Association Rules – FP Trees and FP-Growth Algorithm

V.S. Subrahmanian
University of Maryland
vs@cs.umd.edu

Basic Idea

- Improve association rule mining by maintaining a special tree that tracks frequent patterns => *Frequent Pattern (FP) tree.*
- Two steps:
  - Construct FP-tree from data
  - Extract association rules.

FP-Tree Node Structure

- Each node has the following fields:
  - Item (item labeling the node. Root is labeled NULL)
  - Count (number of items labeling the node)
  - Next (link to next node with the same item)
  - LCHILD (left child)
  - RCHILD (right child)
- A separate *Itemset* table has schema (*Item,First*) where First is a pointer to the first node labeled with *Item*.

Running Example

Transaction	Items
1	A,B,D,E,F
2	A,B,E,F,G
3	A,B,C,F,H
4	C,E,D,G,H
5	A,C,E,F
6	A,B,E,H
7	B,C,F,G
8	A,D,E,F,G
9	A,C,D,H
10	B,E,F,G,H

FPT Construction, I

Step 1. Build a table with a COUNT of each item (i.e. how many transactions each item occurs in.

Transaction	Items
1	A,B,D,E,F
2	A,B,E,F,G
3	A,B,C,F,H
4	C,E,D,G,H
5	A,C,E,F
6	A,B,E,H
7	B,C,F,G
8	A,D,E,F,G
9	A,C,D,H
10	B,E,F,G,H

Item	Count
A	7
B	6
C	5
D	4
E	7
F	7
G	5
H	5

FPT Construction, II

Step 2. Sort all items in descending order of count.

Transaction	Items
1	A,B,D,E,F
2	A,B,E,F,G
3	A,B,C,F,H
4	C,E,D,G,H
5	A,C,E,F
6	A,B,E,H
7	B,C,F,G
8	A,D,E,F,G
9	A,C,D,H
10	B,E,F,G,H

Item	Count
A	7
B	6
C	5
D	4
E	7
F	7
G	5
H	5

SORTED LIST: A,E,F,B,C,G,H,D

FPT Construction, III

Step 3. Sort item lists in transaction table in descending order., ignoring cases where count of an item is below support threshold. E.g. Support Threshold = 5. So D is ignored.

Transaction	Items
1	A,B,D,E,F
2	A,B,E,F,G
3	A,B,C,F,H
4	C,E,D,G,H
5	A,C,E,F
6	A,B,E,H
7	B,C,F,G
8	A,D,E,F,G
9	A,C,D,H
10	B,E,F,G,H

Transaction	Items
1	A,E,F,B,D
2	A,E,F,B,G
3	A,F,B,C,H
4	E,C,G,H,D
5	A,E,F,C
6	A,E,B,H
7	F,B,C,G
8	A,E,F,G, D
9	A,C,H,D
10	E,F,B,G,H

SORTED LIST: A,E,F,B,C,G,H,D

FPT Construction, IV

Step 4. Build Tree, processing one tuple at a time

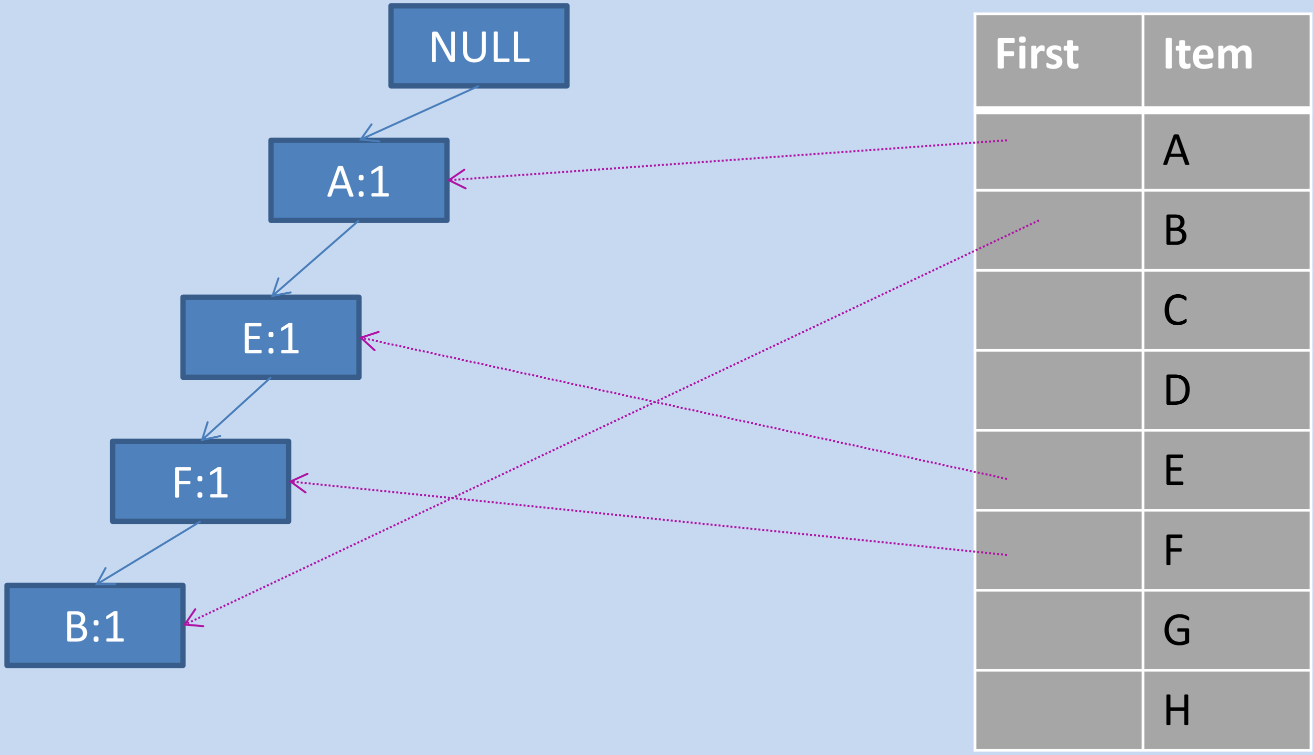
NULL

Item	First
A	
B	
C	
D	
E	
F	
G	
H	

Trans	Items
1	A,E,F,B,D
2	A,E,F,B,G
3	A,F,B,C,H
4	E,C,G,H,D
5	A,E,F,C
6	A,E,B,H
7	F,B,C,G
8	A,E,F,G,D
9	A,C,H,D
10	E,F,B,G,H

FPT Construction, IV

Step 4. Process transaction 1



NULL

A:1

E:1

F:1

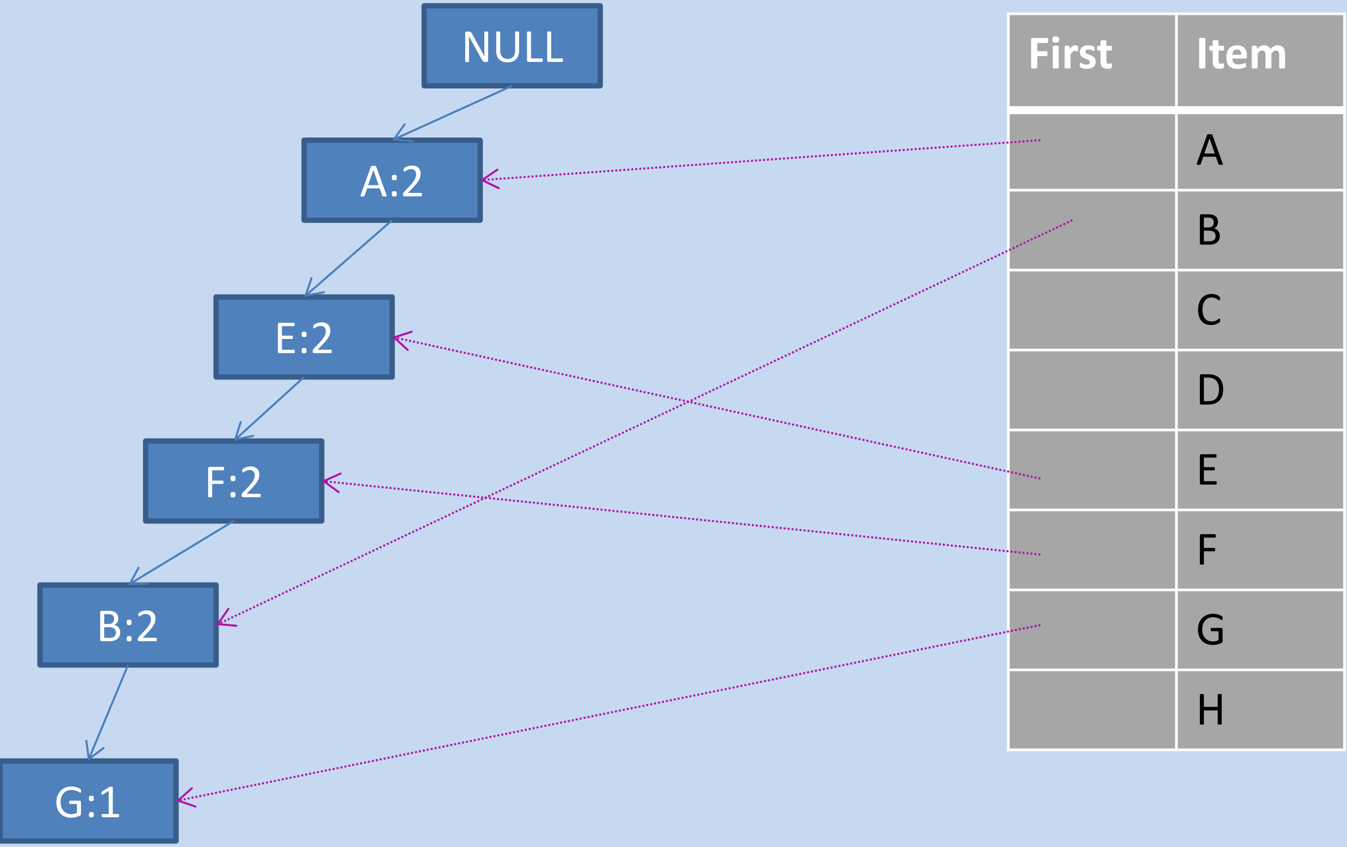
B:1

First	Item
	A
	B
	C
	D
	E
	F
	G
	H

Trans	Items
1	A,E,F,B,D
2	A,E,F,B,G
3	A,F,B,C,H
4	E,C,G,H,D
5	A,E,F,C
6	A,E,B,H
7	F,B,C,G
8	A,E,F,G,D
9	A,C,H,D
10	E,F,B,G,H

FPT Construction, IV

Step 4. Process transaction 2

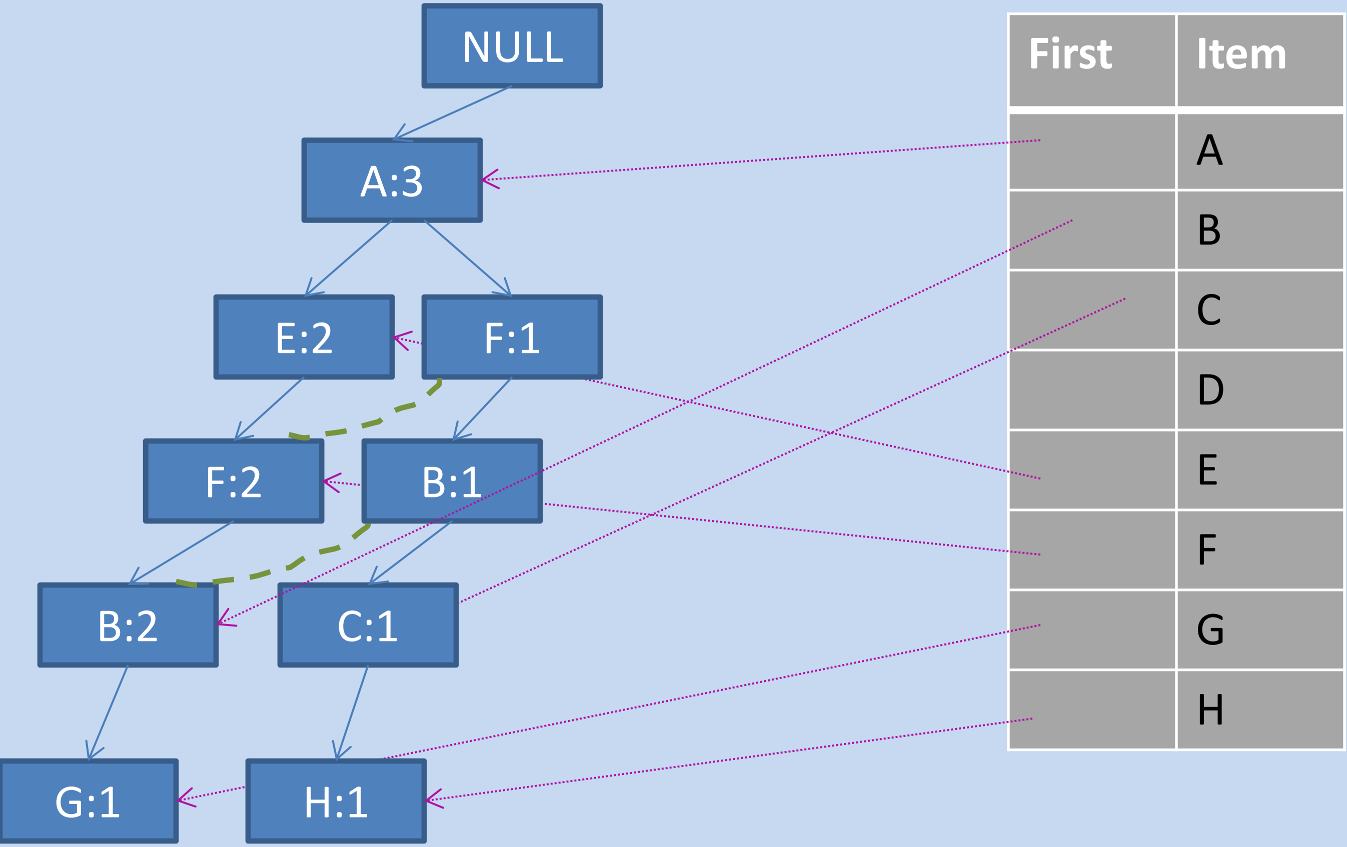


First	Item
	A
	B
	C
	D
	E
	F
	G
	H

Trans	Items
1	A,E,F,B,D
2	A,E,F,B,G
3	A,F,B,C,H
4	E,C,G,H,D
5	A,E,F,C
6	A,E,B,H
7	F,B,C,G
8	A,E,F,G,D
9	A,C,H,D
10	E,F,B,G,H

FPT Construction, IV

Step 4. Process transaction 2



First	Item
	A
	B
	C
	D
	E
	F
	G
	H

Trans	Items
1	A,E,F,B,D
2	A,E,F,B,G
3	A,F,B,C,H
4	E,C,G,H,D
5	A,E,F,C
6	A,E,B,H
7	F,B,C,G
8	A,E,F,G,D
9	A,C,H,D
10	E,F,B,G,H

Example Continued

Build rest of FP-tree in class on board.

Key Properties of FP-tree

- *What is the max height of an FP-tree?* Bounded above by the max number of items in a transaction that survive the support requirement.
- *What is the max number of nodes in an FP-tree?* Bounded above by the sum of the items in each transaction.

FP-Growth: Finding the Frequent Itemsets

- The algorithm follows three steps.
  - *Step 1:* Here, we look at the *Itemset* table. For each item, we construct a *conditional pattern base (CPB).*
  - *Step 2:* We build a tree using the CPB above. This is called the *conditional pattern tree.*
  - *Step 3:* We analyze this tree.

FP-Growth: Step 1

- We create a table, starting from the “bottom” of the *Itemset* table (after sorting in descending order of counts and ignoring any items that don’t meet support criteria).

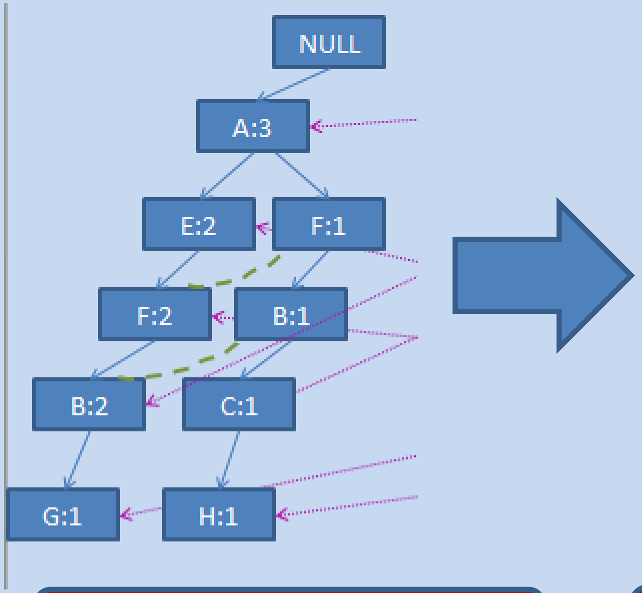
Item	Count
A	7
B	6
C	5
D	4
E	7
F	7
G	5
H	5

Item	Count
A	7
E	7
F	7
B	6
C	5
G	5
H	5
~~D~~	~~4~~

FP-Growth: Step 1

- So in this case, we start with H.
- We create a new table which shows all paths from Root to H (not incl. H) and the count of H

Item	Count
A	7
E	7
F	7
B	6
C	5
G	5
H	5
~~D~~	~~4~~



Incomplete FP-tree

Pattern bases

Node	Paths
H	AFBC:1,…
G	AEFB:1,…
C	AFB:1,…
B	AEF:2,AF:1,…

Incomplete conditional pattern base

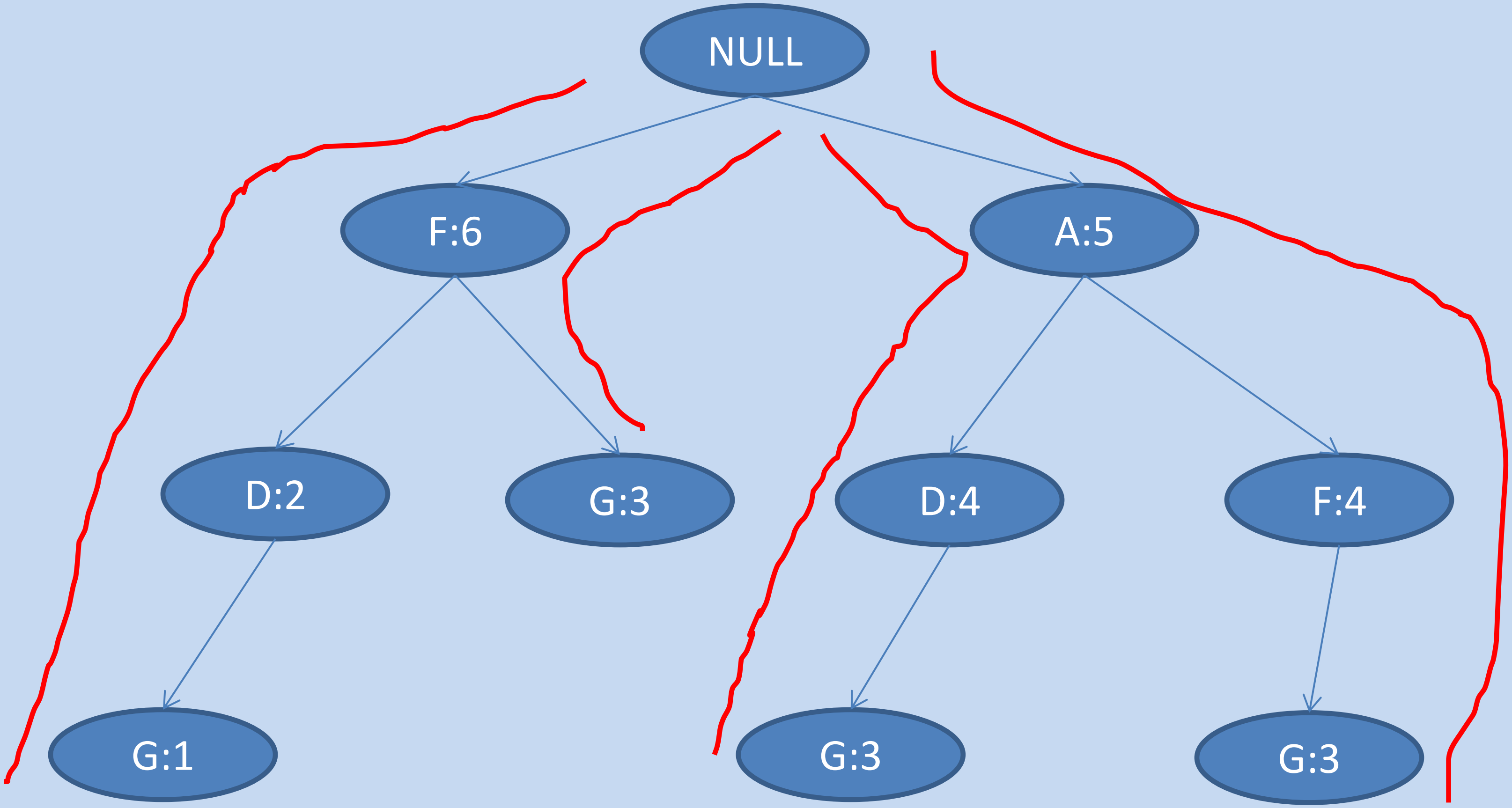
Example Continued

Build rest of Conditional Pattern Base in class on board.

FP-Growth: Step 1

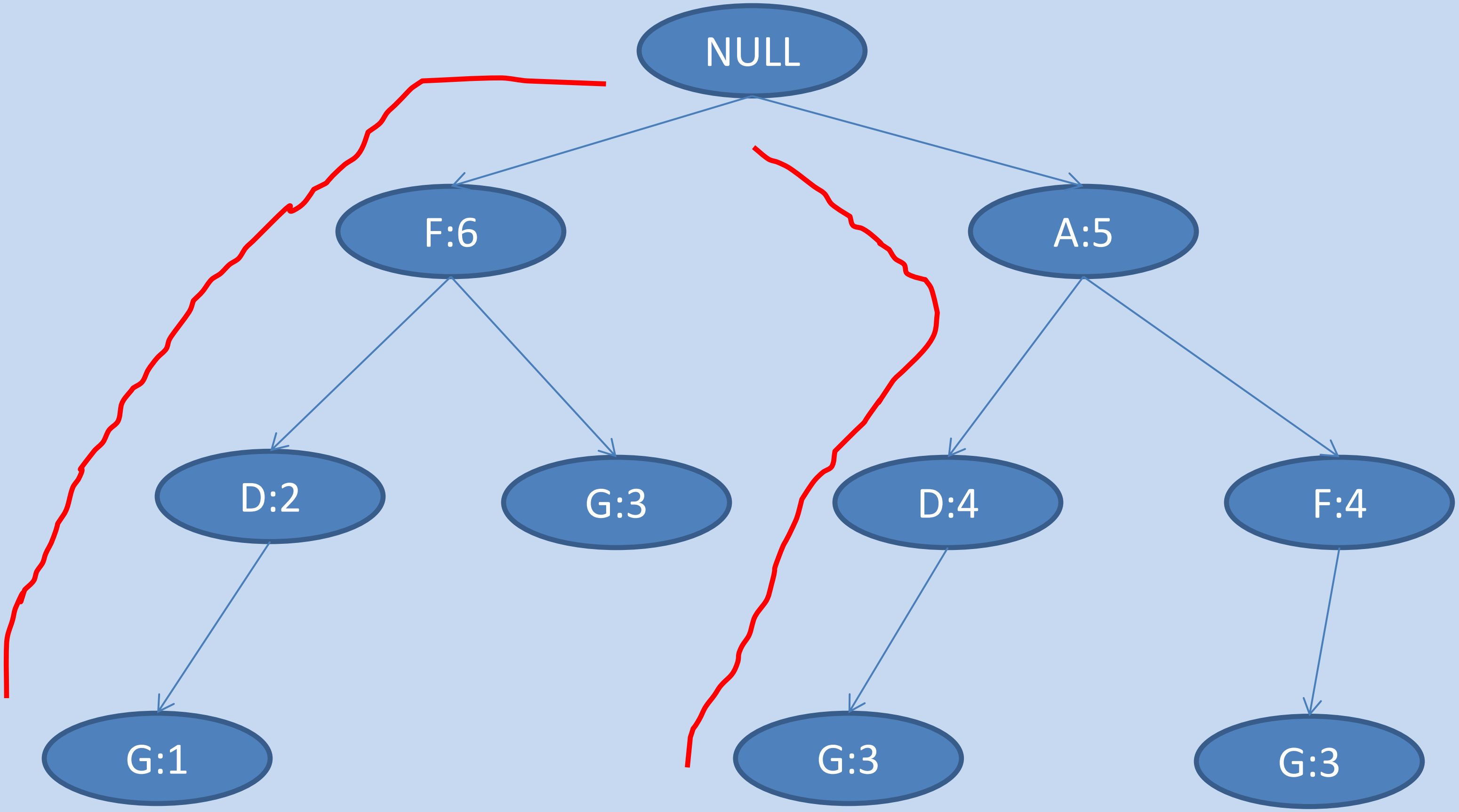
- Extract frequent itemsets, starting from the leaves of the FP-tree and working its way up.
- For each node N:
  - Use Itemsets Table to find places where N occurs. Work your way up.
  - Build a “prefix path subtree”
- Later, we can use the prefix path subtree to extract paths ending in suffixes, not just ending in a node.

Example: Prefix Path Subtree for G



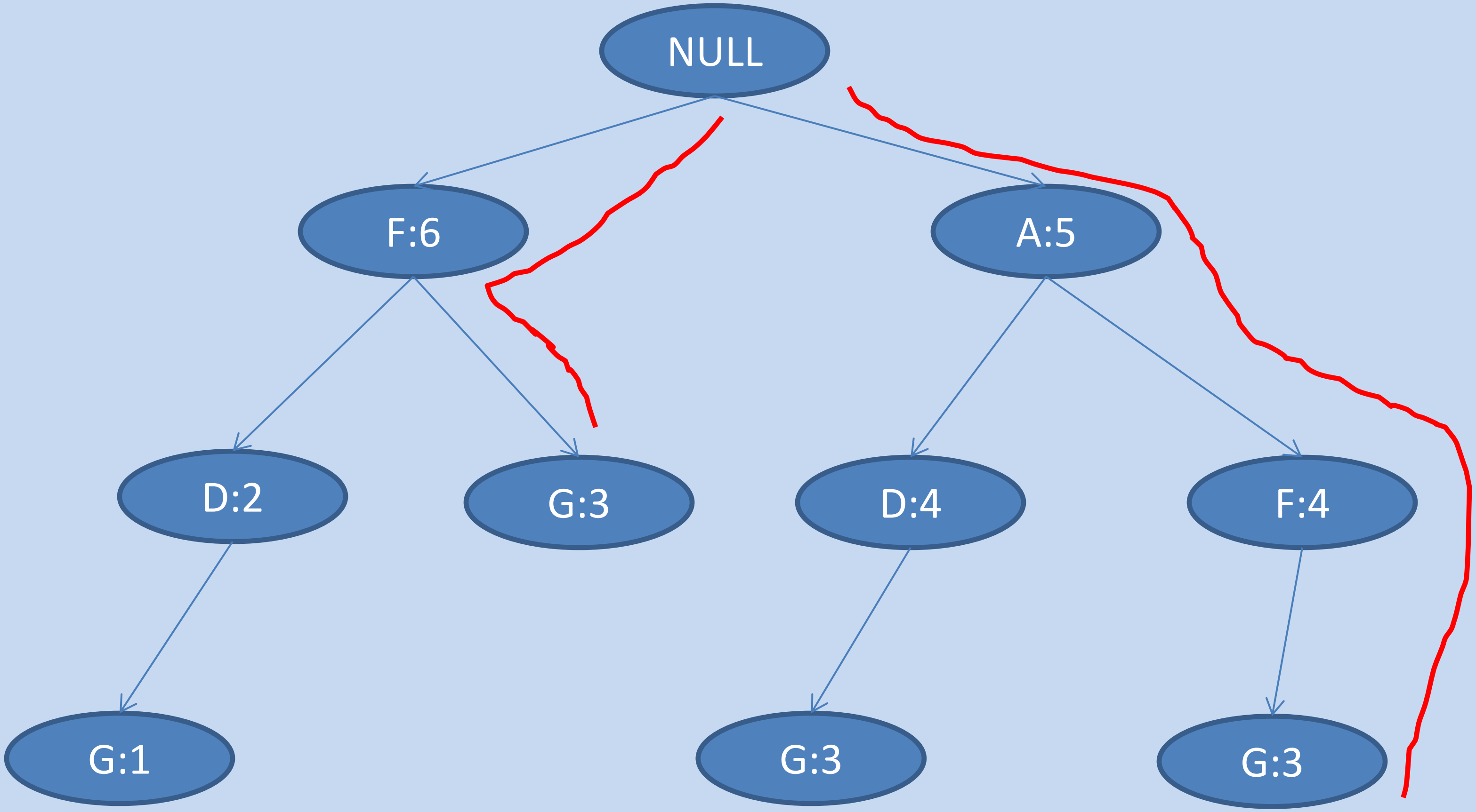
These are all paths ending in G. Clearly, any paths ending in PG must be drawn from these for different pre-fixes I.

Example: Prefix Path Subtree for DG



These are all paths ending in G. Clearly, any paths ending in PG must be drawn from these for different pre-fixes I.

Example: Prefix Path Subtree for FG

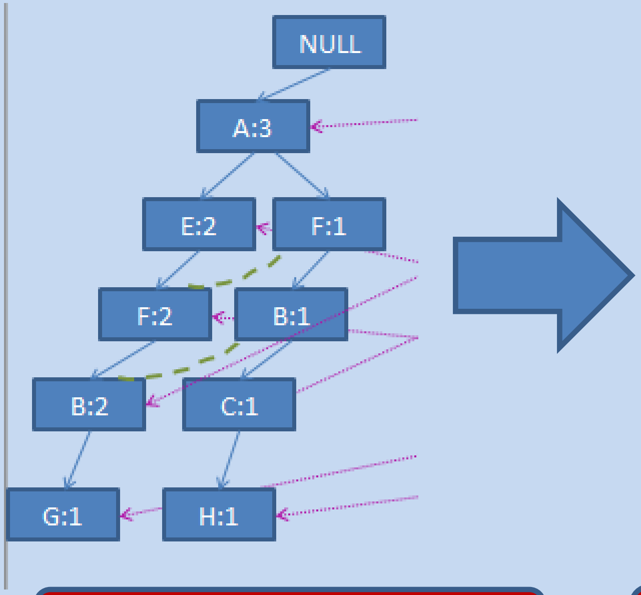


These are all paths ending in G. Clearly, any paths ending in PG must be drawn from these for different pre-fixes I.

Conditional FP-Tree: Step 2

- Consider this situation again.

Item	Count
A	7
E	7
F	7
B	6
C	5
G	5
H	5
~~D~~	~~4~~



Incomplete FP-tree

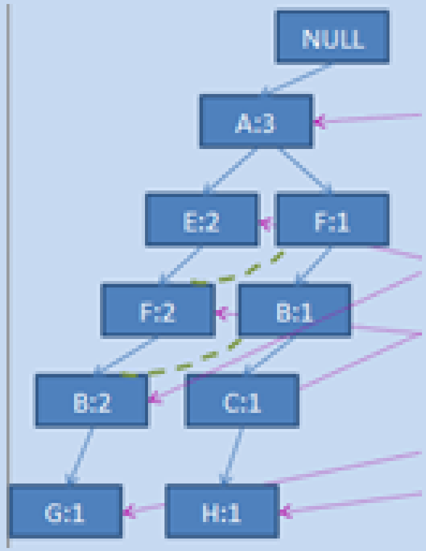
Pattern bases

Node	Paths
H	AFBC:1,…
G	AEFB:1,…
C	AFB:1,…
B	AEF:2,AF:1,..

Incomplete conditional pattern base

Conditional FP-Tree: Step 2

- Condition FP-tree for H:
- A:1->F:1->B:1->C:1



Item	Count	Node	Paths
A	7	H	AFBC:1,…
E	7	G	AEFB:1,…
F	7	C	AFB:1,…
B	6	B	AEF:2,AF:1,…
C	5		
G	5		
H	5		

Reference

Also consult slides from: Mining Frequent Patterns without Candidate Generation

http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCQQFjAA&url=http%3A%2F%2Fwww.cs.uvm.edu%2F~xwu%2Fkdd%2FFPTree-09.ppt&ei=SaAMU8G-J6S6yAG78oHoDw&usg=AFQjCNF_RmxJN4MYEfpeNnR76dbAJyYb5A&bvm=bv.61725948,d.cWc&cad=rja