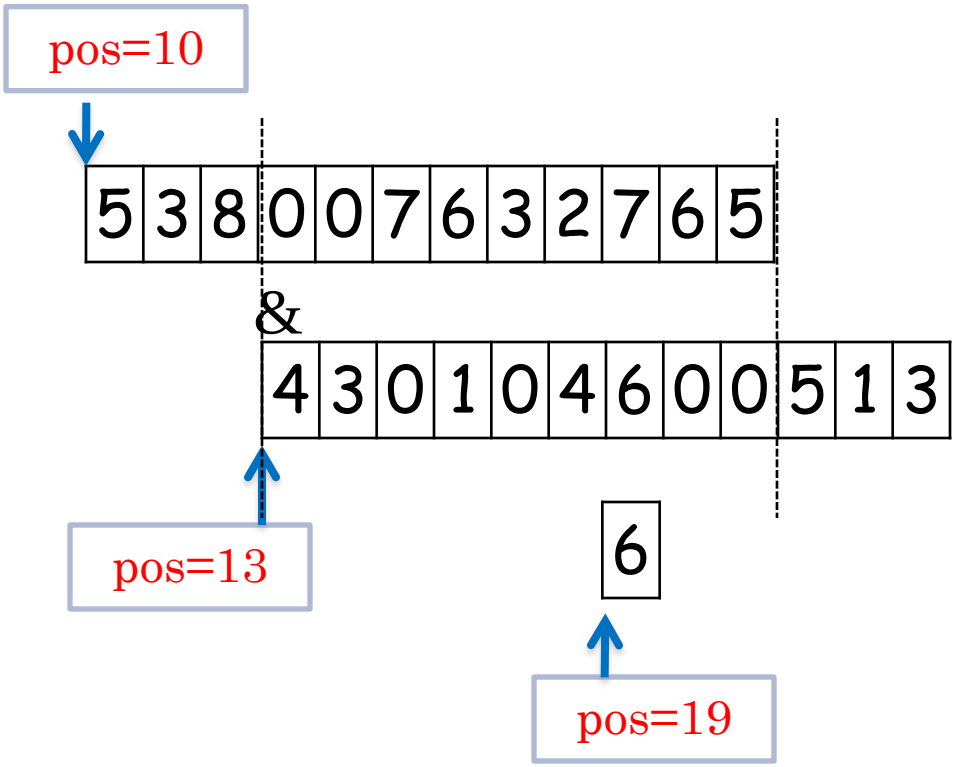
- pos: store the first none zero byte in bit string.
- Bitlist: byte-array of an itemset after removing 0 bytes at the begin and the end.



Vo, B., Hong, T.P., Le, B. (2012). DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets. Expert Systems with Applications 39 (8), 7196-7206.

DYNAMIC BIT VECTORs

Intersection between 2 DBVs



Vo, B., Hong, T.P., Le, B. (2012). DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets. Expert Systems with Applications 39 (8), 7196-7206.

DYNAMIC BIT VECTORS

- DBV-Miner (Vo et al., 2012) – FCI mining
- CloFS-DBV (Tran et al., 2015) – Closed sequence pattern mining
- ClosedISP (Le et al., 2015) – Closed inter-sequence pattern mining

Vo, B., Hong, T.P., Le, B. (2012). DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets. Expert Systems with Applications 39 (8), 7196-7206.

Tran, M.T., Le, B., Vo, B. (2015). Combination of dynamic bit vectors and transaction information for mining frequent closed sequences efficiently. Engineering Applications of Artificial Intelligence 38, 183-189.

Le, B., Tran, M.T., Vo, B. (2015). Mining frequent closed inter-sequence patterns efficiently using dynamic bit vectors. Applied Intelligence 43 (1), 74-84.

QUANTITATIVE DATABASES

item TID	A	B	C	D	E
T_1	0	0	16	0	1
T_2	0	12	0	2	1
T_3	2	0	1	0	1
T_4	1	0	0	2	1
T_5	0	0	4	0	2
T_6	1	2	0	0	0
T_7	0	20	0	2	1
T_8	3	0	25	6	1
T_9	1	2	0	0	0
T_{10}	0	12	2	0	2

Item	Benefit
A	3
B	5
C	1
D	3
E	5

How to mine patterns that their utility is greater than or equal to a threshold?

Mining High Utility Itemsets (HUIs)

DEFINITIONS

Definition 1. *The utility of an itemset*

The utility of an itemset, denoted u(X), is the sum of the local profits of each item in X in all transactions containing X.

$$u(X) = \sum_{i_p \in X} \sum_{t_q \in t(X)} f(x_{pq}, y_p)$$

Definition 2. *High utility itemset*

Itemset X is called a high utility itemset if u(X) ≥ minutil (minutil is the utility threshold).

Definition 3. *High utility itemset mining*

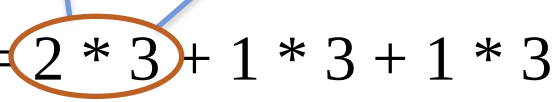
Mining high utility itemset is discovered the collection H that contains all itemsets satisfying the given minutil threshold:

$$H = \{X \mid u(X) \geq \text{minutil}\}$$

EXAMPLE

item / TID	A	B	C	D	E
T_1	0	0	16	0	1
T_2	0	12	0	2	1
T_3	2	0	1	0	1
T_4	1	0	0	2	1
T_5	0	0	4	0	2
T_6	1	2	0	0	0
T_7	0	20	0	2	1
T_8	3	0	25	6	1
T_9	1	2	0	0	0
T_{10}	0	12	2	0	2

Item	Benefit
A	3
B	5
C	1
D	3
E	5



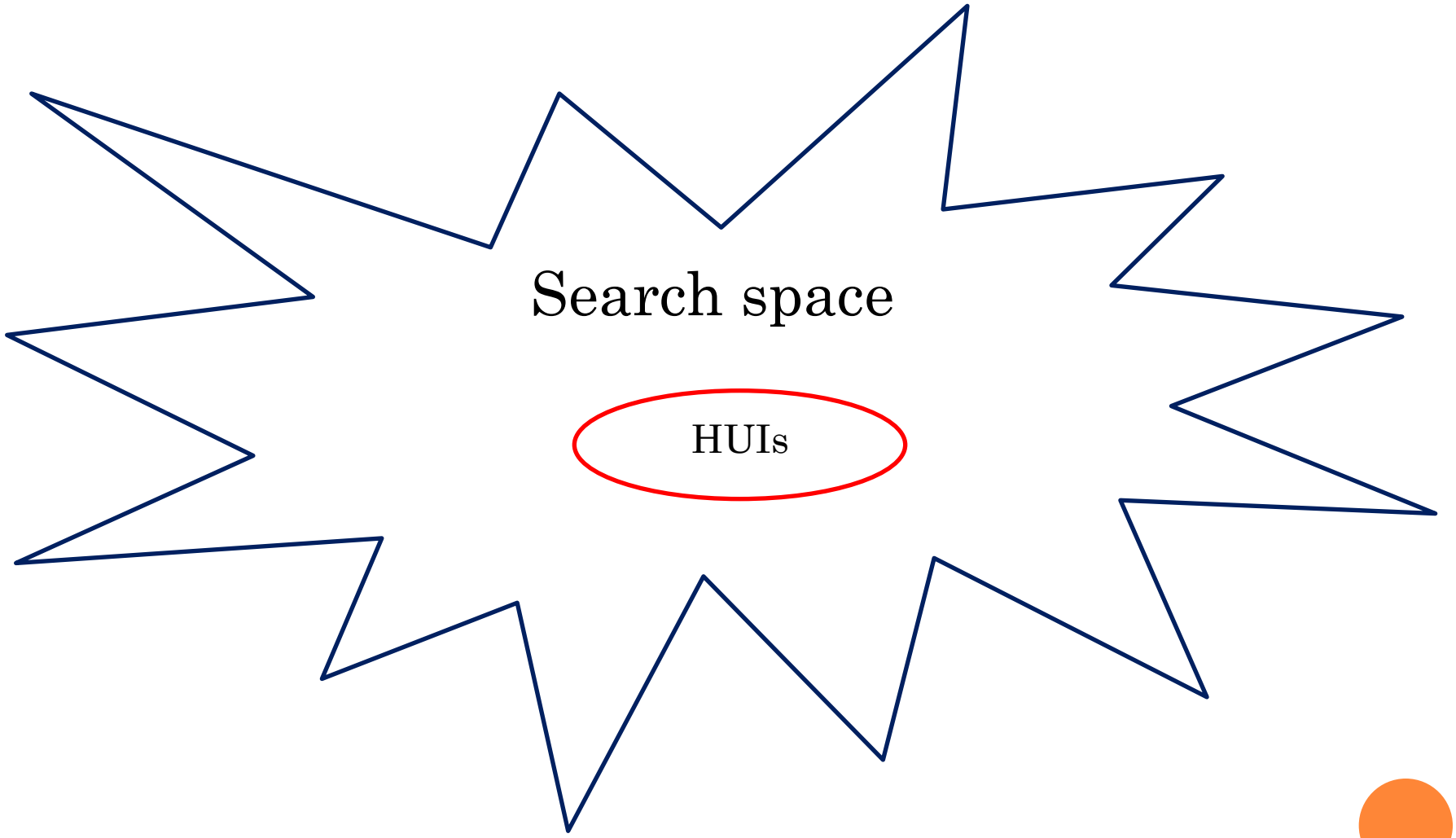
$u(A) = 2 * 3 + 1 * 3 + 1 * 3 + 3 * 3 + 1 * 3 = 24$

$u(AE) = (2*3 + 1 * 5) + (1*3 + 1*5) + (3*3 + 1*5) = 33$

$\Rightarrow$ utility does not satisfy the downward closure property.

$\Rightarrow$ Huge search space!!!

MINING HUIs



How to reduce the search space?

ALGORITHMs FOR MINING HUIs

Item	Benefit
A	3
B	5
C	1
D	3
E	5

item / TID	A	B	C	D	E	TU
T_1	0	0	16	0	1	21
T_2	0	12	0	2	1	71
T_3	2	0	1	0	1	12
T_4	1	0	0	2	1	14
T_5	0	0	4	0	2	14
T_6	1	2	0	0	0	13
T_7	0	20	0	2	1	111
T_8	3	0	25	6	1	57
T_9	1	2	0	0	0	13
T_{10}	0	12	2	0	2	72

twu(A) = 12 + 14 + 13 + 57 + 13 = 109

twu(AE) = 83.

$\Rightarrow$ twu of itemset satisfies the downward closure property(DCP).

$\Rightarrow$ We can use FP mining algorithms to mine HUIs.

Mining HUIs

2004 **Problem statement** (Hamilton et al.)

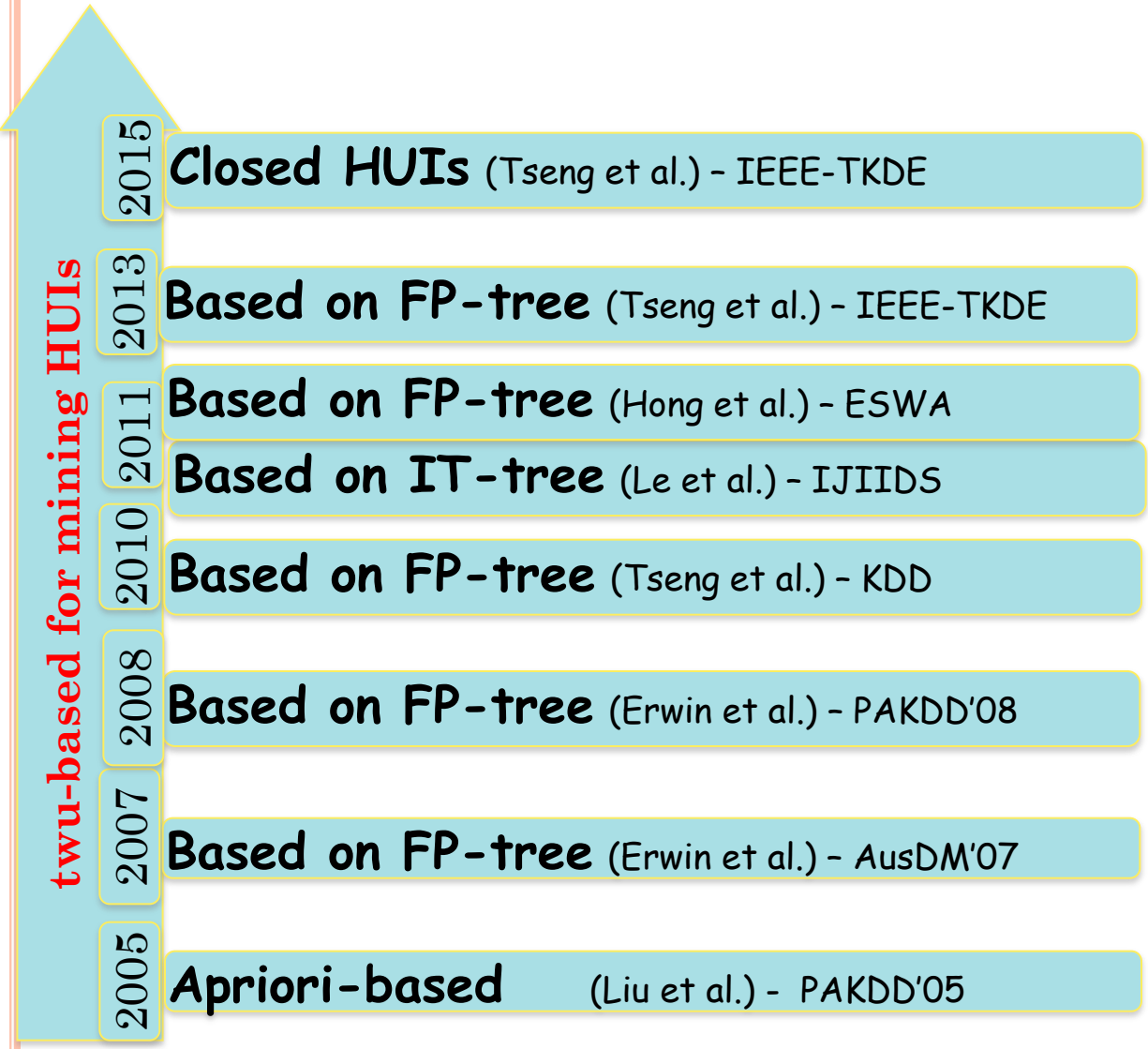
2005 **First UBDM** (USA)

TWU (Liu et al.)

twu: Transaction weight utility

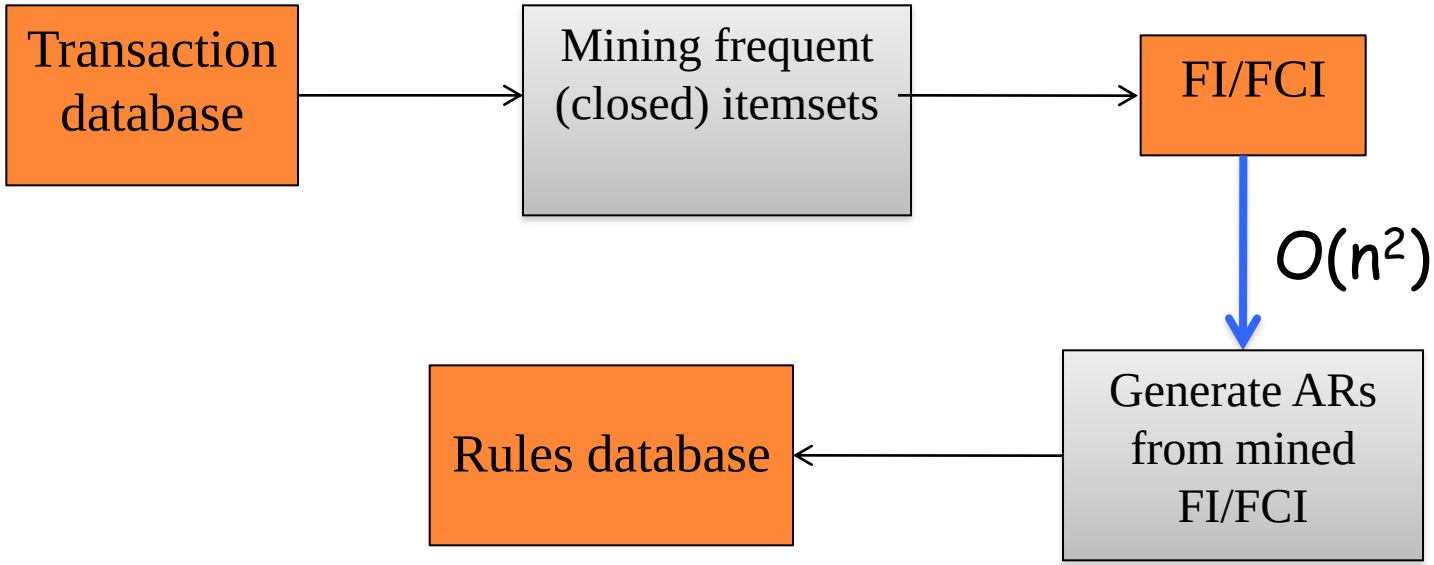
Upper bound: does not Satisfy the DCP

ALGORITHMs FOR MINING HUIs



MINING ASSOCIATION RULES (ARs)

Traditional approaches

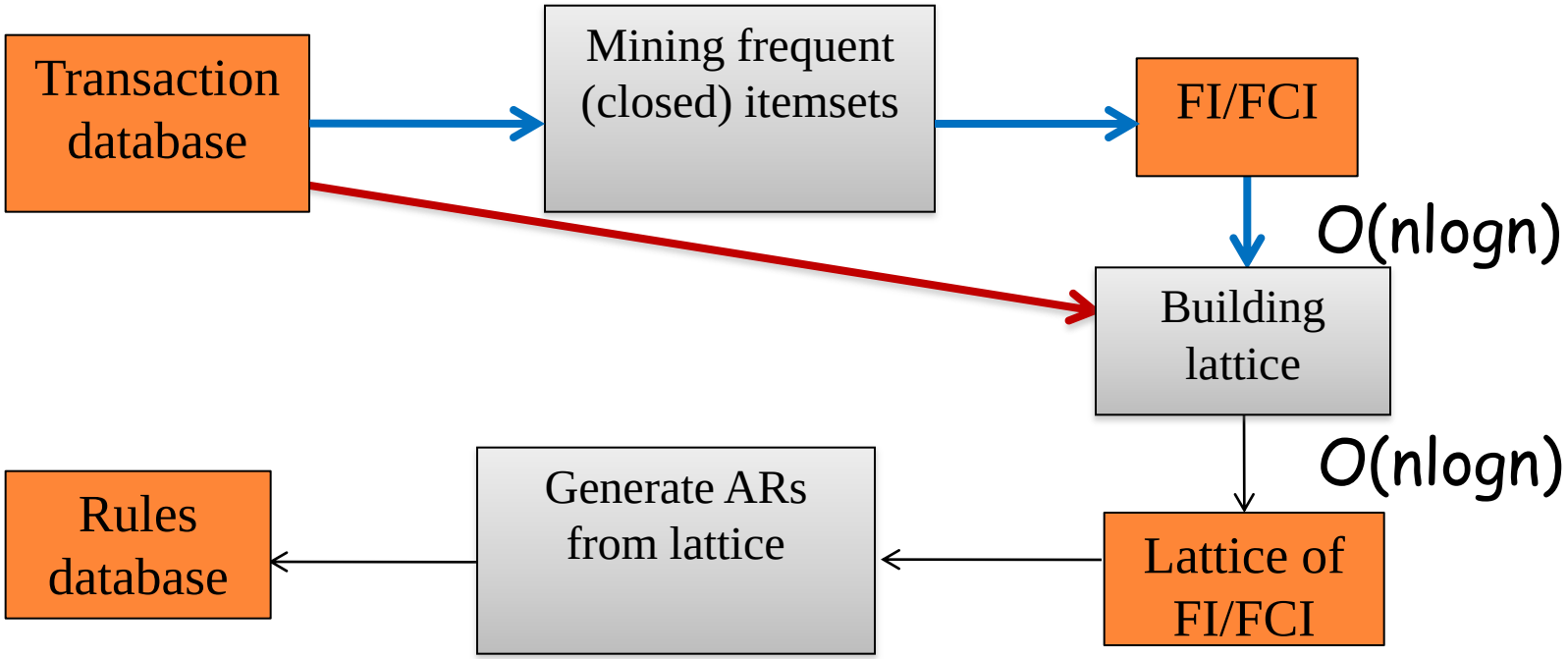


Agrawal, R., Srikant, R. (1994). Fast algorithms for mining association rules in large databases. VLDB'94, 487-499

Bastide, Y. et al. (2000). Mining minimal non-redundant association rules using closed frequent itemsets. 1st International Conference on Computational Logic, 972-986

Zaki, M.J. (2004). Mining non-redundant association rules. Data Mining and Knowledge Discovery 9 (3), 223–248.

LATTICE-BASED FOR MINING ARS



Vo, B., Hong, T.P, Le, B.(2013). A lattice-based approach for mining most generalization association rules. Knowledge-Based Systems 45, 20-30.

Vo, B., Le, T., Hong, T.P, Le, B. (2014). An effective approach for maintenance of pre-large-based frequent-itemset lattice in incremental mining. Applied Intelligence 41 (3), 759-775.

Vo, B., Le, B. (2011). Interestingness measures for association rules: Combination between lattice and hash tables. Expert Systems with Applications 38 (9), 11630-11640.

Vo, B., Le, B. (2011). Mining minimal non-redundant association rules using frequent itemsets lattice. International Journal of Intelligent Systems Technology and Applications 10 (1), 92 - 106.

Vo, B., Le, B. (2009). Mining traditional association rules using frequent itemsets lattice. 39th International Conference on CIE, Troyes, France, 1401-1406 (IEEE).

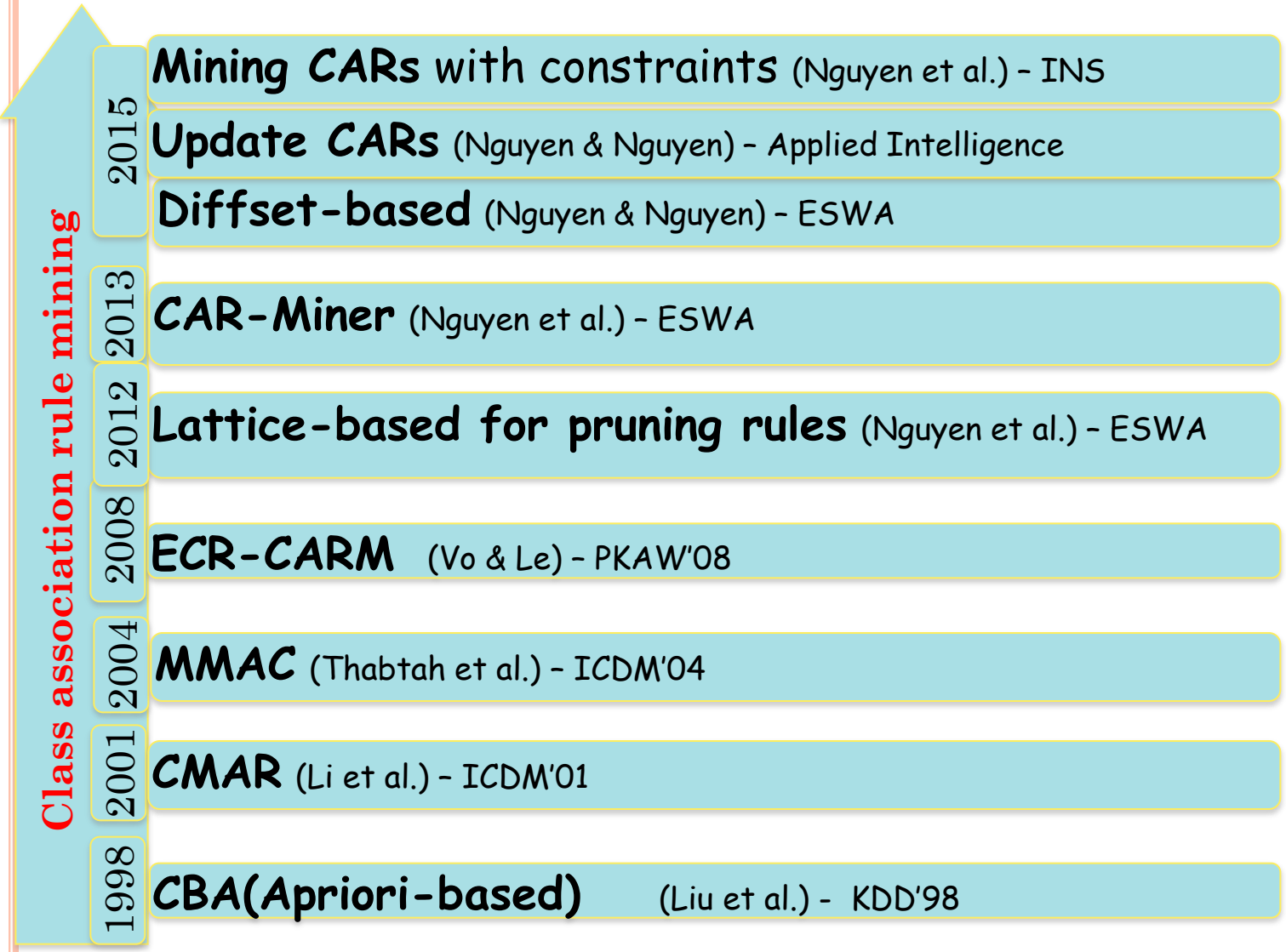
CLASS ASSOCIATION RULES

A class association rule (CARs) is an association rule form X $\rightarrow$ y, where X is an itemset, y is a class label.

Two approaches

1. Mine all association rules and after that select the class association rules.
2. Only mine class association rules (put the class constraint into the mining process).

CLASS ASSOCIATION RULES



FUTURE RESEARCH DIRECTIONS

- Continue working on current research directions with:
  - Incremental and sequence databases
  - Quantitative & hierarchical databases
  - Graph databases
  - Parallel computing
- Apply to social networks and bioinformatics
- Research on text mining, subspace clustering, etc.

THANK YOU FOR YOUR ATTENTION!