DATABASE SYSTEMS

Practice Exercises on Normalization

CMPT 354, Course Section of Dr. E. Ternovska

Problem 1. Consider a schema with attributes A, B, C, D, E, F and FDs

$$D \to A \; , \qquad \qquad F \to B \; , \qquad \qquad DF \to E \; , \qquad \qquad B \to C$$

This is the same schema of Problem 2 in Practice 5.

- (a) Find a lossless BCNF decomposition. Is it dependency-preserving?
- (b) Is the schema in 3NF? If not, apply the 3NF synthesis algorithm to obtain a lossless, dependency-preserving 3NF decomposition.

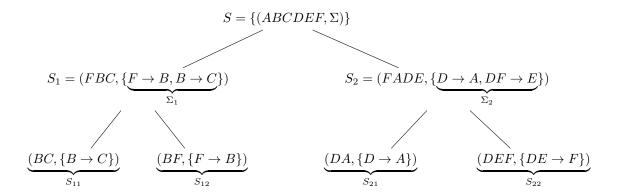
Solution. Let $S = (ABCDEF, \Sigma)$, where Σ is the given set of FDs.

- (a) From the previous practice, we know that the FDs $D \to A$, $F \to B$ and $B \to C$ violate BCNF (the l.h.s. is not a key), therefore S is not in BCNF. To decompose into BCNF we choose one violation: $F \to B$. We have $C_{\Sigma}(F) = FBC$, so we split S into S_1 and S_2 with attributes FBC and FADE, respectively. The FDs for S_1 are given by the projection of Σ onto FBC: $\Sigma_1 = \{F \to B, B \to C\}$. Observe that there is no other FD on attributes FBC that can be derived from Σ but not from Σ_1 (e.g., $F \to C$ can be derived from both Σ and Σ_1 , that's why I did not include it). The FDs for S_2 are given by the projection of Σ onto FADE: $\Sigma_2 = \{D \to A, DF \to E\}$. Observe that there is no other FD on attributes FADE that can be derived from Σ but not from Σ_2 . So now we have $S_1 = (FBC, \Sigma_1)$ and $S_2 = (FADE, \Sigma_2)$, and neither schema is in BCNF:
 - B is not a key for S_1 , so the FD $B \to C$ violates BCNF;
 - D is not a key for S_2 , so the FD $D \to A$ violates BCNF.

Using $B \to C$ to decompose S_1 , we get $C_{\Sigma_1}(B) = BC$ and so we split S_1 into schemas S_{11} and S_{12} with attributes BC and BF, respectively. The only FD for S_{11} is $B \to C$ and the only FD for S_{12} is $F \to B$. Observe that from Σ_1 we can derive $F \to C$, but this is on attributes FC, which is not a subset of either BC or BF. Clearly, both S_{11} and S_{12} are in BCNF.

Using $D \to A$ to decompose S_2 , we get $C_{\Sigma_2}(D) = DA$ and so we split S_2 into schemas S_{21} and S_{22} with attributes DA and DEF, respectively. The only FD for S_{21} is $D \to A$ and the only FD for S_{22} is $DF \to E$. Clearly, both S_{21} and S_{22} are in BCNF.

The decomposition process can be represented as the following tree:



The final lossless BCNF decomposition of S is given by the leaves of the above tree. If we take the union of all FDs in the decomposed schema, we obtain exactly Σ , so all dependencies are trivially preserved.

As an additional exercise, try to decompose S using $D \to A$ first and $B \to C$ after: you should obtain the same decomposition. This is not the case in general: a different choice of violations on which to split may lead to different decompositions.

- (b) From the previous practice we know that the only candidate key is DF and so the prime attributes are D and F. The schema S is not in 3NF, because the l.h.s. of the FDs $D \to A$ is not a key, and its r.h.s. is not prime. To synthesize a 3NF decomposition we need a minimal cover of Σ .
 - 1. The FDs in Σ are already in standard form (only one attribute in the r.h.s.).
 - 2. The only l.h.s. we could minimize is that of $DF \to E$, but DF is a candidate key so we cannot remove any attribute without compromising equivalence to Σ .
 - 3. It is easy to see that we cannot remove any FD:
 - $\{F \to B, DF \to E, B \to C\} \not\models D \to A$
 - $\{D \rightarrow A, DF \rightarrow E, B \rightarrow C\} \not\models F \rightarrow B$
 - $\{D \rightarrow A, F \rightarrow B, B \rightarrow C\} \not\models DF \rightarrow E$
 - $\{D \rightarrow A, F \rightarrow B, DF \rightarrow E\} \not\models B \rightarrow C$

So the given set of FDs Σ is already a minimal cover. We now apply the 3NF synthesis algorithm:

1. For each FD in Σ we create a relation:

$$(DA, \{D \rightarrow A\}), (FB, \{F \rightarrow B\}), (DFE, \{DF \rightarrow E\}), (BC, \{B \rightarrow C\})$$

- 2. Since there is already a relation, namely $(DEF, \{DF \to E\})$, whose set of attributes is a key for the *original schema S*, we don't need to add one.
- 3. None of the above relations have a set of attributes which is contained in the set of attributes of another, so we are done.