1 FP-Growth Examples

1.1 Example 1

First we will see a simple example of FP-Growth where we are not doing repeated recursion on the projected databases. Consider the transaction database in Table 1 for which we have to identify frequent itemsets with minimum support equal to 3.

ID	Transaction
1	ABDE
2	BCE
3	ABDE
4	ABCE
5	ABCDE
6	BCD

Table 1: Transaction Database

1.1.1 Step 1.

Sort the items in the transaction database by their support (see Table 2).

Item	Support
B	6
E	5
A	4
C	4
D	4

Table 2: Sorted item support for transaction database in Table 1.

1.1.2 Step 2.

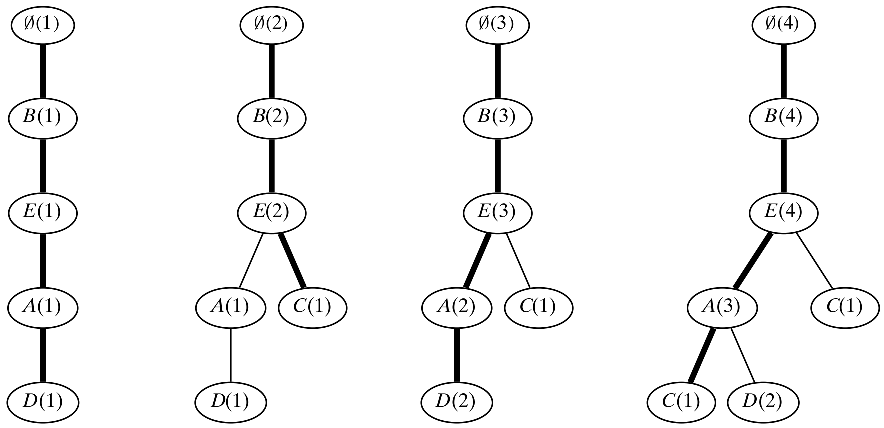
Sort the order of the items in each transaction of the database with descending order of support (see Table 3). For example, for the first transaction $ABDE$ we sort the items in the transaction with descending order of support $B(6)\,E(5)\,A(4)\,D(4)$.

ID	Transaction
1	BEAD
2	BEC
3	BEAD
4	BEAC
5	BEACD
6	BCD

Table 3: Transactions with items sorted by their support for transaction database in Table 1.

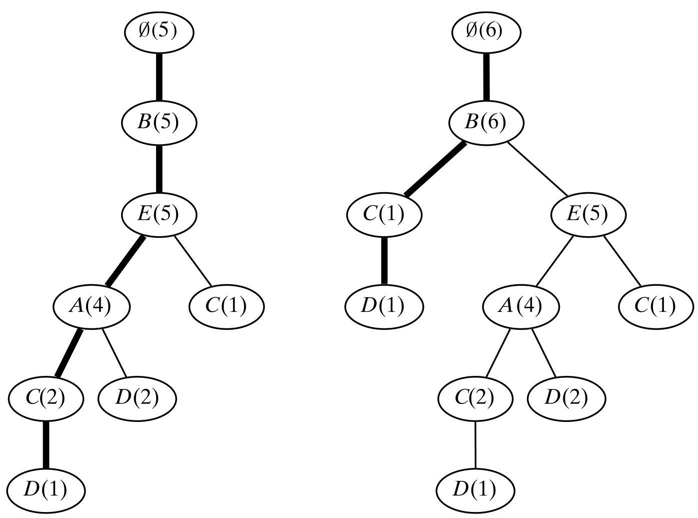
1.1.3 Step 3.

Construct the FP-Tree (call it R) step-by-step by adding each of the transactions (where items in them are reordered with descending support). The FP-tree R serves as an index in lieu of the original database.



(a) $\langle 1, BEAD \rangle$ (b) $\langle 2, BEC \rangle$ (c) $\langle 3, BEAD \rangle$ (d) $\langle 4, BEAC \rangle$

Figure 1: FP-Tree construction steps (a) to (d) for Table 1.



(e) $\langle 5, BEACD \rangle$ (f) $\langle 6, BCD \rangle$

Figure 2: FP-Tree construction steps (e) to (f) for Table 1.

1.1.4 Step 4.

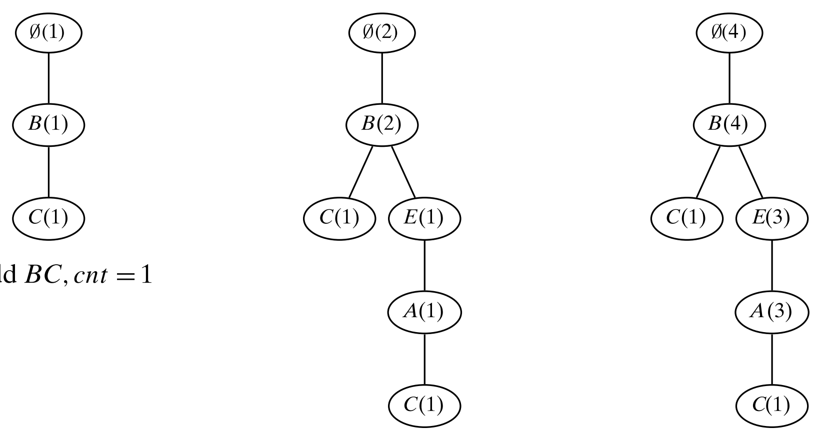
Given the FP-Tree (R), projected (conditional) FP-trees (databases) are built for each frequent item i in R in increasing order of support. That is, we now project the FP-Tree R for the following items in order: $D(4)\, C(4)\, A(4)\, E(5)\, B(6)$.

Projected FP-Tree for D. First, we perform the projection for $D(4)$. For the projection note down all the paths that lead up to D in the FP-Tree R and also note down the count that is associated with D in that path. This is also known as **Conditional Sub-Database** that we will use for the projection. For D we have,

Path	Count
BCD	1
BEACD	1
BEAD	2

Table 4: Paths in the FP-Tree R leading up to D. Also known as **Conditional Sub-Database** for D.

For creating the projected FP-Tree for D we follow the same procedure as we did for creating the initial FP-Tree R. That is, we keep adding the paths we have in the "Conditional Sub-Database" for D (see Table 4) in a new tree — call it R_D. However, whenever we insert a path we omit D and increase the count of all the items encountered along the path with count observed in the "Conditional Sub-Database". This projected FP-Tree is also known as **Conditional FP-Tree**. See Figure 3.



(a) Add $BC, cnt = 1$

(b) Add $BEAC, cnt = 1$

(c) Add $BEA, cnt = 2$

Figure 3: Projected FP-Tree for D.

Projected FP-Tree for C. Second, we perform the projection for $C(4)$. For the projection note down all the paths that lead up to C in the FP-Tree R

and also note down the count that is associated with C in that path. For C we have,

Path	Count
BC	1
BEAC	2
BEC	1

Table 5: Paths in the FP-Tree R leading up to C. Also known as **Conditional Sub-Database** for C.

Projected FP-Tree for A. Third, we perform the projection for $A(4)$:

Path	Count
BEA	4

Table 6: Paths in the FP-Tree R leading up to A. Also known as **Conditional Sub-Database** for A.

Projected FP-Tree for E. Third, we perform the projection for $E(5)$:

Path	Count
BE	5

Table 7: Paths in the FP-Tree R leading up to E. Also known as **Conditional Sub-Database** for E.

Projected FP-Tree for B. Fourth and finally, we perform the projection for $B(5)$:

Path	Count
B	6

Table 8: Paths in the FP-Tree R leading up to B. Also known as **Conditional Sub-Database** for B.

The projected FP-Trees (not null) are shown in Figure 4.

1.1.5 Step 5.

Having projected the original FP-Tree R for all the items we are now ready to mine the projected FP-Trees for frequent itemsets.

Frequent Itemsets for D. First, we look at the projected FP-Tree of D. Here, notice that in the leaves of R_D we have C which is an infrequent item (support count is equal to 2 in the projected / conditional sub-database), which we can remove. Thus, the remaining part of the tree is a path ending the recursive procedure. We can now mine the frequent itemsets with prefix $P = D$ by

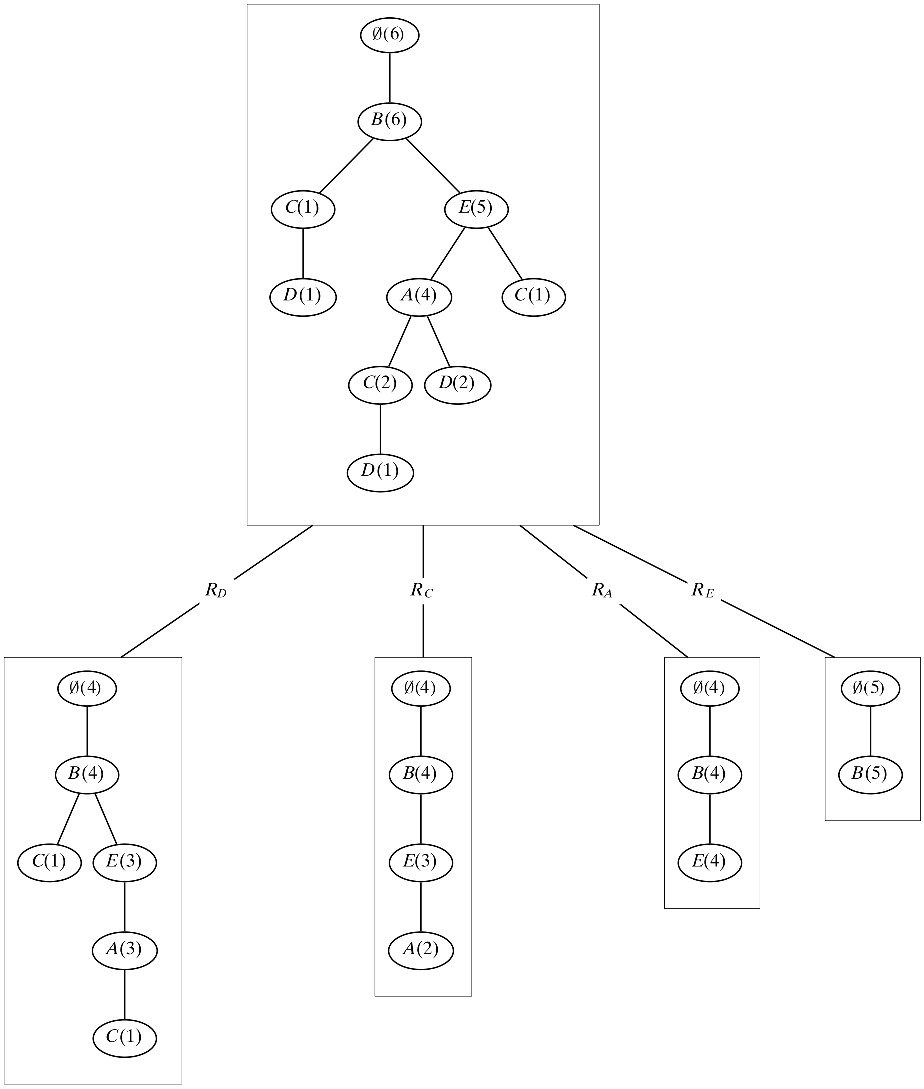


Figure 4: Projected FP-Tree for all the items.

enumerating all the possible subsets from this path as: $\{DA(3), DE(3), DB(4), DAE(3), DAB(3), DEB(3), DBAE(3)\}$.

Frequent Itemsets for C. Second, we look at the projected FP-Tree of C. Here, notice that in the leaves of R_C we have A which is an infrequent item (support count is equal to 2 in the projected / conditional sub-database), which we can remove. Thus, the remaining part of the tree is a path ending the recursive procedure. We can now mine the frequent itemsets with prefix $P = C$ by enumerating all the possible subsets from this path as: $\{CE(3), CB(4), CBE(4)\}$.

Frequent Itemsets for A. Third, we look at the projected FP-Tree of A. Since the projected FP-Tree is a path we terminate the recursive procedure. We can now mine the frequent itemsets with prefix $P = A$ by enumerating all the possible subsets from this path as: $\{AE(4), AB(4), ABE(4)\}$.

Frequent Itemsets for E. Fourth and finally, we look at the projected FP-Tree of E. Since the projected FP-Tree is a path we terminate the recursive procedure. We can now mine the frequent itemsets with prefix $P = E$ by enumerating all the possible subsets from this path as: $\{EB(5)\}$.

1.1.6 Table Representation Summarizing the Procedure

Alternatively, we may represent this entire process in a Tabular format as below:

Item	**Conditional Pattern Base**	**Conditional FP-Tree**	**Frequent Itemsets**
D	{B,C,D:1}, {B,E,A,C,D:1}, {B,E,A,D:2}	⟨B:4, C:1⟩ ⟨B:4, E:3, A:3, C:1⟩	{D,A:3}, {D,E:3}, {D,B:4}, {D,A,E:3}, {D,A,B:3}, {D,E,B:3}, {D,B,A,E:3}
C	{B,C:1}, {B,E,A,C:2}, {B,E,C:1}	⟨B:4, E:3, A:2⟩	{C,E:3}, {C,B:4}, {C,B,E:3}
A	{B,E,A:4}	⟨B:4, E:4⟩	{A,E:4}, {A,B:4}, {A,B,E:4}
E	{B, E:5}	⟨B:5⟩	{E,B:5}

Table 9: Summarizing the FP-Growth algorithm results.

1.2 Example 2

Second we will see an example of FP-Growth where we are doing repeated recursion on the projected databases. Consider the transaction database in Table 10 for which we have to identify frequent itemsets with minimum support equal to 3.

ID	**Transaction**
1	ABG
2	ABCD
3	ACJ
4	BC
5	ACH
6	BCL
7	ABCD
8	ABCDE
9	ABK

Table 10: Transaction Database

1.2.1 Step 1.

Sort the items in the transaction database by their support (see Table 11).

Item	Support
A	7
B	7
C	7
D	3
E	1
G	1
H	1
J	1
K	1
L	1

Table 11: Support item support for transaction database in Table 10.

1.2.2 Step 2.

Sort the order of the items in each transaction of the database with descending order of support (see Table 3). For example, for the first transaction ABG we sort the items in the transaction with descending order of support $A(7)\,B(7)\,G(1)$.

ID	Transaction
1	ABG
2	ABCD
3	ACJ
4	BC
5	ACH
6	BCL
7	ABCD
8	ABCDE
9	ABK

Table 12: Transactions with items sorted by their support for transaction database in Table 10.

1.2.3 Step 3.

Construct the FP-Tree (call it R) step-by-step by adding each of the transactions (where items in them are reordered with descending support). The FP-tree R serves as an index in lieu of the original database. The complete FP-Tree for the transaction database in Table 12 is given in Figure 5.

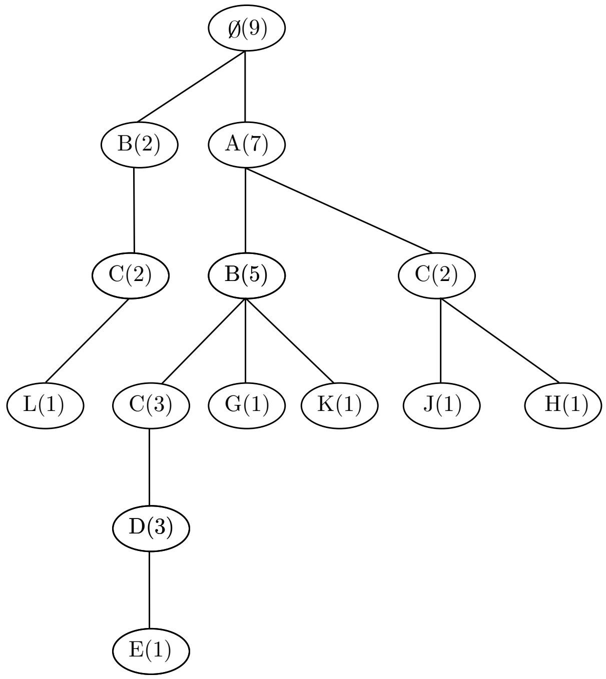


Figure 5: Projected FP-Tree for all the items for transaction database in Table 12.

Note that in the FP-Tree in Figure 5, since the support of the items $\{L, E, G, K, J, H\}$ is 1 (less than minimum support of 3) we can remove them from the tree.

1.2.4 Step 4.

Given the FP-Tree (R), projected (conditional) FP-trees (databases) are built for each frequent item i in R in increasing order of support. That is, we now project the FP-Tree R for the following items in order: $D(3)\,C(7)\,B(7)\,A(7)$ (note that we have already removed the infrequent items).

Projected FP-Tree for D. First, we perform the projection for $D(4)$. For the projection note down all the paths that lead up to D in the FP-Tree R and also note down the count that is associated with D in that path. This is also known as **Conditional Sub-Database** that we will use for the projection. For D we have,

Path	Count
ABCD	3

Table 13: Paths in the FP-Tree R leading up to D. Also known as **Conditional Sub-Database** for D.

Since the conditional sub-database for D consists of only one transaction the projected tree is a path $\emptyset(3) \rightarrow A(3) \rightarrow B(3) \rightarrow C(3)$.

Projected FP-Tree for C. For C we have the conditional sub-database is given in Table 14.

Path	Count
AC	2
ABC	3
BC	2

Table 14: Paths in the FP-Tree R leading up to C. Also known as **Conditional Sub-Database** for C.

The projected/ conditional FP-Tree for C — R_C is given in Figure 6.

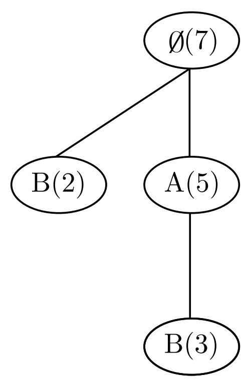


Figure 6: Projected FP-Tree for all the items for transaction database in Table 12.

Projected FP-Tree for CB. Note that since the projected tree for C is not a path we have to recursively project R_C. To do so, we take the next item with the lowest support count in the sub-database for C: $\{B(5)\}$. For R_{CB} we have, the following conditional sub-database:

Path	Count
B	2
AB	3

Table 15: Conditional Sub-Database for R_{CB}.

The projected FP-Tree R_{CB} will consist of only one path: $\emptyset(5) \rightarrow A(3)$.

Projected FP-Tree for CA. The only transaction for the projected FP-Tree R_{CA} is: $A(5)$. Consequently, the projected FP-Tree R_{CA} is $\emptyset(5)$.

Projected FP-Tree for B. For the projected tree for B we have,

Path	Count
B	2
AB	5

Table 16: Paths in the FP-Tree R leading up to B. Also known as **Conditional Sub-Database** for B.

From Table 16, we can construct the projected FP-Tree R_B which will consist of only one path: $\emptyset(7) \rightarrow A(5)$.

Projected FP-Tree for A. For the projected tree for A we have,

Path	Count
A	7

Table 17: Paths in the FP-Tree R leading up to A. Also known as **Conditional Sub-Database** for A.

From Table 17, we can construct the projected FP-Tree R_A which is essentially an empty set: $\emptyset(7)$.

1.2.5 Step 5.

Having projected the original FP-Tree R for all the items we are now ready to mine the projected FP-Trees for frequent itemsets.

Frequent Itemsets for D. Since, the projected FP-Tree R_D is a path, we can end the recursive procedure. We can now mine the frequent itemsets with prefix $P = D$ by enumerating all the possible subsets from this path as: $\{DC(3), DB(3), DA(3), DBC(3), DAB(3), DAC(3), DABC(3)\}$.

Frequent Itemsets for C. Second, we look at the projected FP-Tree of C. Here, notice that in the leaves of R_C we have B which is a frequent item (support count is equal to 5 in the projected / conditional sub-database), which we can not remove. Thus, we recursively project to obtain the projected FP-Tree R_{CB}. Since, R_{CB} is a path we can now mine the frequent itemsets with prefix $P = CB$ by enumerating all the possible subsets from this path as: $\{CB(5), CBA(3)\}$.

We also recursively project R_C to obtain the projected FP-Tree R_{CA}. Since, R_{CA} is a path we can now mine the frequent itemsets with prefix $P = CA$ by enumerating all the possible subsets from this path as: $\{CA(5)\}$.

Frequent Itemsets for B. Third and finally, we look at the projected FP-Tree of B. Since the projected FP-Tree is a path we terminate the recursive procedure. We can now mine the frequent itemsets with prefix $P = B$ by enumerating all the possible subsets from this path as: $\{BA(5)\}$.

1.2.6 Table Representation Summarizing the Procedure

Alternatively, we may represent this entire process in a Tabular format as below:

Item	**Conditional Pattern Base**	**Conditional FP-Tree**	**Frequent Itemsets**
D	{A,B,C,D:3}	$\langle$A:3, B:3, C:3$\rangle$	{D,C:3}, {D,B:3}, {D,A:3}, {D,B,C:3}, {D,A,B:3}, {D,A,C:3}, {D,A,B,C:3}
C	{A,C:2}, {A,B,C:3}, {B,C:2}	$\langle$A:5, B:3$\rangle$$\langle B:2\rangle$	
CB	{B:2}, {A,B:3}	$\langle$A:3$\rangle$	{C,B:5}, {C,B,A:3}
CA	{A:5}	$\langle \emptyset : 5\rangle$	{C,A:5}
B	{B:2}, {A,B:5}	$\langle$A:5$\rangle$	{B,A:5}
A	{A:7}	$\langle \emptyset : 7\rangle$	{A:7}

Table 18: Summarizing the FP-Growth algorithm results.