ASSOCIATION RULE: TUTORIAL 2

Cx4032-Data Analytics and Mining
(Data Mining)

Q1

Explain the following observation for PCY algorithm

- If a bucket contains a frequent pair, then the bucket is surely frequent
- However, even without any frequent pair, a bucket can still be frequent

PCY Algorithm – First Pass

New in PCY (marks the inner FOR loop over pairs)

```
FOR (each basket) :
    FOR (each item in the basket) :
        add 1 to item's count;
    FOR (each pair of items) :
        hash the pair to a bucket;
        add 1 to the count for that bucket;
```

- **Few things to note:**
  - Pairs of items need to be generated from the input file; they are not present in the file
  - We are not just interested in the presence of a pair, but we need to see whether it is present at least ***s*** (support) times

Observations about Buckets

- **Observation: If a bucket contains a frequent pair, then the bucket is surely frequent**
- However, even without any frequent pair, a bucket can still be frequent ☹
  - So, we cannot use the hash to eliminate any member (pair) of a "frequent" bucket
- **But, for a bucket with total count less than s, none of its pairs can be frequent** ☺
  - Pairs that hash to this bucket can be eliminated as candidates (even if the pair consists of 2 frequent items)

- **Pass 2:**
  Only count pairs that hash to frequent buckets

Example: PCY Algorithm 2nd Pass

For $\{i, j\}$ to be a **candidate pair**:

1. Both i and j are frequent items
2. The pair $\{i, j\}$ hashes to a bucket whose bit in the bit vector is **1** (i.e., a **frequent bucket**)

without any frequent pair, a bucket can still be frequent

$\cancel{(1,4)}, (2,3) \rightarrow h(i,j) = 0$

$\cancel{(1,5), (2,4) \rightarrow h(i,j) = 1}$

$(2,5), \cancel{(3,4)} \rightarrow h(i,j) = 2$

$(1,2), (3,5) \rightarrow h(i,j) = 3$

$(1,3), \cancel{(4,5)} \rightarrow h(i,j) = 4$

Candidate Pairs & Counts

Pair	Count
(2,3)	4
(2,5)	3
(1,2)	2
(3,5)	2
(1,3)	4

Frequent Itemsets are: $\{1\}, \{2\}, \{3\}, \{5\}, \{1,3\}, \{2,3\}, \{2,5\}$

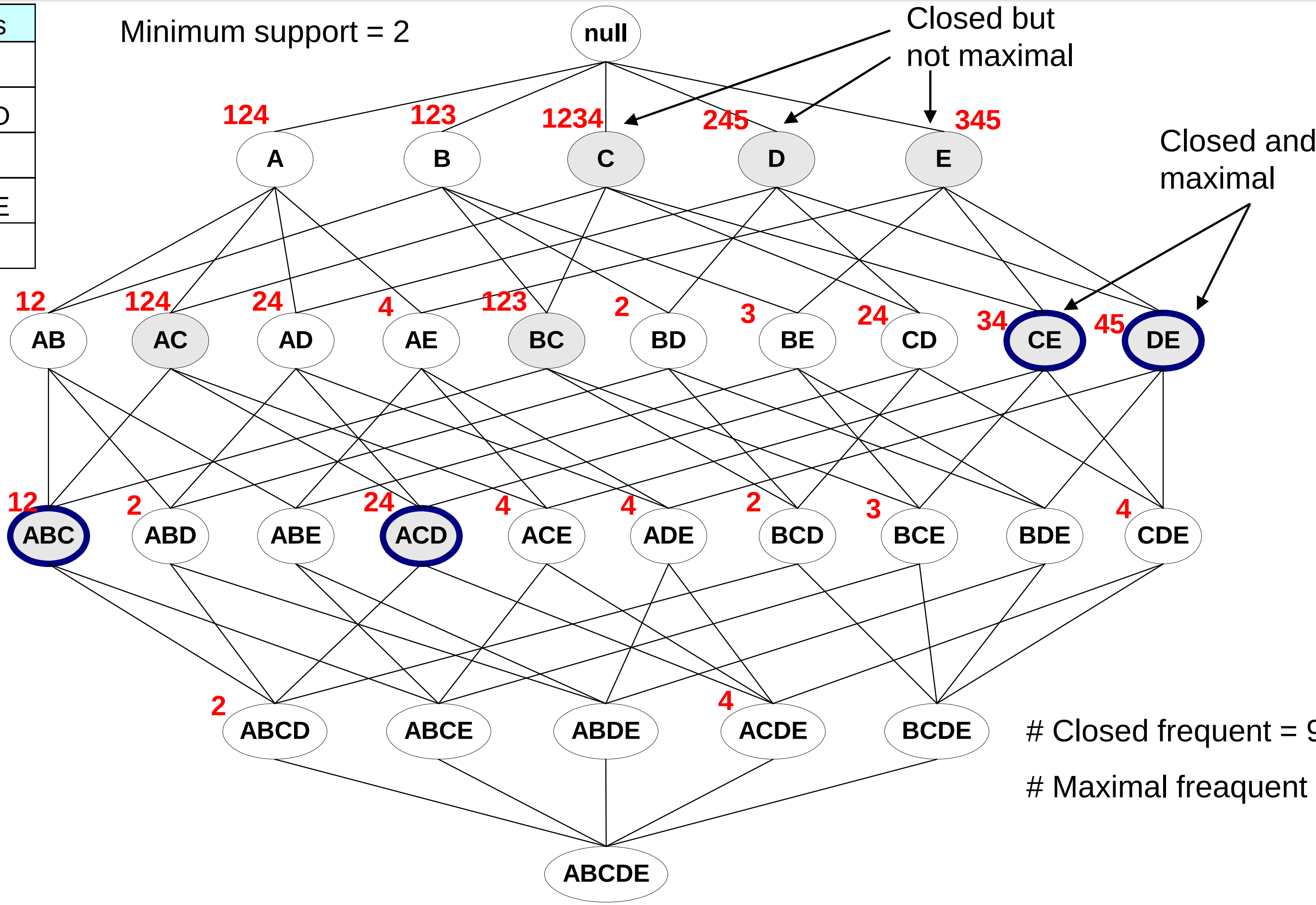
Q2

- Given a dataset, minsup threshold, which of the following has the largest number of itemsets? Which has the smallest number of itemsets?
  - Frequent itemsets
  - Maximal frequent itemsets
  - Closed frequent itemsets

Maximal Frequent vs Closed Frequent Itemsets

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE

Minimum support = 2



Closed frequent = 9

Maximal freaquent = 4

Maximal vs Closed Itemsets

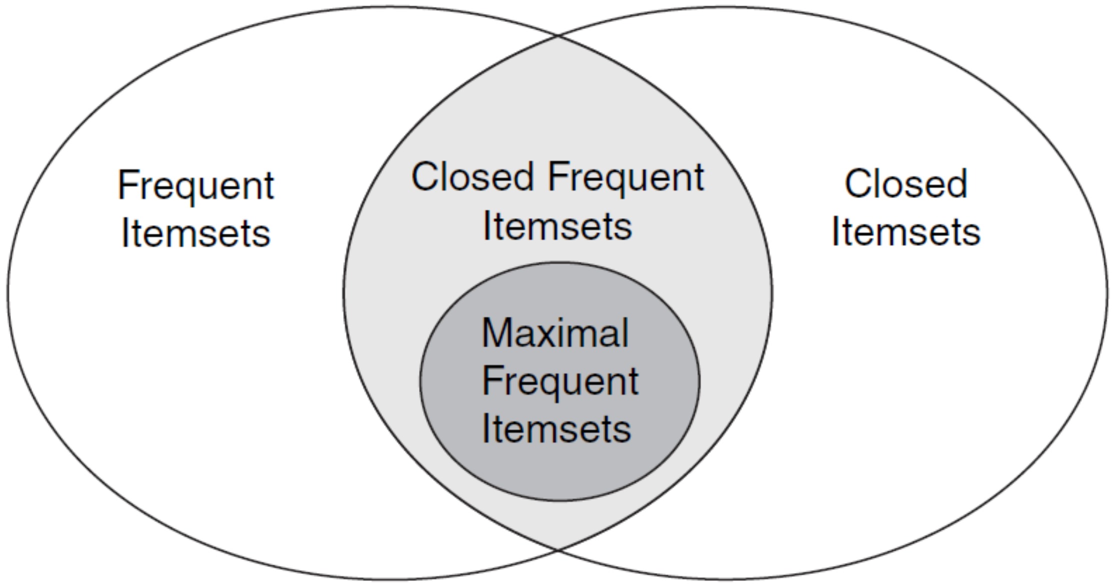


Figure 5.18. Relationships among frequent, closed, closed frequent, and maximal frequent itemsets.

Q3

- Discuss the impact of the following characteristics of a transaction table on the use of the FP tree to mine frequent itemsets from the table:
- (a) Number of unique items in table
- (b) Average number of items in a transaction
- (c) Number of transactions in table

Q3-Ans

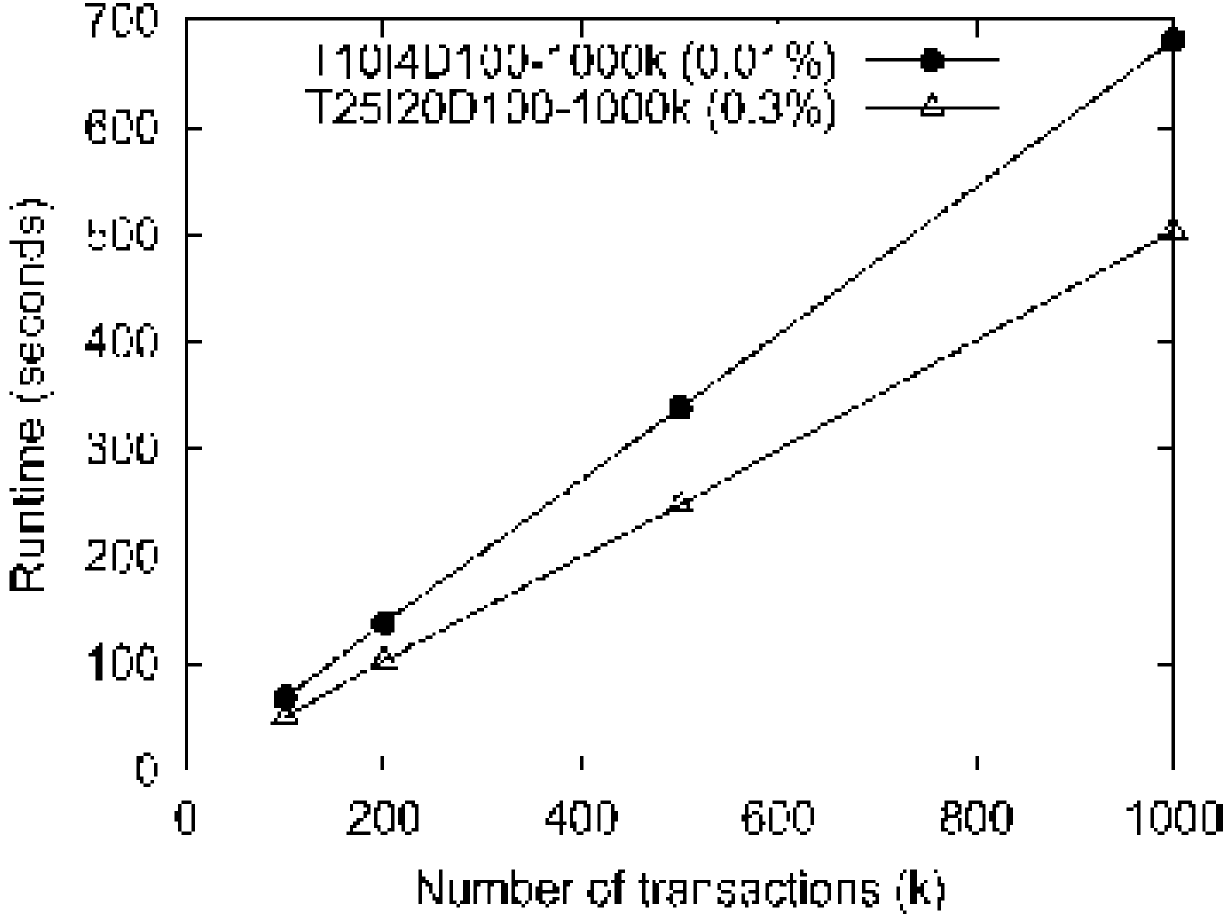
- (a) Number of unique items in table
  - (1) the header table increases linearly,
  - (2) the number of possible patterns to be mined (i.e., FP-tree) increases non-linearly,
  - (3) time spent in exploring fp-tree (i.e., fp-growth) increases non-linearly.

Q3-Ans

- (b) Average number of items in a transaction
  - the height of FP-tree is limited by the maximal length of the transactions. We may find more longer patterns and the number of possible patterns to be mined may increase

- (c) Number of transactions in table
  - Time of building Fp-tree increases linearly

Example experimental results of Fp-tree



Q4

- A database has four transactions. Let *min_sup* = 60% (equivalent to 2.4 out of 4) and *min_conf* = 80%.

TID	Date	Items_bought
T100	20006-01-01	{*K*, *A*, *D*, *B*}
T200	20006-01-01	{*D*, *A*, *C*, *E*, *B*}
T300	20006-01-01	{*C*, *A*, *B*, *E*}
T400	20006-01-01	{*B*, *A*, *D* }

- Find all frequent itemsets using FP-growth.

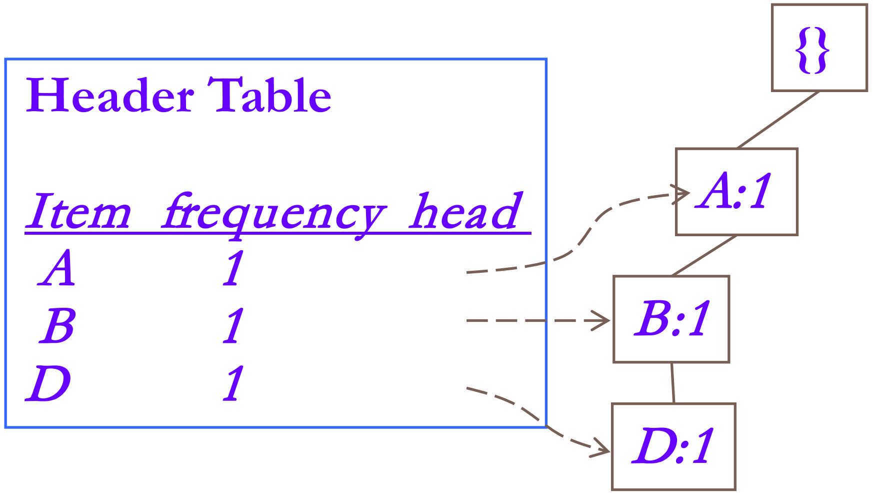
TID	Date	Items_bought
T100	2006-01-01	{*K*, *A*, *D*, *B*}
T200	2006-01-01	{*D*, *A*, *C*, *E*, *B*}
T300	2006-01-01	{*C*, *A*, *B*, *E*}
T400	2006-01-01	{*B*, *A*, *D* }

C1

A	4
B	4
C	2
D	3
E	2
K	1

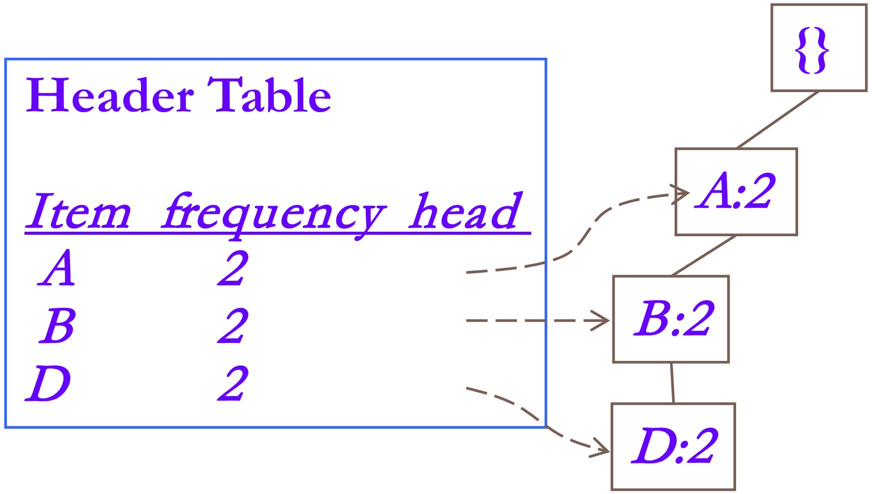
TID	Items_bought
T100	*{A, B, D}*
T200	*{A, B, D}*
T300	*{A, B}*
T400	*{A, B, D }*

TID	Items_bought
T100	*{A, B, D}*
T200	*{A, B, D}*
T300	*{A, B}*
T400	*{A, B, D }*



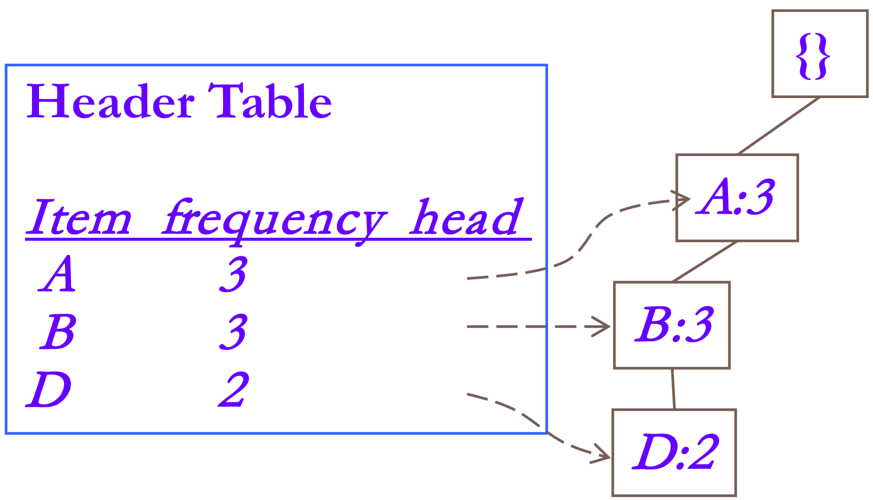
Insert *T100*

TID	Items_bought
T100	*{A, B, D}*
T200	*{A, B, D}*
T300	*{A, B}*
T400	*{A, B, D }*



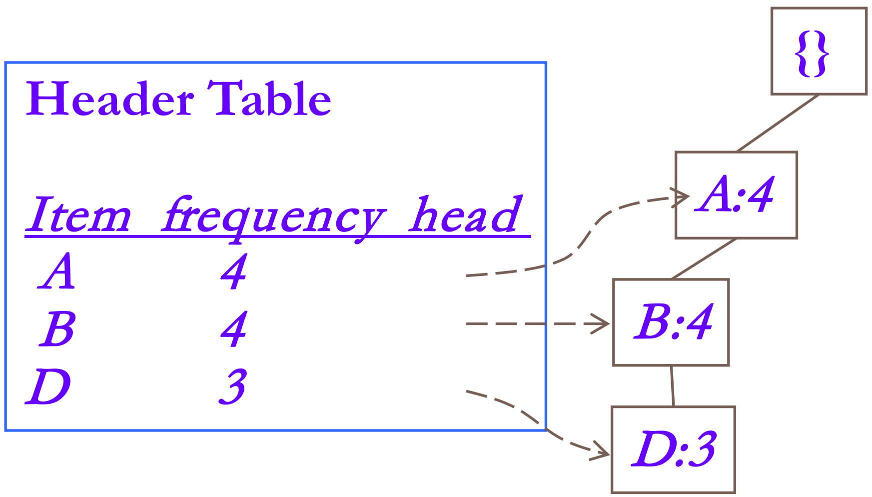
Insert *T200*

TID	Items_bought
T100	*{A, B, D}*
T200	*{A, B, D}*
T300	*{A, B}*
T400	*{A, B, D }*



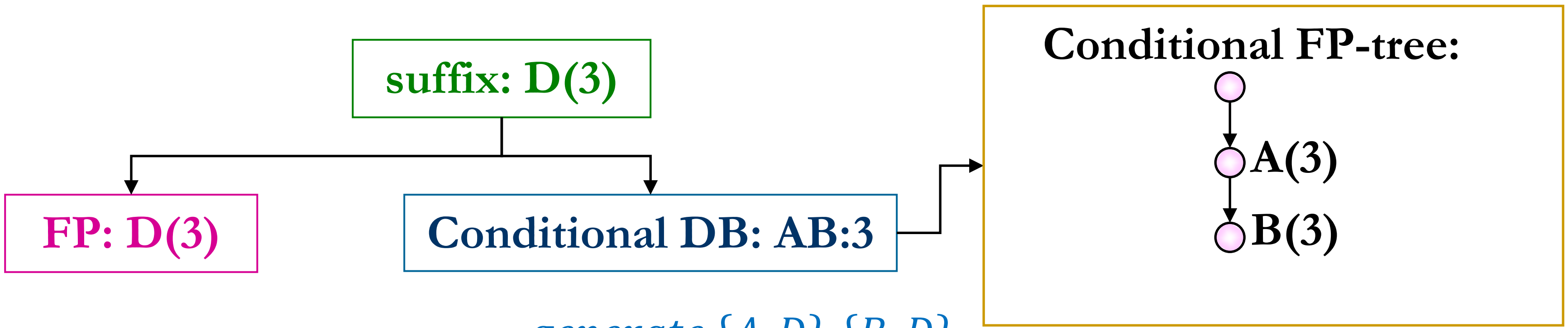
Insert ***T300***

TID	Items_bought
T100	*{A, B, D}*
T200	*{A, B, D}*
T300	*{A, B}*
T400	*{A, B, D }*



Insert ***T400***

Collect all patterns that ends at *D*

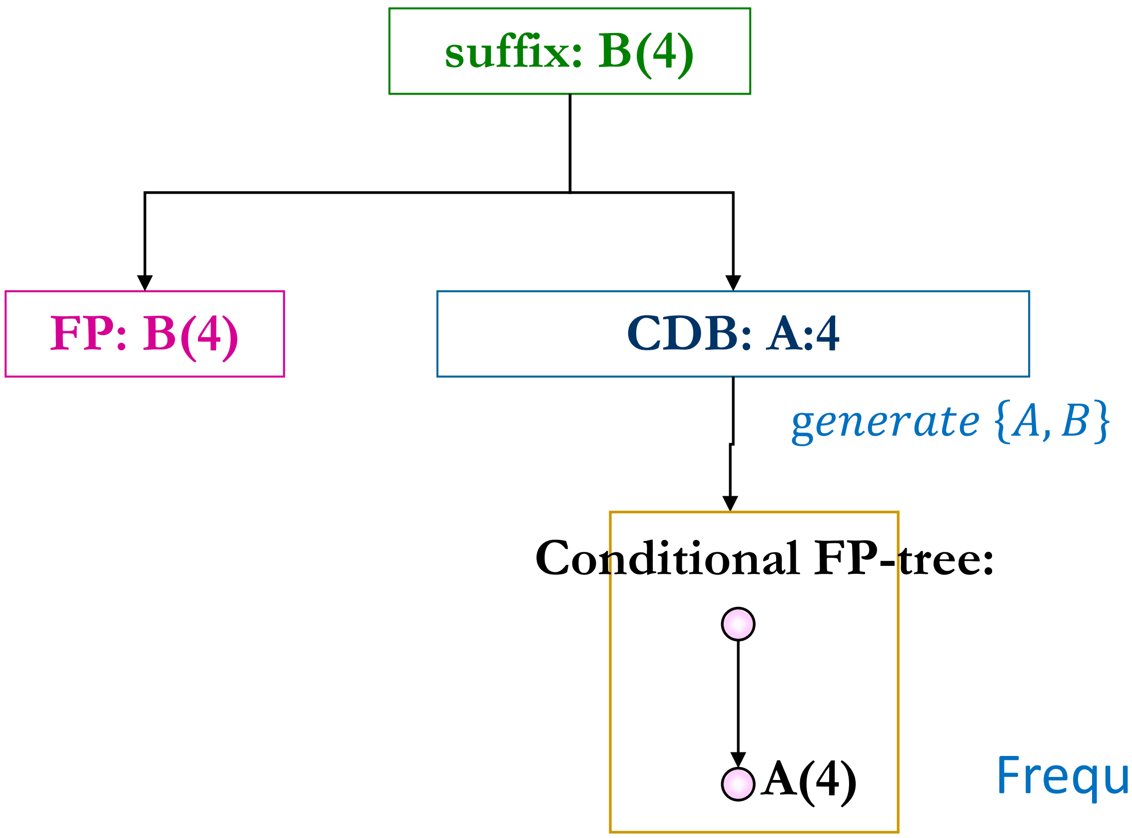


generate $\{A, D\}, \{B, D\}$

If the conditional FP-tree contains a single path, simply enumerate all patterns.

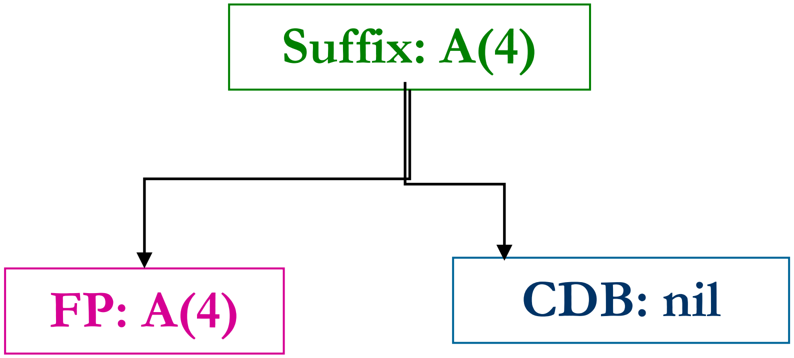
Frequent patterns contain D:
$\{\{D\}, \{A, D\}, \{B, D\}, \{A, B, D\}\}$

Collect all patterns that ends at B



Frequent patterns contain B:
$\{\{B\}, \{A, B\}\}$

Collect all patterns that ends at *A*



Q5 Candidate Generation

- <{a},{b},{c}> can be merged with <{b},{c},{f}> to produce <{a},{b},{c},{f}>
- <{a},{b},{c}> cannot be merged with <{b,c},{f}>
- <{a},{b},{c}> can be merged with <{b},{c,f}> to produce <{a},{b},{c,f}>
- <{a,b},{c}> can be merged with <{b},{c,f}> to produce <{a,b},{c,f}>
- <{a,b,c}> can be merged with <{b,c,f}> to produce <{a,b,c,f}>
- <{a}{b}{a}> can be merged with <{b}{a}{b}> to produce <{a},{b},{a},{b}>
- <{b}{a}{b}> can be merged with <{a}{b}{a}> to produce <{b},{a},{b},{a}>