A Pattern Growth Approach

FP-growth: Frequent Pattern-Growth

- Adopts a divide and conquer strategy
- Compress the database representing frequent items into a **frequent –pattern tree** or **FP-tree**
  - → Retains the itemset association information
- Divide the compressed database into a set of conditional databases, each associated with one frequent item
- Mine each such databases separately

Example: FP-growth

- The first scan of data is the same as Apriori
- Derive the set of frequent 1-itemsets

Item ID	Support count
I1	6
I2	7
I3	6
I4	2
I5	2

- Let min-sup=2
- Generate a set of ordered items (apply condition (min-sup=2) & write in descending order)

Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2

Transactional Database

TID	List of item IDS
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

Construct the FP-Tree

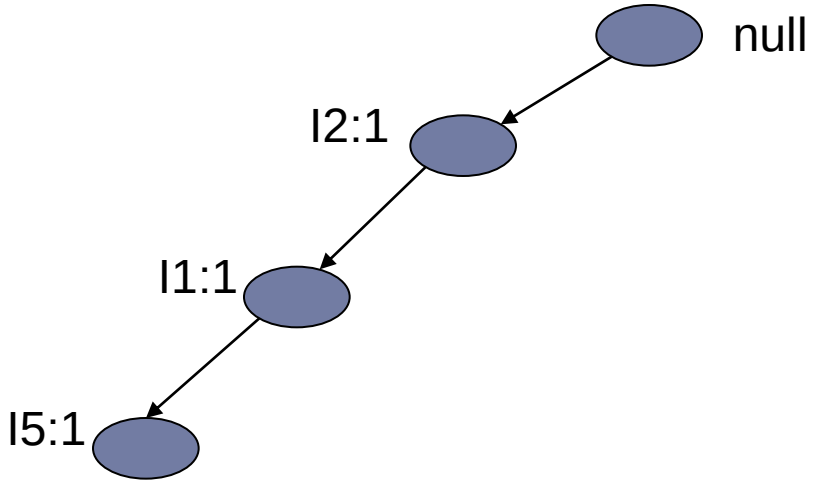
Transactional Database

TID	Items	TID	Items	TID	Items
T100	I1,I2,I5	T400	I1,I2,I4	T700	I1,I3
T200	I2,I4	T500	I1,I3	T800	I1,I2,I3,I5
T300	I2,I3	T600	I2,I3	T900	I1,I2,I3

- Create a branch for each transaction
- Items in each transaction are processed in order

1- Order the items T100: {I2,I1,I5}
2- Construct the first branch: <I2:1>, <I1:1>,<I5:1>

Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2



Construct the FP-Tree

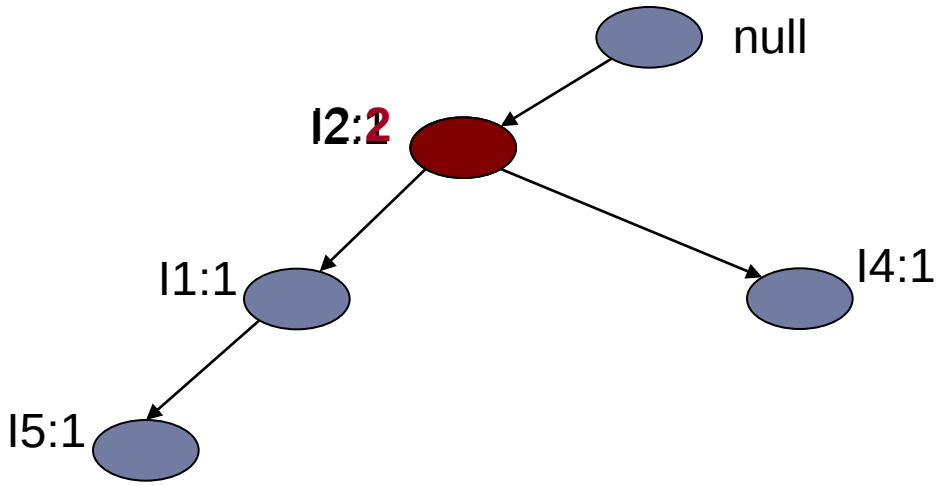
Transactional Database

TID	Items	TID	Items	TID	Items
T100	I1,I2,I5	T400	I1,I2,I4	T700	I1,I3
T200	I2,I4	T500	I1,I3	T800	I1,I2,I3,I5
T300	I2,I3	T600	I2,I3	T900	I1,I2,I3

- Create a branch for each transaction
- Items in each transaction are processed in order

1- Order the items T200: {I2,I4}
2- Construct the second branch: <I2:1>, <I4:1>

Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2



Construct the FP-Tree

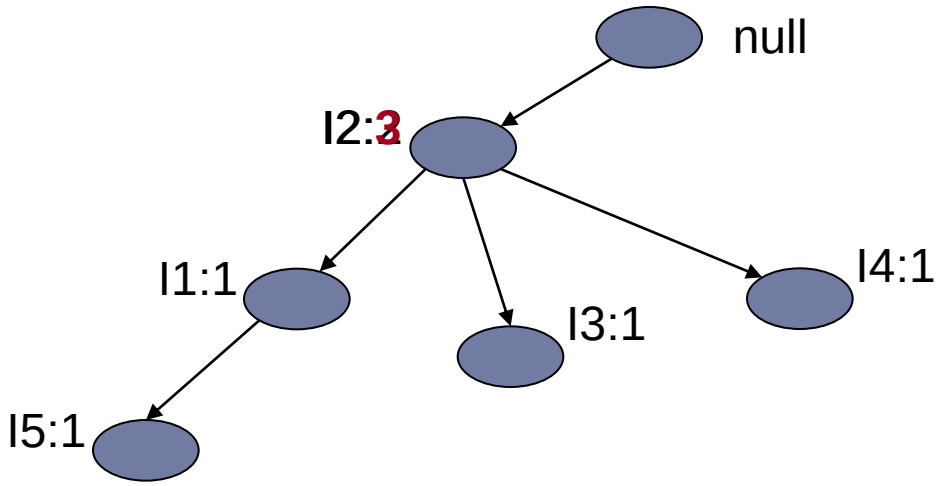
Transactional Database

TID	Items	TID	Items	TID	Items
T100	I1,I2,I5	T400	I1,I2,I4	T700	I1,I3
T200	I2,I4	T500	I1,I3	T800	I1,I2,I3,I5
T300	I2,I3	T600	I2,I3	T900	I1,I2,I3

- Create a branch for each transaction
- Items in each transaction are processed in order

1- Order the items T300: {I2,I3}
2- Construct the third branch: <I2:2>, <I3:1>

Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2



Construct the FP-Tree

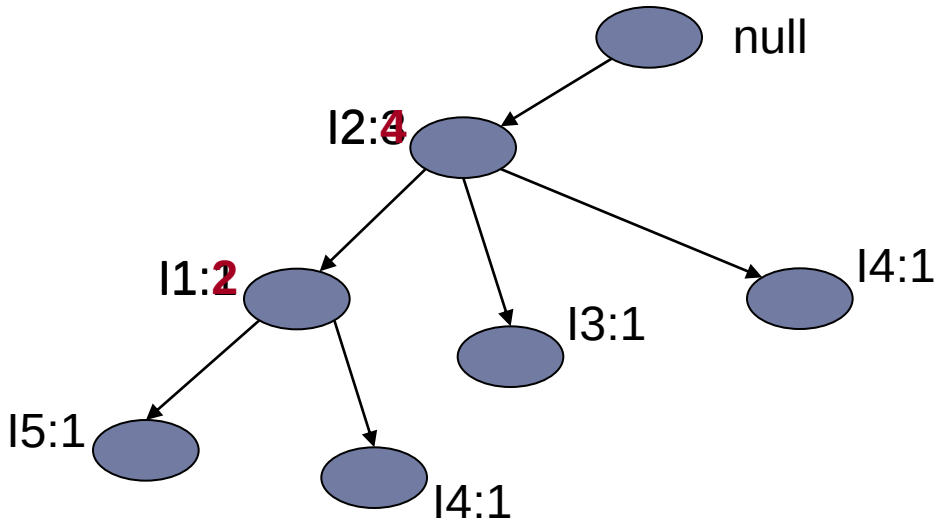
Transactional Database

TID	Items	TID	Items	TID	Items
T100	I1,I2,I5	T400	I1,I2,I4	T700	I1,I3
T200	I2,I4	T500	I1,I3	T800	I1,I2,I3,I5
T300	I2,I3	T600	I2,I3	T900	I1,I2,I3

- Create a branch for each transaction
- Items in each transaction are processed in order

1- Order the items T400: {I2,I1,I4}
2- Construct the fourth branch: <I2:3>, <I1:1>,<I4:1>

Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2



Construct the FP-Tree

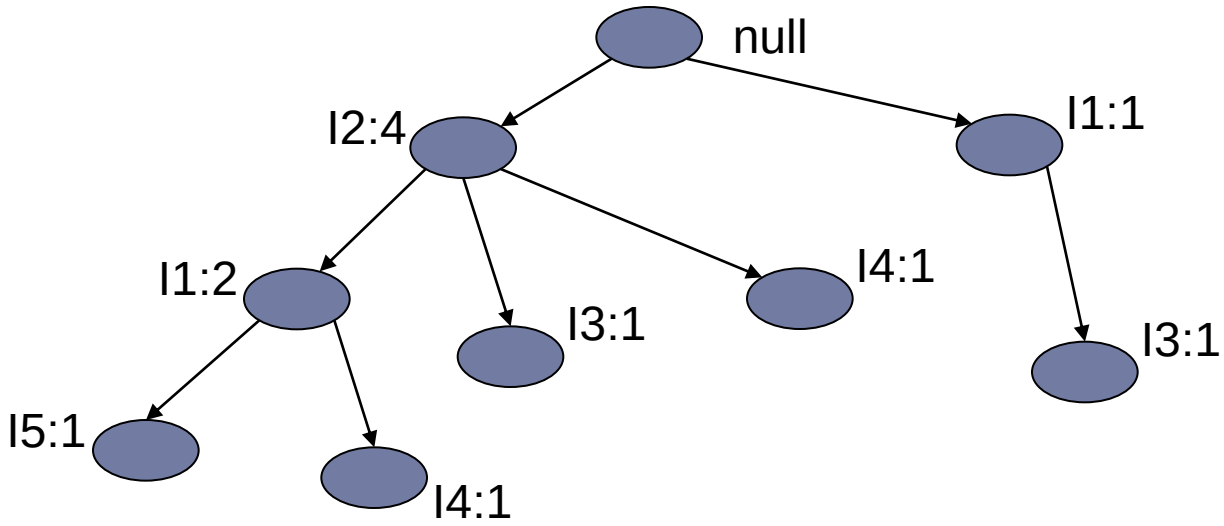
Transactional Database

TID	Items	TID	Items	TID	Items
T100	I1,I2,I5	T400	I1,I2,I4	T700	I1,I3
T200	I2,I4	T500	I1,I3	T800	I1,I2,I3,I5
T300	I2,I3	T600	I2,I3	T900	I1,I2,I3

- Create a branch for each transaction
- Items in each transaction are processed in order

1- Order the items T400: {I1,I3}
2- Construct the fifth branch: <I1:1>, <I3:1>

Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2

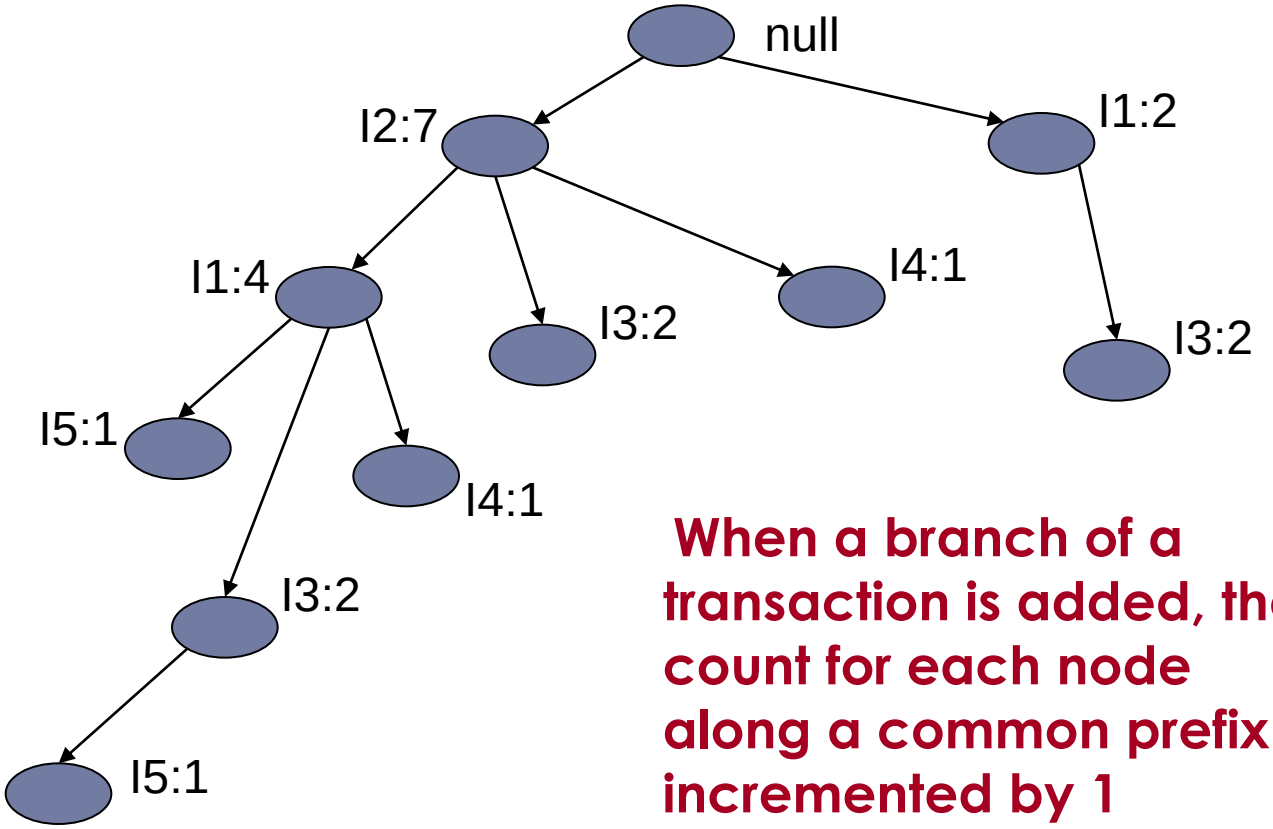


Construct the FP-Tree

Transactional Database

TID	Items	TID	Items	TID	Items
T100	I1,I2,I5	T400	I1,I2,I4	T700	I1,I3
T200	I2,I4	T500	I1,I3	T800	I1,I2,I3,I5
T300	I2,I3	T600	I2,I3	T900	I1,I2,I3

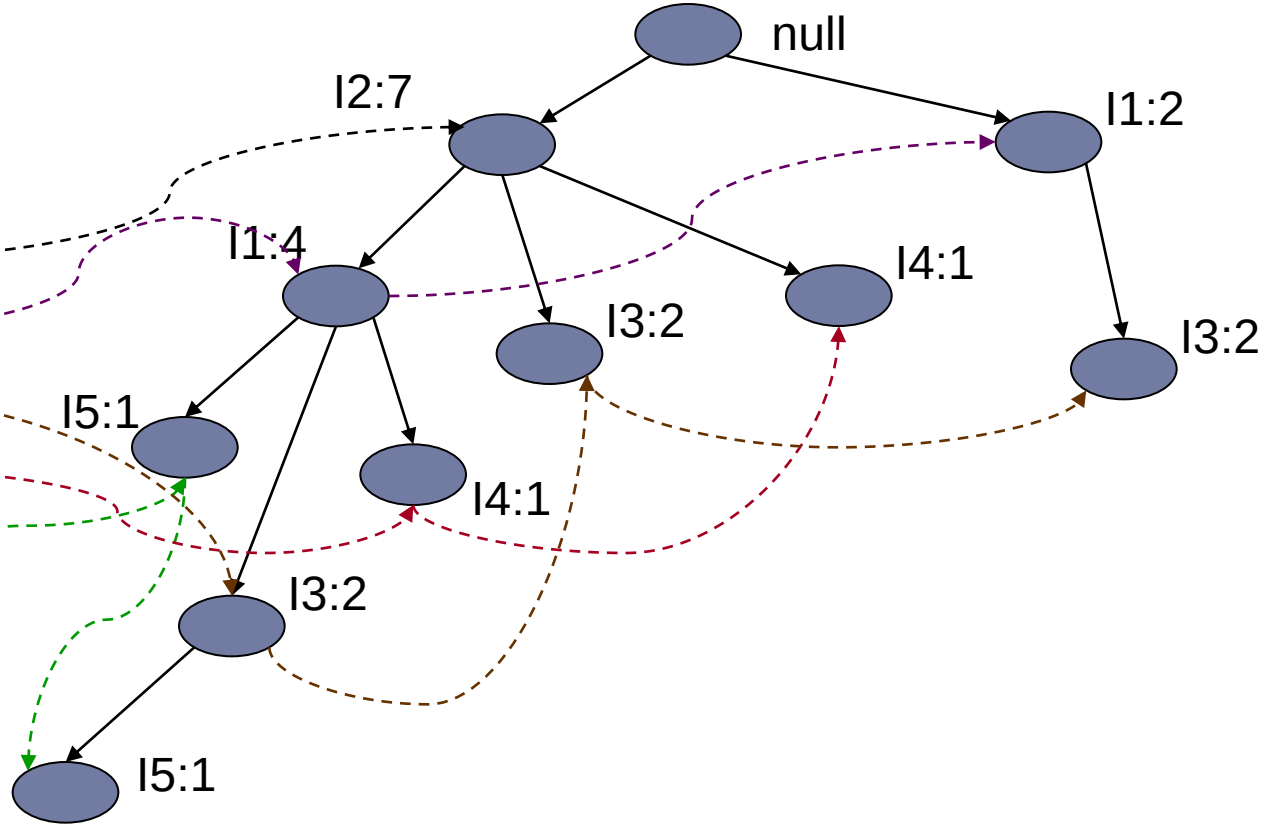
Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2



When a branch of a transaction is added, the count for each node along a common prefix is incremented by 1

Construct the FP-Tree

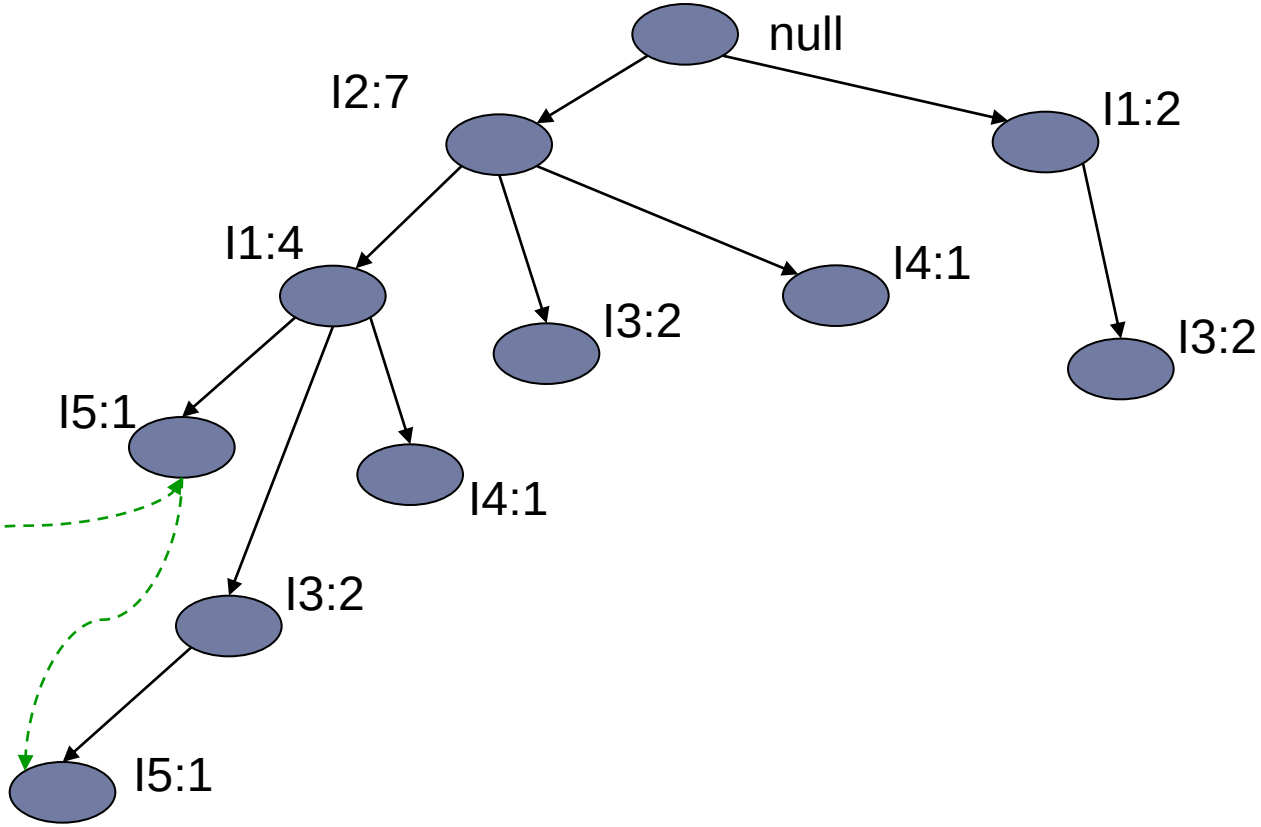
Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2



The problem of mining frequent patterns in databases is transformed to that of mining the FP-tree

Construct the FP-Tree

Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2



-Occurrences of I5: <I2,I1,I5> and <I2,I1,I3,I5>

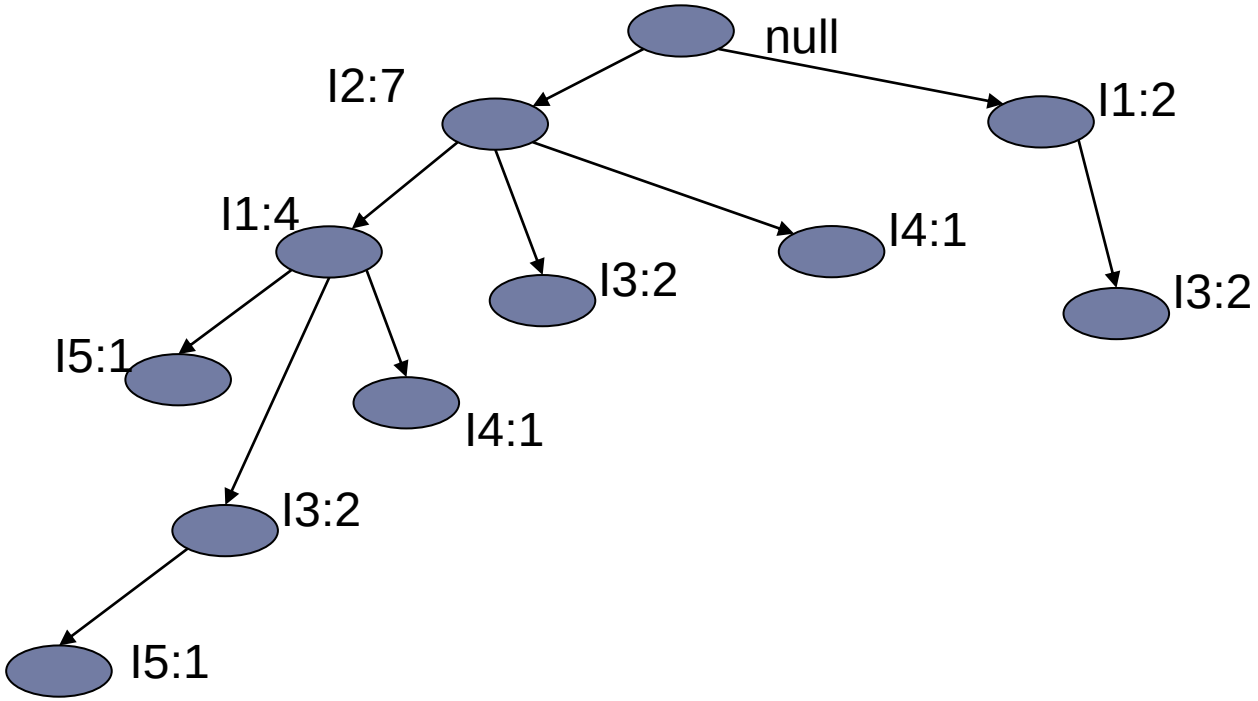
-Two prefix Paths <I2, I1: 1> and <I2,I1,I3: 1>

-Conditional FP tree contains only <I2: 2, I1: 2>, I3 is not considered because its support count of 1 is less than the minimum support count.

-Frequent patterns {I2,I5:2}, {I1,I5:2},{I2,I1,I5:2}

Construct the FP-Tree

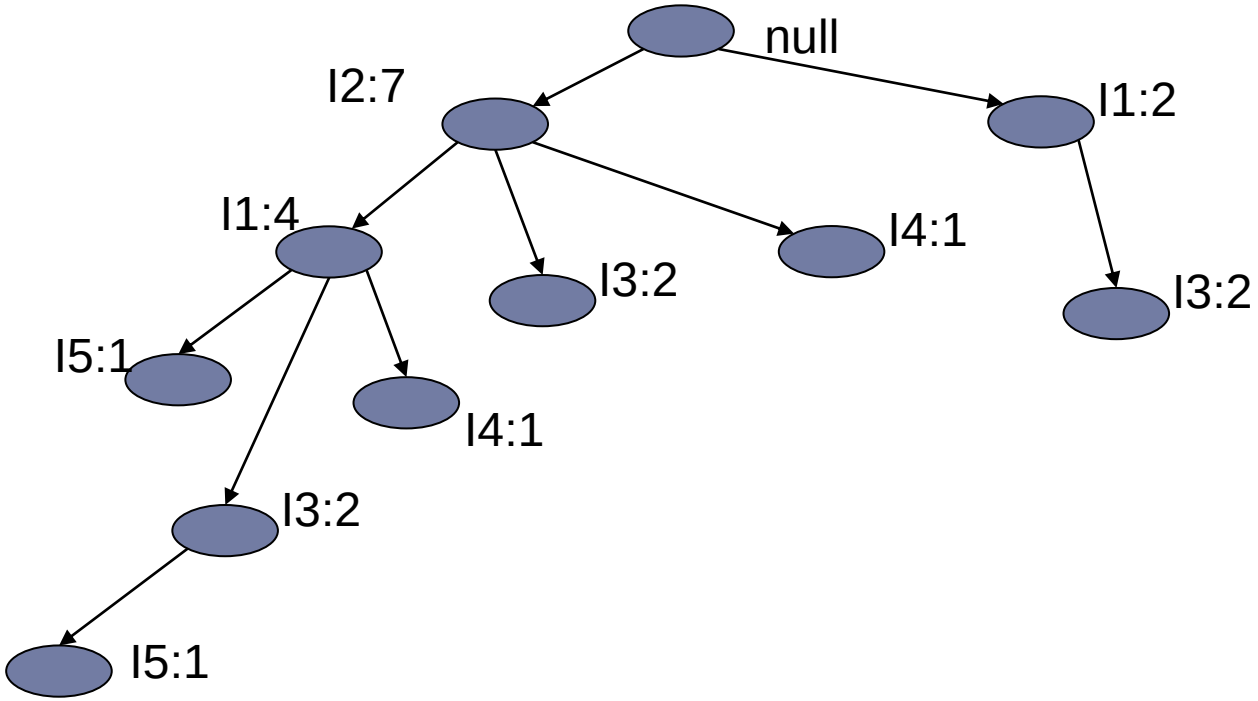
Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2



Item ID	Conditional Pattern Base	Conditional FP-tree
I5	{{I2,I1:1},{I2,I1,I3:1}}	<I2:2,I1:2>
I4	{{I2,I1:1},{I2:1}}	<I2:2>
I3	{{I2,I1:2},{I2:2}, {I1:2}}	<I2:4,I1:2>,<I1:2>
I1	{I2:4}	<I2:4>

Construct the FP-Tree

Item ID	Support count
I2	7
I1	6
I3	6
I4	2
I5	2



TID	Conditional FP-tree	Frequent Patterns Generated
I5	<I2:2,I1:2>	{I2,I5:2}, {I1,I5:2},{I2,I1,I5:2}
I4	<I2:2>	{I2,I4:2}
I3	<I2:4,I1:2>,<I1:2>	{I2,I3:4},{I1,I3:4},{I2,I1,I3:2}
I1	<I2:4>	{I2,I1:4}

FP-growth properties

- FP-growth transforms the problem of finding long frequent patterns to searching for shorter ones recursively and concatenating the suffix
- It uses the least frequent suffix offering a good selectivity
- It reduces the search cost
- If the tree does not fit into main memory, partition the database
- Efficient and scalable for mining both long and short frequent patterns

Generating Association Rules

- Once the frequent itemsets have been found, it is straightforward to generate **strong** association rules that satisfy:

  → **minimum** support
  → **minimum** confidence

- Relation between support and confidence:

$$\text{Confidence}(A \Rightarrow B) = P(B \mid A) = \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}$$

  → **Support_count(A∪B)** is the number of transactions containing the itemsets A ∪ B
  → **Support_count(A)** is the number of transactions containing the itemset A.

Generating Association Rules

- For each frequent itemset **L**, generate all non empty subsets of **L**
- *For* every non empty subset **S** of **L**, output the rule:

$$S \Rightarrow (L-S)$$

If (support_count(L)/support_count(S)) >= min_conf

(or) Confidence

Example

- Suppose the frequent Itemset **L={I1,I2,I5}**
- Subsets of L are: (i.e.)

S = {I1,I2}, {I1,I5},{I2,I5},{I1},{I2},{I5}

S ⇒ (L-S)

- Association rules :

I1 ∧ I2 ⇒ I5	confidence = 2/4= 50%
I1 ∧ I5 ⇒ I2	confidence=2/2=100%
I2 ∧ I5 ⇒ I1	confidence=2/2=100%
I1 ⇒ I2 ∧ I5	confidence=2/6=33%
I2 ⇒ I1 ∧ I5	confidence=2/7=29%
I5 ⇒ I2 ∧ I2	confidence=2/2=100%

If the minimum confidence =70%

Transactional Database

TID	List of item IDS
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

If support_count(L)/support_count(S)) >= min_conf

support_count(L) = 2

support_count(S) = 4

2/4 =0.5 (or) 50% this value should be >=70% (not satisfying)