

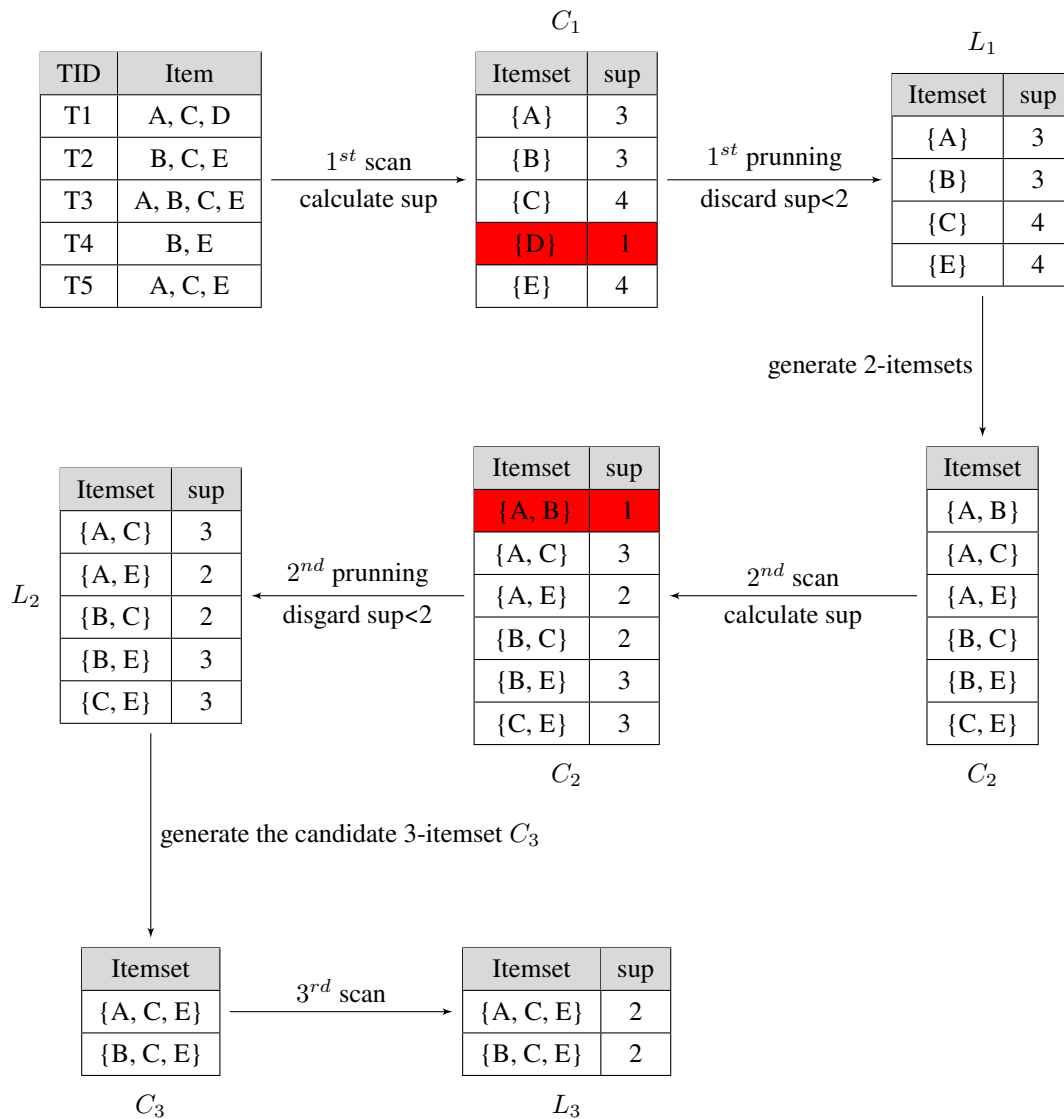
# DSAA 5002 - HW1

50015976 Ruiming ZHANG

## Q1 [15 Marks]

Given the transaction database below, set the minimum support count to 2 and the minimum confidence level to 60% to find the strong association rule. Generate the set  $C_3$  of the candidate 3-itemset, using pruning on Apriori principle.

### Solution:



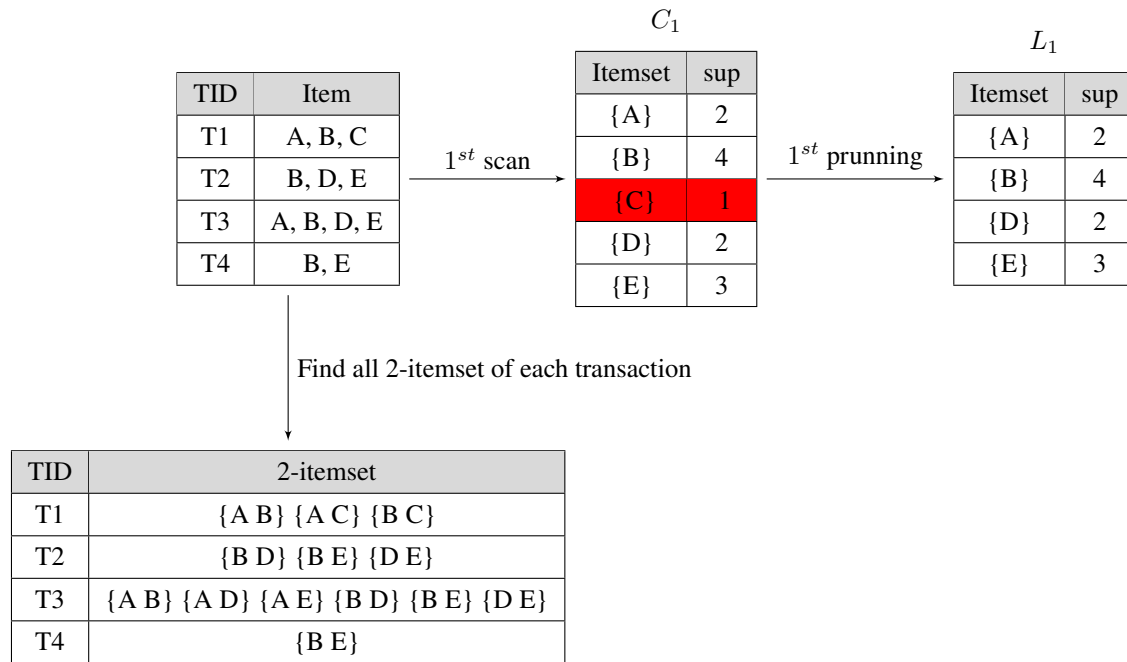
Thus we get the candidate 3-itemset  $C_3$ .

## Q2 [15 Marks]

Reducing the transactions using dynamic hashing and pruning(DHP) algorithm. Set the minimum support count to 2.

Hash function bucket # =  $h(\{xy\}) = ((\text{order of } x) * 10 + (\text{order of } y)) \% 7$

### Solution



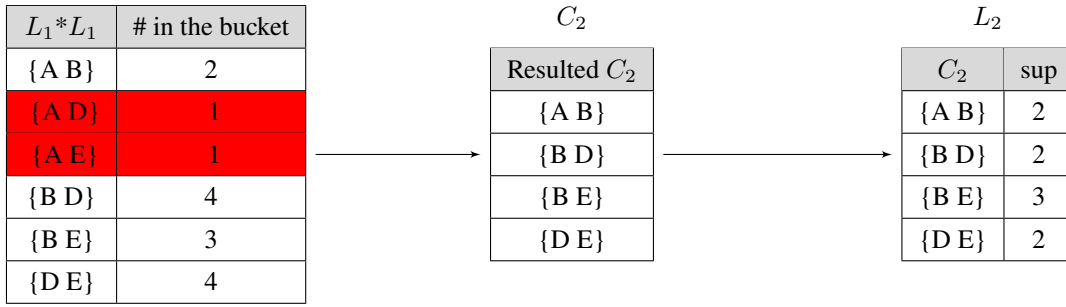
Because we have:

- Items = A, B, C, D, E,
- Order = 1, 2, 3, 4, 5
- Hash function :  $h(\{xy\}) = ((\text{order of } x) * 10 + (\text{order of } y)) \% 7$

Thus we have the hash table below:

bucket	0	1	2	3	4	5	6
count	1	1	1	4	3	2	1
2-itemset	{A D}	{A E}	{B C}	{B D} {D E} {B D} {D E}	{B E} {B E} {B E}	{A B} {A B}	{A C}

Let  $L_1 * L_1$  to generate a 2-itemset table, and choose the itemsets where the number of content in its bucket is above the minimum support.



Because if an item occurs in a frequent  $(k+1)$ -itemset, it must occur in at least  $k$  candidate  $k$ -itemsets.

TID	Item	2-itemset occurs	
T1	A, B, C	{A B}	Disgard
T2	B, D, E	{B D} {B E} {D E}	Keep {B D E}
T3	A, B, D, E	{A B} {B D} {B E} {D E}	Keep {B D E}
T4	B, E	{E E}	Disacrd

$\longrightarrow$

TID	Item
T2	B, D, E
T3	B, D, E

Thus we have reduced the transactions.

### Q3 [35 Marks]

An itemset  $X$  is said to be a frequent itemset if the frequency count of  $X$  is at least a given support threshold.

An itemset  $Y$  is a proper super-itemset of  $X$  if  $X \subset Y$  and  $X \neq Y$ .

An itemset  $X$  is said to be a closed frequent itemset if (1)  $X$  is frequent and (2) there exists no proper super-itemset  $Y$  of  $X$  such that  $Y$  is frequent and  $Y$  has the same frequency count as  $X$ .

An itemset  $X$  is said to be a maximal frequent itemset if (1)  $X$  is frequent and (2) there exists no proper super-itemset  $Y$  of  $X$  such that  $Y$  is frequent.

Let  $F_c$  be the set of (traditional) frequent itemsets each of which is associated with a frequency in the dataset.

For example, if there are three frequent itemsets,  $\{I_1\}$  with frequency 4,  $\{I_2\}$  with frequency 5, and  $\{I_1, I_2\}$  with frequency 3,  $F = \{\{I_1\}, \{I_2\}, \{I_1, I_2\}\}$  and  $F_c = \{<\{I_1\}, 4>, <\{I_2\}, 5>, <\{I_1, I_2\}, 3>\}$ .

Similarly, let  $C$  be the set of closed frequent itemsets without specifying the frequency of itemsets.

Let  $C_c$  be the set of closed frequent itemsets each of which is associated with a frequency of itemsets.

Let  $M$  be the set of maximal frequent itemsets without specifying the frequency of itemsets.

Ler  $M_c$  be the set of maximal frequent itemsets each of which is associated with a frequency in the dataset.

The following shows six transactions with four items. Each row corresponds to a transaction where 1 corresponds to a presence of an item and 0 corresponds to an absence.

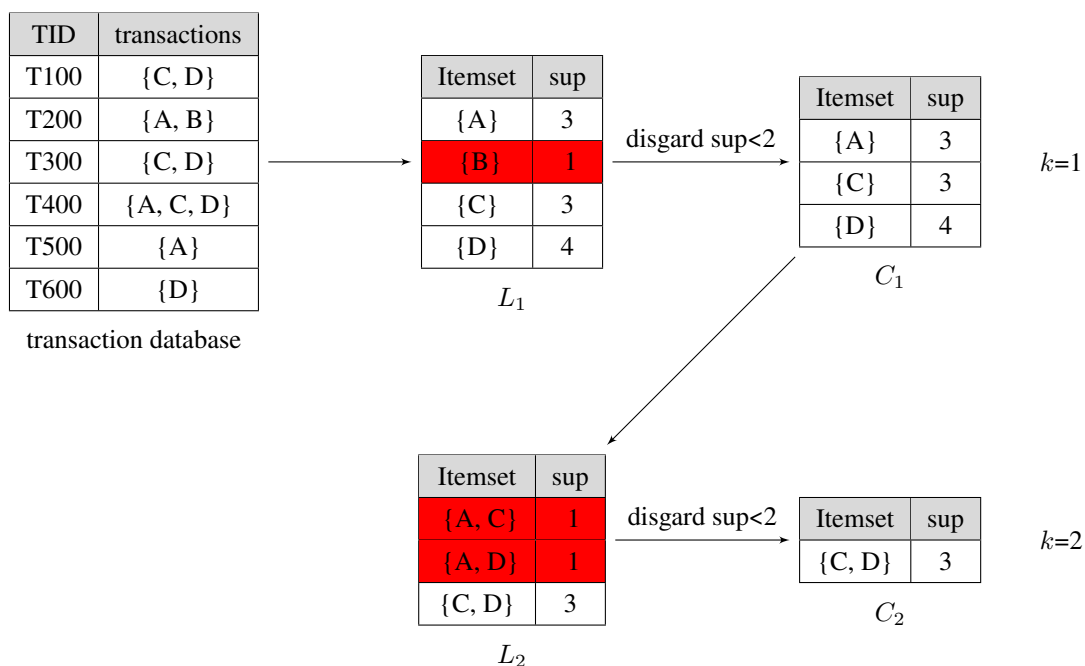
A	B	C	D
0	0	1	1
1	1	0	0
0	0	1	1
1	0	1	1
1	0	0	0
0	0	0	1

Suppose that the support threshold is 2.

- (a) (i) What is  $F_C$ ? (ii) What is  $C_c$ ? (iii) What is  $M_c$ ? (5 Marks)
- (b) (i) What are the advantages and the disadvantages of using closed frequent itemsets compared with traditional frequent itemsets? (5 Marks)
- (ii) What are the advantages and the disadvantages of using closed frequent itemsets compared with maximal frequent itemsets? (5 Marks)
- (c) Please adapt algorithm FP-growth with the use of the FP-tree to find all closed frequent item set. Please write down how to adapt algorithm FP-growth and illustrate the adapted algorithm with the above example. (20 Marks)

## Solution

- (a) According to the topic, we have the following transaction database. And we generate all the  $k$ -itemsets which might be frequent itemsets.



i: We have  $F_c = \{ \langle \{A\}, 3 \rangle, \langle \{C\}, 3 \rangle, \langle \{D\}, 4 \rangle, \langle \{C, D\}, 3 \rangle \}$

ii: We have  $C_c = \{ \langle \{A\}, 3 \rangle, \langle \{D\}, 4 \rangle, \langle \{C, D\}, 3 \rangle \}$

iii: We have  $M_c = \{ \langle \{A\}, 3 \rangle, \langle \{C, D\}, 3 \rangle \}$

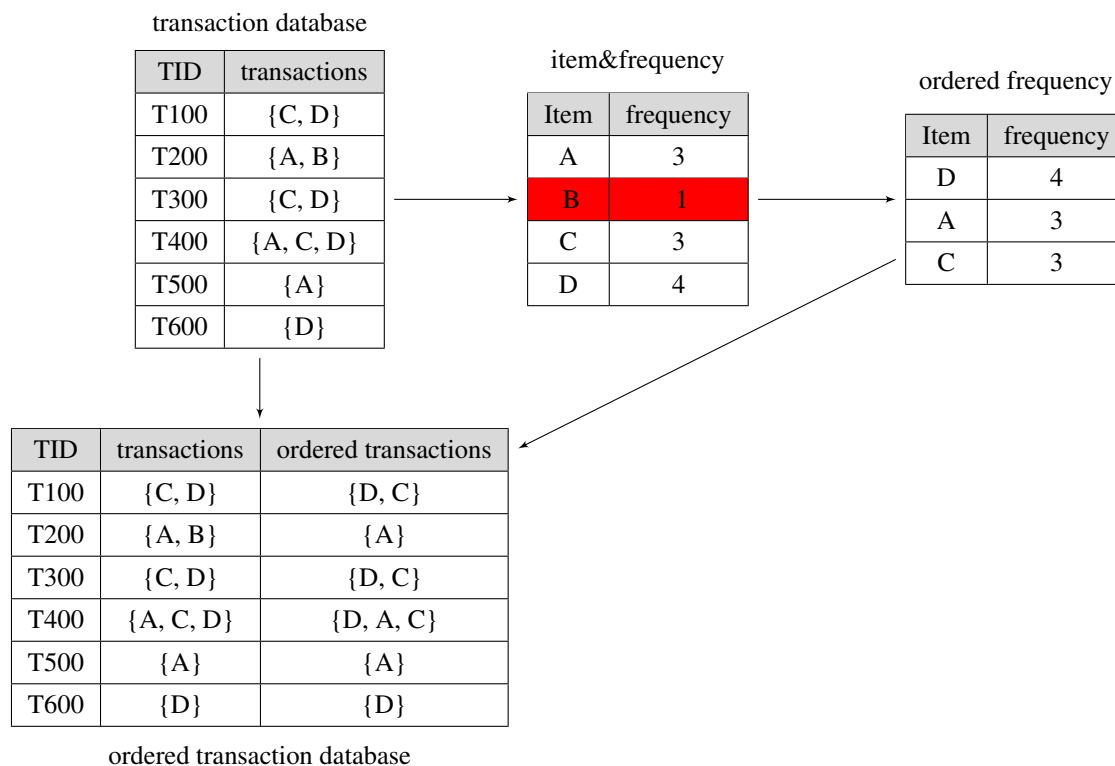
(b) i:

- Advantages: Compared with traditional frequent itemsets, inclosed itemsets are eliminated, which reduces the number of itemsets. And because a lot of frequent itemsets can be considered as subsets of a closed super-itemset, it does not lose much information.
- Disadvantages: Finding frequent closed itemsets need more computation, increasing the time complexity.

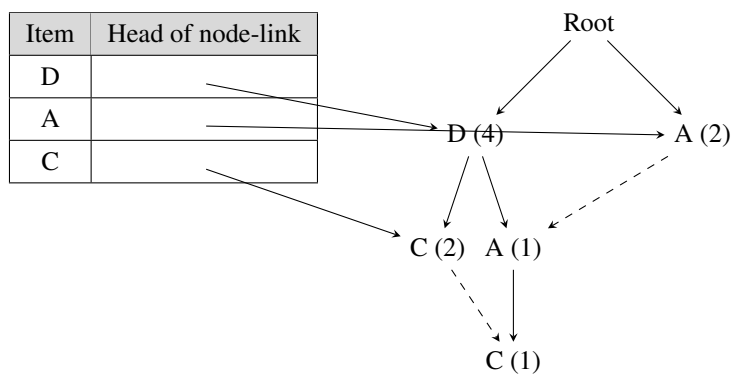
ii:

- Advantages: Closed itemsets do not lose much information because the closed super-itemsets have same frequency as the sub-itemsets. But maximal frequent itemsets may lose some information.
- Disadvantages: Maximal frequent itemsets are more efficient than closed itemsets, because it does not need to check the frequency of the super-itemsets.

(c) Generate a FP-tree. Firstly, deduce the ordered frequent items.

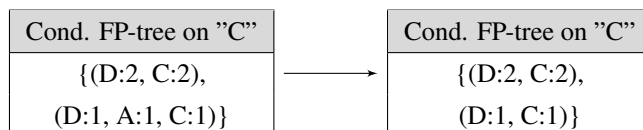


Then we construct the FP-tree from the above data.



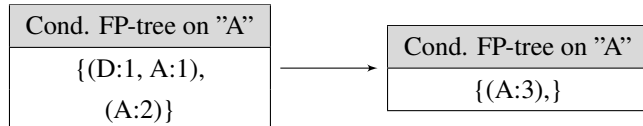
Construct the FP-conditional tree for  $C, A, D$ .

**for  $C$**



Because for every path lying  $C$ ,  $D$  also lies in the path, so  $C$  is not a closed frequent itemset. But  $\{D, C\}$  is a closed frequent itemset.

**for  $A$**



Because there is no other itemsets lies in the right path lying  $A$ , so  $A$  is a closed frequent itemset.

**for  $D$**



Because there is a path only lying  $D$ , so  $D$  is a closed frequent itemset.

Thus we have  $C_c = \{ \langle \{D, C\}, 3 \rangle, \langle \{A\}, 3 \rangle, \langle \{D\}, 4 \rangle \}$ .

## Q4 [35 Marks]

A GSP example: Suppose now we have 5 events: 'Upload Songs', 'Add Tags', 'Share', 'Listen' and 'Comment'. Let min-support be 40%. The sequence database of a Music Platform is shown in following table:

Object	Sequence
A	<{'Upload Songs', 'Add Tags'}>
B	<{'Upload Songs', 'Share'}>
C	<{'Upload Songs'}, {'Share', 'Listen'}>
D	<{'Upload Songs'}, {'Upload Songs', 'Add Tags'}, {'Listen'}>
E	<{'Listen'}, {'Add Tags', 'Comment'}, {'Share', 'Listen'}>

Please answer the following questions:

- Make the first pass over the sequence database to yield all the 1-element **frequent** sequences and what is the corresponding support? **5 Marks**
- Based on (a), do the 2-sequences Candidate Generation and Candidate Pruning. **10 Marks**
- What is the **frequent** 2-sequences based on the result of (b)? **5 Marks**
- Based on (c), do the 3-sequences Candidate Generation and Candidate Pruning. When a sequence should be pruned, you need to explain why. **10 Marks**
- What is the frequent 3-sequences based on the result of (d)? Please calculate the support. **5 Marks**

**Remember: For frequent k-sequences, the support  $\geq$  min-support**

### Solution

For easier reading, we denote 'Upload Songs', 'Add Tags', 'Share', 'Listen' and 'Comment' by U, A, S, L and C respectively. And if there are 2 items in 1 set, we have arranged it in alphabetical order by the first letter. Thus we have:

Object	Sequence		Object	Sequence
A	<{'Upload Songs', 'Add Tags'}>		A	<{A, U}>
B	<{'Upload Songs', 'Share'}>		B	<{S, U}>
C	<{'Upload Songs'}, {'Share', 'Listen'}>	→	C	<{U}, {L, S}>
D	<{'Upload Songs'}, {'Upload Songs', 'Add Tags'}, {'Listen'}>		D	<{U}, {A, U}, {L}>
E	<{'Listen'}, {'Add Tags', 'Comment'}, {'Share', 'Listen'}>		E	<{L}, {A, C}, {L, S}>

- (a) Candidate 1-sequences are:

<{A}>, <{C}>, <{L}>, <{S}>, <{U}>

According to the database above, we have:

Sequence	Sup		Sequence	Sup
<{A}>	60%	disgard sup<40%	<{A}>	60%
<{C}>	20%		<{L}>	60%
<{L}>	60%		<{S}>	60%
<{S}>	60%		<{U}>	80%
<{U}>	80%			

The 1-element frequent sequences and the corresponding support are:

<{'Add Tags'}> (support=60%)

<{'Listen'}> (support=60%)

<{'Share'}> (support=60%)

<{'Upload Songs'}> (support=80%)

**(b)** Base case ( $k = 2$ ): Merging two frequent 1-sequences  $\langle\{i_1\}\rangle$  and  $\langle\{i_2\}\rangle$  will produce two candidate 2-sequences:  $\langle\{i_1\} \{i_2\}\rangle$  and  $\langle\{i_1, i_2\}\rangle$

Candidate 2-sequences are:

$\langle\{A, L\}\rangle, \langle\{A, S\}\rangle, \langle\{A, U\}\rangle, \langle\{L, S\}\rangle, \langle\{L, U\}\rangle, \langle\{S, U\}\rangle,$

$\langle\{A\}, \{A\}\rangle, \langle\{A\}, \{L\}\rangle, \langle\{A\}, \{S\}\rangle, \langle\{A\}, \{U\}\rangle,$

$\langle\{L\}, \{A\}\rangle, \langle\{L\}, \{L\}\rangle, \langle\{L\}, \{S\}\rangle, \langle\{L\}, \{U\}\rangle,$

$\langle\{S\}, \{A\}\rangle, \langle\{S\}, \{L\}\rangle, \langle\{S\}, \{S\}\rangle, \langle\{S\}, \{U\}\rangle,$

$\langle\{U\}, \{A\}\rangle, \langle\{U\}, \{L\}\rangle, \langle\{U\}, \{S\}\rangle, \langle\{U\}, \{U\}\rangle$

All the 1-sequences we generate 2-sequences from are frequent. So after candidate pruning, the 2-sequences should remain the same.

**(c)** After candidate elimination, frequent 2-sequences are:

$\langle\{A, U\}\rangle$  (support=40%),

$\langle\{L, S\}\rangle$  (support=40%),

$\langle\{A\}, \{L\}\rangle$  (support=40%),

$\langle\{U\}, \{L\}\rangle$  (support=40%)

**(d)** Generate 3-sequences from the remaining 2-sequences, 3-sequences are:

$\langle\{A, U\}, \{L\}\rangle$  (generate from  $\langle\{A, U\}\rangle$  and  $\langle\{U\}, \{L\}\rangle$  or from  $\langle\{A, U\}\rangle$  and  $\langle\{A\}, \{L\}\rangle$ ),



$\langle \{A\}, \{L, S\} \rangle$  (generate from  $\langle \{A\}, \{L\} \rangle$  and  $\langle \{L\}, \{S\} \rangle$ ),

$\langle \{U\}, \{L, S\} \rangle$  (generate from  $\langle \{U\}, \{L\} \rangle$  and  $\langle \{L\}, \{S\} \rangle$ )

Pruning:

$\langle \{A\}, \{L, S\} \rangle$  should be pruned because one 2-subsequence  $\langle \{A\}, \{S\} \rangle$  is not frequent.

$\langle \{U\}, \{L, S\} \rangle$  should be pruned because one 2-subsequence  $\langle \{U\}, \{S\} \rangle$  is not frequent.

(e)  $\langle \{A, U\}, \{L\} \rangle$  (support=20% < 40%, should be eliminated)

Thus, there is no frequent 3-sequence.