**COMP4710 Project Report**

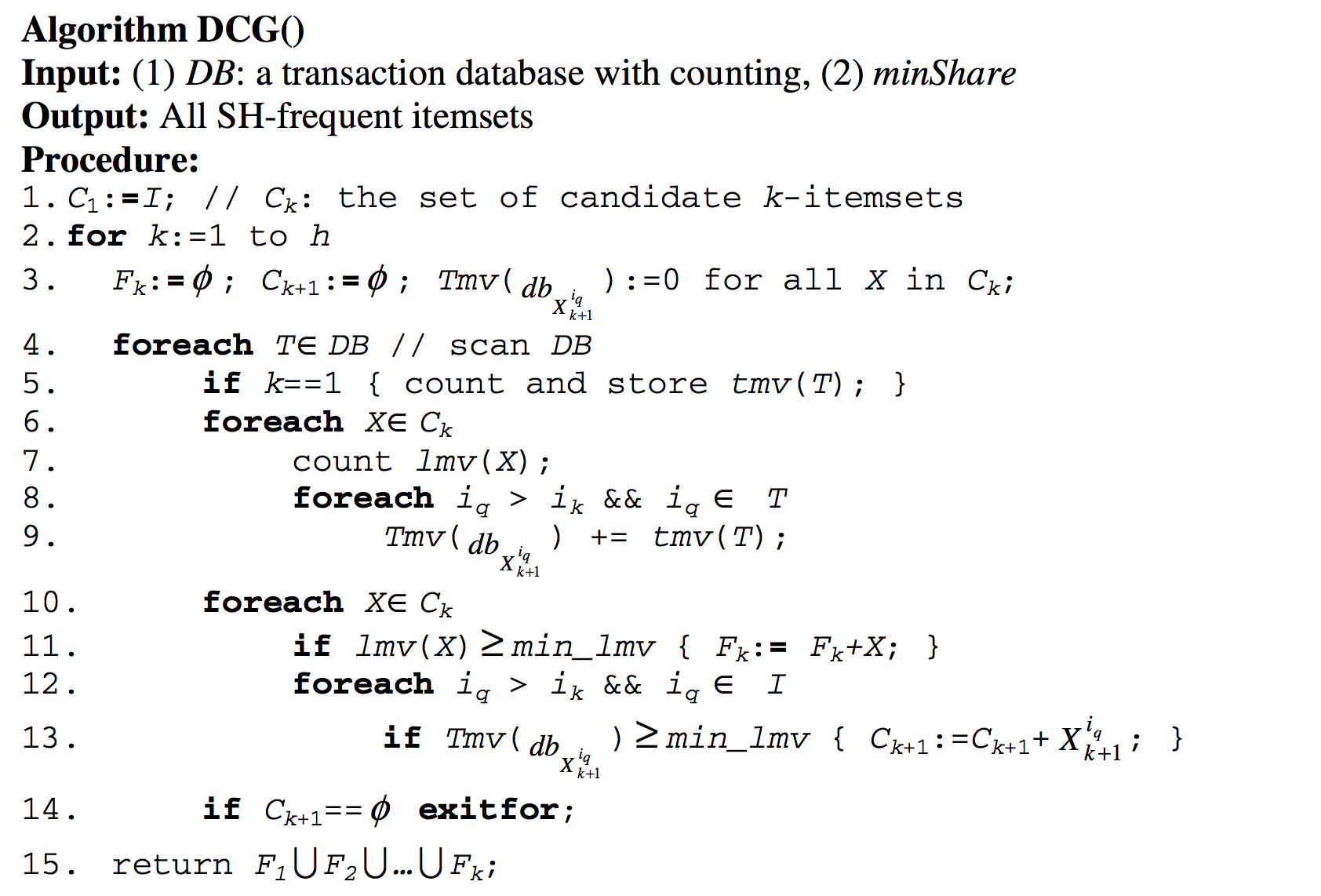
by Noel Ganotisi 6721898 and Ian Armit 7637713

**Description:**

The Direct Candidates Generation (DCG) algorithm directly generates candidates without the prune and the join steps, as well as, generating less candidates than in previous algorithms such as ShFSM.[1]

Below is the pseudo code for the DCG algorithm of the original specification.

**Figure 1: [1]**



**Deviations from the original specification:**

* The min\_lmv and tmv(T) are calculated before getting into the DCG algorithm.
* Because of how we stored each TMV value, it is unnecessary to need a nested loop to generate the next set of candidates (line 12-13 of pseudo code), instead we just need to loop through the hashmap of tmv values outside of looping through each candidate item.

**How to run the algorithm:**

Refer to README.txt

**Experimental results:**

**Running times:**

100 transactions took 10 msec

10,000 transactions took 130 msec

100,000 transactions took 1260 msec

1,000,000 transactions took 12130 msec

This algorithm was run on an Intel I7- 4700MQ, 2.4 Ghz processor with 4 GB RAM on an Ubuntu Virtual Machine. Based on the running time, this appears to be a linear algorithm. Note that the data from the dcgtable.txt was replicated several times to increase the number of transactions, so running times may be skewed.

**References:**

[1] Y. Li et al (2005). Direct Candidates Generation: A Novel Algorithm for Discovering Complete Share-Frequent Itemsets. Available: http://proxy2.lib.umanitoba.ca/login?url=http://dx.doi.org/10.1007/11540007\_67