# Frequent itemsets: FP-growth

Lecture 15

# Main ideas

- Compressed index of transactions – Frequent Pattern tree – FP-tree
- Frequent patterns are extracted from FP-tree recursively – by projections for each item

# FP-tree construction

1. Scan DB, count C1, produce F1
2. Sort items in F1 in decreasing order of support counts. Create indexing header for this sorted list
3. Second DB scan, sort frequent itemsets in each transaction in order corresponding to the header, map each transaction to a path in FP-tree, updating counts of items encountered on this path
4. Preserve links for the same item from the header table to all occurrences of this item in different paths

# 2. Header table

| 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} |

Header

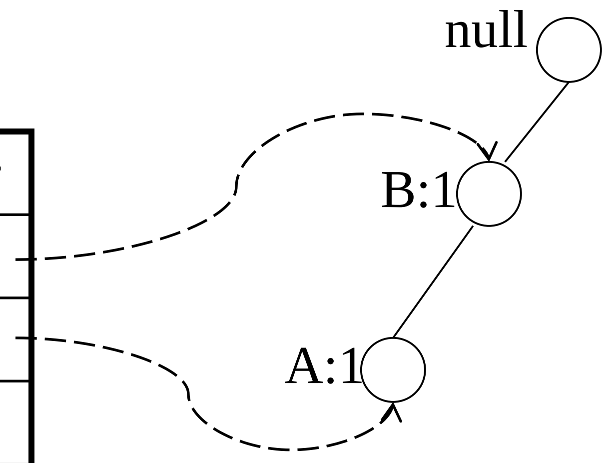
| | |
|---|---|
| B | 8 |
| A | 7 |
| C | 7 |
| D | 5 |
| E | 3 |

Decreasing order of support counts

# Inserting transactions: TID 1

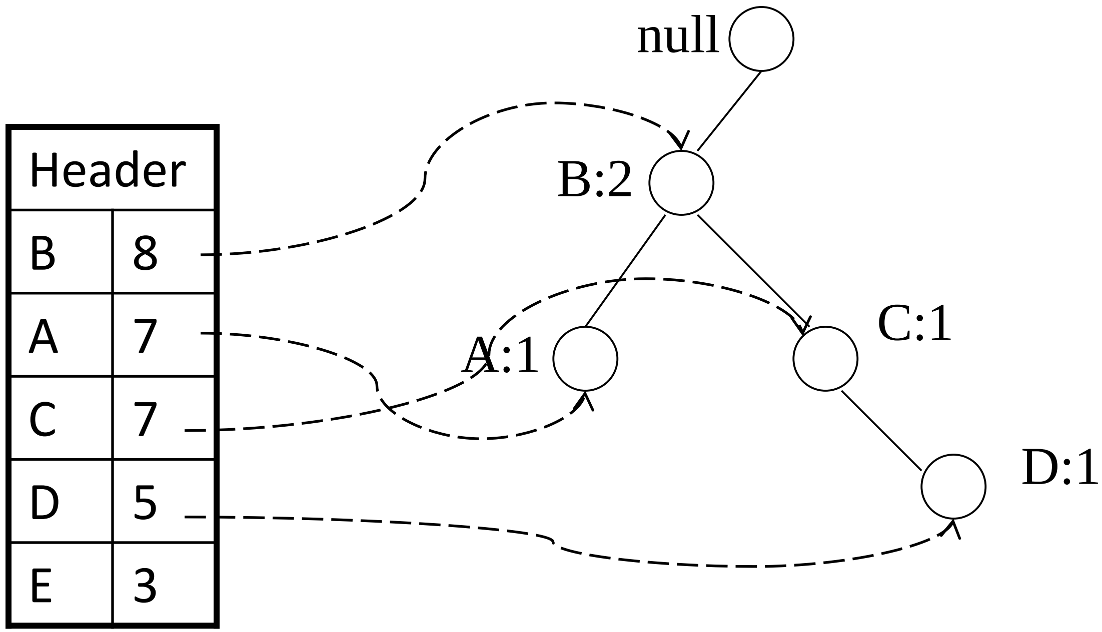
| TID | Items |
|---|---|
| 1 | {A,B} |
| 2 | {B,C,D} |
| 3 | {A,C,D,E} |
| 4 | {A,D,E} |
| 5 | {A,B,C} |
| 6 | {A,B,C,D} |
| 7 | {B,C} |
| 8 | {A,B,C} |
| 9 | {A,B,D} |
| 10 | {B,C,E} |

| Header | |
|---|---|
| B | 8 |
| A | 7 |
| C | 7 |
| D | 5 |
| E | 3 |



# Inserting transactions: TID 2

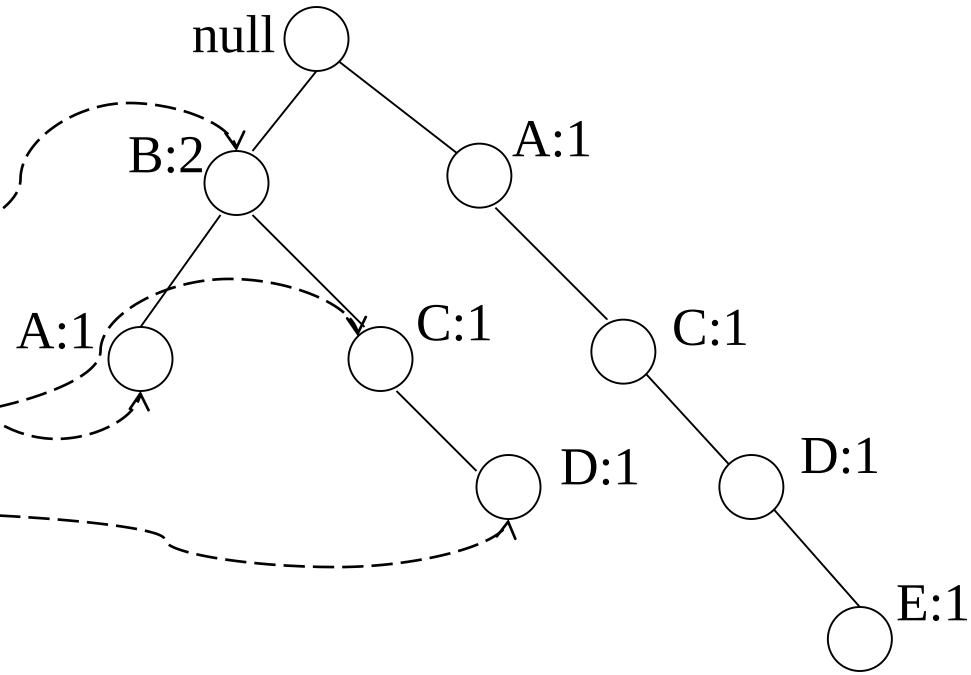
| TID | Items |
|---|---|
| 1 | {A,B} |
| 2 | {B,C,D} |
| 3 | {A,C,D,E} |
| 4 | {A,D,E} |
| 5 | {A,B,C} |
| 6 | {A,B,C,D} |
| 7 | {B,C} |
| 8 | {A,B,C} |
| 9 | {A,B,D} |
| 10 | {B,C,E} |



# Inserting transactions: TID 3

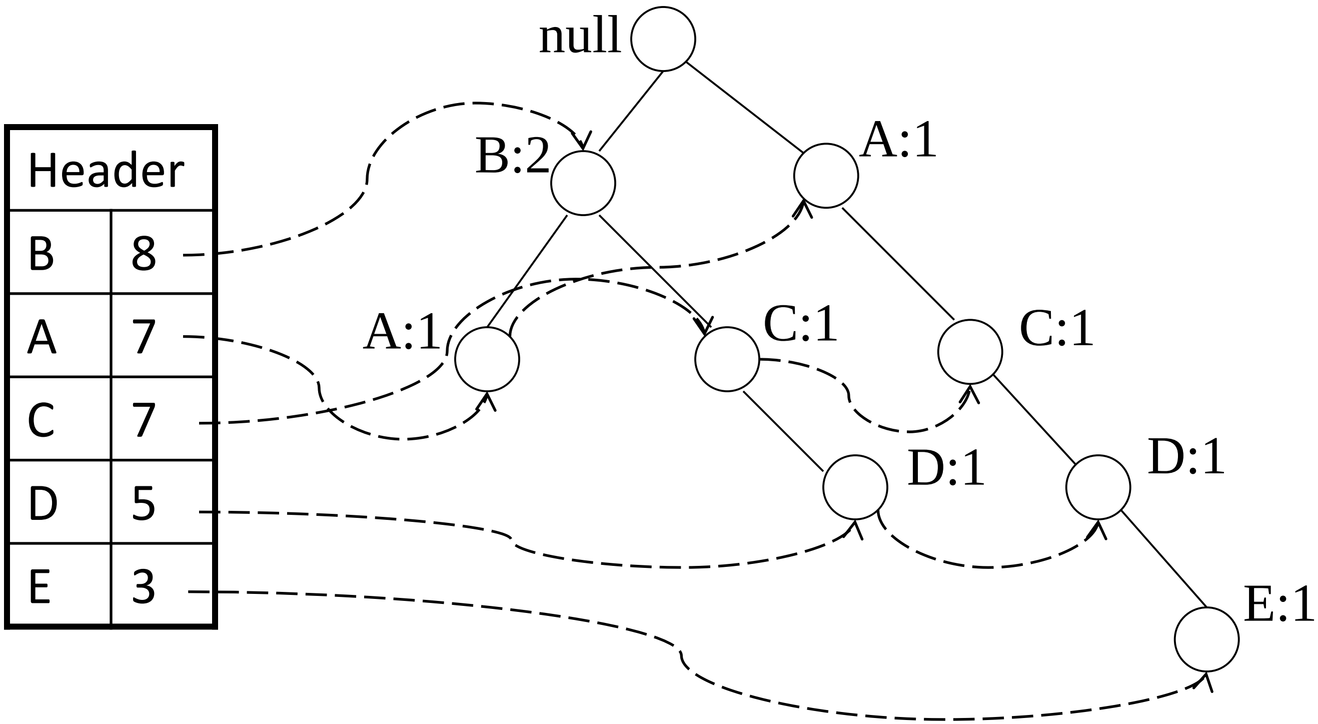
| TID | Items |
|---|---|
| 1 | {A,B} |
| 2 | {B,C,D} |
| 3 | {A,C,D,E} |
| 4 | {A,D,E} |
| 5 | {A,B,C} |
| 6 | {A,B,C,D} |
| 7 | {B,C} |
| 8 | {A,B,C} |
| 9 | {A,B,D} |
| 10 | {B,C,E} |

| Header | |
|---|---|
| B | 8 |
| A | 7 |
| C | 7 |
| D | 5 |
| E | 3 |



# Inserting transactions: TID 3

| 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} |

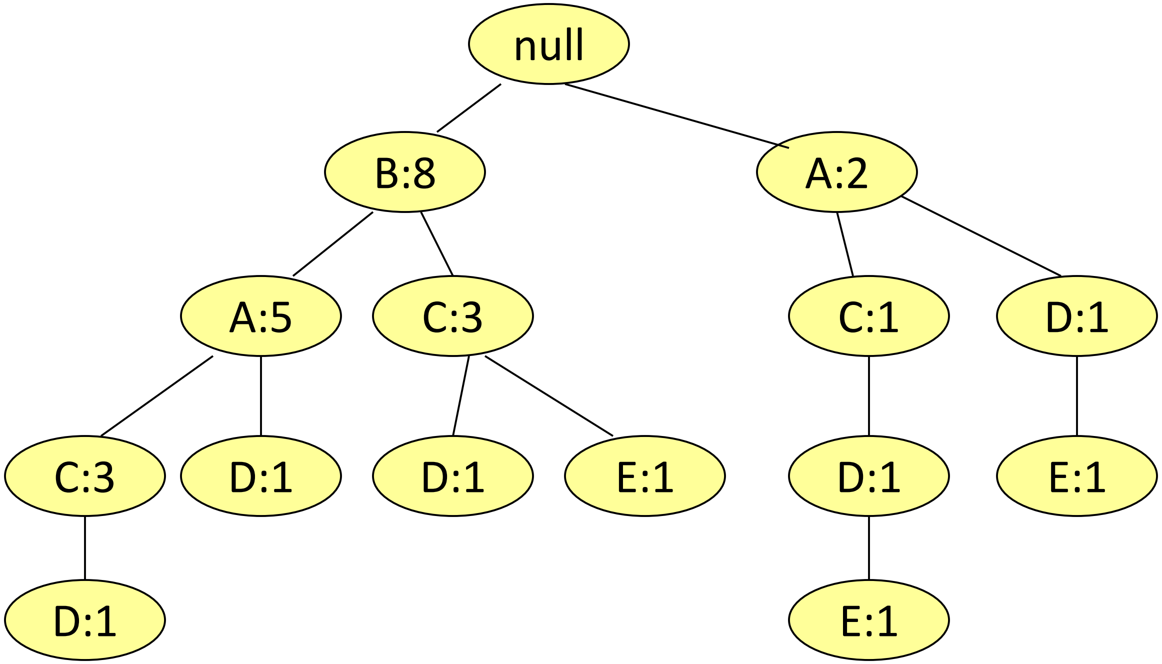



And of course the chain pointers

# Final FP-tree

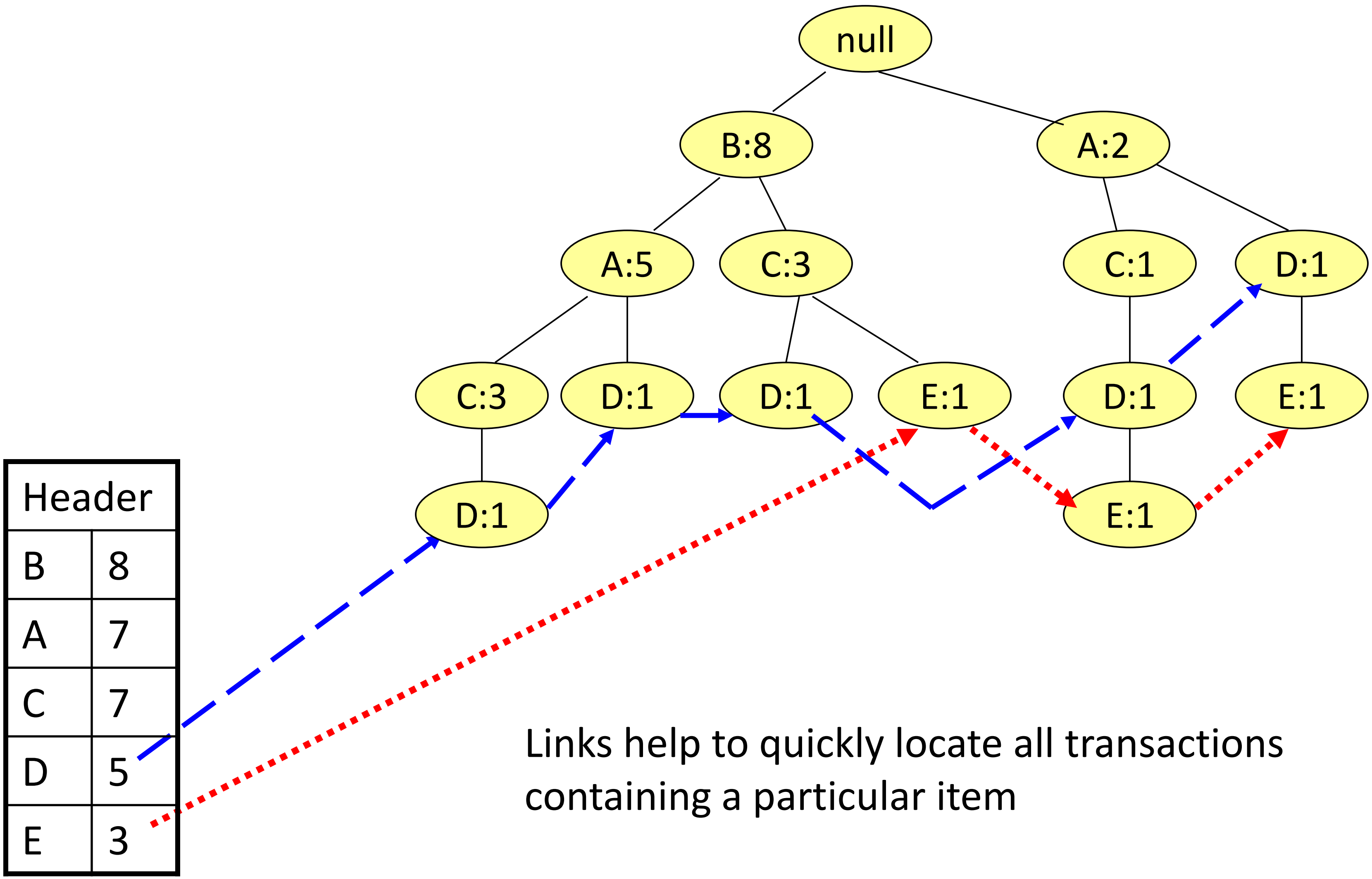
| TID | Items |
|---|---|
| 1 | {A,B} |
| 2 | {B,C,D} |
| 3 | {A,C,D,E} |
| 4 | {A,D,E} |
| 5 | {A,B,C} |
| 6 | {A,B,C,D} |
| 7 | {B,C} |
| 8 | {A,B,C} |
| 9 | {A,B,D} |
| 10 | {B,C,E} |

| Header | |
|---|---|
| B | 8 |
| A | 7 |
| C | 7 |
| D | 5 |
| E | 3 |




At this point, we use FP-tree instead of database

# Final FP-tree




Links help to quickly locate all transactions containing a particular item

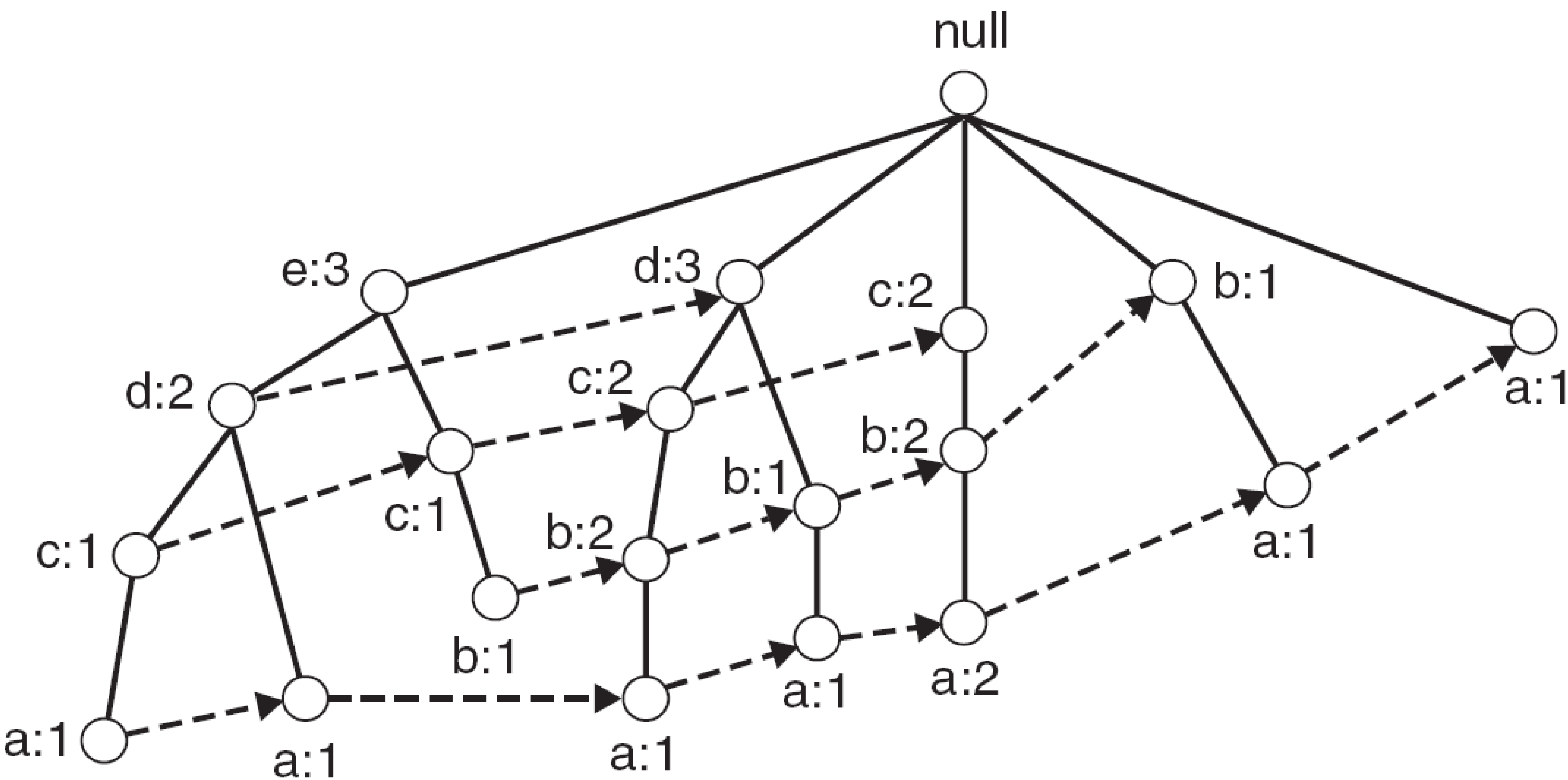
# The size of FP-tree

- Best case: identical items in all transactions – 1 path
- Worst case: no overlapping items in transactions – the tree is as big as the original database

- Normal case: the size is significantly smaller, and in many cases fits into the main memory

# Sorting heuristic

If the sorting is reversed (in ascending order of support counts), then the tree is in most cases much larger

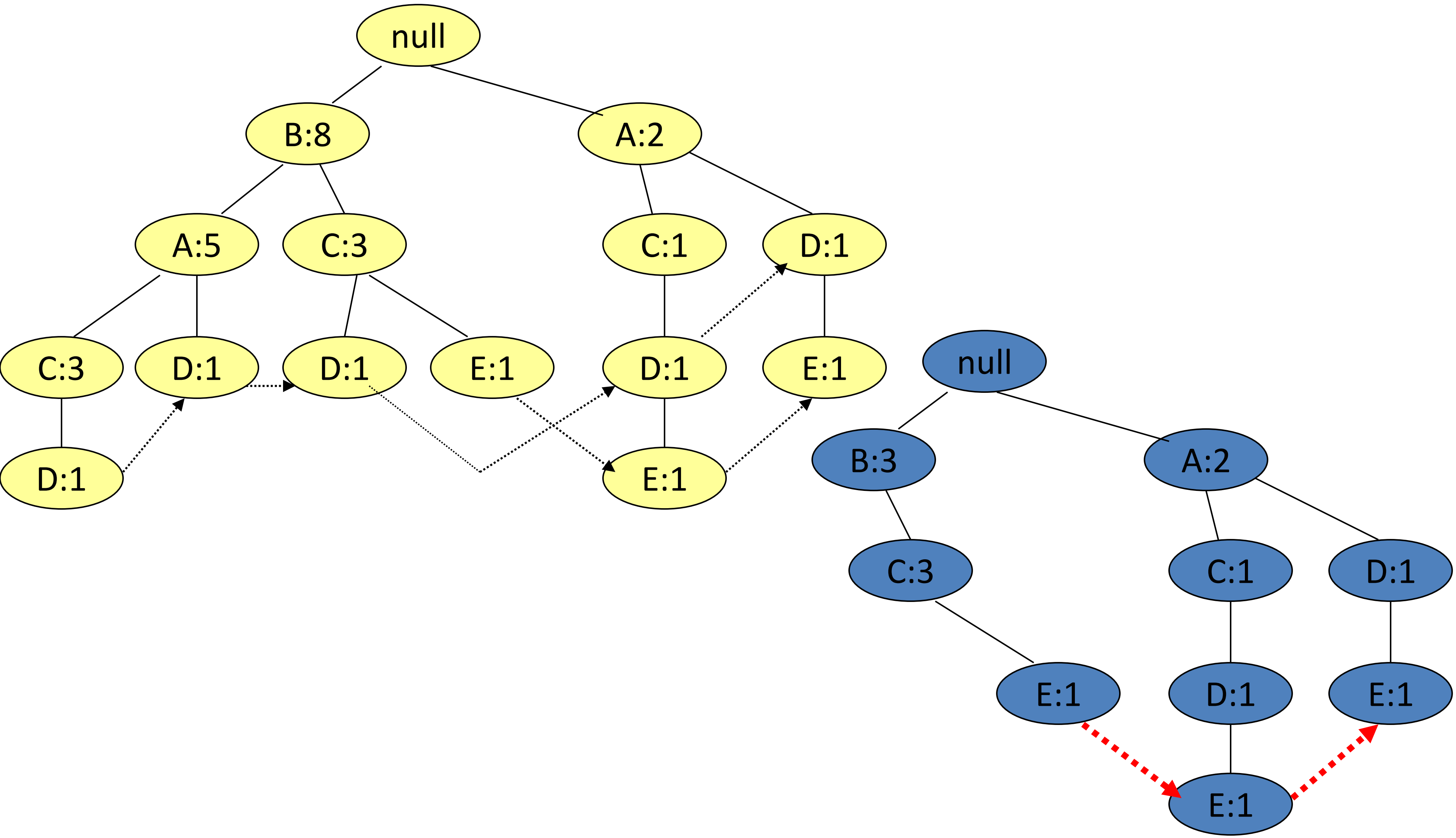


# Mining frequent patterns from FP-tree: FP-growth algorithm

- Strategy: divide-and-conquer: splits the problem into smaller sub-problems
- Finds frequent itemsets ending in particular item by processing all paths ending in E first, then paths ending in D etc.
- To mine frequent itemsets ending in E (with *suffix* E), only the paths associated with E are observed
- The paths are accessed rapidly due to the chain pointers

| Header | |
|---|---|
| B | 8 |
| A | 7 |
| C | 7 |
| D | 5 |
| E | 3 |

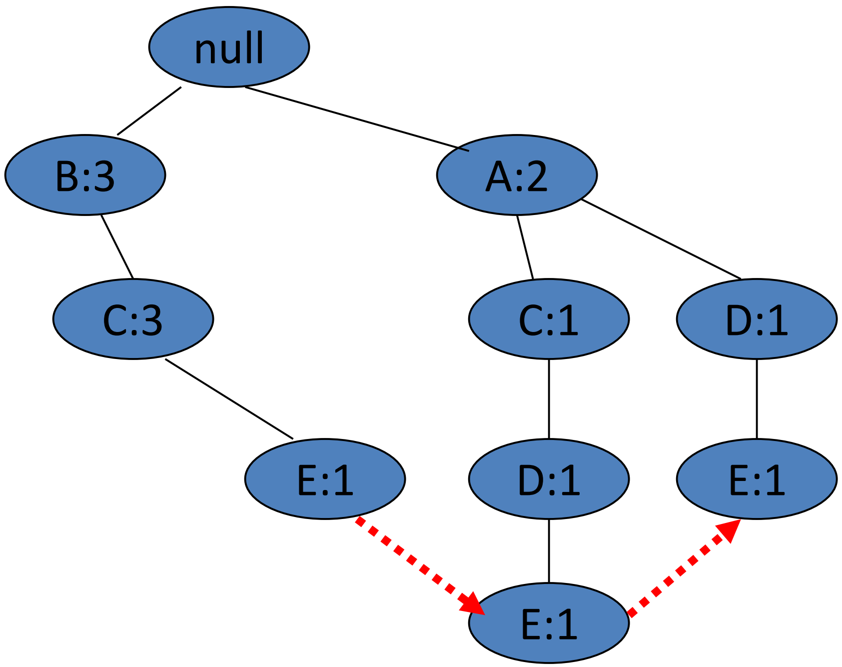
# Paths containing node E



# Conditional FP-tree on E: projection on E

- Using paths containing E, we perform two operations:
  - Collect counts of all 1-itemsets in the projection, build a new header
  - Build a smaller FP-tree by using each path as an input transaction, reading a path bottom-up

# Conditional on E: header table

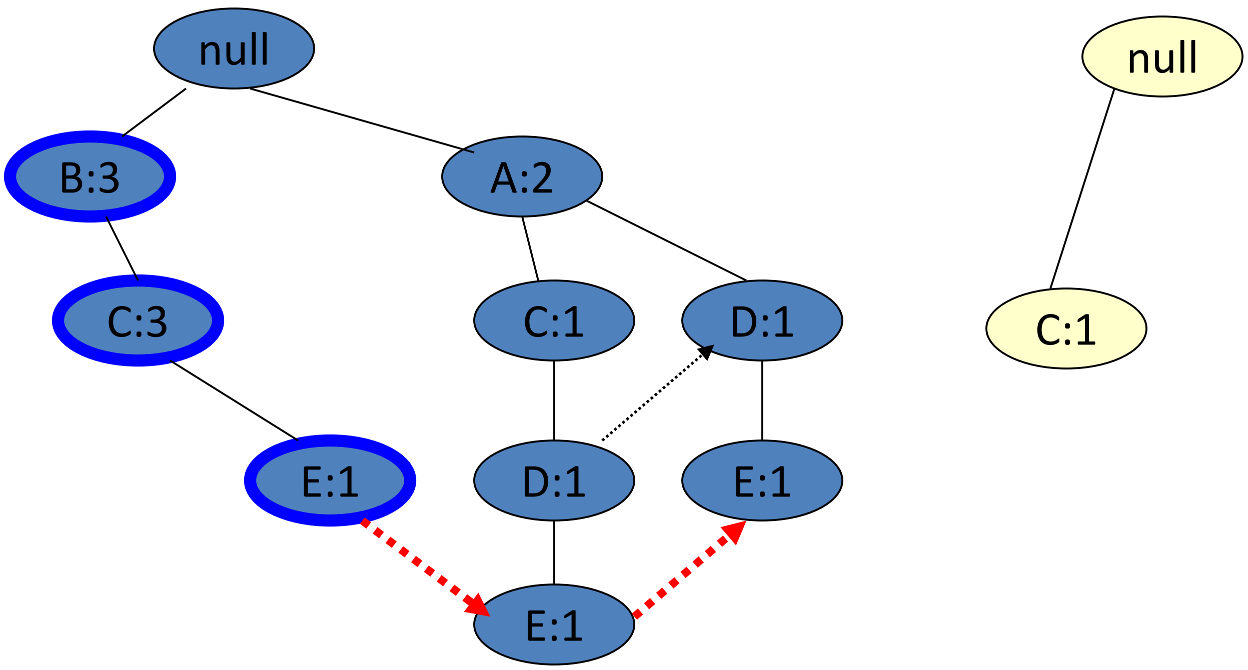



| Header | |
|---|---|
| A | 2 |
| C | 2 |
| D | 2 |
| ~~B~~ | ~~1~~ |

At this point we know that {A,E}, {C,E} and {D,E} are frequent itemsets

Frequent itemsets ending in E: {A,E}, {C,E}, {D,E}

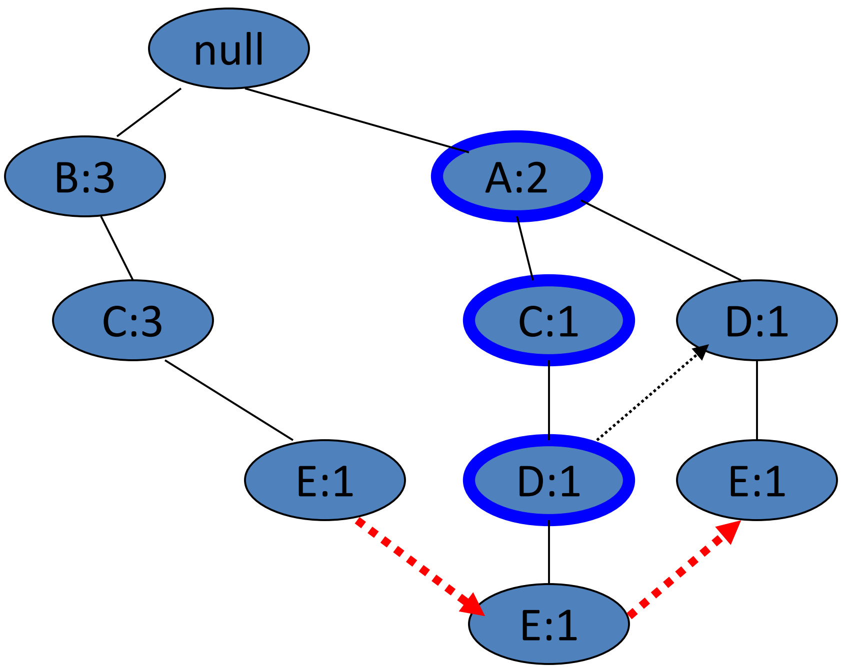
# Conditional on E: insert transactions 1

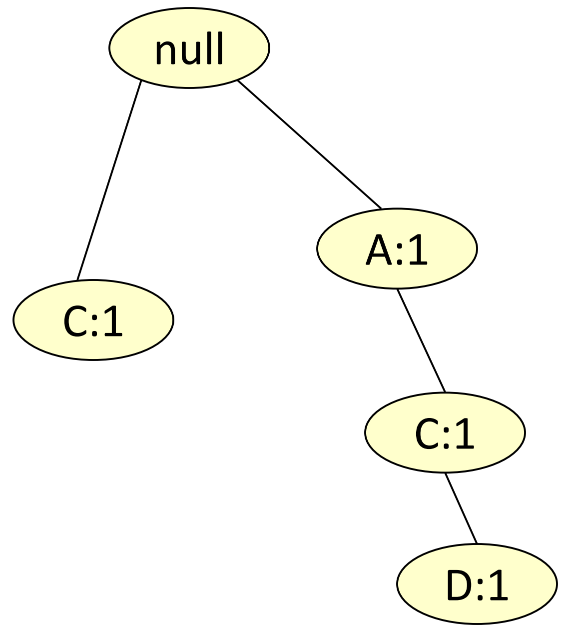



| Header | |
|---|---|
| A | 2 |
| C | 2 |
| D | 2 |

Frequent itemsets ending in E: {A,E}, {C,E}, {D,E}

# Conditional on E: insert transactions 2

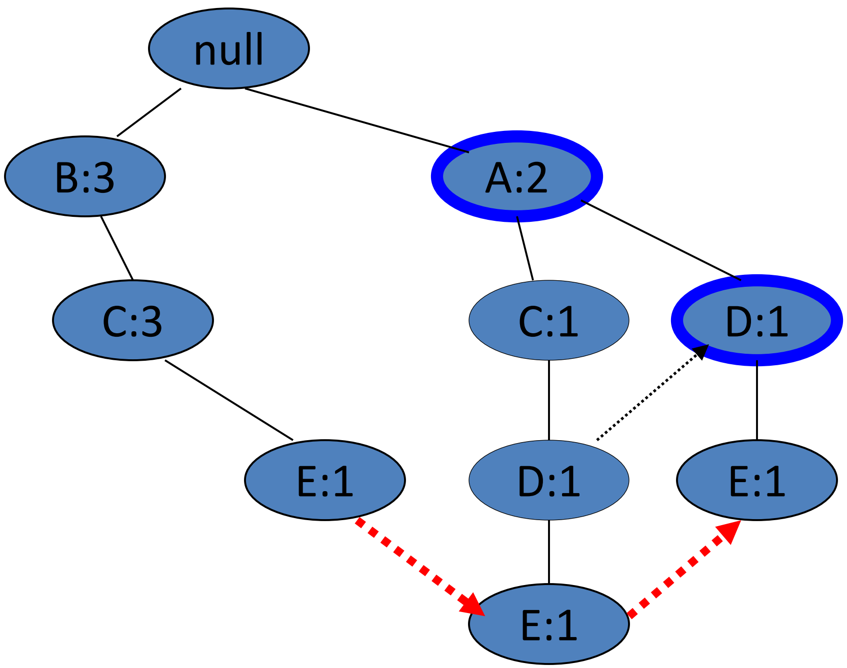


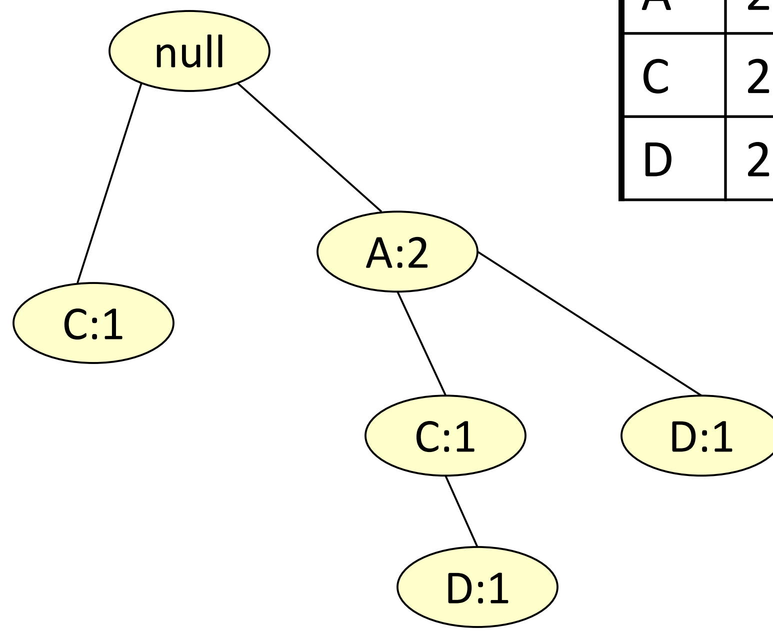




| Header | |
|---|---|
| A | 2 |
| C | 2 |
| D | 2 |

Frequent itemsets ending in E: {A,E}, {C,E}, {D,E}

# Conditional on E: insert transactions 3

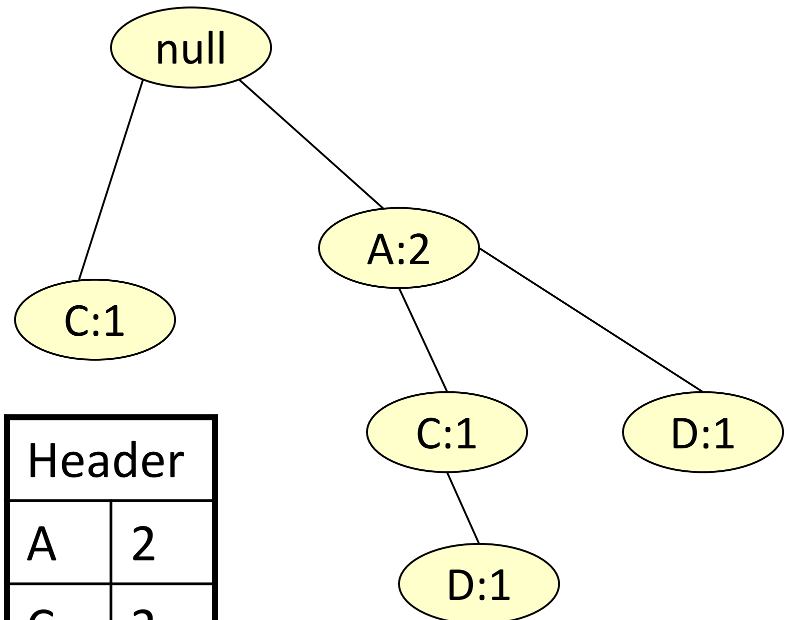



| Header | |
|---|---|
| A | 2 |
| C | 2 |
| D | 2 |

Frequent itemsets ending in E: {A,E}, {C,E}, {D,E}

# Mine FP-tree conditional on E: recursion

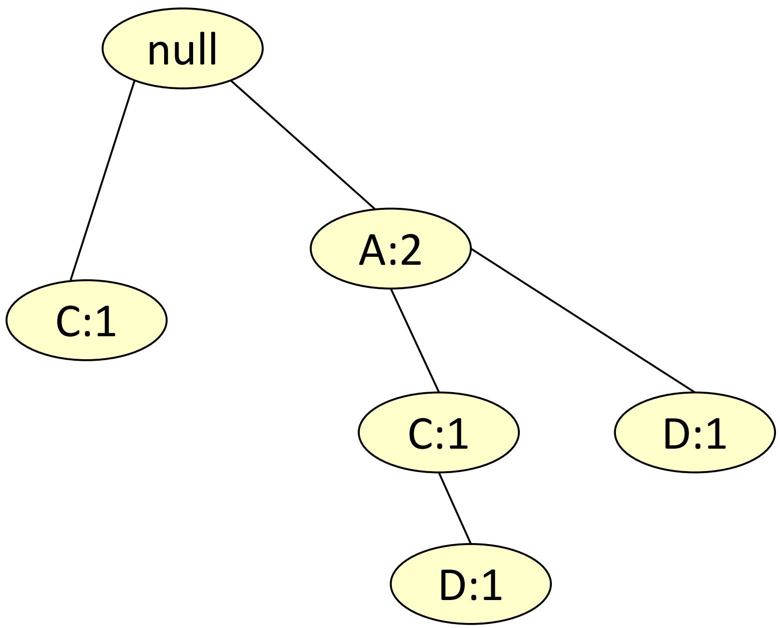



| Header | |
|---|---|
| A | 2 |
| C | 2 |
| D | 2 |

- Continue the same process treating FP-tree (E) as a regular FP-tree
- Remember that these are frequent itemsets ending in E

Frequent itemsets ending in E: {A,E}, {C,E}, {D,E}

# Conditional on DE: header

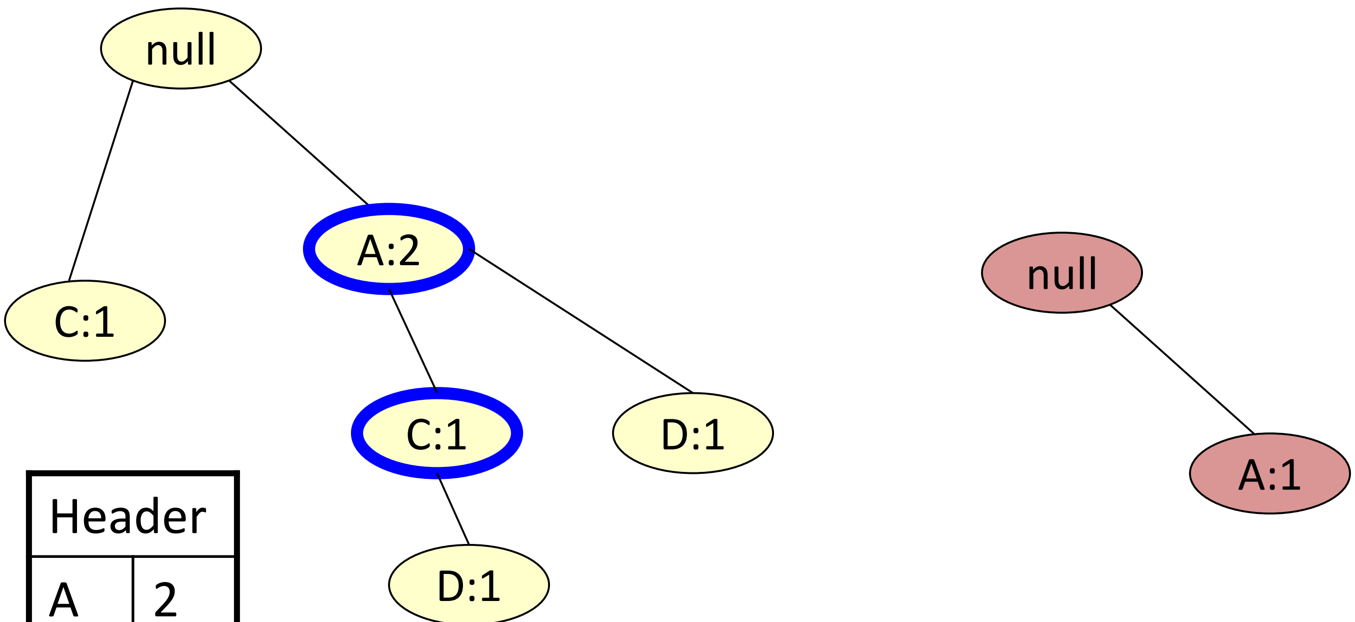



| Header | |
|---|---|
| A | 2 |
| ~~C~~ | ~~1~~ |

At this point we know that {A,D,E} is frequent

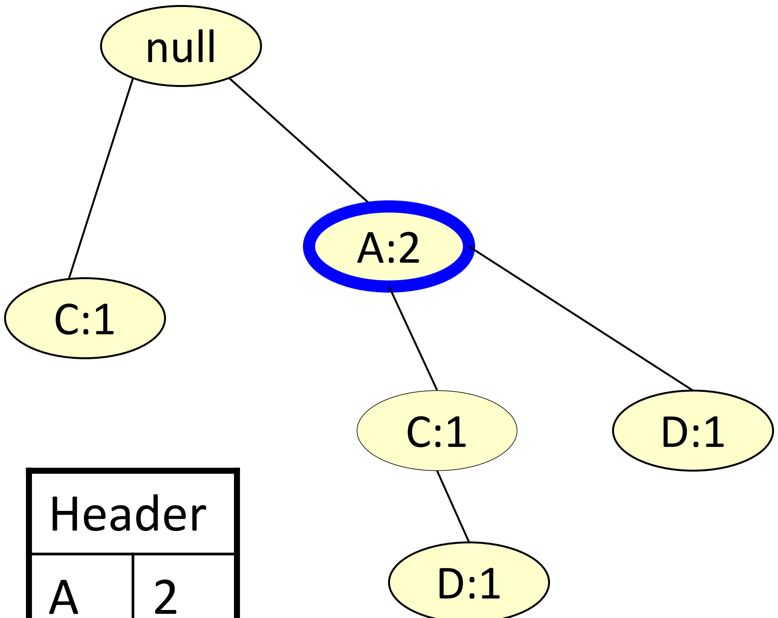
Frequent itemsets ending in E: {A,E}, {C,E}, {D,E} + {A,D,E}

# Conditional on DE: insert transaction 1

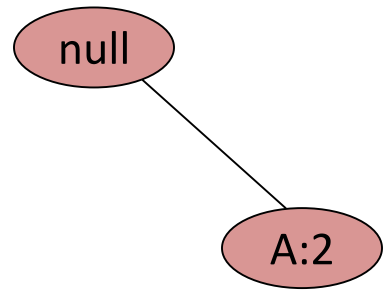



Frequent itemsets ending in E: {A,E}, {C,E}, {D,E} ,{A,D,E}

# Conditional on DE: insert transaction 2




| Header | |
|---|---|
| A | 2 |

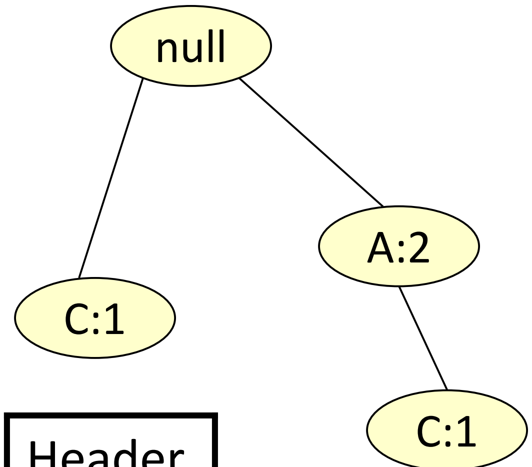



End of recursion: single path

We have collected all frequent itemsets which contain item E

Frequent itemsets ending in E: {A,E}, {C,E}, {D,E} ,{A,D,E}

# Back to mine FP-tree conditional on E: recursion, but now D is removed

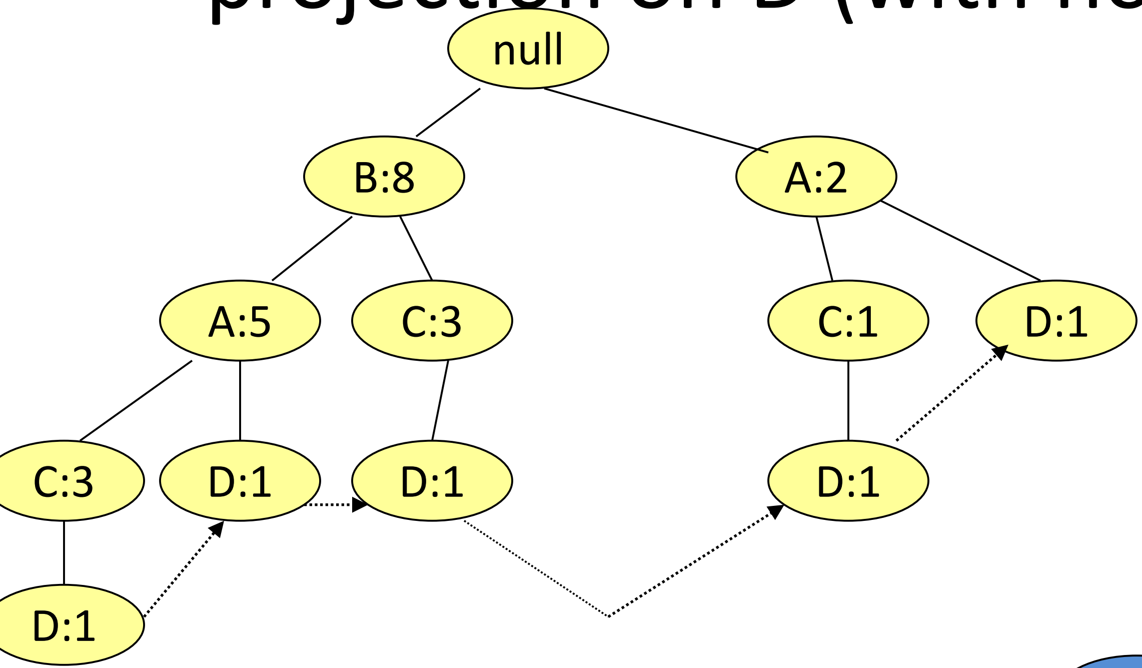


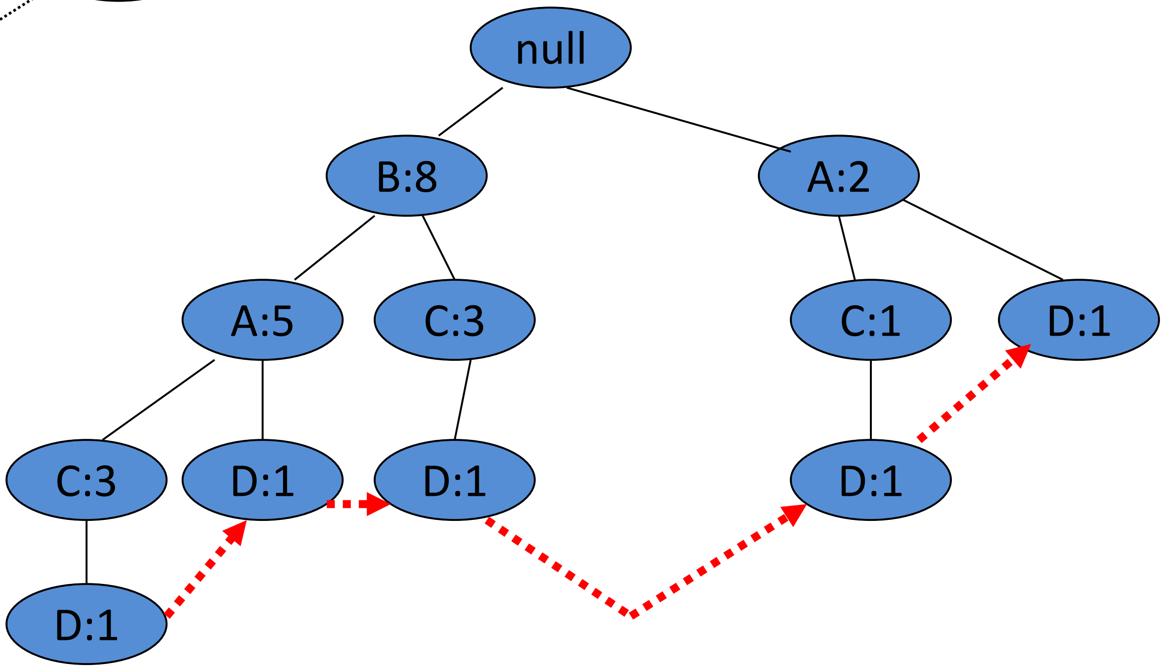

| Header | |
|---|---|
| A | 2 |
| C | 2 |
| D | 2 |

- Now we build an FP-tree conditional on (CE) and treat it as a regular FP-tree
- Remember that these are frequent itemsets ending in CE
- No frequent itemsets

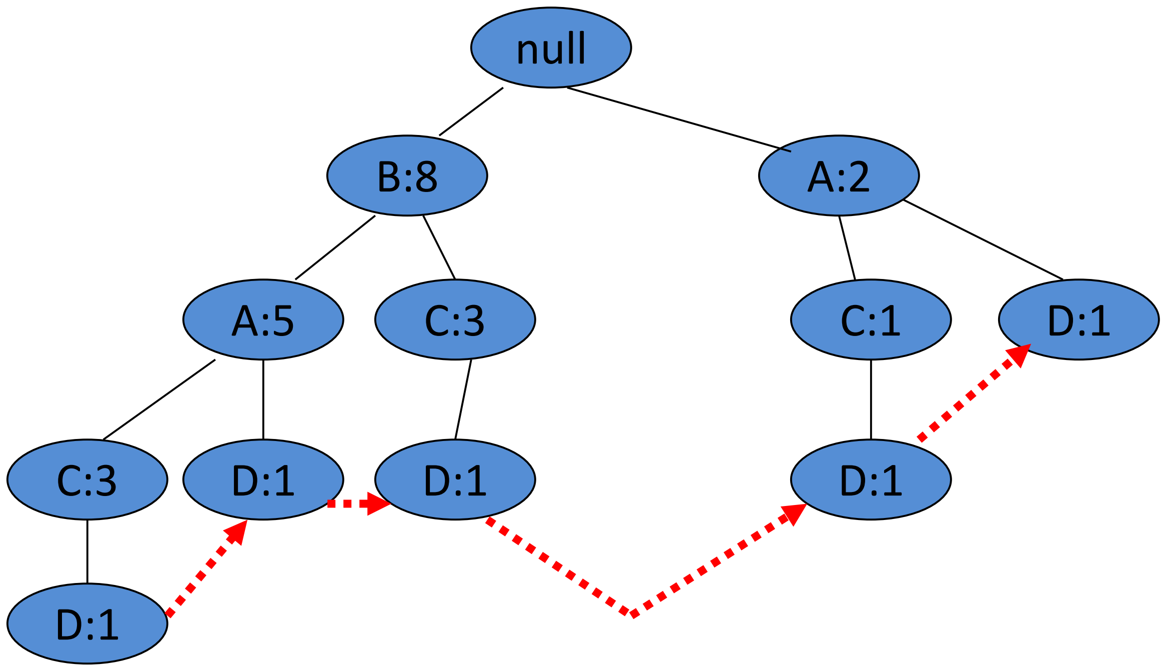
Frequent itemsets ending in E: {A,E}, {C,E}, {D,E} ,{A,D,E}

# Back to the original FP-tree: projection on D (with nodes E removed)



# Conditional on D: header table



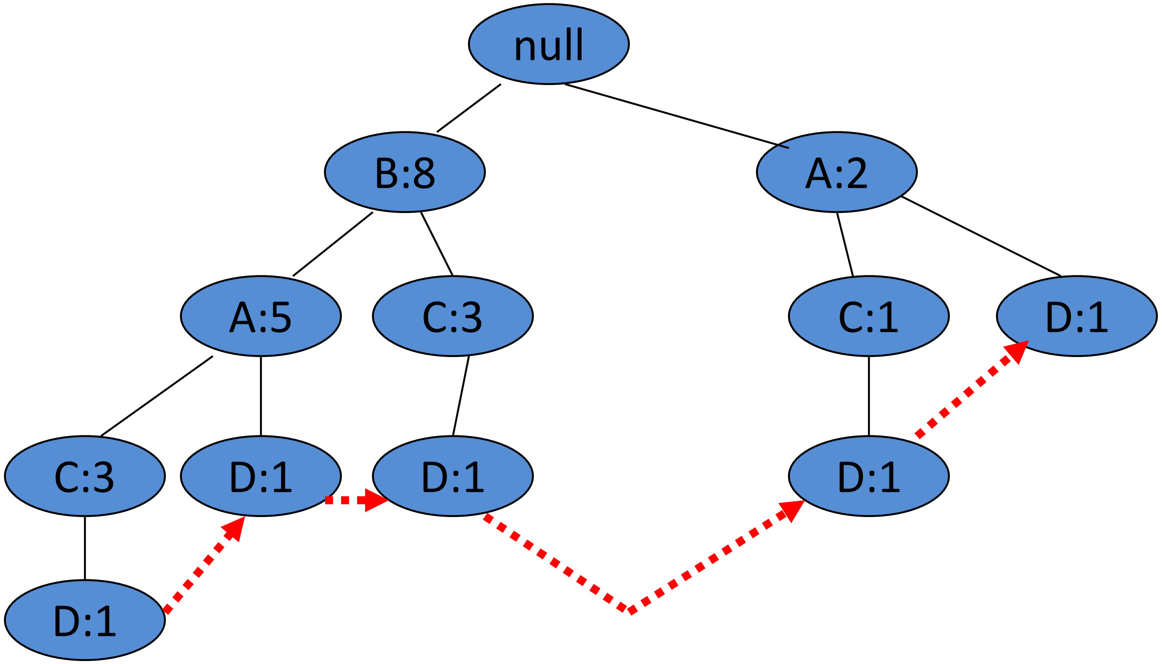

| Header | |
|---|---|
| A | 4 |
| B | 3 |
| C | 3 |

At this point we know that {A,D} , {B,D} and {C,D} are frequent

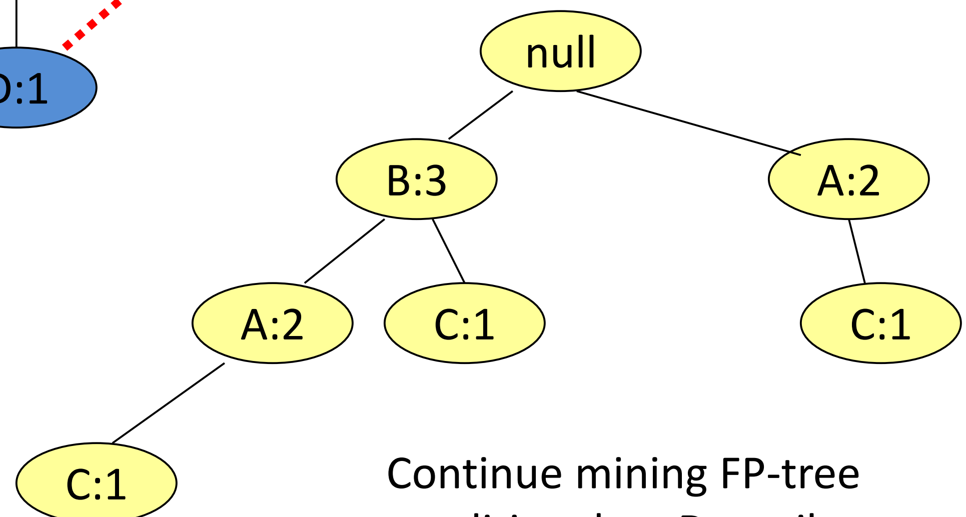
Frequent itemsets ending in E: {A,E}, {C,E}, {D,E} ,{A,D,E}

Frequent itemsets ending in D: {A,D}, {B,D}, {C,D}

# Conditional on D: FP-tree




| Header | |
|---|---|
| A | 4 |
| B | 3 |
| C | 3 |



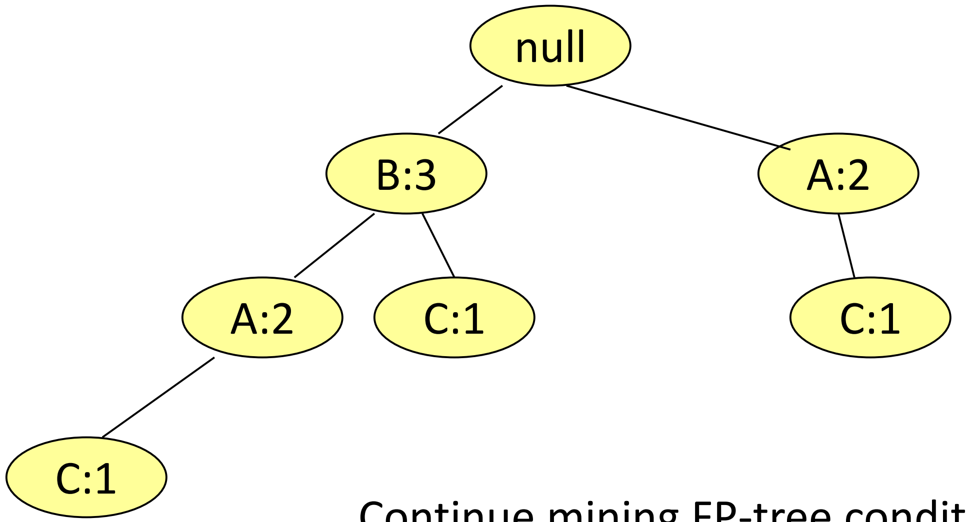

Continue mining FP-tree conditional on D, until a single path left ...

And the same for C, A, B in this order

Frequent itemsets ending in E: {A,E}, {C,E}, {D,E} ,{A,D,E}

Frequent itemsets ending in D: {A,D}, {B,D}, {C,D}

# Conditional on D: FP-tree




Continue mining FP-tree conditional on D, until a single path left ...

And the same for C, A, B in this order

Frequent itemsets ending in E: {A,E}, {C,E}, {D,E} ,{A,D,E}

Frequent itemsets ending in D: {A,D}, {B,D}, {C,D}

# FP-Tree Another Example

Transactions

A B C E F O
A C G
E I
A C D E G
A C E G L
E J
A B C E F P
A C D
A C E G M
A C E G N

Freq. 1-Itemsets.
Supp. Count ≥2

| | |
|---|---|
| A:8 | |
| C:8 | |
| E:8 | |
| G:5 | |
| B:2 | |
| D:2 | |
| F:2 | |

Transactions with items sorted based on frequencies, and ignoring the infrequent items.

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G

# FP-Tree after reading 1$^{st}$ transaction

**A C E B F**

A C G

E

A C E G D

A C E G

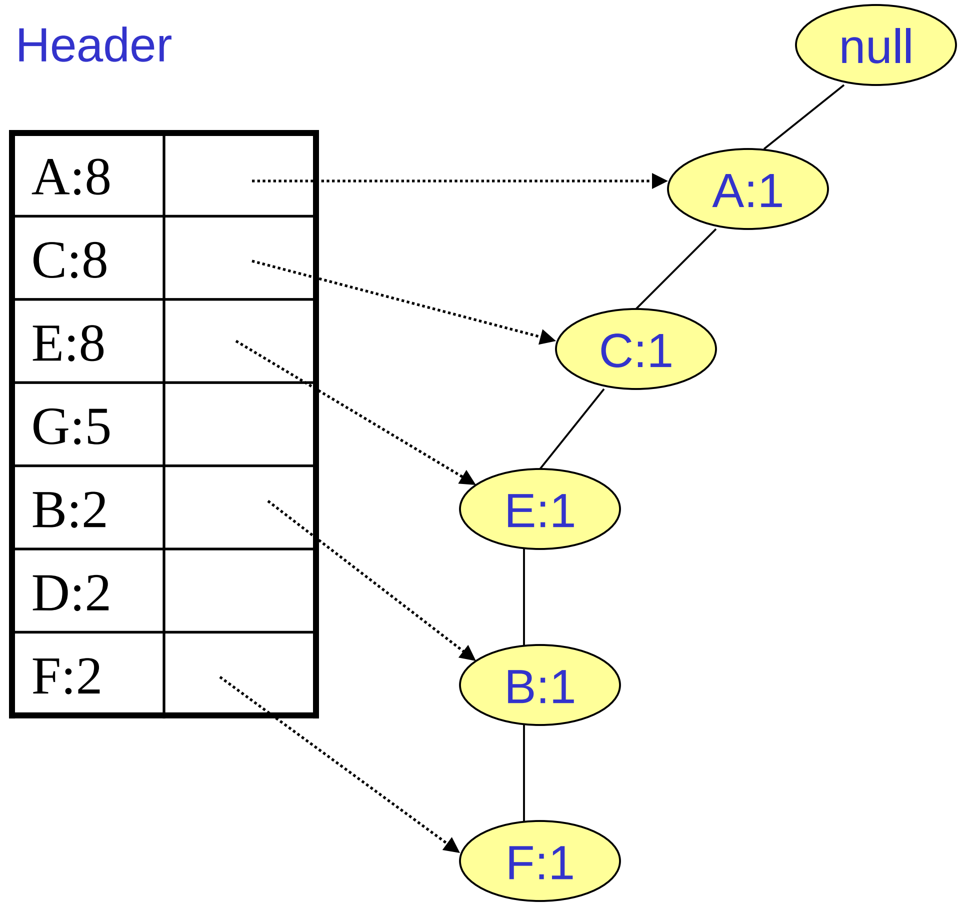
E

A C E B F

A C D

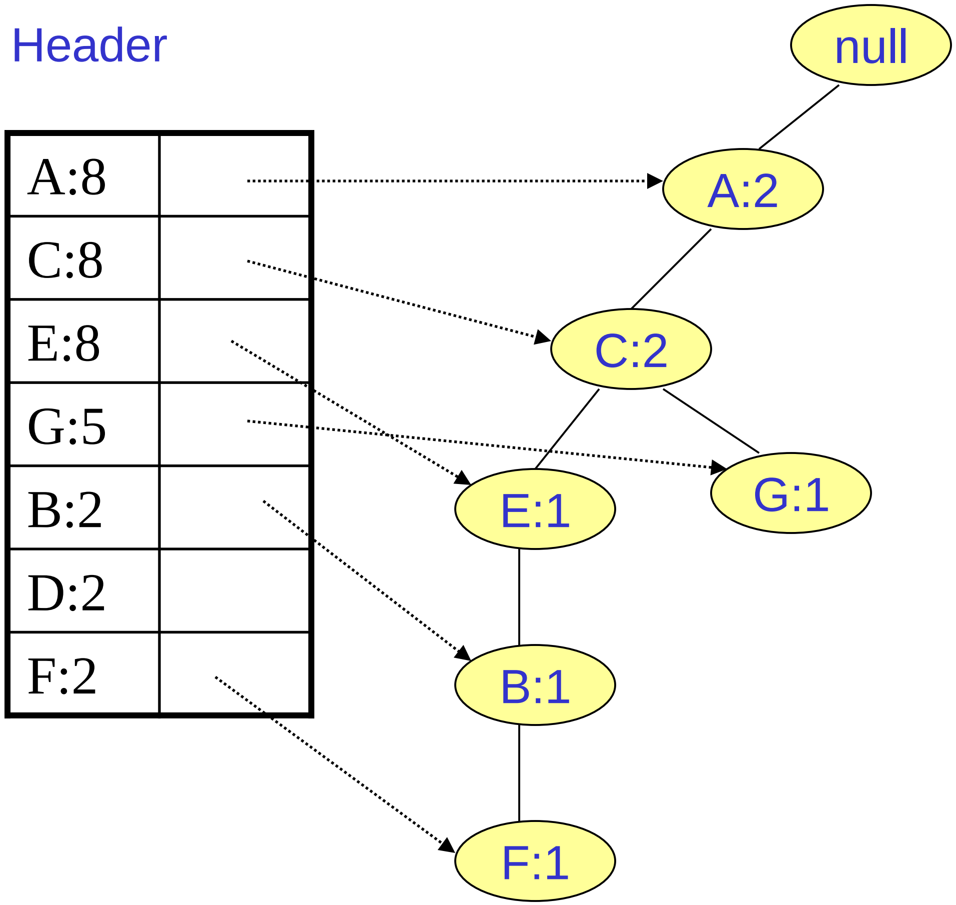
A C E G

A C E G



# FP-Tree after reading 2nd transaction

A C E B F
**A C G**
E
A C E G D
A C E G
E
A C E B F
A C D
A C E G
A C E G



# FP-Tree after reading 3rd transaction

A C E B F

A C G

**E**

A C E G D

A C E G

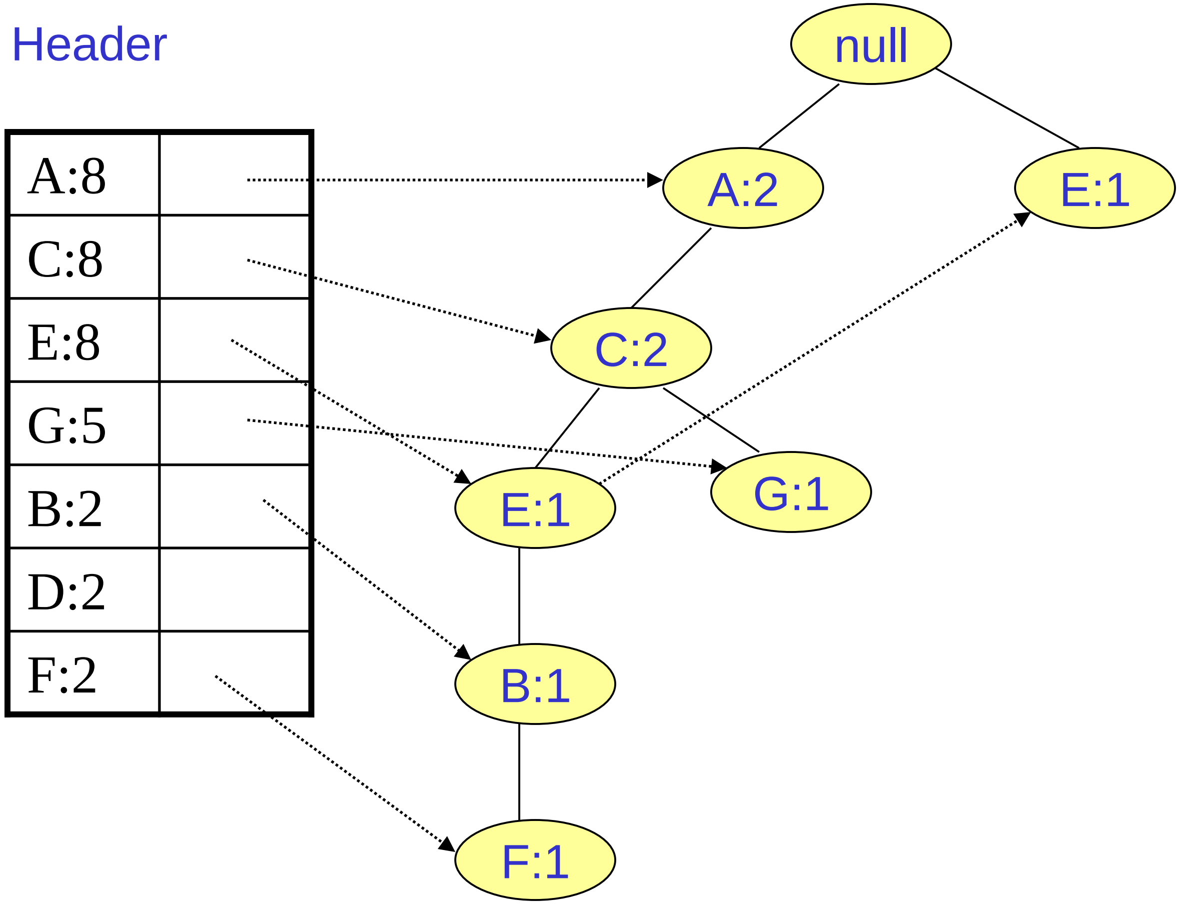
E

A C E B F

A C D

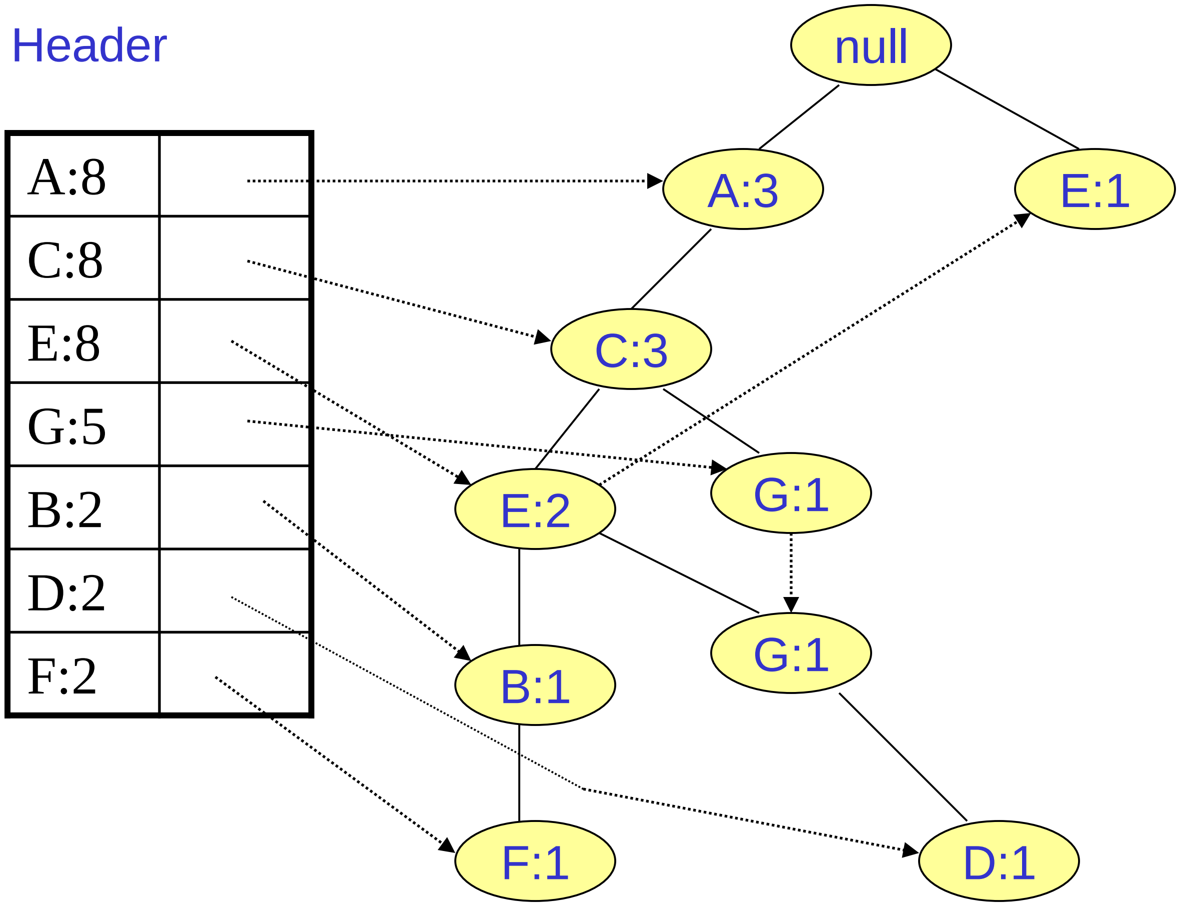
A C E G

A C E G



# FP-Tree after reading 4th transaction

A C E B F
A C G
E
**A C E G D**
A C E G
E
A C E B F
A C D
A C E G
A C E G



# FP-Tree after reading 5th transaction

A C E B F

A C G

E

A C E G D

**A C E G**

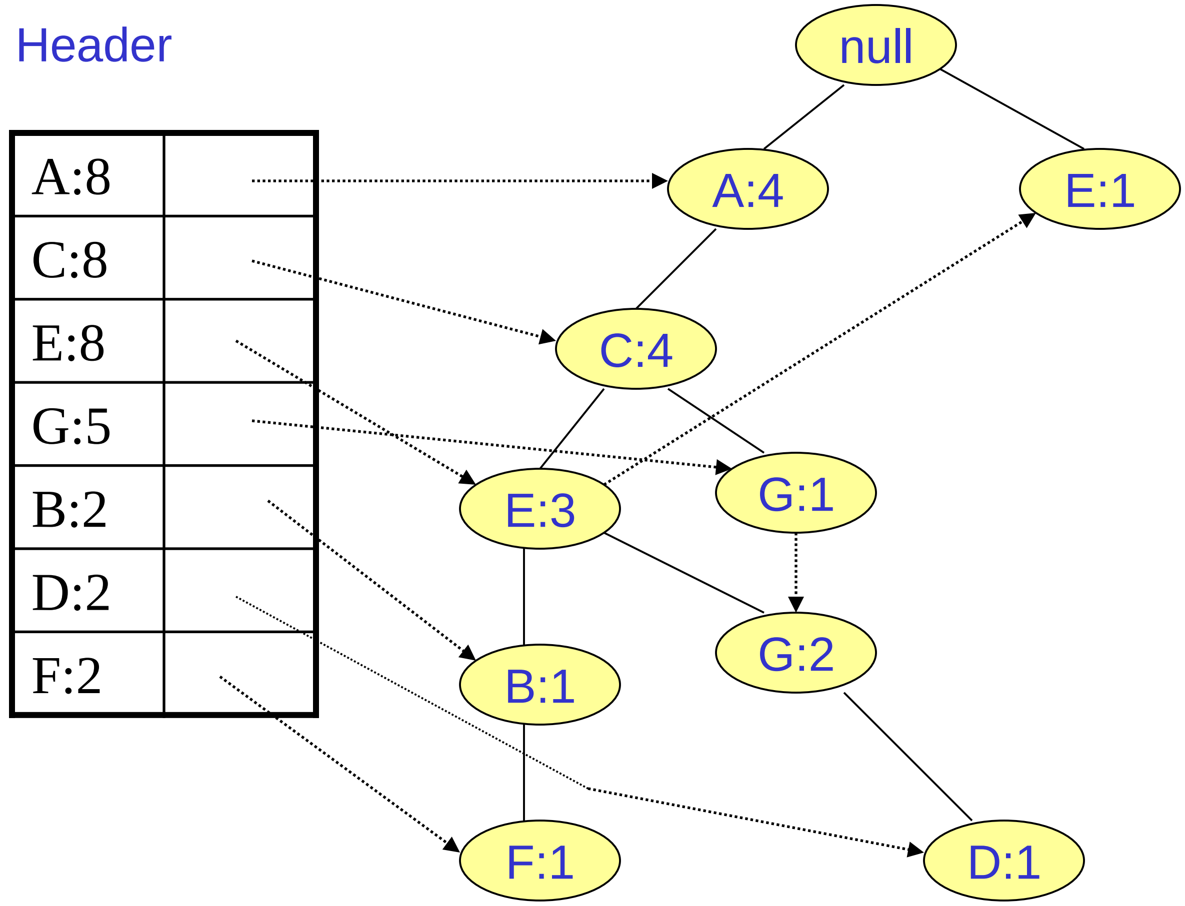
E

A C E B F

A C D

A C E G

A C E G



# FP-Tree after reading 6$^{th}$ transaction

A C E B F

A C G

E

A C E G D

A C E G

**E**

A C E B F

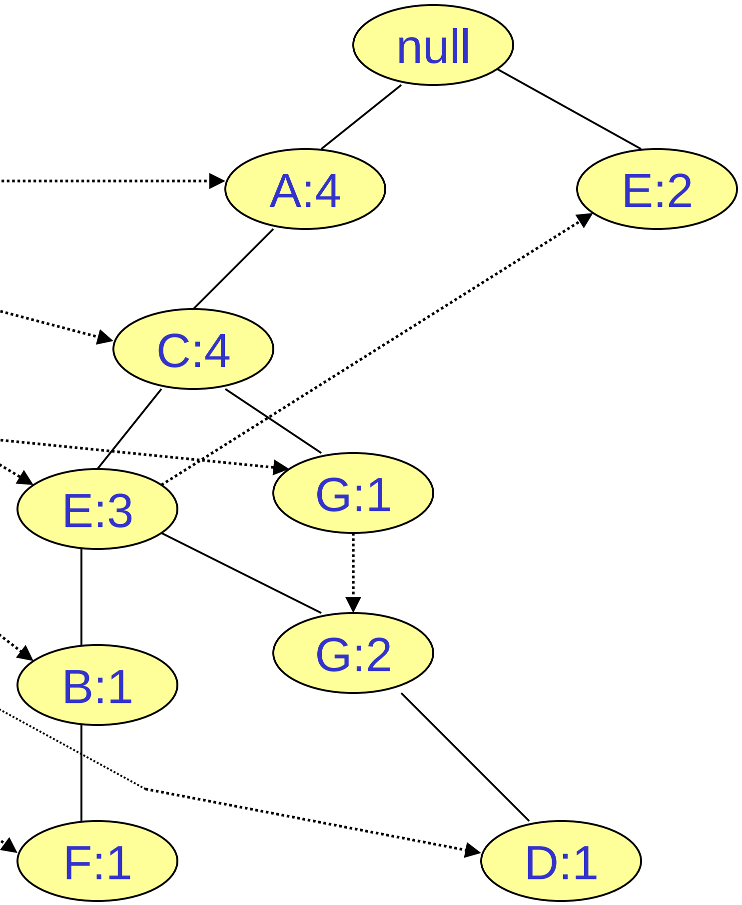
A C D

A C E G

A C E G

Header

| | |
|---|---|
| A:8 | |
| C:8 | |
| E:8 | |
| G:5 | |
| B:2 | |
| D:2 | |
| F:2 | |



# FP-Tree after reading 7$^{th}$ transaction

A C E B F

A C G

E

A C E G D

A C E G

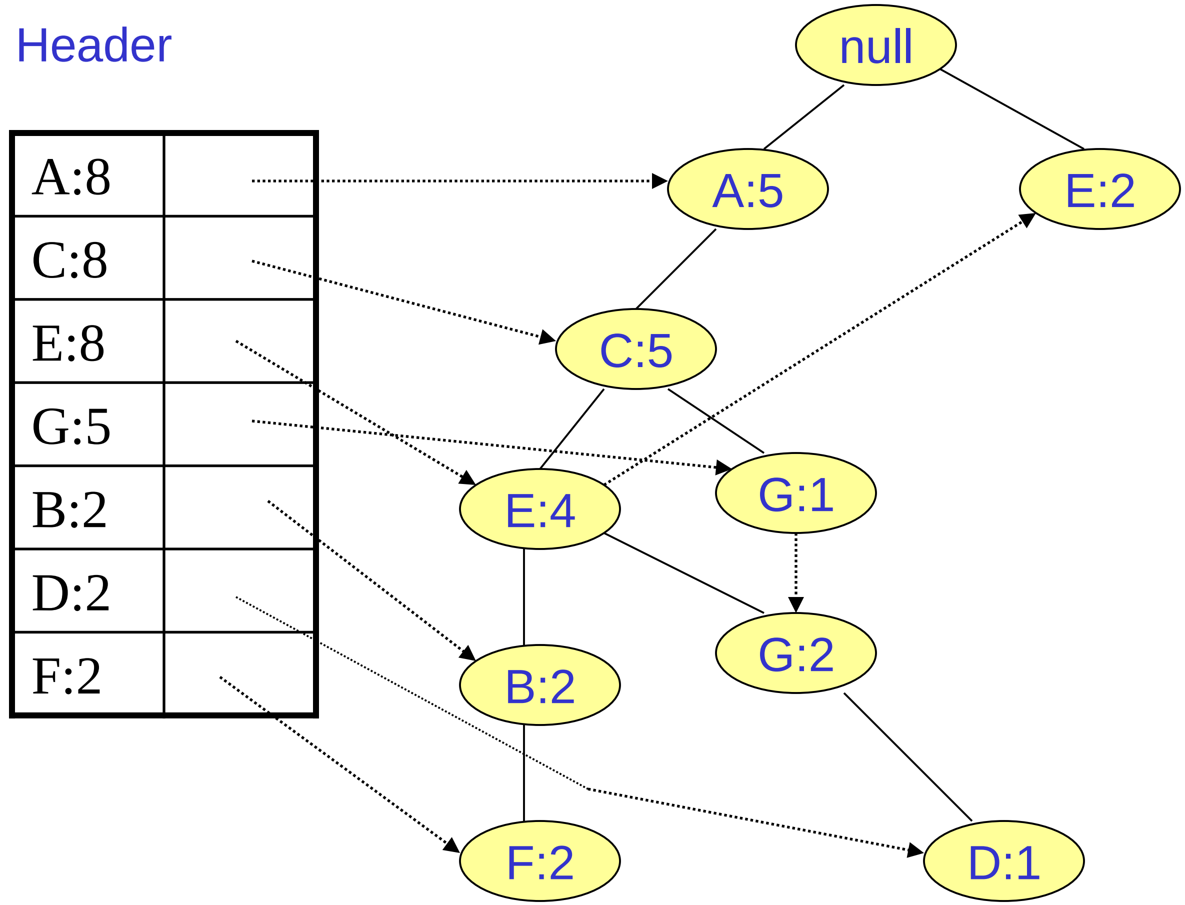
E

**A C E B F**

A C D

A C E G

A C E G



# FP-Tree after reading 8th transaction

A C E B F

A C G

E

A C E G D

A C E G

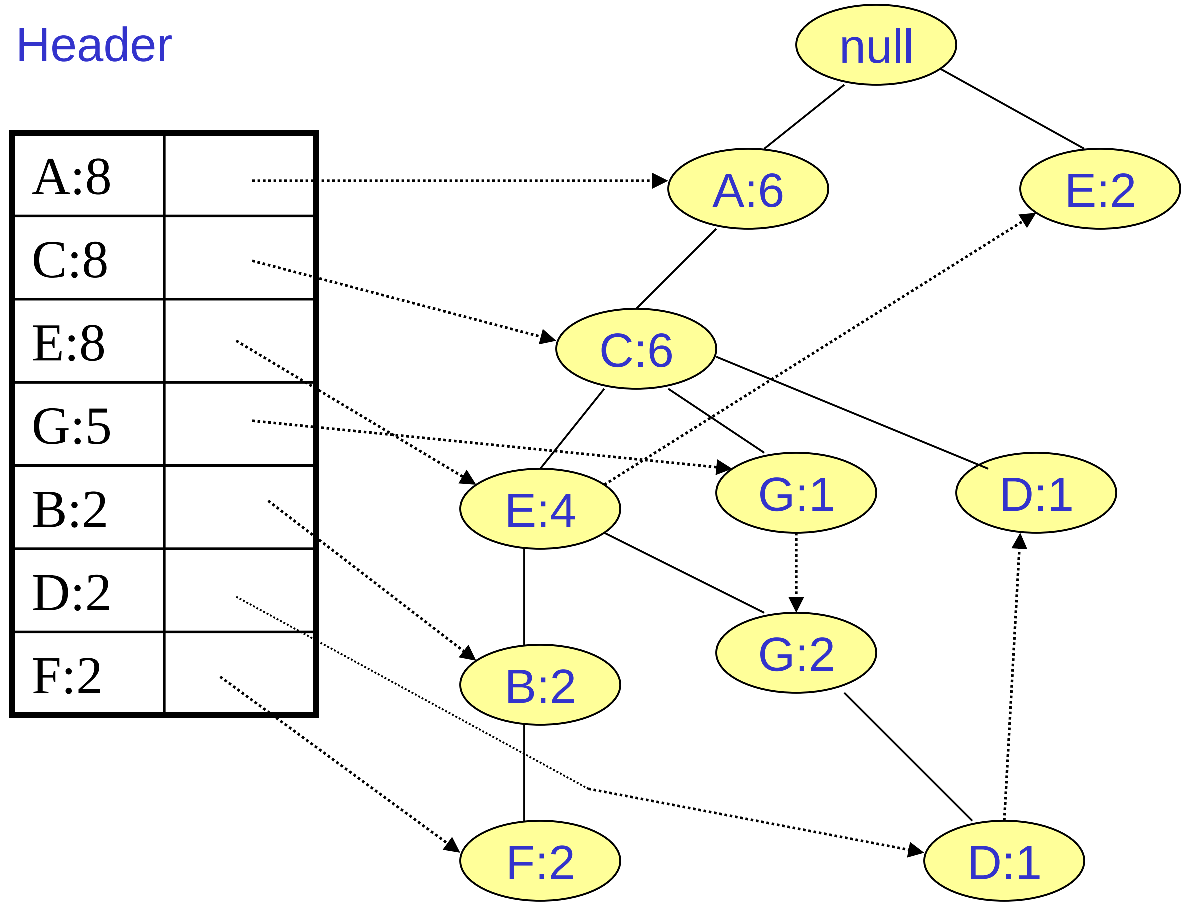
E

A C E B F

**A C D**

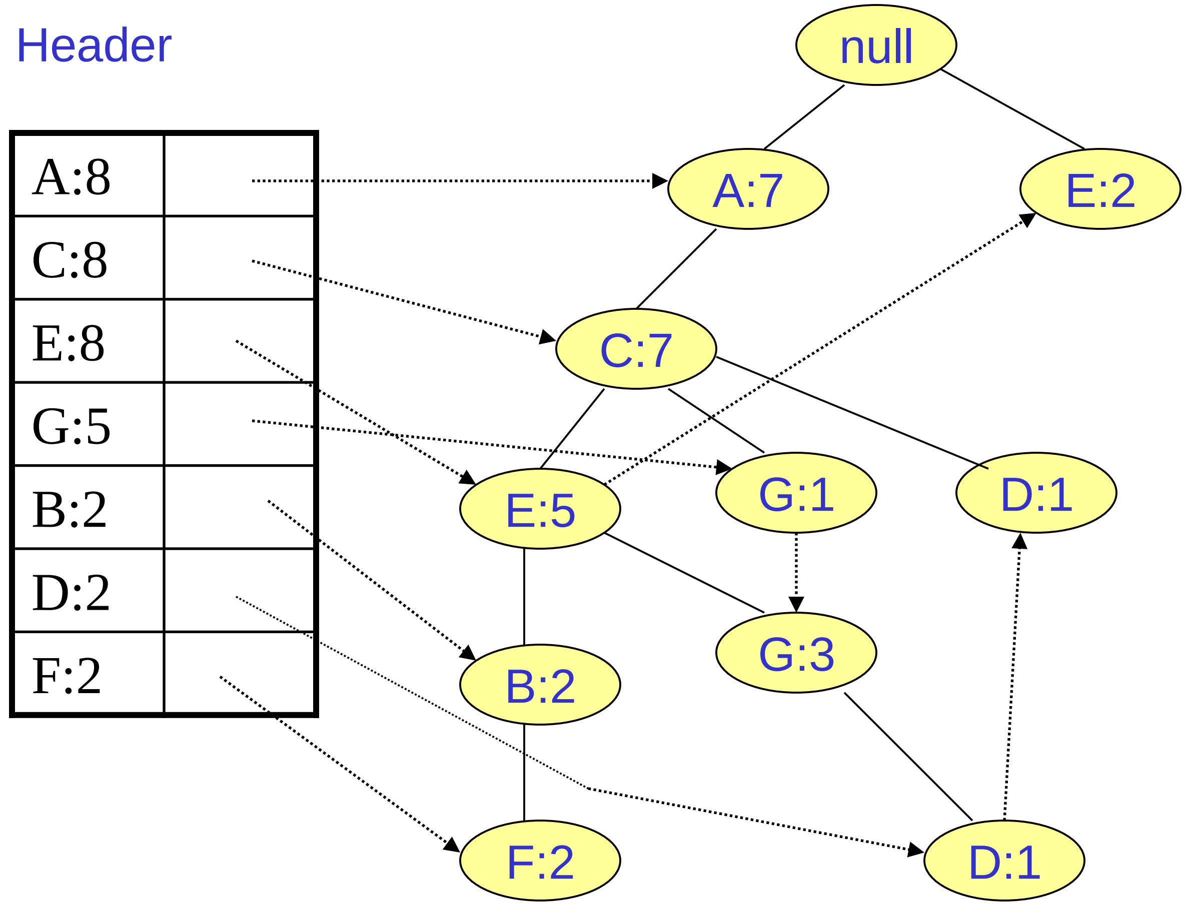
A C E G

A C E G



# FP-Tree after reading 9th transaction

A C E B F
A C G
E
A C E G D
A C E G
E
A C E B F
A C D
**A C E G**
A C E G



# FP-Tree after reading 10th transaction

A C E B F

A C G

E

A C E G D

A C E G

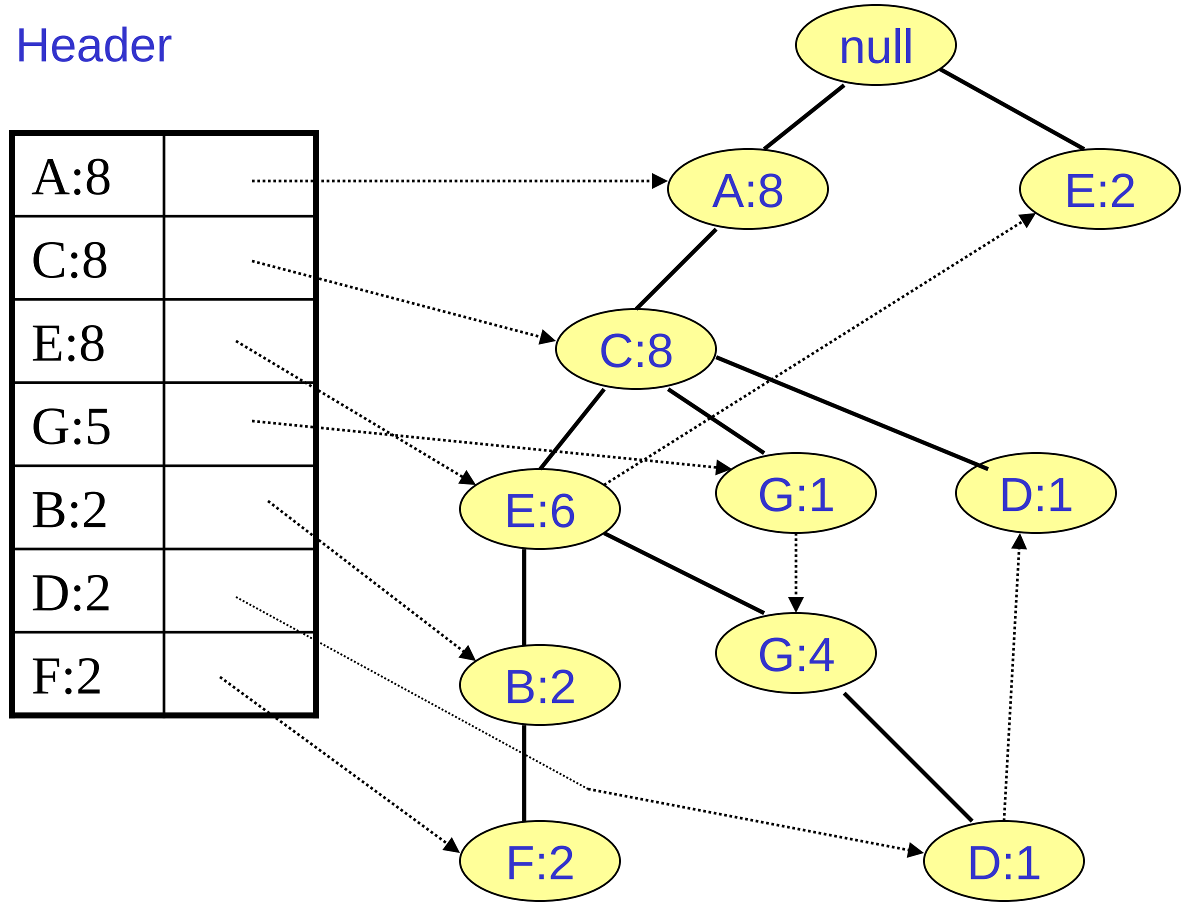
E

A C E B F

A C D

A C E G

**A C E G**



# Conditional FP-Trees

Build the conditional FP-Tree for each of the items.

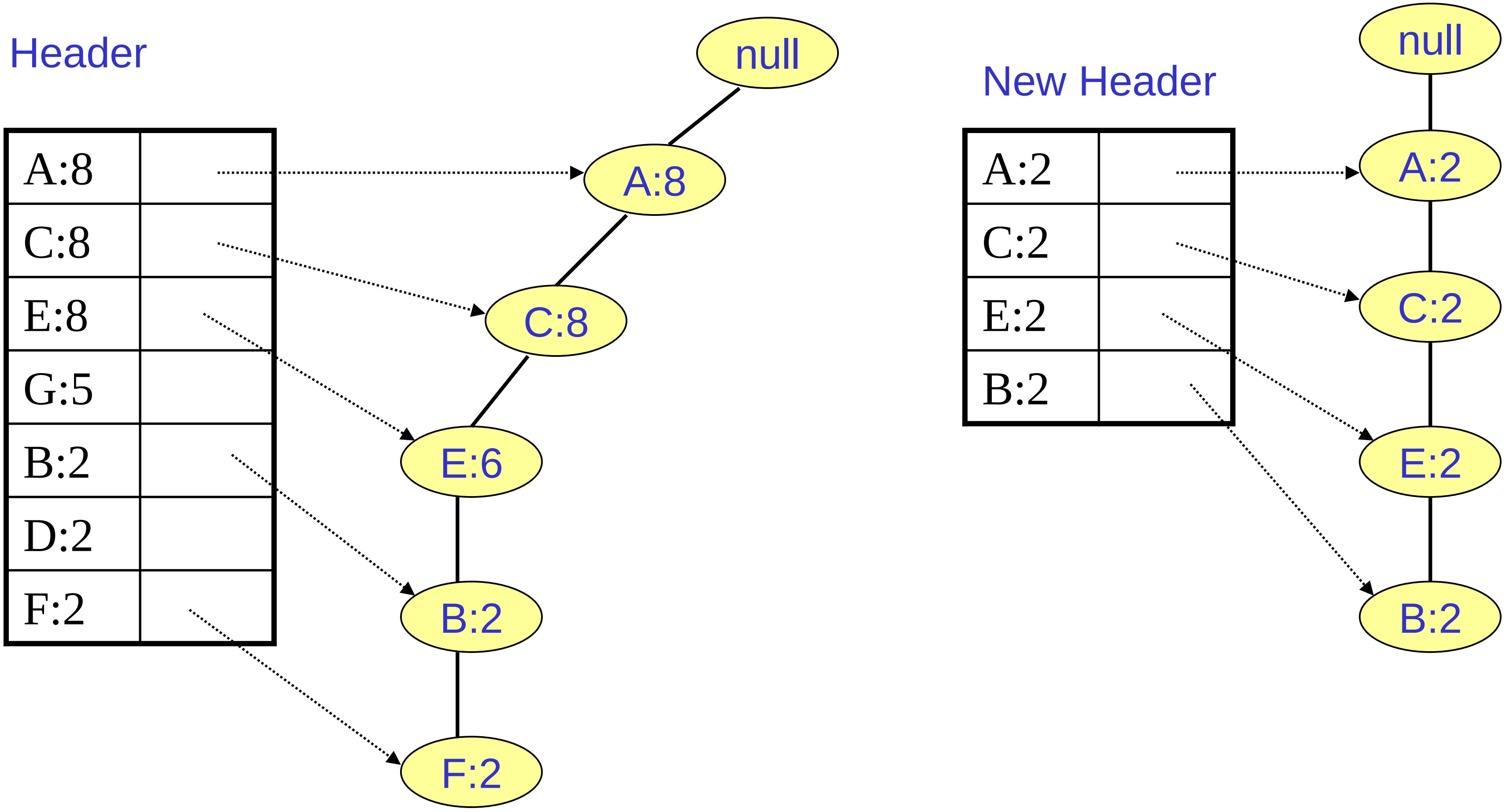
For this:

1. Find the paths containing on focus item. With those paths we build the conditional FP-Tree for the item.

2. Read again the tree to determine the new counts of the items along those paths. Build a new header.

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

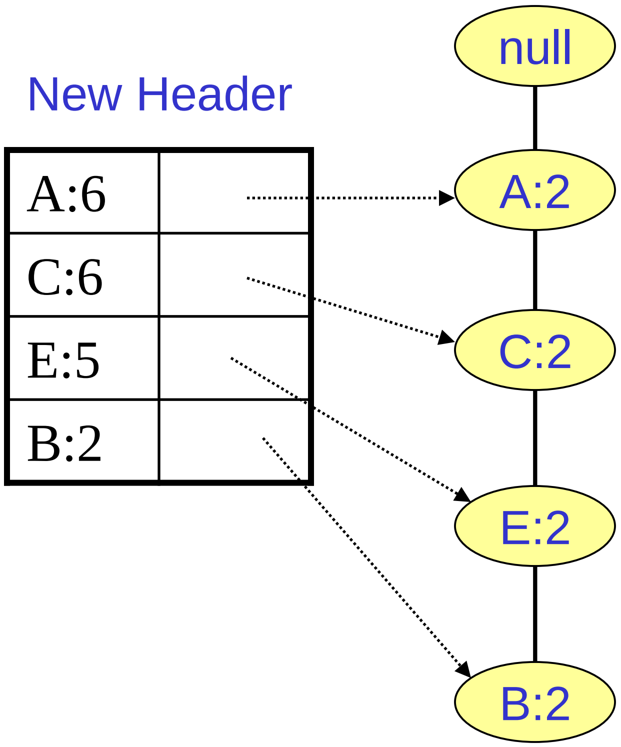
# Conditional FP-Tree for F




There is only a single path containing F

# Recursion

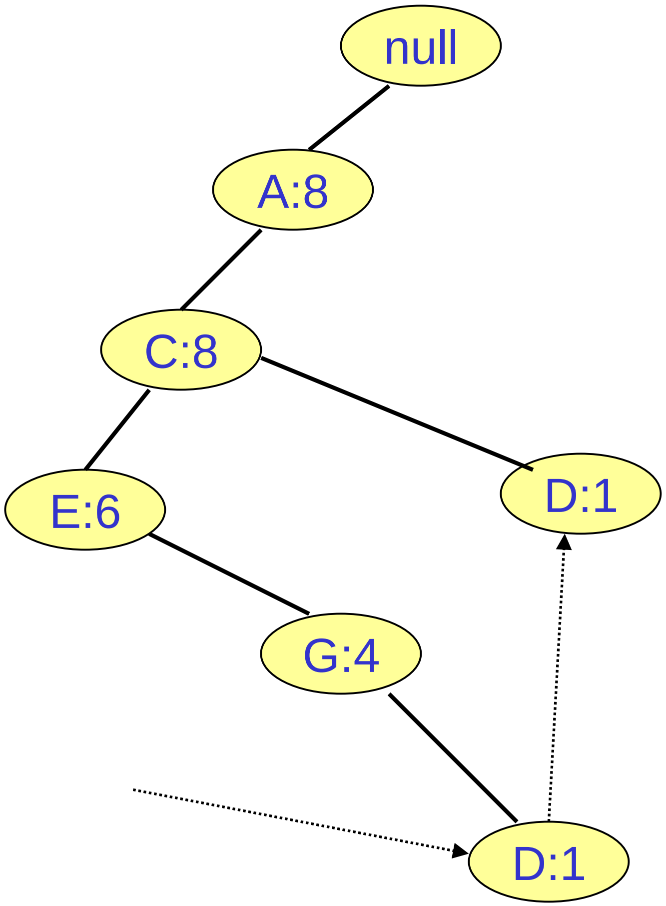
- We continue recursively on the conditional FP-Tree for F.
- However, when the tree is just a single path it is the base case for the recursion.
- So, we just produce all the subsets of the items on this path merged with F.



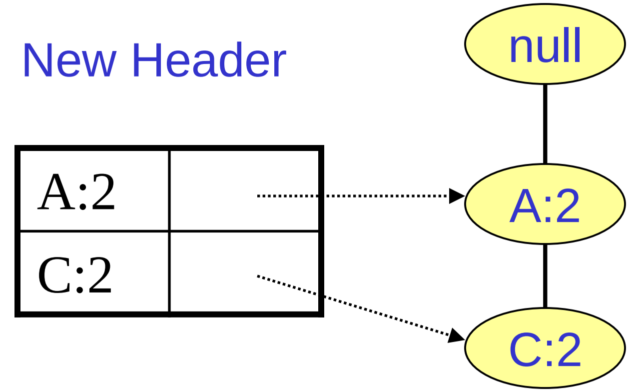

{F} {A,F} {C,F} {E,F} {B,F} {A,C,F}, …,

{A,C,E,F}

# Conditional FP-Tree for D




Paths containing D after updating the counts




The other items are removed as infrequent.

The tree is just a single path; it is the base case for the recursion.

So, we just produce all the subsets of the items on this path merged with D.

{D} {A,D} {C,D} {A,C,D}

Exercise: Complete the example.