

Level2-candidate-gen-SPM

Function level2-candidate-gen(L, φ)

```
1   $C_2 \leftarrow \emptyset$ ; // initialize the set of candidates
2  for each item  $l$  in  $L$  in the same order do
3      if  $l.count/n \geq MIS(l)$  then
4          for each item  $h$  in  $L$  that is after  $l$  do
5              if  $h.count/n \geq MIS(l)$  and  $|sup(h) - sup(l)| \leq \varphi$  then
6                   $C_2 \leftarrow C_2 \cup \{l, h\}$ ; // insert the candidate  $\{l, h\}$  into  $C_2$ 
```

Fig. 2.7. The level2-candidate-gen function

Note: In sequential pattern mining, you need to combine a sequence with itself in the join step

MSCandidate-gen-SPM

condition 1:

if the MIS value of the first item in a sequence (denoted by $s1$) is less than ($<$) the MIS value of every other item in $s1$ **then** // $s1$ and $s2$ can be equal

Sequence $s1$ joins with $s2$ **if** (1) the subsequences obtained by dropping the second item of $s1$ and the last item of $s2$ are the same, **and** (2) the MIS value of the last item of $s2$ is greater than **or equal to** that of the first item of $s1$. Candidate sequences are generated by extending $s1$ with the last item of $s2$:

if the last item l in $s2$ is a separate element **then**

$\{l\}$ is appended at the end of $s1$ as a separate element to form a candidate sequence $c1$.

if (the length and the size of $s1$ are both 2) **AND** (the last item of $s2$ is greater than the last item of $s1$) **then** // maintain lexicographic order

l is added at the end of the last element of $s1$ to form another candidate sequence $c2$.

else if ((the length of $s1$ is 2 and the size of $s1$ is 1) **AND** (the last item of $s2$ is greater than the last item of $s1$)) **OR** (the length of $s1$ is greater than 2) **then**

the last item in $s2$ is added at the end of the last element of $s1$ to form the candidate sequence $c2$.

condition 2:

else if the MIS value of the last item in a sequence (denoted by s2) is less than (<) the MIS value of every other item in s2 **then** // s1 and s2 can be equal

[A similar method to the one above can be used in reverse order.]

Sequence s1 joins with s2 **if** (1) the subsequences obtained by dropping the *second last* item of s2 and the *first* item of s1 are the same, **and** (2) the MIS value of the *first* item of s1 is greater than that of the *last* item of s2. Candidate sequences are generated by *prepending* s2 with the *first* item of s1:

if the *first* item l in s1 is a separate element **then**

{l} is *prepended* at the *beginning* of s2 as a separate element to form a candidate sequence c1.

if (the length and the size of s2 are both 2) **AND** (the *first* item of s1 is greater than the *first* item of s2) **then** // maintain lexicographic order

l is added at the *beginning* of the *first* element of s2 to form another candidate sequence c2.

else if ((the length of s2 is 2 and the size of s2 is 1) **AND** (the *first* item of s1 is greater than the *first* item of s2)) **OR** (the length of s2 is greater than 2) **then**

the *first* item in s1 is added at the *beginning* of the *first* element of s2 to form the candidate sequence c2.