CS5421: Normalization Project

August 13, 2019

1 Preview

This report main explain the algorithms used to all the reachable minimal covers of a set functional dependencies Σ under schema R. The algorithms are consistent with those taught in the class and tutorial. The new thing is that I filter the repetitions of the functional dependencies and elimination case between each step. Because there would be so many functional dependencies and elimination case when we try to get Σ'' and Σ''' that the computation complexity is extremely large if we do not do the cases and functional dependencies.

2 Detailed Explanation

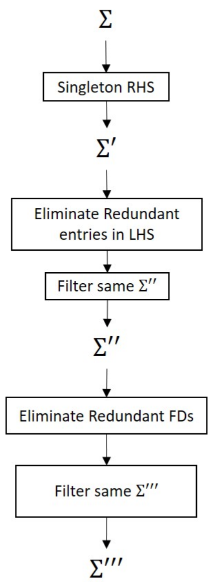
The algorithms in Question 2.b and Question 2.c are almost the same as getting all the minimal covers of the R with FDs (Question 2.c) means that get the reachable minimal covers of the Σ^+ (Σ in Question 2.b). We just need to add one step to get the Σ^+ for Question 2.c. We show the main body of function `min_covers(R,FD)` and `all_min_covers(R,FD)` and explain what we do in each step.

2.1 Σ to Σ'

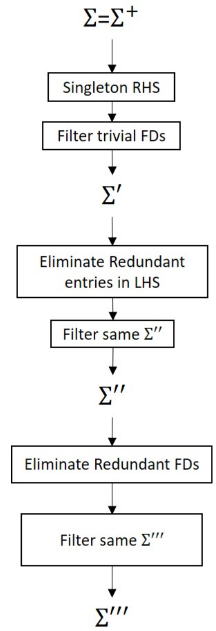
In this step, we mainly make the RHS of each FD as singleton. Note that this will generate many FDs if the number of entries in the RHS of the FDs are huge. Because the trivial FDs do no help to the later steps if the input is the Σ^+, we filter all the trivial ones among the Σ' to get the filtered Σ' when we try to get all the minimal covers of Σ. When we try to get all the reachable covers of Σ, the filter may cause missing some cases in Σ' to Σ''. We do not apply the trivial entry filter in the `min_covers(R,FD)`.

2.2 Σ' to Σ''

In this step, we remove the redundant entries in LHS of each FD. For example, for a FD $X \rightarrow Y \in \Sigma$, we update X by removing A from X if $\{A\} \subset (X - \{A\})^+$ or $Y \subset (X - \{A\})^+$ until there is no A in X such that $\{A\} \subset (X - \{A\})^+$ or $Y \subset (X - \{A\})^+$. Here, we use the recursion structure to get the results. The elimination of the redundant entries may have different cases. For example, $R = \{A, B, C\}$, $\Sigma = \{A \rightarrow B, B \rightarrow A, \{A, B\} \rightarrow C\}$ has two ways for $\{A, B\} \rightarrow C$, i.e., $A \rightarrow C$ or $B \rightarrow C$. Consequently, we get different cases of Σ. There may be some same cases among all the cases. We filter the same cases to reduce the number of cases the next step need deal with.



(a) Algorithms for Question 2.b



(b) Algorithms for Question 2.c

Figure 1: Diagrams

2.3 Σ'' to Σ'''

In this step, we eliminate the redundant FDs among all the FDs in each Σ'' cases. For example, we will eliminate $X \to Y \in \Sigma$ if $X \to Y$ can be obtained for $\Sigma - \{X \to Y\}$ until there is no $X \to Y \in \Sigma$ such that $X \to Y$ can be obtained from $\Sigma - \{X \to Y\}$. Here, we use the recursion structure to get the results. Like the last step Σ to Σ'', there are also different ways to eliminate the FDs. For example, $R = \{A, B, C\}$, $\Sigma = \{A \to B, B \to A, A \to C, B \to C\}$ has two ways, i.e., $\Sigma = \{A \to B, B \to A, A \to C\}$ or $\Sigma = \{A \to B, B \to A, B \to C\}$. There are also some same cases among all the cases. We filter the same cases in the end to get Σ.