TDT4300 - Assignment 2

Andreas B. Berg

19.03.2021

1 Apriori Algorithm

Given the data in Table 1 (market basket transaction), your task is to describe the purchasing behavior of customers in the form of association rules. First, you need to generate the frequent itemsets and second, you need to generate the association rules. Apply the Apriori algorithm for following tasks and describe thoroughly the process and the outcome of each step.

a

Generate frequent 2-, 3- and 4-itemsets using the $F_{k-1} \times F_{k-1}$ method. Consider the support threshold $minsup = 0.5$ and use the data presented in Table 1.

We know the formula for support s is as follows:

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \tag{1}$$

where N is the total number of transactions and σ is the count of the itemset. With $minsup = 0.5$ and 10 entries, this means that an itemset must be counted at least $0.5 * 10 = 5$ times to be considered frequent.

K = 1

- Create a table with all frequent 1-itemsets
- Prune candidates with count less than minimum count (grayed out in the cells below

Itemset	**Count**
{A}	6
{B}	7
{C}	10
{D}	3
{E}	6
{F}	2
{G}	8
{H}	7

K = 2

- Merge two frequent (k-1)-itemsets if their first (k-2) items are identical (see table below)
- Prune candidate itemsets containing infrequent subsets of length k (grayed out and not counted below)

- Remove candidates with count less than minimum count (grayed out in the table below)

Itemset	Count
{A, B}	5
{A, C}	6
{A, E}	2
{A, G}	6
{A, H}	5
{B, C}	7
{B, E}	4
{B, G}	7
{B, H}	4
{C, E}	6
{C, G}	8
{C, H}	7
{E, G}	4
{E, H}	3
{G, H}	5

K = 3

- Merge two frequent (k-1)-itemsets if their first (k-2) items are identical (see table below)
- Prune candidate itemsets containing infrequent subsets of length k (grayed out and not counted below)
- Remove candidates with count less than minimum count (grayed out in the table below)

Itemset	Count
{A, B, C}	5
{A, B, G}	5
{A, B, H}	
{A, C, G}	6
{A, C, H}	5
{A, G, H}	5
{B, C, G}	7
{C, E, G}	
{C, E, H}	
{C, G, H}	5

K = 4

- Create a table with all combinations of F_{k-1} that gives a K-tuple
- Prune candidate itemsets containing infrequent subsets of length k (grayed out and not counted below)
- Remove candidates with count less than minimum count (grayed out in the table below)

Itemset	Count
{A, B, C, G}	5
{A, C, G, H}	5

b

Using the frequent 4-itemsets from the previous task, generate association rules. Consider the confidence threshold $minconf = 0.8$.

We have the following two 4-itemsets from (a):

Itemset	**Count**
{A, B, C, G}	5
{A, C, G, H}	5

We know that we can calculate confidence c using the following formula:

$$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{2}$$

Starting with itemset {A, B, C, G}, we have the following itemsets, with their sigma. Note that confidence have an anti-monotone property in some cases:

$$c(AB \rightarrow C) \geq c(A \rightarrow BC). \tag{3}$$

This means that we can eliminate some itemsets without directly calculating their confidence. These are grayed out in the table.

Nr.	**X**	**Y**	$\sigma(X \cup Y)$	$\sigma(X)$	$c(X \rightarrow Y)$
1	{A, B, C}	{G}	5	8	0.625
2	{A, B, G}	{C}	5	10	0.5
3	{A, C, G}	{B}	5	7	0.714
4	{B, C, G}	{A}	5	6	0.833
5	{A, B}	{C, G}	Rule (3) on row 1 and 2		
6	{A, C}	{B, G}	Rule (3) on row 1 and 3		
7	{A, G}	{B, C}	Rule (3) on row 2 and 3		
8	{B, C}	{A, G}	Rule (3) on row 1		
9	{B, G}	{A, C}	Rule (3) on row 2		
10	{C, G}	{A, B}	Rule (3) on row 3		

Due to rule (3), there is no need to calculate any further itemsets, as none will have confidence above 0.8. From this, we can see that the only association rule we can get from this itemset is {B, C, G} $\rightarrow$ {A}, with a confidence of 0.833.

Starting with itemset {A, C, G, H}, we have the following itemsets, with their sigma.

Nr.	**X**	**Y**	$\sigma(X \cup Y)$	$\sigma(X)$	$c(X \rightarrow Y)$
1	{A, C, G}	{H}	5	6	0.833
2	{A, C, H}	{G}	5	5	1
3	{A, G, H}	{C}	5	5	1
4	{C, G, H}	{A}	5	5	1
5	{A, C}	{G, H}	5	6	0.833
6	{A, G}	{C, H}	5	6	0.833
7	{C, G}	{A, H}	5	8	0.625
8	{A, H}	{C, G}	5	5	1
9	{C, H}	{A, G}	5	7	0.714
10	{G, H}	{A, C}	5	5	1
11	{A}	{C, G, H}	5	6	0.833
12	{C}	{A, G, H}	Rule (3) on row 7 and 9 (0.5)		
13	{G}	{A, C, H}	Rule (3) on row 7 (0.625)		
14	{H}	{A, C, G}	Rule (3) on row 9 (0.714)		

From these two itemsets, we get the following association rules:

Rule	Confidence
{A, C, G} → {H}	0.833
{A, C, H} → {G}	1
{A, G, H} → {C}	1
{B, C, G} → {A}	0.833
{C, G, H} → {A}	1
{A, C} → {G, H}	0.833
{A, G} → {C, H}	0.833
{A, H} → {C, G}	1
{G, H} → {A, C}	1
{A} → {C, G, H}	0.833

2 FP-Growth Algorithm

Use the Frequent Pattern Growth algorithm to discover the frequent itemsets in the given transaction dataset (Table 1). Consider the support threshold $minsup = 0.5$. Construct an FP-tree and mine the frequent itemsets by creating conditional (sub)pattern bases. Use the table notation with columns: item, conditional pattern base, conditional FP-tree, frequent patterns generated. The recursive steps of the FP-Growth algorithm must be clearly captured using the aforementioned table notation. Sort items alphabetically in case of ties in the item support. **Describe thoroughly the process and the outcome of each step.**

Note: With $minsup = 0.5$ and 10 transactions, we need to get support of 5 to consider itemsets to be frequent.

Step 1 - Obtain the transaction database

See table 1 in assignment.

Step 2 - Sort the items in the transaction database by their support

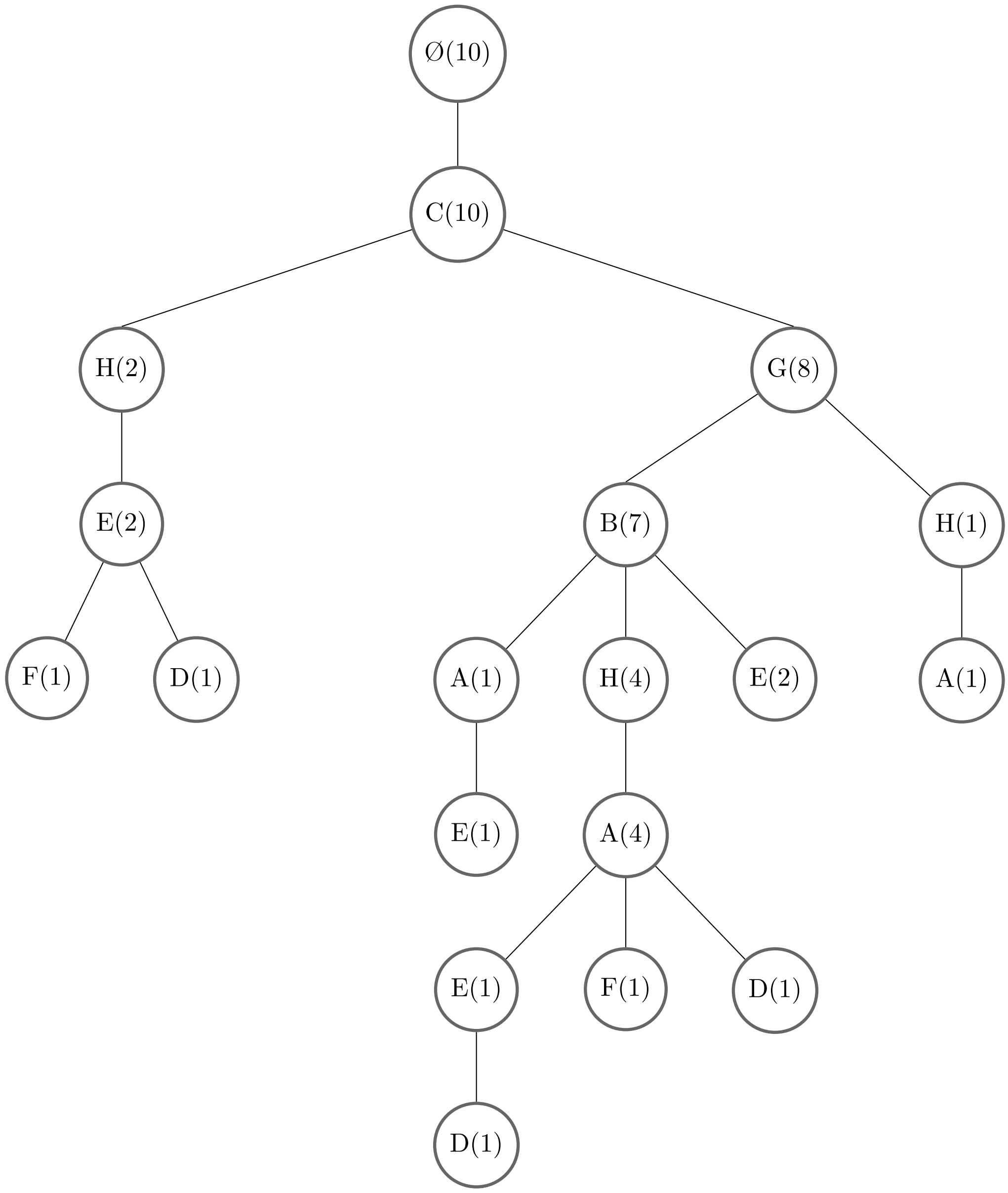
Item	Support count
C	10
G	8
B	7
H	7
A	6
E	6
D	3
F	2

Step 3 - Sort the order of the items in each transaction with descending order of support

TID	Items
110	C, G, B, H, A, F
111	C, G, B, A, E
112	C, H, E, F
113	C, G, B, H, A
114	C, H, E, D
115	C, G, B, E
116	C, G, B, H, A, D
117	C, G, B, E
118	C, G, H, A
119	C, G, B, H, A, E, D

Step 4 - Construct FP-tree step-by-step by adding transactions

We start with the most common item (in this case, C) as root, and then add paths found in the transactions. To avoid very a very long paper, I will only show the resulting FP-tree.

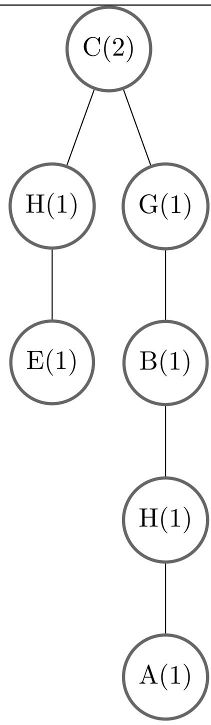
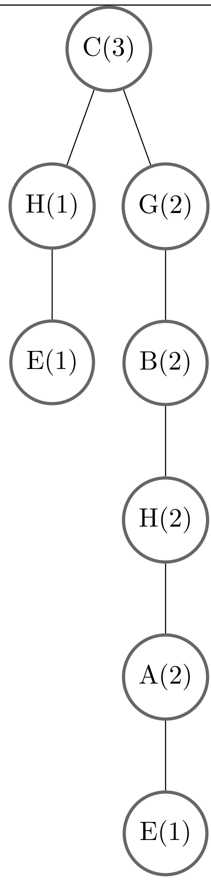


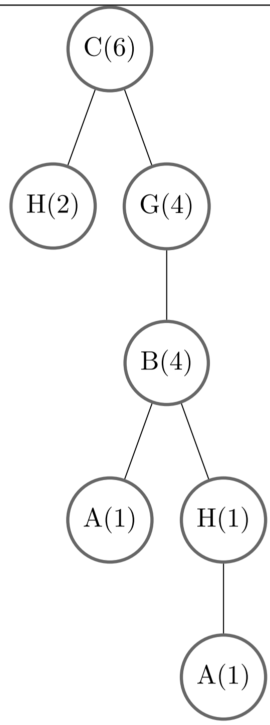
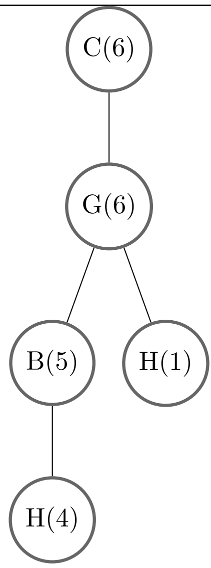
Note: The order here might not make sense. This is simply because I struggled with space in my notes while sketching this, and then transferred the sketch almost directly from my notebook to this document.

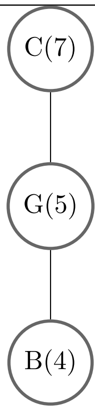
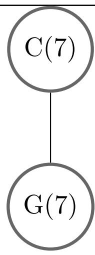
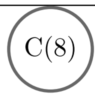
Step 5 - Build conditional FP-trees for each frequent item in increasing order of support

Note: I have, according with the algorithm, excluded all sub-trees that do not satisfy the minimum support count requirement of 5. This means that the column "Conditional FP-tree" in the table below only contains nodes with a count greater than or equal to 5.

I have opted to not redraw the conditional FP-trees in a similar fashion as above, as they are all linear. Instead, each {name : count} represents one node, with item name and count as elements.

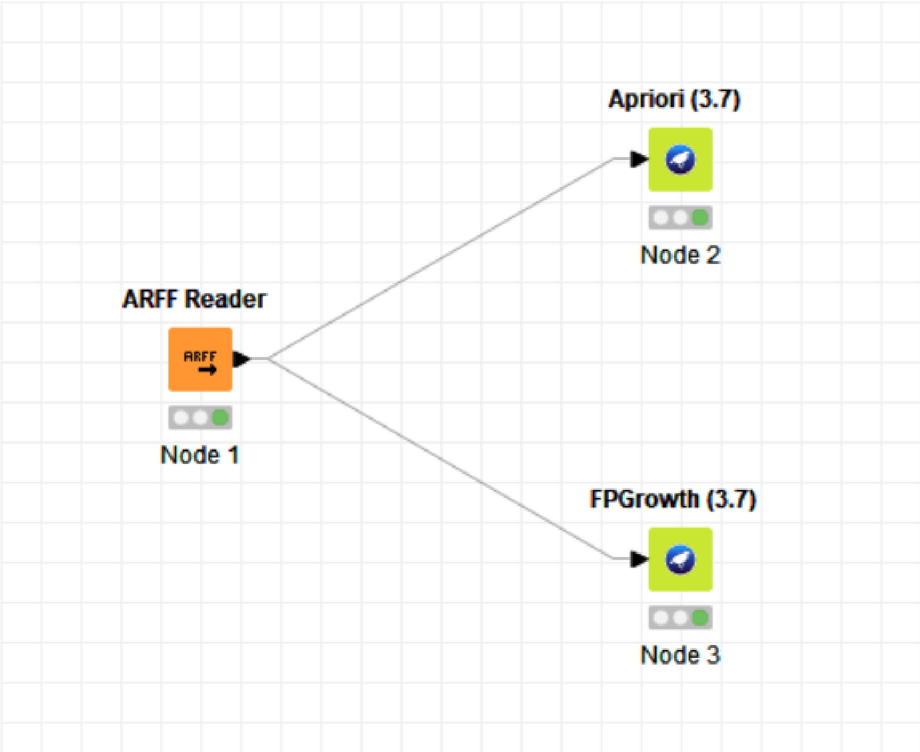
Item	**Conditional pattern base**	**Conditional FP-tree**	**Frequent patterns generated**
F	{C, H, E} : 1 {C, G, B, H, A} : 1		None, as count < minimum support count of 5 for all nodes
D	{C, H, E} : 1 {C, G, B, H, A} : 1 {C, G, B, H, A, E} : 1		None, as count < minimum support count of 5 for all nodes

Item	**Conditional pattern base**	**Conditional FP-tree**	**Frequent patterns generated**
E	{C, H} : 2 {C, G, B, A} : 1 {C, G, B} : 2 {C, G, B, H, A} : 1		{C, E} : 6
A	{C, G, H} : 1 {C, G, B} : 1 {C, G, B, H} : 4		{A, C} : 6 {A, G} : 6 {A, B} : 5 {A, H} : 5 {A, C, G} : 6 {A, B, C} : 5 {A, B, G} : 5 {A, C, H} : 5 {A, G, H} : 5 {A, B, C, G} : 5 {A, C, G, H} : 5

Item	**Conditional pattern base**	**Conditional FP-tree**	**Frequent patterns generated**
H	{C} : 2 {C, G} : 1 {C, G, B} : 4		{C, H} : 7 {G, H} : 5 {C, G, H} : 5
B	{C, G} : 7		{B, C} : 7 {B, G} : 7 {B, C, G} : 7
G	{C} : 8		{C, G} : 8
C			

3 KNIME

Workflow



Note: KNIME found multiple rules form the dataset that I didn't find. These are based on the 1, 2 and 3-itemsets it found. As the task in (1) and (2) was to find rules based on only the 4-itemsets, these are skipped above. I hope I have read the assignment correctly!

All rules based on the 4-itemsets are similar as the ones found above.

Apriori output

apriori.txt

```
Apriori
=======

Minimum support: 0.5 5 instances
Minimum metric <confidence>: 0.8
Number of cycles performed: 10

Generated sets of large itemsets:

Size of set of large itemsets L1: 6

Large Itemsets L1:
A=t 6
B=t 7
C=t 10
E=t 6
G=t 8
H=t 7

Size of set of large itemsets L2: 10

Large Itemsets L2:
A=t B=t 5
A=t C=t 6
A=t G=t 6
A=t H=t 5
B=t C=t 7
B=t G=t 7
C=t E=t 6
C=t G=t 8
C=t H=t 7
G=t H=t 5

Size of set of large itemsets L3: 7

Large Itemsets L3:
A=t B=t C=t 5
A=t B=t G=t 5
A=t C=t G=t 6
A=t C=t H=t 5
A=t G=t H=t 5
B=t C=t G=t 7
C=t G=t H=t 5

Size of set of large itemsets L4: 2

Large Itemsets L4:
A=t B=t C=t G=t 5
A=t C=t G=t H=t 5

Best rules found:

  1. G=t 8 ==> C=t 8    <conf:1> lift:1 lev:0 [0] conv:0
  2. B=t 7 ==> C=t 7    <conf:1> lift:1 lev:0 [0] conv:0
  3. B=t 7 ==> G=t 7    <conf:1> lift:1.25 lev:0.14 [1] conv:1.4
  4. H=t 7 ==> C=t 7    <conf:1> lift:1 lev:0 [0] conv:0
  5. B=t G=t 7 ==> C=t 7    <conf:1> lift:1 lev:0 [0] conv:0
  6. B=t C=t 7 ==> G=t 7    <conf:1> lift:1.25 lev:0.14 [1] conv:1.4
  7. B=t 7 ==> C=t G=t 7    <conf:1> lift:1.25 lev:0.14 [1] conv:1.4
  8. A=t 6 ==> C=t 6    <conf:1> lift:1 lev:0 [0] conv:0
  9. A=t 6 ==> G=t 6    <conf:1> lift:1.25 lev:0.12 [1] conv:1.2
 10. E=t 6 ==> C=t 6    <conf:1> lift:1 lev:0 [0] conv:0
 11. A=t G=t 6 ==> C=t 6    <conf:1> lift:1 lev:0 [0] conv:0
 12. A=t C=t 6 ==> G=t 6    <conf:1> lift:1.25 lev:0.12 [1] conv:1.2
 13. A=t 6 ==> C=t G=t 6    <conf:1> lift:1.25 lev:0.12 [1] conv:1.2
 14. A=t B=t 5 ==> C=t 5    <conf:1> lift:1 lev:0 [0] conv:0
 15. A=t B=t 5 ==> G=t 5    <conf:1> lift:1.25 lev:0.1 [0] conv:1
```

```
16. A=t H=t 5 ==> C=t 5    <conf:1> lift:1 lev:0 [0] conv:0
17. G=t H=t 5 ==> A=t 5    <conf:1> lift:1.67 lev:0.2 [2] conv:2
18. A=t H=t 5 ==> G=t 5    <conf:1> lift:1.25 lev:0.1 [0] conv:1
19. G=t H=t 5 ==> C=t 5    <conf:1> lift:1 lev:0 [0] conv:0
20. A=t B=t G=t 5 ==> C=t 5    <conf:1> lift:1 lev:0 [0] conv:0
21. A=t B=t C=t 5 ==> G=t 5    <conf:1> lift:1.25 lev:0.1 [0] conv:1
22. A=t B=t 5 ==> C=t G=t 5    <conf:1> lift:1.25 lev:0.1 [0] conv:1
23. C=t G=t H=t 5 ==> A=t 5    <conf:1> lift:1.67 lev:0.2 [2] conv:2
24. A=t G=t H=t 5 ==> C=t 5    <conf:1> lift:1 lev:0 [0] conv:0
25. A=t C=t H=t 5 ==> G=t 5    <conf:1> lift:1.25 lev:0.1 [0] conv:1
26. G=t H=t 5 ==> A=t C=t 5    <conf:1> lift:1.67 lev:0.2 [2] conv:2
27. A=t H=t 5 ==> C=t G=t 5    <conf:1> lift:1.25 lev:0.1 [0] conv:1
28. G=t 8 ==> B=t 7    <conf:0.88> lift:1.25 lev:0.14 [1] conv:1.2
29. C=t G=t 8 ==> B=t 7    <conf:0.88> lift:1.25 lev:0.14 [1] conv:1.2
30. G=t 8 ==> B=t C=t 7    <conf:0.88> lift:1.25 lev:0.14 [1] conv:1.2
31. A=t 6 ==> B=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
32. A=t 6 ==> H=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
33. A=t C=t 6 ==> B=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
34. A=t 6 ==> B=t C=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
35. A=t G=t 6 ==> B=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
36. A=t 6 ==> B=t G=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
37. A=t C=t 6 ==> H=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
38. A=t 6 ==> C=t H=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
39. A=t G=t 6 ==> H=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
40. A=t 6 ==> G=t H=t 5    <conf:0.83> lift:1.67 lev:0.2 [2] conv:1.5
41. A=t C=t G=t 6 ==> B=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
42. A=t G=t 6 ==> B=t C=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
43. A=t C=t 6 ==> B=t G=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
44. A=t 6 ==> B=t C=t G=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
45. A=t C=t G=t 6 ==> H=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
46. A=t G=t 6 ==> C=t H=t 5    <conf:0.83> lift:1.19 lev:0.08 [0] conv:0.9
47. A=t C=t 6 ==> G=t H=t 5    <conf:0.83> lift:1.67 lev:0.2 [2] conv:1.5
48. A=t 6 ==> C=t G=t H=t 5    <conf:0.83> lift:1.67 lev:0.2 [2] conv:1.5
49. C=t 10 ==> G=t 8    <conf:0.8> lift:1 lev:0 [0] conv:0.67
```

FP-Growth output

FP-growth.txt

```
FPGrowth found 48 rules

 1. [G=t]: 8 ==> [C=t]: 8   <conf:1> lift:1 lev:0 conv:0
 2. [H=t]: 7 ==> [C=t]: 7   <conf:1> lift:1 lev:0 conv:0
 3. [B=t]: 7 ==> [C=t]: 7   <conf:1> lift:1 lev:0 conv:0
 4. [E=t]: 6 ==> [C=t]: 6   <conf:1> lift:1 lev:0 conv:0
 5. [A=t]: 6 ==> [C=t]: 6   <conf:1> lift:1 lev:0 conv:0
 6. [B=t]: 7 ==> [G=t]: 7   <conf:1> lift:1.25 lev:0.14 conv:1.4
 7. [A=t]: 6 ==> [G=t]: 6   <conf:1> lift:1.25 lev:0.12 conv:1.2
 8. [G=t, H=t]: 5 ==> [C=t]: 5   <conf:1> lift:1 lev:0 conv:0
 9. [B=t]: 7 ==> [C=t, G=t]: 7   <conf:1> lift:1.25 lev:0.14 conv:1.4
10. [C=t, B=t]: 7 ==> [G=t]: 7   <conf:1> lift:1.25 lev:0.14 conv:1.4
11. [G=t, B=t]: 7 ==> [C=t]: 7   <conf:1> lift:1 lev:0 conv:0
12. [A=t]: 6 ==> [C=t, G=t]: 6   <conf:1> lift:1.25 lev:0.12 conv:1.2
13. [C=t, A=t]: 6 ==> [G=t]: 6   <conf:1> lift:1.25 lev:0.12 conv:1.2
14. [G=t, A=t]: 6 ==> [C=t]: 6   <conf:1> lift:1 lev:0 conv:0
15. [H=t, A=t]: 5 ==> [C=t]: 5   <conf:1> lift:1 lev:0 conv:0
16. [B=t, A=t]: 5 ==> [C=t]: 5   <conf:1> lift:1 lev:0 conv:0
17. [G=t, H=t]: 5 ==> [A=t]: 5   <conf:1> lift:1.67 lev:0.2 conv:2
18. [H=t, A=t]: 5 ==> [G=t]: 5   <conf:1> lift:1.25 lev:0.1 conv:1
19. [B=t, A=t]: 5 ==> [G=t]: 5   <conf:1> lift:1.25 lev:0.1 conv:1
20. [G=t, H=t]: 5 ==> [C=t, A=t]: 5   <conf:1> lift:1.67 lev:0.2 conv:2
21. [C=t, G=t, H=t]: 5 ==> [A=t]: 5   <conf:1> lift:1.67 lev:0.2 conv:2
22. [H=t, A=t]: 5 ==> [C=t, G=t]: 5   <conf:1> lift:1.25 lev:0.1 conv:1
23. [C=t, H=t, A=t]: 5 ==> [G=t]: 5   <conf:1> lift:1.25 lev:0.1 conv:1
24. [G=t, H=t, A=t]: 5 ==> [C=t]: 5   <conf:1> lift:1 lev:0 conv:0
25. [B=t, A=t]: 5 ==> [C=t, G=t]: 5   <conf:1> lift:1.25 lev:0.1 conv:1
26. [C=t, B=t, A=t]: 5 ==> [G=t]: 5   <conf:1> lift:1.25 lev:0.1 conv:1
27. [G=t, B=t, A=t]: 5 ==> [C=t]: 5   <conf:1> lift:1 lev:0 conv:0
28. [G=t]: 8 ==> [B=t]: 7   <conf:0.88> lift:1.25 lev:0.14 conv:1.2
29. [G=t]: 8 ==> [C=t, B=t]: 7   <conf:0.88> lift:1.25 lev:0.14 conv:1.2
30. [C=t, G=t]: 8 ==> [B=t]: 7   <conf:0.88> lift:1.25 lev:0.14 conv:1.2
31. [A=t]: 6 ==> [H=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
32. [A=t]: 6 ==> [B=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
33. [A=t]: 6 ==> [C=t, H=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
34. [C=t, A=t]: 6 ==> [H=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
35. [A=t]: 6 ==> [C=t, B=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
36. [C=t, A=t]: 6 ==> [B=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
37. [A=t]: 6 ==> [G=t, H=t]: 5   <conf:0.83> lift:1.67 lev:0.2 conv:1.5
38. [G=t, A=t]: 6 ==> [H=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
39. [A=t]: 6 ==> [G=t, B=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
40. [G=t, A=t]: 6 ==> [B=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
41. [A=t]: 6 ==> [C=t, G=t, H=t]: 5   <conf:0.83> lift:1.67 lev:0.2 conv:1.5
42. [C=t, A=t]: 6 ==> [G=t, H=t]: 5   <conf:0.83> lift:1.67 lev:0.2 conv:1.5
43. [G=t, A=t]: 6 ==> [C=t, H=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
44. [C=t, G=t, A=t]: 6 ==> [H=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
45. [A=t]: 6 ==> [C=t, G=t, B=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
46. [C=t, A=t]: 6 ==> [G=t, B=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
47. [G=t, A=t]: 6 ==> [C=t, B=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
48. [C=t, G=t, A=t]: 6 ==> [B=t]: 5   <conf:0.83> lift:1.19 lev:0.08 conv:0.9
```

4 Compact Representation of Frequent Itemsets

Given the compact representation of frequent itemsets in Table 2, use the appropriate algorithm to generate all frequent itemsets including the support counts. **Describe thoroughly each step of the algorithm and present the final result.**

Closed Frequent Itemsets	Support Count
{b}	10
{d}	13
{a, d}	11
{b, d}	8
{b, e}	6
{a, d, e}	7
{a, c, d}	6
{b, d, e}	5
{a, c, d, e}	5

Table 2: Closed frequent itemsets.

To solve this, I will apply "Algorithm 5.4 - Support counting using closed frequent itemsets" from Tan et al. (page 388).

Step 1 - find all frequent itemsets with maximum size

We can see that the maximum size (k_{max}) is 4, as the largest closed frequent itemset is {a, c, d, e}

Frequent itemset	Support count
{a, c, d, e}	5

Step 2 - Iterate through k $\in [k_{max} - 1, 1]$

- Find all frequent itemsets of size k
- Iterate through all itemsets found above
- If support count of an itemset is not given, it is the maximum of the support counts of its immediate supersets

Note that in the tables below, I have simply listed the frequent itemsets and their count. If the support count is not given in the assignment, then I have indicated which superset it comes from.

Note that several itemsets of size 2 and 3 are excluded (such as {a, b, c} and {a, b}). These have neither a count in Table 2 nor a frequent superset, and as such their support count must be below the *minsup*.

k = 3

Frequent itemsets of size 3

Itemset	Support count	Found from
{a, c, d}	6	Table 2
{a, c, e}	5	{a, c, d, e}
{a, d, e}	7	Table 2
{b, d, e}	5	Table 2
{c, d, e}	5	{a, c, d, e}

k = 2

Frequent itemsets of size 2

Itemset	Support count	Found from
{a, c}	6	{a, c, d}
{a, d}	11	Table 2
{a, e}	7	{a, d, e}
{b, d}	8	Table 2
{b, e}	6	Table 2
{c, d}	6	{a, c, d}
{c, e}	5	{a, c, e}, {c, d, e}
{d, e}	7	{a, d, e}

k = 1

Frequent itemsets of size 1

Itemset	Support count	Found from
{a}	11	{a, d}
{b}	10	Table 2
{c}	6	{a, c}, {c, d}
{d}	13	Table 2
{e}	7	{a, e}, {d, e}