Sondre Schirmer-Mikalsen, Arash Aminpour

TDT4300 Exercise 2

Task 1

For the first candidate set, we created a table containing the support count of each item present in the dataset. The first item set was created by filtering out the U since it did not surpass the minimum support of 2. Item sets were used to create the next candidate sets, and the process was repeated until there was only one row in the third item set.

Task 1: Apriori algorithm

a)

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

Minimum support = 33.33% means that the itemsets must occur at least 2 times (2/6 = 1/3). Hence, 2 is the minimum support count

C1 candidate set (K=1)

	Sup_count
B	2
C	3
H	4
I	3
K	2
U	1

I1 Item set

	Sup_count
B	2
C	3
H	4
I	3
K	2

C2 candidate set (K=2)

B,C	0
B,H	2
B,I	0
B,K	1
C,H	2
C,I	3
C,K	0
H,I	2
H,K	1
I,K	1

I2 Item set

B,H	2
C,H	2
C,I	3
H,I	2

C3 candidate set (K=3)

B,C,H	0
C,H,I	2

I3 Item set

C,H,I	2

The item sets with minimum support 33.3% are (B,H),(C,H),(C,I),(H,I),(C,H,I) since they all have support count >= 2

b)

Rule generation for element set {HCI}

Rule	Confidence
{CH} -> {I}	2/2 = 100%
{HI} -> {C}	2/2 = 100%
{CI} -> {H}	2/3 = 66.7%
{H} -> {CI}	2/4 = 50%
{I} -> {CH}	2/4 = 50%
{C} -> {HI}	2/3 = 66.7%

Based on the confidence threshold c = 60%, we can see 4 association rules ({CH} -> {I}, {HI} -> {C}, {CI} -> {H}, {C} -> {HI}.

The numbers from the confidence calculation results from the formula

$c(X \rightarrow Y) = \sigma(X \cup Y) / \sigma(X)$

Sondre Schirmer-Mikalsen, Arash Aminpour

Task 2

Task 2: FP-Growth Algorithm

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

Minimum support = 22% means that the itemsets must occur at least 2 times (2/9 = 22.22%). Hence, 2 is the minimum support count

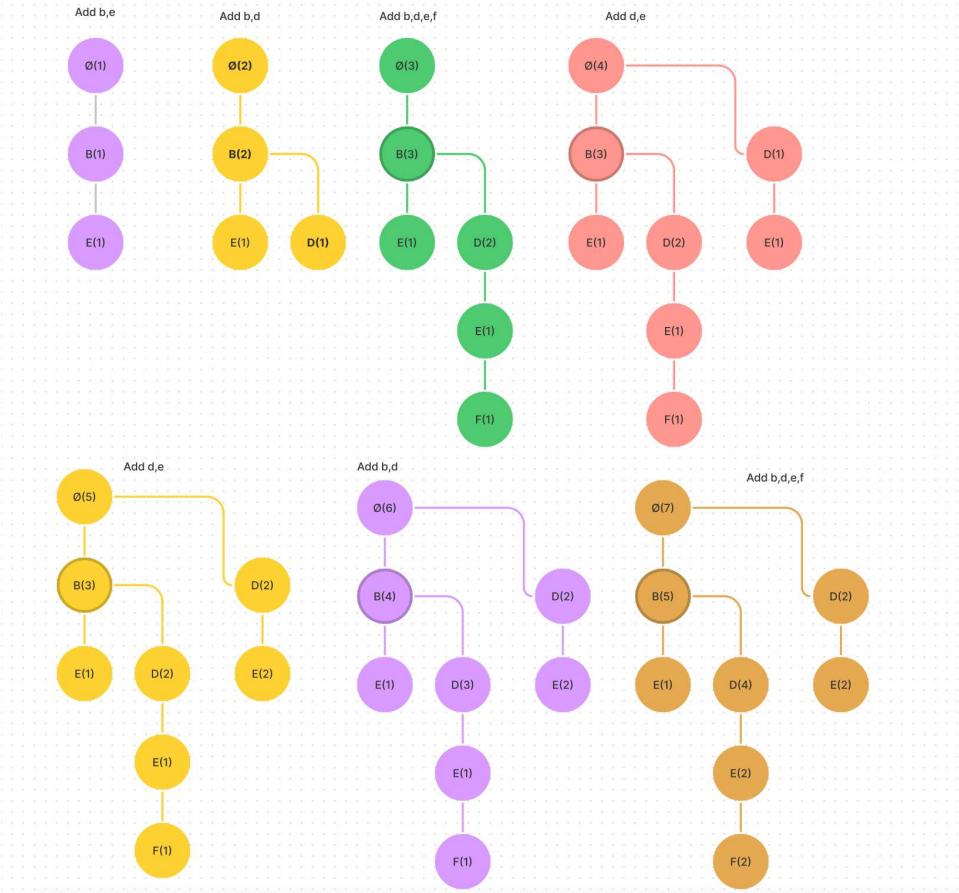
Support count	
a	1
b	7
c	1
d	7
e	7
f	3
g	1
h	1
i	1
j	1

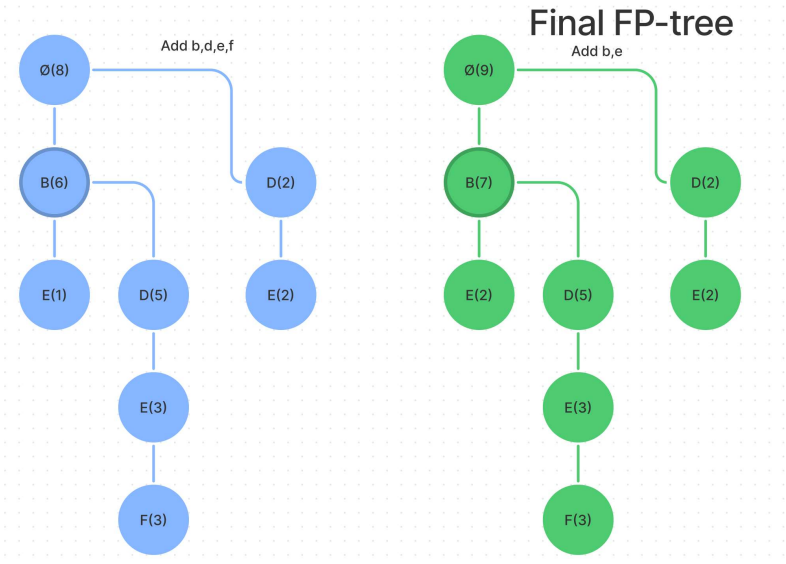
Sorted frequent pattern set	
b	7
d	7
e	7
f	3

TID	Ordered-item set
T1	b,e
T2	b,d
T3	b,d,e,f
T4	d,e
T5	d,e
T6	b,d
T7	b,d,e,f
T8	b,d,e,f
T9	b,e

- First we have counted the support count for every item.
- Next we removed the items that had lower support count than minimum support count of 22%
- Then we sorted the remaining items in descending and alphabetical order
- Then we sorted the different transactions by the item with the highest support count

- Next we constructed the FP-tree as you can see below, by adding one by one transaction into the tree from the ordered item-set.





- Next we created the conditional sub-databases and the table representation for the conditional FP-trees as you can see below. We choose to split up the conditional sub-database for E into the paths R_ed and R_eb, since the conditional path for E was not a path.

Conditional sub-databases:

f(3) paths:

Path	Count
b,d,e,f	3

d(7) paths:

Path	Count
b,d	5
d	2

e(7) paths:

Path	Count
b,e	2
b,d,e	3
d,e	2

b(7) paths:

Path	Count
b	7

Recursivly project Re, takes the next item with lowest support count in the sub-database for E: {D(5)}

Red

Path	Count
d	2
b,d	3

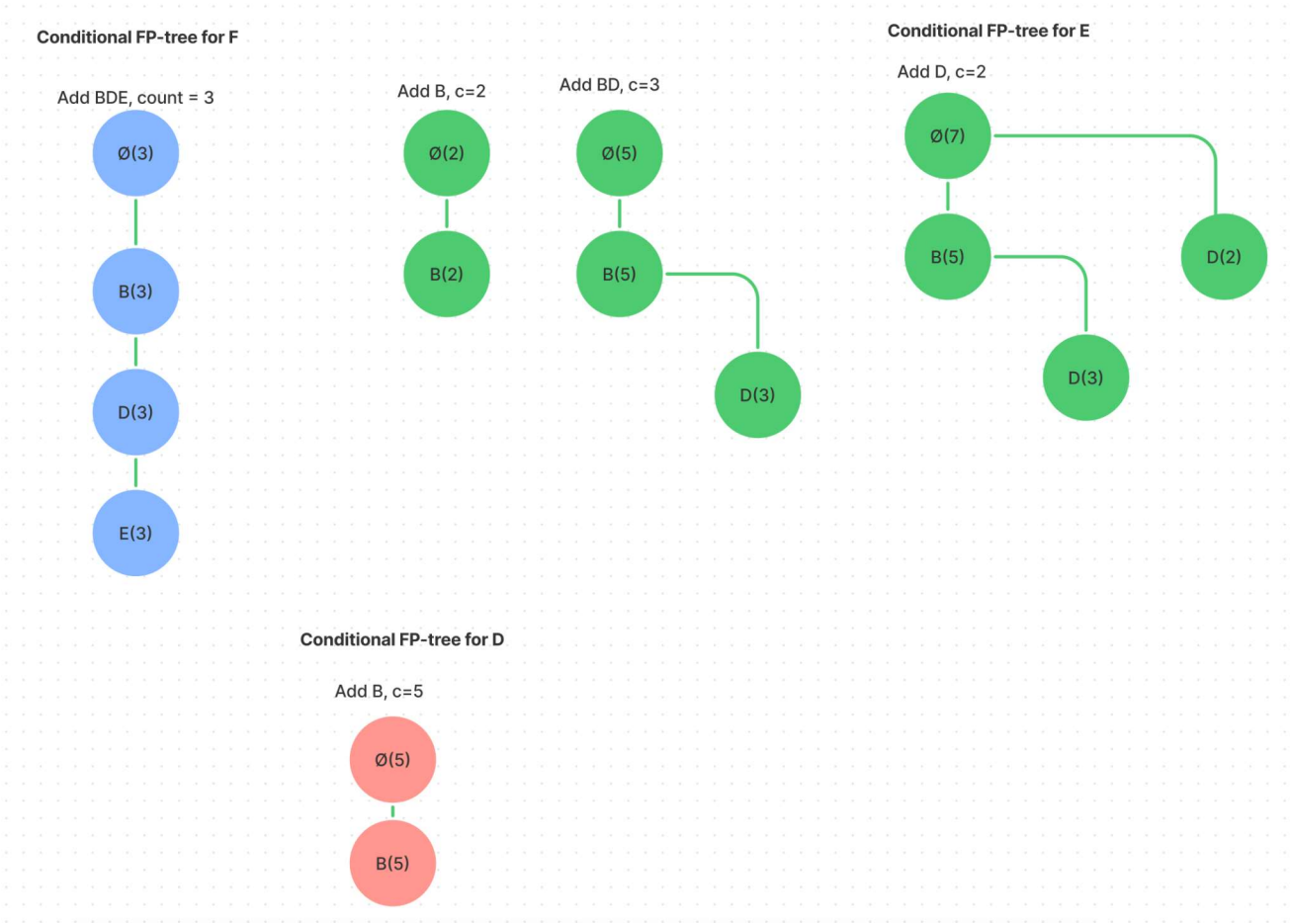
Reb

Path	Count
b	5

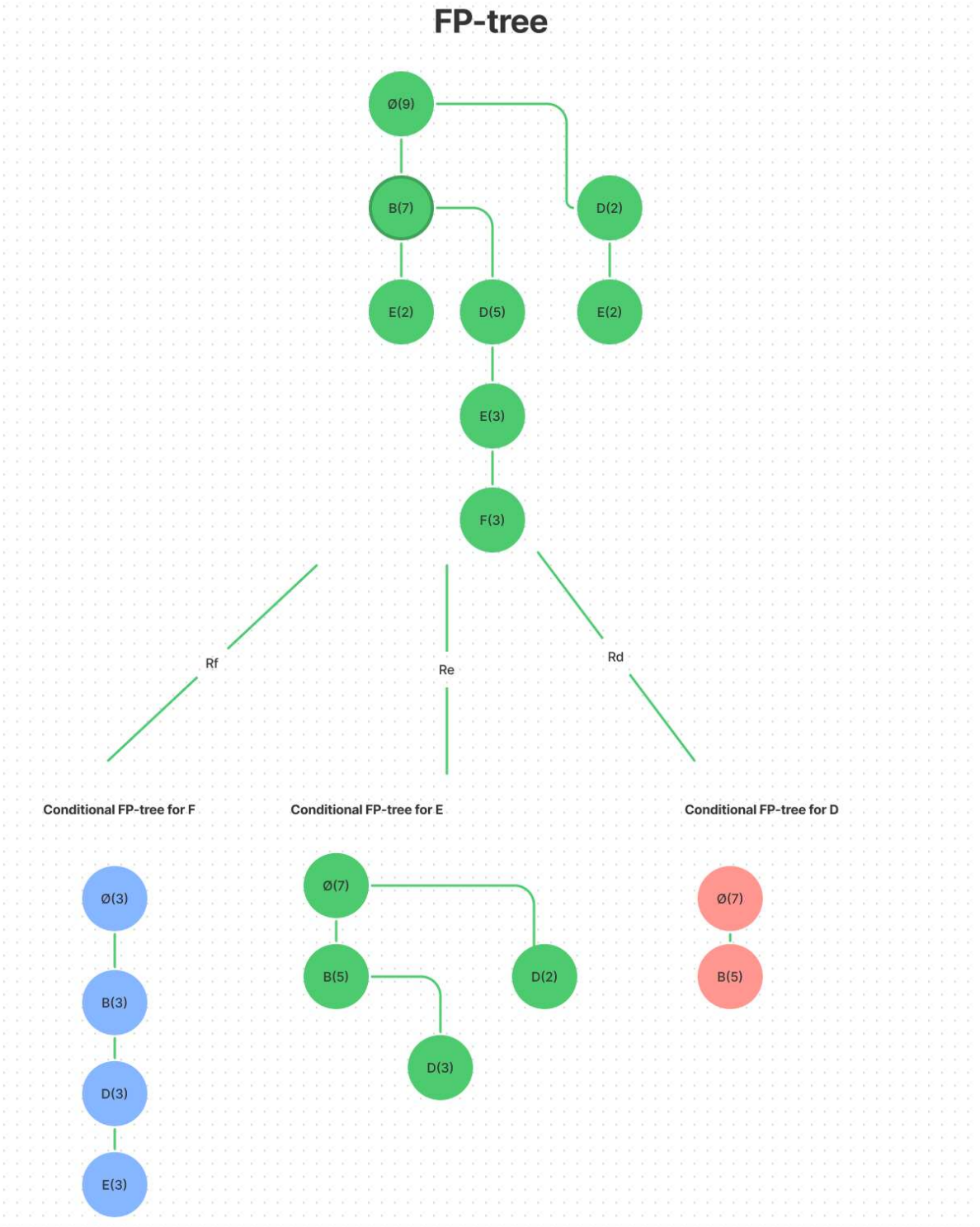
Table representation for conditional FP-trees

Item	**Conditional Pattern database**	**Conditional FP-tree**	**Frequent itemsets**
f	{b,d,e,f:3}	<b:3,d:3,e:3>	{b,f:3}, {d,f:3},{e,f:3}, {b,d,f:3},{d,e,f:3}, {b,e,f:3},{b,d,e,f:3}
e	{b,e:2} {b,d,e:3} {d,e:2}	<b:5,d:3> <d:2>	
ed	{d:2} {b,d : 3}	<b: 3>	{e,d: 5}, {e,d,b: 3}
eb	{b: 5}	<ø(5)>	{e,b:5 }
d	{b,d:5} {d,2}	<b:5>	{b,d:5}

Below you can also graphs for the conditional FP-trees and how they are constructed.

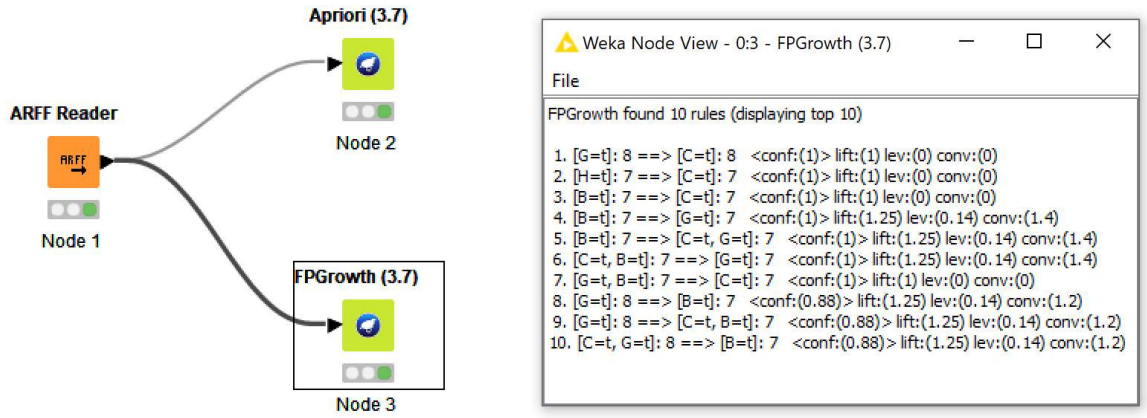


Here you can see the FP-tree and the conditional FP-trees connected to it

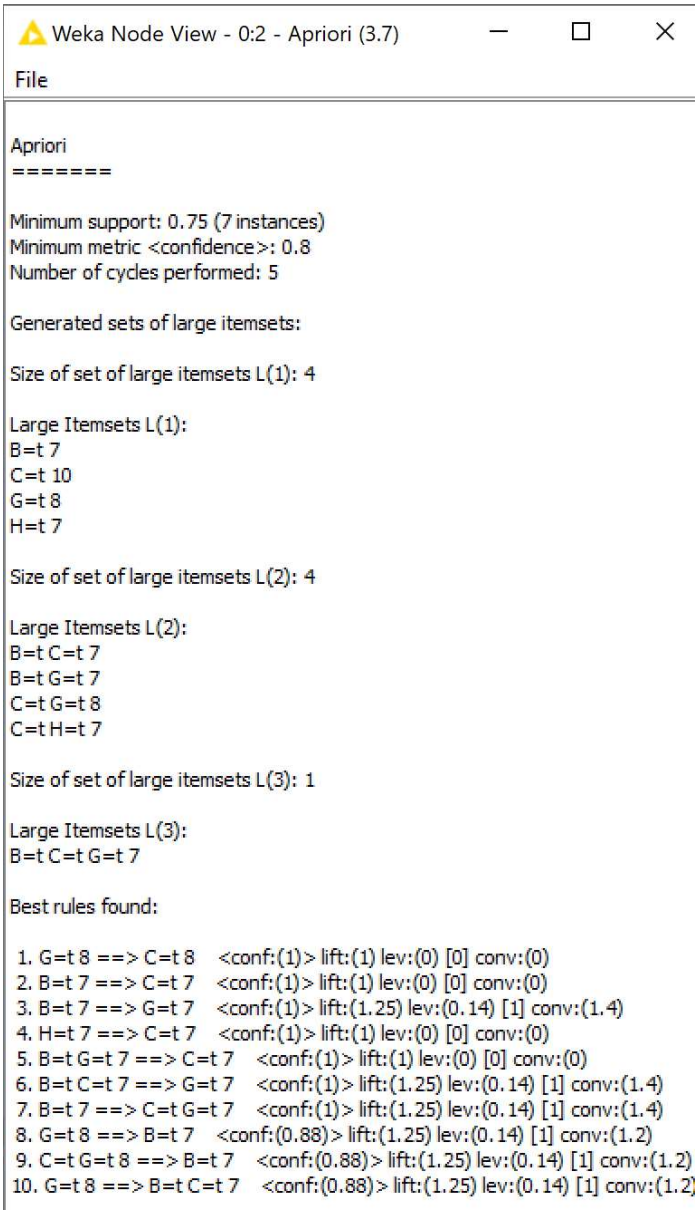


Task 3: Knime

In the workflow, we added the ARFF reader in order to receive and process data from the ARFF file. The data was passed on through to the Weka Apriori and FPGrowth nodes. The node settings were changed to set the minimum support to 0.5t and the minimum confidence to 80%. The nodes were executed, and the first 10 results were displayed in the Weka node view.



Workflow, diagram including the nodes and the results from the FPGrowth node.



Results from the Apriori node.

Task 4: Compact representation of frequent item sets

Frequent element set	Support		Reasoning
a	11		a,d = 11
b	10		
c	6		a,c,d = 6
d	13		
e	8		b,e = 8
a,b	7		a,b,e = 7
a,c	6		a,c,d = 6
a,d	11		
a,e	7		a,b,e = 7
b,d	7		
b,e	8		
c,d	6		a,c,d = 6
c,e	5		a,c,d,e = 5
d,e	6		
a,b,e	7		
a,c,d	6		
a,c,e	5		a,c,d,e = 5
a,d,e	5		a,c,d,e = 5
b,d,e	4		
c,d,e	5		a,c,d,e = 5
a,c,d,e	5		

Frequent element sets were generated by taking the union of all the subsets of the closed element sets. The support of the non-frequent element sets was found by finding the superset with the highest support, as shown in the rightmost column.