FP-Growth Case Study: FP-Tree Construction Step by Step

Question

Consider the following transactional dataset:

Transaction ID	Items Purchased
T1	Milk, Bread, Butter
T2	Bread, Butter, Eggs, Cheese
T3	Milk, Bread, Eggs, Cheese
T4	Milk, Bread, Butter, Cheese
T5	Bread, Eggs

Minimum support threshold is **0.4**. Construct the FP-tree **step by step** and then find all frequent itemsets.

Solution

1) Minimum Support Count

There are $N = 5$ transactions. With minimum support $= 0.4$:

$$\text{min support count} = 0.4 \times 5 = 2$$

So, an itemset is frequent if its support count is ≥ 2.

2) First Scan: Item Frequency Count

Count the occurrence of each item across all transactions:

Item	Support Count
Bread	5
Milk	3
Butter	3
Eggs	3
Cheese	3

All items have support ≥ 2, hence all are **frequent**.

3) Global Item Order (Descending Frequency)

FP-Growth sorts items by descending support (ties broken consistently). We use:

$$\text{Bread} > \text{Milk} > \text{Butter} > \text{Eggs} > \text{Cheese}$$

4) Reorder Each Transaction by the Global Order

Each transaction is filtered to frequent items (all items remain) and then reordered:

TID	Ordered Transaction
T1	Bread, Milk, Butter
T2	Bread, Butter, Eggs, Cheese
T3	Bread, Milk, Eggs, Cheese
T4	Bread, Milk, Butter, Cheese
T5	Bread, Eggs

FP-tree construction rule:

- Start from the root (null node).
- Insert each ordered transaction as a path.
- If a node already exists on the path, increment its count.
- Otherwise, create a new node with count 1.

5) FP-Tree Construction (All Intermediate Steps)

Notation

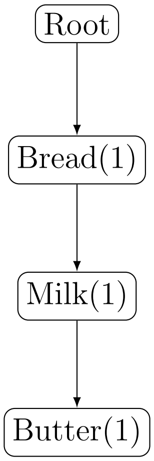
A node is written as **Item(count)**.

Step 0: Initialize

Start with an empty tree containing only the root:

Step 1: Insert T1 = (Bread, Milk, Butter)

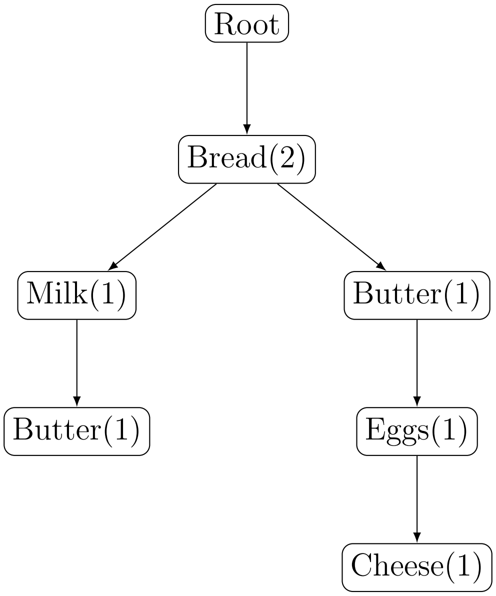
Explanation: No nodes exist yet, so we create a new path: Root → Bread(1) → Milk(1) → Butter(1).



Step 2: Insert T2 = (Bread, Butter, Eggs, Cheese)

Explanation:

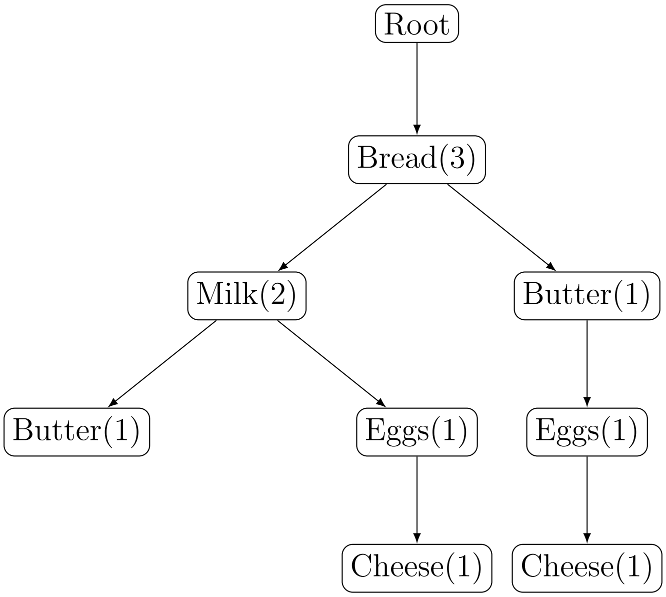
- Root → Bread already exists, so increment Bread: 1 → 2.
- Next item is Butter. Under Bread, there is currently only Milk, so create a new child Butter(1).
- Then create Eggs(1) under this Butter.
- Then create Cheese(1) under Eggs.



Step 3: Insert T3 = (Bread, Milk, Eggs, Cheese)

Explanation:

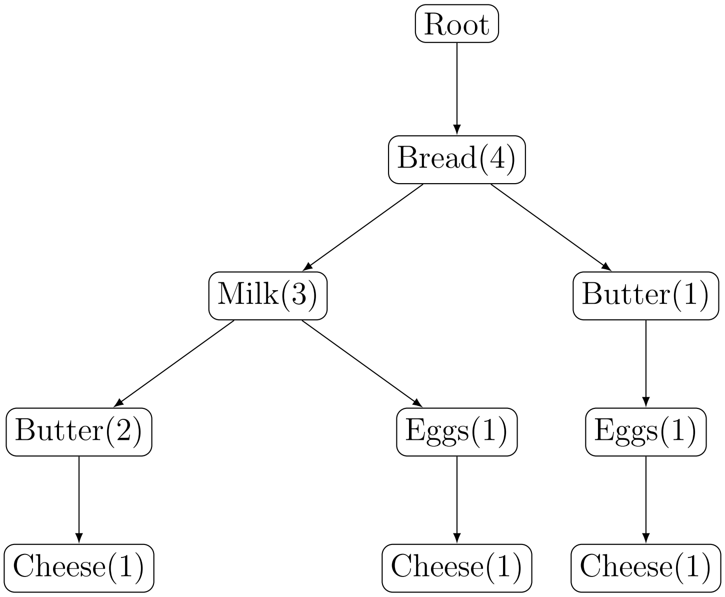
- Bread exists: 2 → 3.
- Under Bread, Milk exists: 1 → 2.
- Under Bread→Milk, Eggs does not exist (only Butter exists), so create Eggs(1).
- Under that Eggs, create Cheese(1).



Step 4: Insert T4 = (Bread, Milk, Butter, Cheese)

Explanation:

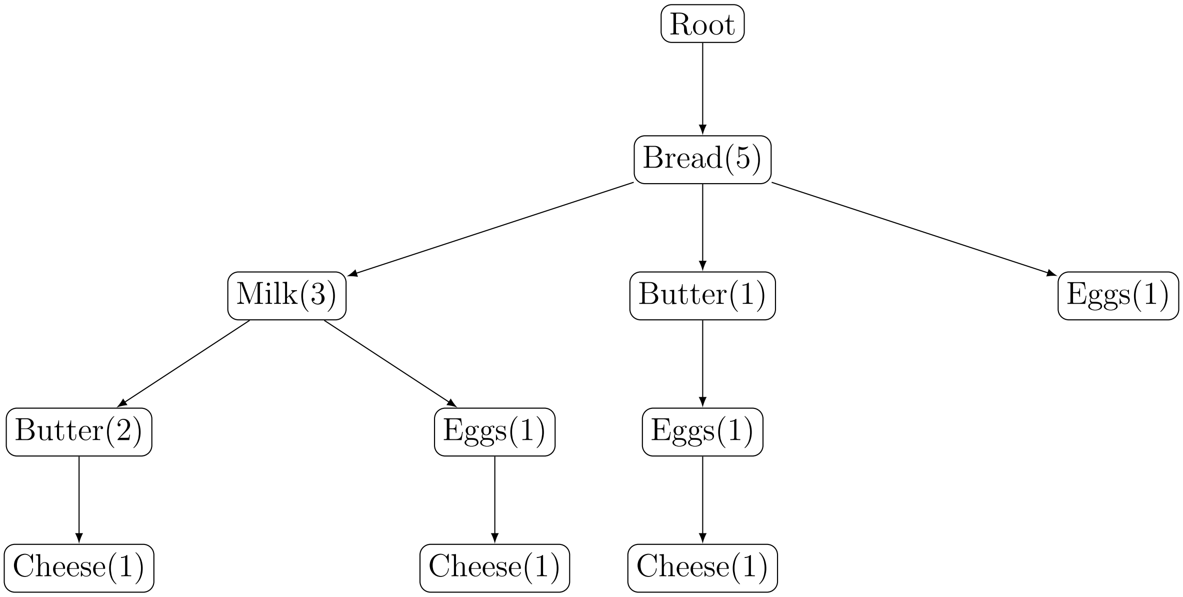
- Bread exists: 3 → 4.
- Under Bread, Milk exists: 2 → 3.
- Under Bread→Milk, Butter exists: 1 → 2.
- Under Bread→Milk→Butter, Cheese does not exist yet, so create Cheese(1).



Step 5: Insert T5 = (Bread, Eggs)

Explanation:

- Bread exists: 4 → 5.
- Under Bread, Eggs does not exist as a direct child (children are Milk and Butter), so create a new child Eggs(1) under Bread.



This is the final FP-tree.

6) Frequent Itemsets (Support ≥ 2)

Below are the frequent itemsets (derived using FP-Growth mining via conditional pattern bases; only itemsets meeting the threshold are listed).

Frequent 1-itemsets

$$\{Bread\}, \{Milk\}, \{Butter\}, \{Eggs\}, \{Cheese\}$$

Frequent 2-itemsets

$$\{Bread, Milk\}, \{Bread, Butter\}, \{Bread, Eggs\}, \{Bread, Cheese\}$$
$$\{Milk, Butter\}, \{Milk, Cheese\}, \{Butter, Cheese\}, \{Eggs, Cheese\}$$

Frequent 3-itemsets

$\{Bread, Milk, Butter\}, \{Bread, Milk, Cheese\}, \{Bread, Butter, Cheese\}, \{Bread, Eggs, Cheese\}$

Summary

- Minimum support count = 2.
- All items are frequent.
- Transactions are sorted by global frequency order.
- FP-tree is built by inserting each ordered transaction, sharing prefixes and incrementing counts.
- Frequent itemsets are mined from the FP-tree using conditional pattern bases (FP-Growth).