MACHINE INTELLIGENCE

Dr. N MEHALA
Department of Computer Science
and Engineering

MACHINE INTELLIGENCE

Module 4 [Unsupervised Learning]

Dr. N MEHALA

Department of Computer Science and Engineering

Apriori VS FP GROWTH ALGORITHM

- Apriori: uses a generate-and-test approach- generates candidate itemsets and tests if they are frequent
- Generation of candidate itemsets is expensive (in terms of both space and time)
- Support counting is expensive
  - Subset checking (computationally expensive)
  - Multiple Database scans (I/O)

FP Growth Algorithm

- Use a compressed representation of the database using an FP-tree
- FP-Growth: allows frequent item set discovery without candidate item set generation

Two step approach:

Step 1: Build a compact data structure called the FP-tree

- Built using 2 passes over the data-set.

Step 2: Extracts frequent item sets directly from the FP-tree

- Traversal through FP-Tree

- Once an FP-tree has been constructed, it uses a recursive divide-and-conquer approach to mine the frequent item sets

FP Growth Algorithm : Example

FP Growth Algorithm : Example

First scan – determine frequent 1-itemsets, then build header

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

B	8
A	7
C	7
D	5
E	3

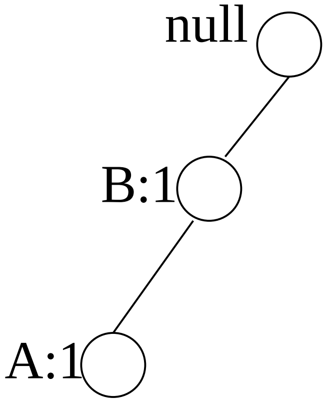
TID	Items
1	{B,A}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{B,A,C}
6	{B,A,C,D}
7	{B,C}
8	{B,A,C}
9	{B,A,D}
10	{B,C,E}

FP Growth Algorithm : Example

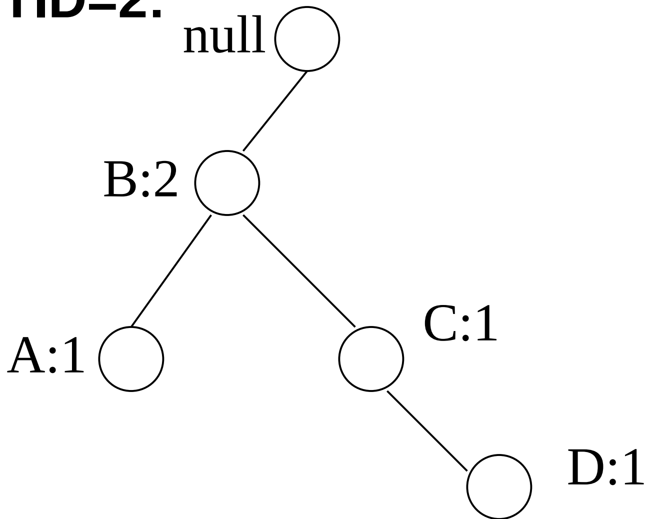
FP-tree construction

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

After reading TID=1:



After reading TID=2:

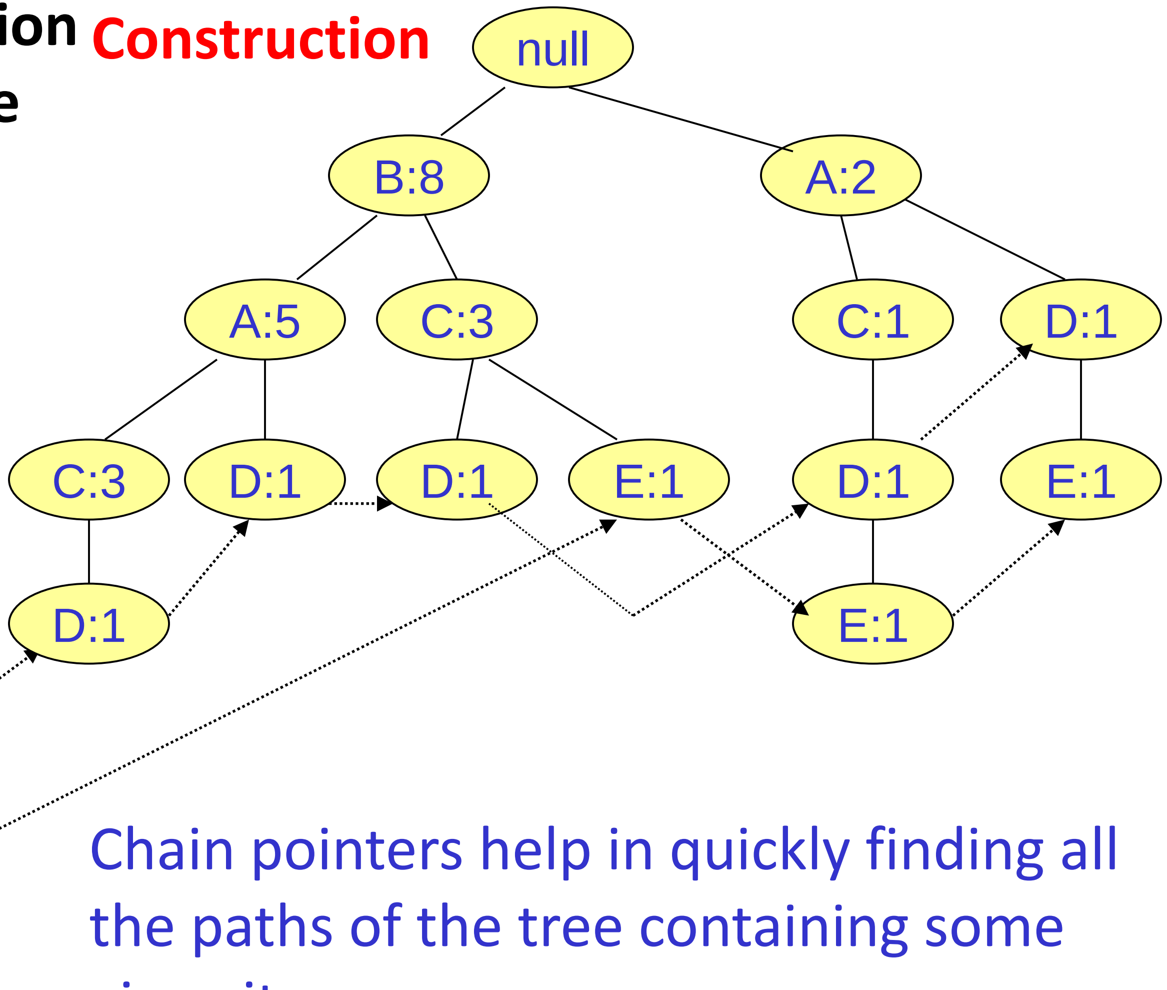


FP Growth Algorithm : Example

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

Transaction Database

FP-Tree Construction

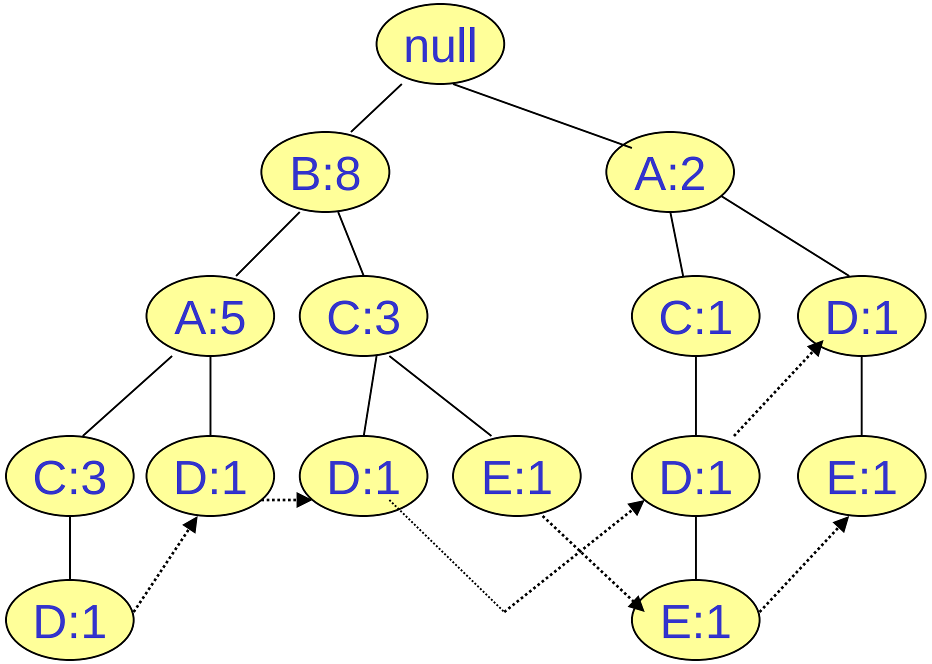


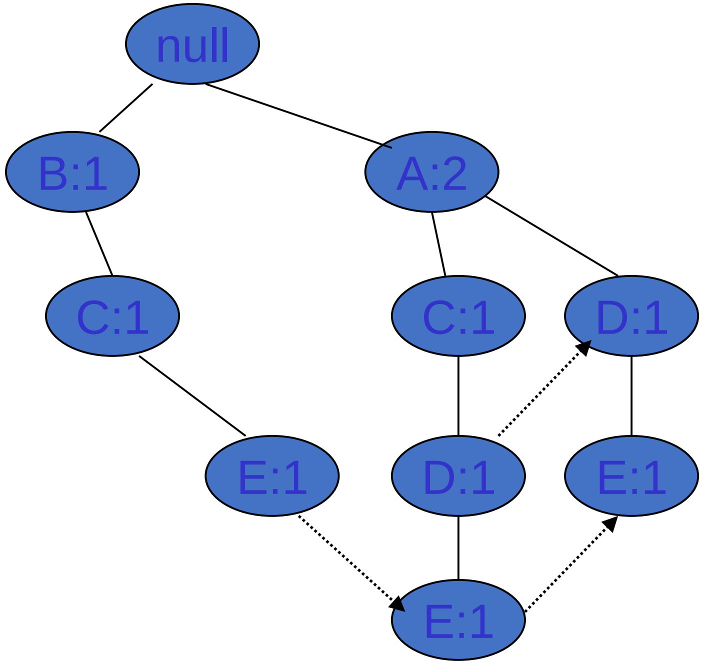
Header table

Item	Pointer
B	8
A	7
C	7
D	5
E	3

Chain pointers help in quickly finding all the paths of the tree containing some given item.

FP Growth Algorithm : Example





Paths containing node E

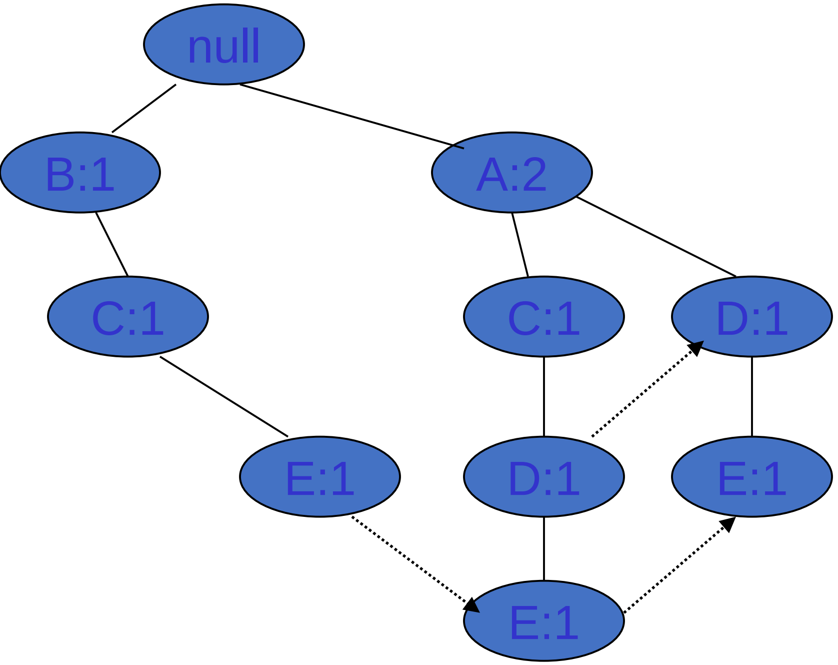
FP Growth Algorithm : Example

Conditional FP-Tree for E

- FP-Growth builds a conditional FP-Tree for E, which is the tree of itemsets ending in E.

- It is not the tree obtained in previous slide as result of deleting nodes from the original tree.
- Because the order of the items can change.
  - Now, C has a higher count than B.

Suffix E



The set of paths ending in E.

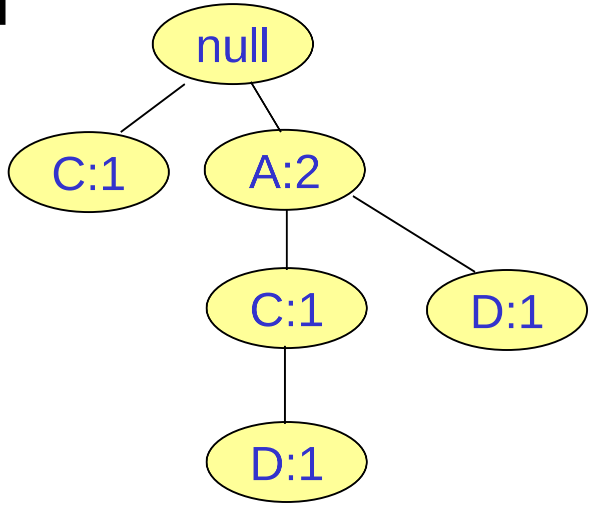
Insert each path (after truncating E) into a new tree.

(New) Header table

A	2
C	2
D	2

B doesn't survive because it has support 1, which is lower than min support of 2.

Conditional FP-Tree for suffix E



We continue recursively.
Base of recursion: When the tree has a single path only.

FP Growth Algorithm : Example

Steps of Building Conditional FP-Trees

1. Find the paths containing on focus item.

2. Read the tree to determine the new counts of the items along those paths.

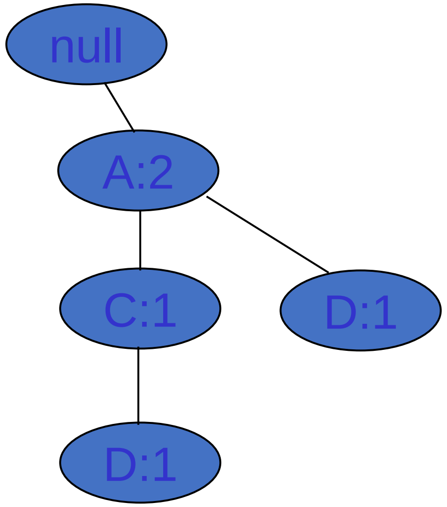
   Build a new header.

3. Read again the tree. Insert the paths in the conditional FP-Tree according to the new order.

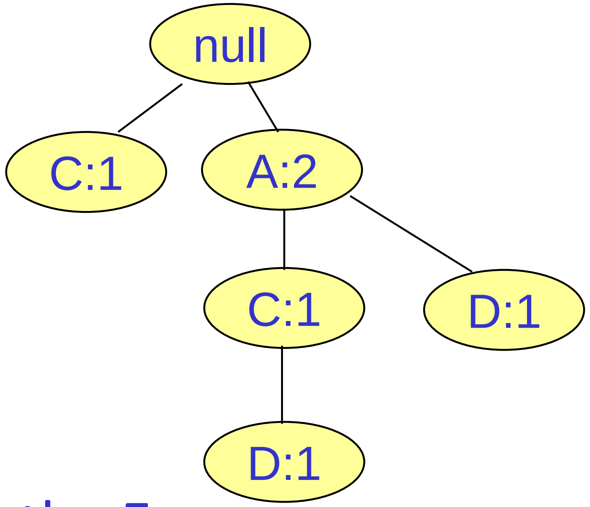
Suffix DE

(New) Header table

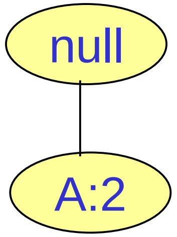
A	2



Conditional FP-Tree for suffix E



The conditional FP-Tree for suffix DE



The set of paths, from the E-conditional FP-Tree, ending in D.

Insert each path (after truncating D) into a new tree.

We have reached the base of recursion.

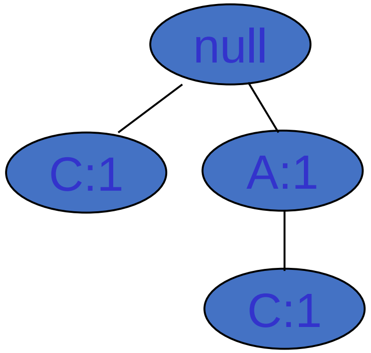
FI: DE, ADE

Base of Recursion

- We continue recursively on the conditional FP-Tree.

- Base case of recursion: when the tree is just a single path.
  - Then, we just produce all the subsets of the items on this path merged with the corresponding suffix.

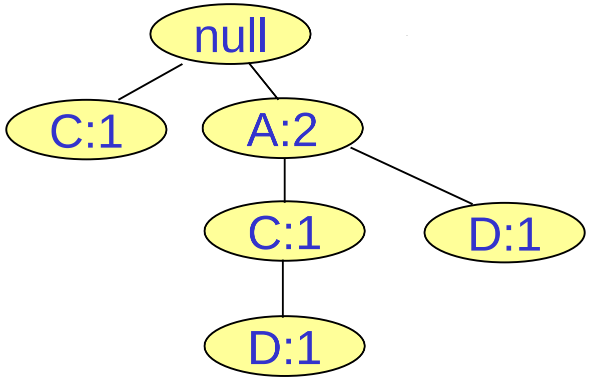
Suffix CE



The set of paths, from the E-conditional FP-Tree, ending in C.

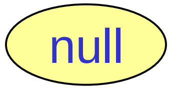
Insert each path (after truncating C) into a new tree.

Conditional FP-Tree for suffix E



(New) Header table

The conditional FP-Tree for suffix CE

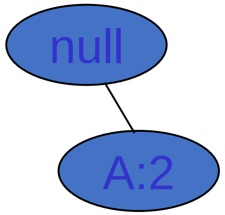


We have reached the base of recursion.

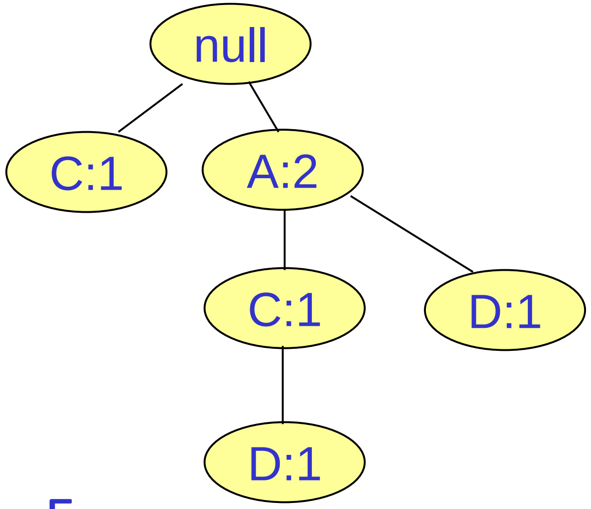
FI: CE

FP Growth Algorithm : Example

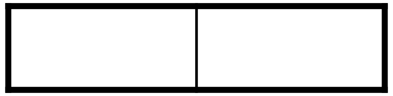
Suffix AE



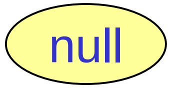
Conditional FP-Tree for suffix E



(New) Header table



The conditional FP-Tree for suffix AE



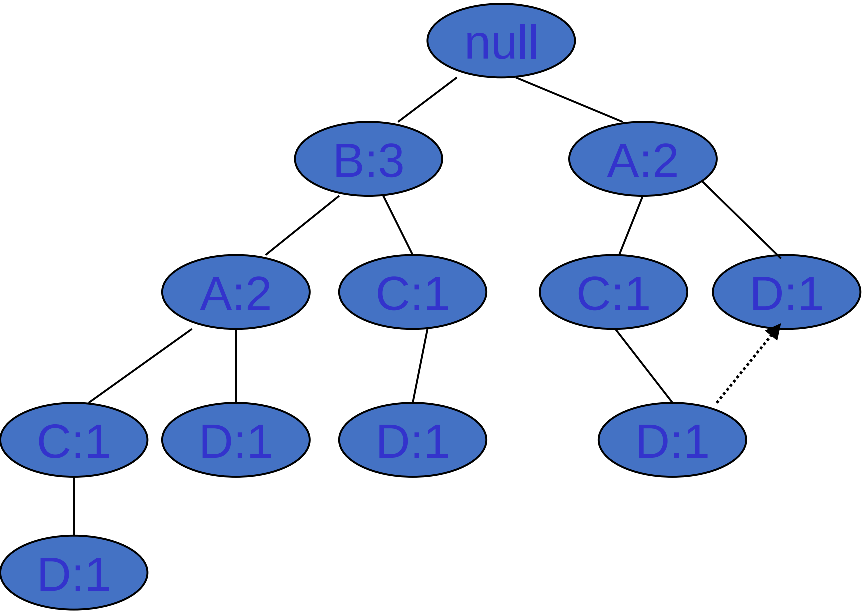
The set of paths, from the E-conditional FP-Tree, ending in A.

Insert each path (after truncating A) into a new tree.

We have reached the base of recursion.

FI: AE

Suffix D



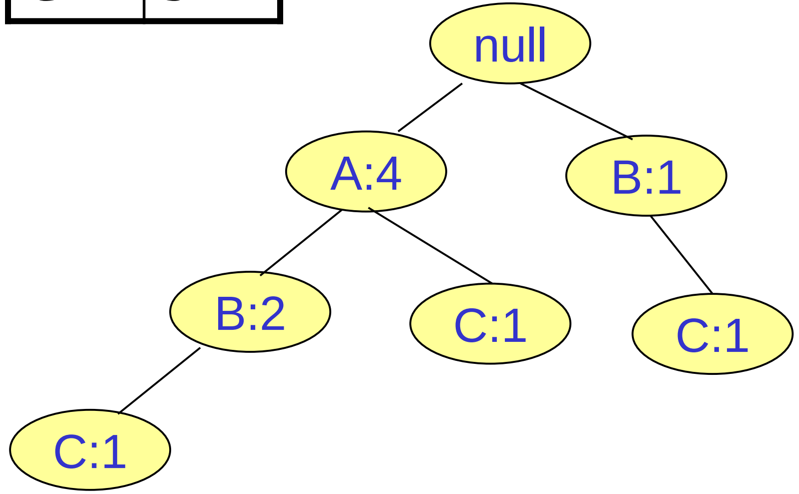
The set of paths ending in D.

Insert each path (after truncating D) into a new tree.

(New) Header table

A	4
B	3
C	3

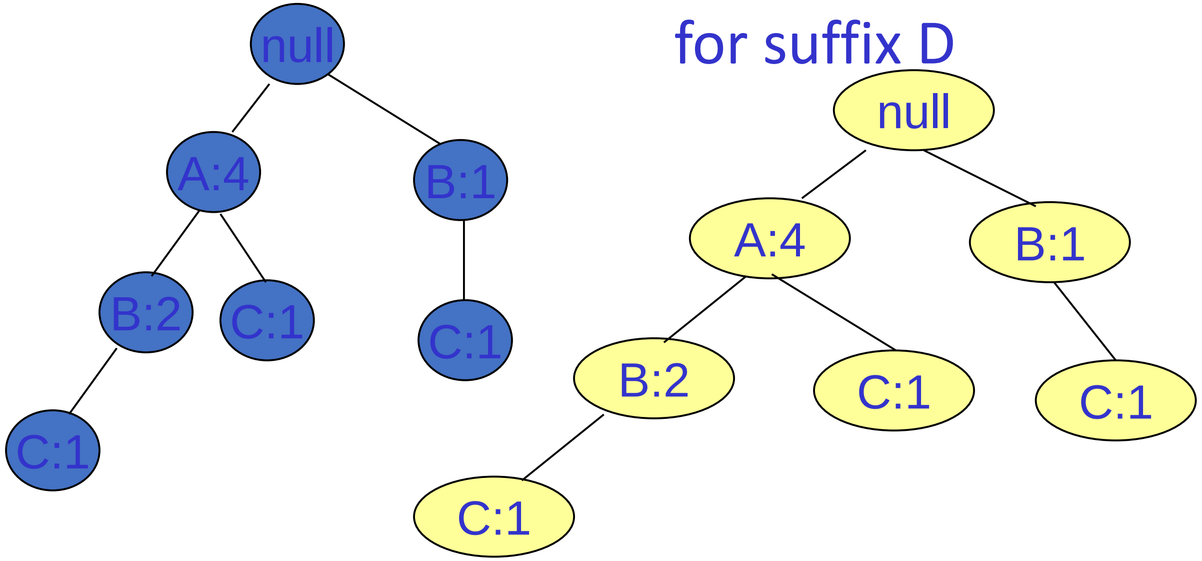
Conditional FP-Tree for suffix D



We continue recursively. Base of recursion: When the tree has a single path only.

Suffix CD

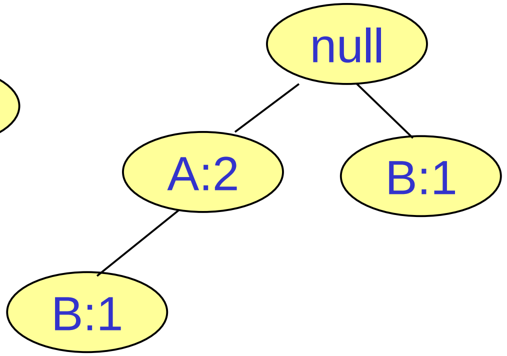
Conditional FP-Tree for suffix D



(New) Header table

A	2
B	2

Conditional FP-Tree for suffix CD

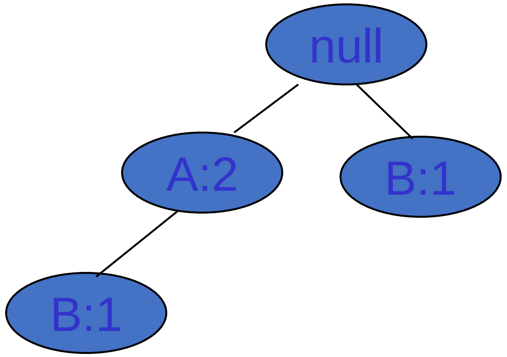


The set of paths, from the D-conditional FP-Tree, ending in C.

Insert each path (after truncating C) into a new tree.

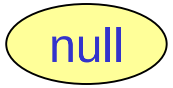
We continue recursively.
Base of recursion: When the tree has a single path only.
FI: CD

Suffix BCD



(New) Header table

Conditional FP-Tree for suffix CDB



The set of paths from the CD-conditional FP-Tree, ending in B.

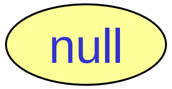
Insert each path (after truncating B) into a new tree.

We have reached the base of recursion.

FI: BCD

null
(New) Header table

Conditional FP-Tree for suffix ACD



Suffix ACD

The set of paths from the CD-conditional FP-Tree, ending in A.

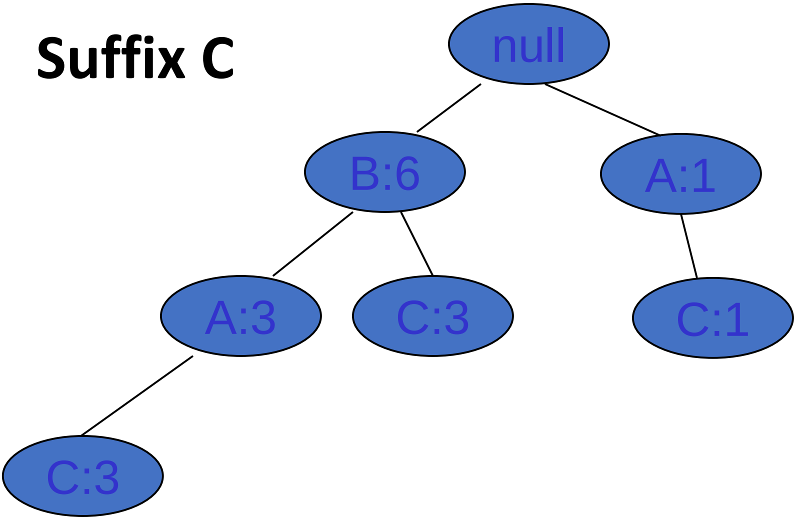
Insert each path (after truncating B) into a new tree.

We have reached the base of recursion.

FI: ACD

FP Growth Algorithm : Example

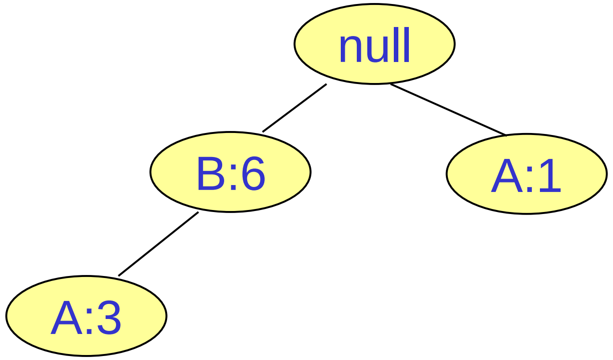
Suffix C



(New) Header table

B	6
A	4

Conditional FP-Tree for suffix C

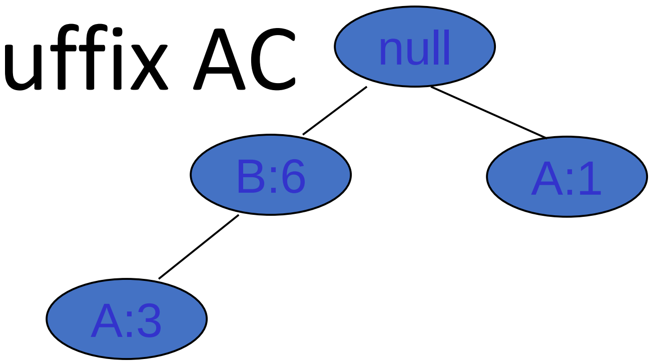


The set of paths ending in C.

Insert each path (after truncating C) into a new tree.

We continue recursively.
Base of recursion: When the tree has a single path only.
FI: C

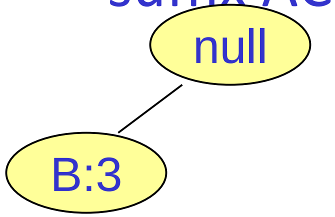
Suffix AC



(New) Header table

B	3

Conditional FP-Tree for suffix AC



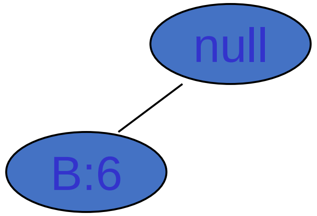
The set of paths from the C-conditional FP-Tree, ending in A.

Insert each path (after truncating A) into a new tree.

We have reached the base of recursion.

FI: AC, BAC

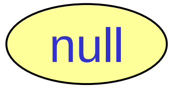
Suffix BC



(New) Header table

B	3

Conditional FP-Tree for suffix BC



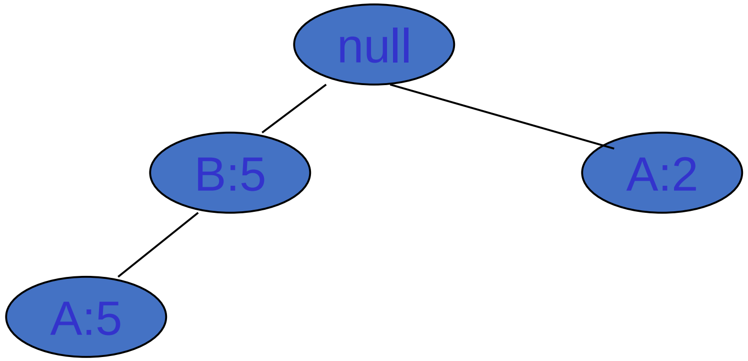
The set of paths from the C-conditional FP-Tree, ending in B.

Insert each path (after truncating B) into a new tree.

We have reached the base of recursion.

FI: BC

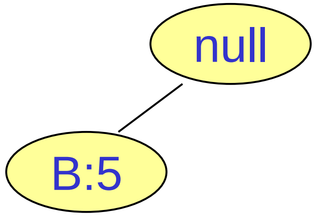
Suffix A



(New) Header table

B	5

Conditional FP-Tree for suffix A



The set of paths ending in A.

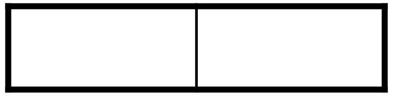
Insert each path (after truncating A) into a new tree.

We have reached the base of recursion.

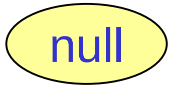
FI: A, BA

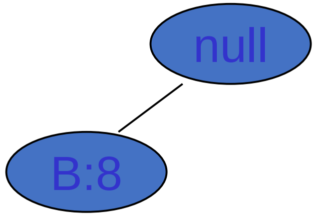
Suffix B

(New) Header table



Conditional FP-Tree for suffix B





The set of paths ending in B.

Insert each path (after truncating B) into a new tree.

We have reached the base of recursion.

FI: B

FP Growth Algorithm : Example

FREQUENT ITEM SETS:

- FI: DE, ADE
- FI: CE
- FI: AE
- FI: CD
- FI: BCD
- FI: ACD
- FI: C
- FI: AC, BAC
- FI: B
- FI: BC
- FI: A, BA

Resources

- http://www2.ift.ulaval.ca/~chaib/IFT-4102-7025/public_html/Fichiers/Machine_Learning_in_Action.pdf
- http://wwwusers.cs.umn.edu/~kumar/dmbook/.
- ftp://ftp.aw.com/cseng/authors/tan
- http://web.ccsu.edu/datamining/resources.html

PES UNIVERSITY ONLINE

THANK YOU

Dr. N MEHALA

Department of Computer Science and Engineering

mehala@pes.edu