Algorithm: The Pincer-Search algorithm

```
$L_0 := \emptyset$; $k := 1$; $C_1 := \{\{i\} \mid i \in I\}$; $S_0 := \emptyset$
MFCS := {{1, 2, . . . , $n$}}; MFS := $\emptyset$
while $C_k \neq \emptyset$
    read database and count supports for $C_k$ and MFCS
    MFS:=MFS U {frequent temsets in MFCS}
    $L_k$:={frequent k-itemset}
    $S_k$ := {infrequent itemsets in $C_k$}
    call the MFCS-gen algorithm if $S_k \neq \emptyset$ //    MFS=MFS U {frequent itemsets in MFCS }
    call MFS-pruning procedure
    generate $C_{k+1}$ from $L_k$ (apriori join)
    if any frequent itemset in $L_k$ is removed in MFS-pruning procedure
            call the recovery procedure to recover candidates to $C_{k+1}$
    call MFCS prune procedure to prune candidates in $C_{k+1}$
    $k := k + 1$
end-while
Answer = $U_k$ $L_k$ U MFS
```

NOTE: MFCS – Maximal Frequent Candidate Set
MFS – Maximal Frequent Set

MFCS-Gen Algorithm

```
for all itemset $s$ in $S_k$
    for all itemsets $m$ in MFCS
        if $s$ is a subset of $m$
            MFCS := MFCS $\setminus$ { $m$ }
            for all items $e$ in itemset $s$
                if $m \setminus \{ e \}$ is not a subset of any itemset in the MFCS
                    MFCS := MFCS $\cup$ { $m \setminus \{ e \}$ }
return MFCS
```

Recovery

```
for all items $l$ in $L_k$
    for all items m in MFS
        if the first k-1 items in $l$ are also in m
         for i from j+1 to |m|
                /*suppose $m.item_j = l.item_{k-1}$ */
            $C_{k+1}=C_{k+1}U\{l.item_1,l.item_2,\ldots,l.item_k, m.item_i\}$
```

MFS-Prune

```
for all items $l$ in $L_k$
          if $l$ is a subset of any itemset in the current MFS
                    delete $l$ from $L_k$
```

MFCS-Prune

```
for all items c in $C_{k+1}$
          if c is not a subset of any itemset in the current MFCS
                    delete c from $C_{k+1}$
```