Norwegian University of Science and Technology

TDT4300 — **Solution to Assignment 2**

ASSOCIATION ANALYSIS

tdt4300-undass@idi.ntnu.no

Spring 2022

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. Use the Apriori algorithm to find all frequent itemsets with minimum support 33.33% (i.e. minimum support count is 2).

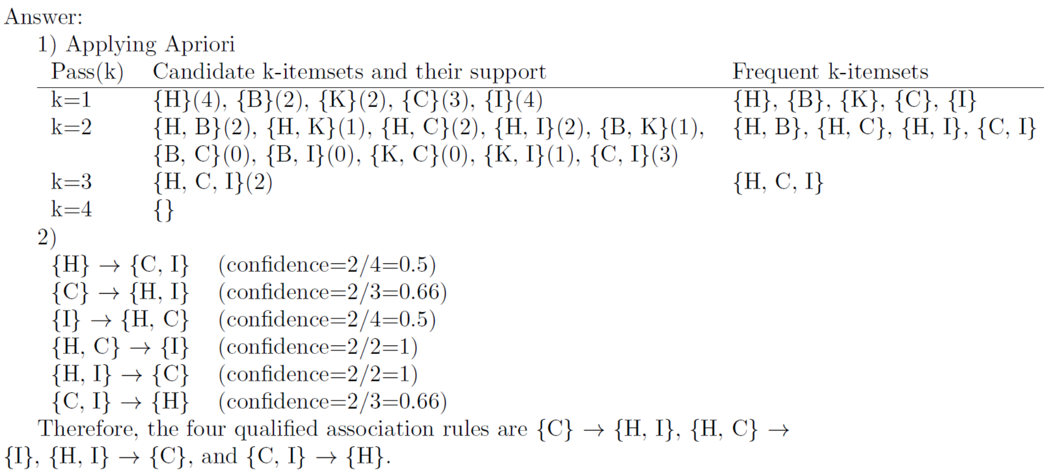
(a) Show thoroughly the steps on how the frequent itemsets are generated.

(b) one of the frequent itemsets is {H,C,I}. Find all association rules based on this set, given confidence threshold $c = 60\%$(it is not necessary to use Apriori to find the association rules, but show how confidence is calculated for each of the candidate rules that are based on {H,C,I}.)

TID	**Items**
T1	H, B, K
T2	H, B
T3	H, C, I
T4	C, I
T5	I, K
T6	H, C, I, U

Table 1: Market basket transactions1.

1.1 Solution



Answer:

1) Applying Apriori

Pass(k)	Candidate k-itemsets and their support	Frequent k-itemsets
k=1	{H}(4), {B}(2), {K}(2), {C}(3), {I}(4)	{H}, {B}, {K}, {C}, {I}
k=2	{H, B}(2), {H, K}(1), {H, C}(2), {H, I}(2), {B, K}(1), {B, C}(0), {B, I}(0), {K, C}(0), {K, I}(1), {C, I}(3)	{H, B}, {H, C}, {H, I}, {C, I}
k=3	{H, C, I}(2)	{H, C, I}
k=4	{}	

2)

{H} → {C, I} (confidence=2/4=0.5)
{C} → {H, I} (confidence=2/3=0.66)
{I} → {H, C} (confidence=2/4=0.5)
{H, C} → {I} (confidence=2/2=1)
{H, I} → {C} (confidence=2/2=1)
{C, I} → {H} (confidence=2/3=0.66)

Therefore, the four qualified association rules are {C} → {H, I}, {H, C} → {I}, {H, I} → {C}, and {C, I} → {H}.

Figure 1: Apriori

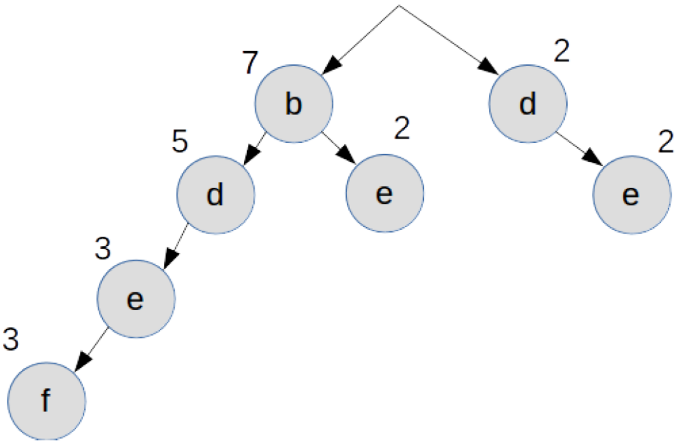
2 FP-Growth Algorithm

Use the Frequent Pattern Growth algorithm to discover the frequent itemsets in the given transaction dataset (Table 2). You are now going to use the FP-growth algorithm in order to find all frequent itemsets with minimum support of 22 % (i.e., minimum support count is 2). 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.** *Click the add symbol on the right and drag it to KNIME. You need to accept cookies first to see this button.*

TID	Items
T1	b, e, g
T2	b, d, i
T3	b, d, e, f
T4	a, d, e
T5	d, e
T6	b, d, j
T7	b, c, d, e, f
T8	b, d, e, f
T9	b, e, h

Table 2: Market basket transactions 2.

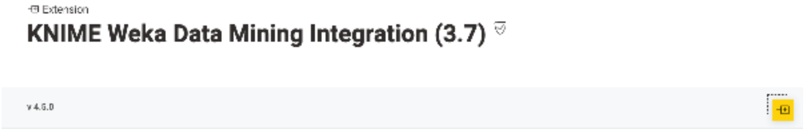
2.1 Solution



2)

Item	Conditional sub-database	Conditional FP-tree	Frequent Item sets
f	{{b,d,e}:3}	$\langle b:3, d:3, e:3\rangle$	{f}:3, {b,f}:3, {d,f}:3, {e,f}:3, {b,d,f}:3, {b,e,f}:3, {d,e,f}:3, {b,d,e,f}:3
e	{{b,d}:3, {b}:2, {d}:2}	$\langle b:5, d:3\rangle$, $\langle d:2\rangle$	{e}:7, {d,e}:5, {b,e}:5
ed	{{b}:3}	b:3	{b,d,e}:3
(eb	empty	empty)	
d	{{b}:5}	b:5	{d}:7, {b,d}:5
b	empty	empty	{b}:7

Figure 2: Fpgrowth



Extension

KNIME Weka Data Mining Integration (3.7)

v 4.6.0

Figure 3: Add Weka rules to KNIME

3 KNIME

For this task you will need to install and use the KNIME[1] data analytics platform. You are given a file *market_basket_transactions.arff* which contains the very same transaction as in Table 1. Your task is to implement two simple workflows for mining association rules, one implementing Apriori algorithm and second implementing FP-Growth algorithm. Use the WEKA nodes both for Apriori and FP-Growth. Use the same parameters as in the previous tasks, e.i. $minsup = 0.5$ and $minconf = 0.8$. **Present pictures of your workflows, and the outputs from both Apriori and FP-Growth nodes. Deliver also the exported KNIME workflows.**

Note: In Knime 4.X Weka nodes are not automatically included. Go to this link[2] and follow the directions under *Add to KNIME Analytics Platform* (drag the extension to your Workflow board in KNIME to install). *Click the add symbol on the right and drag it to KNIME. You need to accept cookies first to see this button.*

3.1 Solution

Figure 4 show the very simple KNIME workflow for mining association rules with Apriori and FP-Growth algorithm.

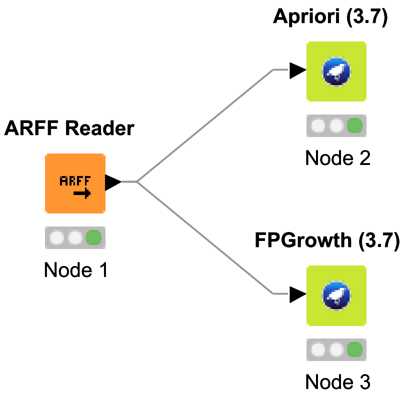


Figure 4: KNIME workflow.

[1] http://www.knime.org/downloads/overview

[2] https://hub.knime.com/knime/extensions/org.knime.features.ext.weka_3.7/latest

Given the *market_basket_transactions.arff* file (and parameters $minsup = 0.5$ and $minconf = 0.8$), the outputs are as follows.

Apriori node output:

```
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 L(1): 6

Size of set of large itemsets L(2): 11

Size of set of large itemsets L(3): 8

Size of set of large itemsets L(4): 2

Best rules found:

  1. B=t 8 ==> C=t 8    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
  2. G=t 8 ==> C=t 8    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
  3. H=t 7 ==> C=t 7    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
  4. B=t G=t 7 ==> C=t 7    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
  5. A=t 6 ==> C=t 6    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
  6. A=t 6 ==> G=t 6    <conf:(1)> lift:(1.25) lev:(0.12) [1] conv:(1.2)
  7. A=t G=t 6 ==> C=t 6    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
  8. A=t C=t 6 ==> G=t 6    <conf:(1)> lift:(1.25) lev:(0.12) [1] conv:(1.2)
  9. A=t 6 ==> C=t G=t 6    <conf:(1)> lift:(1.25) lev:(0.12) [1] conv:(1.2)
 10. E=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 11. A=t B=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 12. A=t B=t 5 ==> G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
 13. A=t H=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 14. G=t H=t 5 ==> A=t 5    <conf:(1)> lift:(1.67) lev:(0.2) [2] conv:(2)
 15. A=t H=t 5 ==> G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
 16. B=t H=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 17. G=t H=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 18. A=t B=t G=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 19. A=t B=t C=t 5 ==> G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
 20. A=t B=t 5 ==> C=t G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
 21. C=t G=t H=t 5 ==> A=t 5    <conf:(1)> lift:(1.67) lev:(0.2) [2] conv:(2)
 22. A=t G=t H=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 23. A=t C=t H=t 5 ==> G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
 24. G=t H=t 5 ==> A=t C=t 5    <conf:(1)> lift:(1.67) lev:(0.2) [2] conv:(2)
 25. A=t H=t 5 ==> C=t G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
 26. G=t 8 ==> B=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
 27. B=t 8 ==> G=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
 28. C=t G=t 8 ==> B=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
 29. B=t C=t 8 ==> G=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
 30. G=t 8 ==> B=t C=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
 31. B=t 8 ==> C=t G=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
 32. A=t 6 ==> B=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
 33. A=t 6 ==> H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 34. A=t C=t 6 ==> B=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
 35. A=t 6 ==> B=t C=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
 36. A=t G=t 6 ==> B=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
 37. A=t 6 ==> B=t G=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 38. A=t C=t 6 ==> H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 39. A=t 6 ==> C=t H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 40. A=t G=t 6 ==> H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 41. A=t 6 ==> G=t H=t 5    <conf:(0.83)> lift:(1.67) lev:(0.2) [2] conv:(1.5)
 42. A=t C=t G=t 6 ==> B=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
 43. A=t G=t 6 ==> B=t C=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
 44. A=t C=t 6 ==> B=t G=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 45. A=t 6 ==> B=t C=t G=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 46. A=t C=t G=t 6 ==> H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 47. A=t G=t 6 ==> C=t H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 48. A=t C=t 6 ==> G=t H=t 5    <conf:(0.83)> lift:(1.67) lev:(0.2) [2] conv:(1.5)
 49. A=t 6 ==> C=t G=t H=t 5    <conf:(0.83)> lift:(1.67) lev:(0.2) [2] conv:(1.5)
 50. C=t 10 ==> B=t 8    <conf:(0.8)> lift:(1) lev:(0) [0] conv:(0.67)
 51. C=t 10 ==> G=t 8    <conf:(0.8)> lift:(1) lev:(0) [0] conv:(0.67)
```

FP-Growth node output:

```
FPGrowth found 49 rules (displaying top 49)

 1. [G=t]: 8 ==> [C=t]: 8   <conf:(1)> lift:(1) lev:(0) conv:(0)
 2. [B=t]: 8 ==> [C=t]: 8   <conf:(1)> lift:(1) lev:(0) conv:(0)
 3. [H=t]: 7 ==> [C=t]: 7   <conf:(1)> lift:(1) lev:(0) conv:(0)
 4. [A=t]: 6 ==> [C=t]: 6   <conf:(1)> lift:(1) lev:(0) conv:(0)
 5. [E=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
 6. [A=t]: 6 ==> [G=t]: 6   <conf:(1)> lift:(1.25) lev:(0.12) conv:(1.2)
 7. [G=t, B=t]: 7 ==> [C=t]: 7   <conf:(1)> lift:(1) lev:(0) conv:(0)
 8. [G=t, H=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
 9. [A=t]: 6 ==> [C=t, G=t]: 6   <conf:(1)> lift:(1.25) lev:(0.12) conv:(1.2)
10. [C=t, A=t]: 6 ==> [G=t]: 6   <conf:(1)> lift:(1.25) lev:(0.12) conv:(1.2)
11. [G=t, A=t]: 6 ==> [C=t]: 6   <conf:(1)> lift:(1) lev:(0) conv:(0)
12. [B=t, H=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
13. [B=t, A=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
14. [H=t, A=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
15. [B=t, A=t]: 5 ==> [G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
16. [G=t, H=t]: 5 ==> [A=t]: 5   <conf:(1)> lift:(1.67) lev:(0.2) conv:(2)
17. [H=t, A=t]: 5 ==> [G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
18. [B=t, A=t]: 5 ==> [C=t, G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
19. [C=t, B=t, A=t]: 5 ==> [G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
20. [G=t, B=t, A=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
21. [G=t, H=t]: 5 ==> [C=t, A=t]: 5   <conf:(1)> lift:(1.67) lev:(0.2) conv:(2)
22. [C=t, G=t, H=t]: 5 ==> [A=t]: 5   <conf:(1)> lift:(1.67) lev:(0.2) conv:(2)
23. [H=t, A=t]: 5 ==> [C=t, G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
24. [C=t, H=t, A=t]: 5 ==> [G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
```

```
25. [G=t, H=t, A=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
26. [G=t]: 8 ==> [B=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
27. [B=t]: 8 ==> [G=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
28. [G=t]: 8 ==> [C=t, B=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
29. [C=t, G=t]: 8 ==> [B=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
30. [B=t]: 8 ==> [C=t, G=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
31. [C=t, B=t]: 8 ==> [G=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
32. [A=t]: 6 ==> [B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
33. [A=t]: 6 ==> [H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
34. [A=t]: 6 ==> [C=t, B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
35. [C=t, A=t]: 6 ==> [B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
36. [A=t]: 6 ==> [C=t, H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
37. [C=t, A=t]: 6 ==> [H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
38. [A=t]: 6 ==> [G=t, B=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
39. [G=t, A=t]: 6 ==> [B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
40. [A=t]: 6 ==> [G=t, H=t]: 5   <conf:(0.83)> lift:(1.67) lev:(0.2) conv:(1.5)
41. [G=t, A=t]: 6 ==> [H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
42. [A=t]: 6 ==> [C=t, G=t, B=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
43. [C=t, A=t]: 6 ==> [G=t, B=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
44. [G=t, A=t]: 6 ==> [C=t, B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
45. [C=t, G=t, A=t]: 6 ==> [B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
46. [A=t]: 6 ==> [C=t, G=t, H=t]: 5   <conf:(0.83)> lift:(1.67) lev:(0.2) conv:(1.5)
47. [C=t, A=t]: 6 ==> [G=t, H=t]: 5   <conf:(0.83)> lift:(1.67) lev:(0.2) conv:(1.5)
48. [G=t, A=t]: 6 ==> [C=t, H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
49. [C=t, G=t, A=t]: 6 ==> [H=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 3, 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}	7
{b, e}	8
{d, e}	6
{a, b, e}	7
{a, c, d}	6
{b, d, e}	4
{a, c, d, e}	5

Table 3: Closed frequent itemsets.

4.1 Solution

The frequent itemsets (including their support counts) were generated following the Algorithm 6.4 in the book Introduction to Data Mining (Tan et al.) at page 357. The calculations bellow depict the states of each iteration of the algorithm. The results are presented in the Table 4.

$F = \{\{a,\, c,\, d,\, e\},\, \{a,\, b,\, e\},\, \{a,\, c,\, d\},\, \{a,\, c,\, e\},\, \{a,\, d,\, e\},\, \{b,\, d,\, e\},\, \{c,\, d,\, e\},\, \{a,\, b\},$

$\{a,\, c\},\, \{a,\, d\},\, \{a,\, e\},\, \{b,\, d\},\, \{b,\, e\},\, \{c,\, d\},\, \{c,\, e\},\, \{d,\, e\},\, \{a\},\, \{b\},\, \{c\},\, \{d\},\, \{e\}\}$

$k_{max} = 4$

$F_{k_{max}} = \{\{a,\, c,\, d,\, e\}\}$

$k = 3$

$F_k = \{\{a,\, b,\, e\},\, \{a,\, c,\, d\},\, \{a,\, c,\, e\},\, \{a,\, d,\, e\},\, \{b,\, d,\, e\},\, \{c,\, d,\, e\}\}$

$sup(\{a,\, c,\, e\}) = max\{sup(\{a,\, c,\, d,\, e\})\} = 5$

$sup(\{a,\, d,\, e\}) = max\{sup(\{a,\, c,\, d,\, e\})\} = 5$

$sup(\{c,\, d,\, e\}) = max\{sup(\{a,\, c,\, d,\, e\})\} = 5$

$k = 2$

$F_k = \{\{a,\, b\},\, \{a,\, c\},\, \{a,\, d\},\, \{a,\, e\},\, \{b{,}d\},\, \{b{,}e\},\, \{c,\, d\},\, \{c,\, e\},\, \{d,\, e\}\}$

$sup(\{a,\, b\}) = max\{sup(\{a,\, b,\, e\})\} = 7$

$sup(\{a,\, c\}) = max\{sup(\{a,\, c,\, d\}){,}sup(\{a,\, c,\, e\})\} = 6$

$sup(\{a,\, e\}) = max\{sup(\{a,\, b,\, e\}){,}sup(\{a,\, c,\, e\}){,}sup(\{a,\, d,\, e\})\} = 7$

$sup(\{c,\, d\}) = max\{sup(\{a,\, c,\, d\}){,}sup(\{c,\, d,\, e\})\} = 6$

$sup(\{c,\, e\}) = max\{sup(\{a,\, c,\, e\}){,}sup(\{c,\, d,\, e\})\} = 5$

$k = 1$

$F_k = \{\{a\},\, \{b\},\, \{c\},\, \{d\},\, \{e\}\}$

$sup(\{a\}) = max\{sup(\{a,\, c\}){,}sup(\{a,\, d\}){,}sup(\{a,\, e\})\} = 11$

$sup(\{c\}) = max\{sup(\{a,\, c\}){,}sup(\{c,\, d\}){,}sup(\{c,\, e\})\} = 6$

$sup(\{e\}) = max\{sup(\{a,\, e\}){,}sup(\{b,\, e\}){,}sup(\{c,\, e\}){,}sup(\{d,\, e\}))\} = 8$

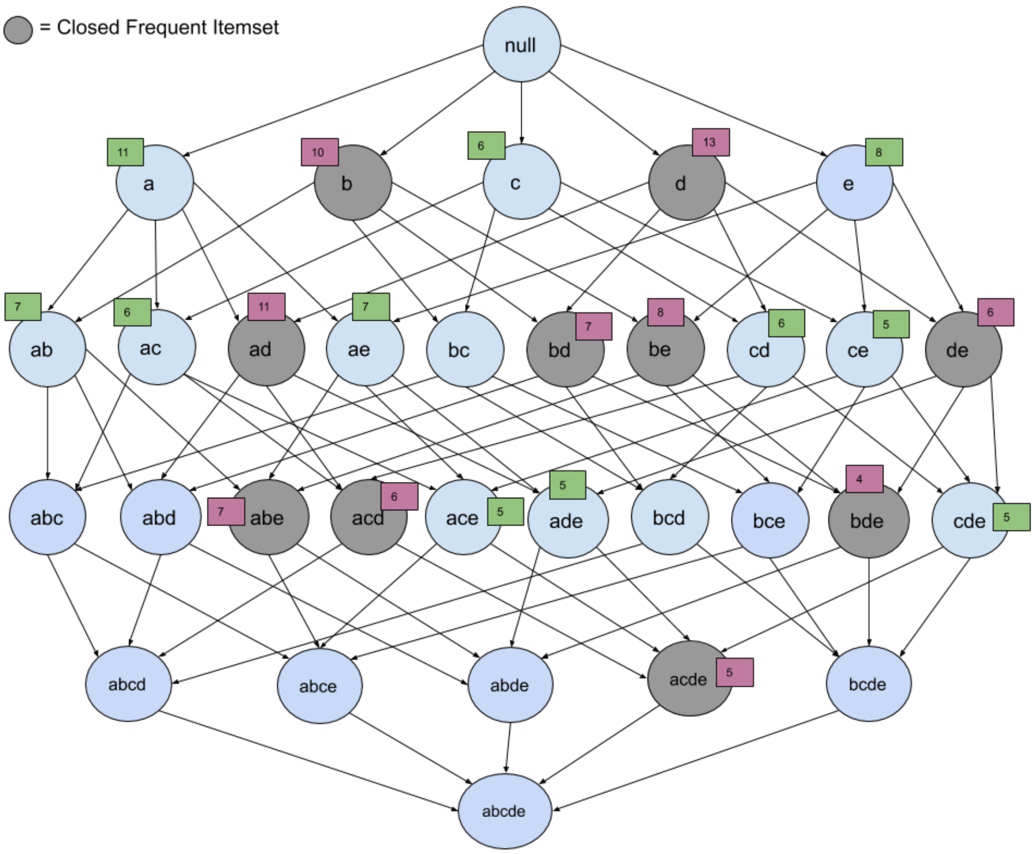


Figure 5: Illustration of closed frequent itemset

Frequent Itemsets	**Support Count**
{a, c, d, e}	5
{a, b, e}	7
{a, c, d}	6
{a, c, e}	5
{a, d, e}	5
{b, d, e}	4
{c, d, e}	5
{a, b}	7
{a, c}	6
{a, d}	11
{a, e}	7
{b, d}	7
{b, e}	8
{c, d}	6
{c, e}	5
{d, e}	6
{a}	11
{b}	10
{c}	6
{e}	8
{d}	13

Table 4: All frequent itemsets (including closed frequent itemsets).