Introduction to Databases (winter term 2004/2005)

Assignment 7

Let $R = (U, F)$ be a relational schema. Prove the following propositions:

a) $R$ is in 3NF $\Rightarrow$ $R$ is in 2NF.

b) $R$ is in 2NF $\nRightarrow$ $R$ is in 3NF.

c) $R$ is in BCNF $\Rightarrow$ $R$ is in 3NF.

d) $R$ is in 3NF $\nRightarrow$ $R$ is in BCNF.

Prove the following proposition:
For relational schemas that have a unique key\(^1\), BCNF and 3NF are equivalent.

Consider a relational schema with the following attributes:

$C$ (Course), $T$ (Teacher), $H$ (Hour), $R$ (Room), $S$ (Student), $G$ (Grade).

Furthermore, the following FDs are assumed:

$$C \rightarrow T, GR \rightarrow C, HT \rightarrow R, CS \rightarrow G, HS \rightarrow R, HR \rightarrow T.$$

Use the synthesis algorithm to decompose the relational schema.

Use the decomposition algorithm to decompose the relational schema of task 7.3.

\(^1\) e.g. there is only one candidate key
Task 7.5 (Lossless Join Property)  
(2 Points)
It is obvious, that any correct and complete set of FD deduction rules (like e.g. the RAP rule set) can be used to decide, whether a decomposition \( D \) of a relational schema \( R = (U, F) \) preserves the dependencies \( F \).

The following algorithm (taken from Elmasri/Navathe "Fundamentals of Database Systems") checks whether a decomposition is loss-free.

### Algorithm 15.2 Testing for the lossless (nonadditive) join property

**Input:** A universal relation \( R \), a decomposition \( D = \{R_1, R_2, ..., R_m\} \) of \( R \), and a set \( F \) of functional dependencies.

1. Create an initial matrix \( S \) with one row \( i \) for each relation \( R_i \) in \( D \), and one column \( j \) for each attribute \( A_j \) in \( R \).
2. Set \( S(i,j) := b_{ij} \) for all matrix entries.
   (* each \( b_{ij} \) is a distinct symbol associated with indices \((i,j)\)*)
3. For each row \( i \) representing relation schema \( R_i \)
   {for each column \( j \) representing attribute \( A_j \)
   {if (relation \( R_i \) includes attribute \( A_j \) then set \( S(i,j) := a_{ij}; \)}
   (* each \( a_{ij} \) is a distinct symbol associated with index \((j)\)*)
4. Repeat the following loop until a complete loop execution results in no changes to \( S \)
   {for all rows in \( S \) which have the same symbols in the columns corresponding to attributes in \( X \)
   {make the symbols in each column that correspond to an attribute in \( Y \) be the same in all these rows as follows: if any of the rows has an "\( a \)" symbol for the column, set the other rows to that same "\( a \)" symbol in the column. If no "\( a \)" symbol exists for the attribute in any of the rows, choose one of the "\( b \)" symbols that appear in one of the rows for the attribute and set the other rows to that same "\( b \)" symbol in the column ;\};}

5. If a row is made up entirely of "\( a \)" symbols, then the decomposition has the lossless join property; otherwise it does not.

Consider the following relational schema:

\[ R = (\ ABCDEFGHI, \]
\[ \{A \rightarrow BDE, B \rightarrow ACDE, DE \rightarrow A, C \rightarrow DF, D \rightarrow I, I \rightarrow G, F \rightarrow G, FD \rightarrow H, D \rightarrow G\} \]

Decide by using the above algorithm, whether the decomposition of \( R \) into the attribute sets

\[ ABCDE, CDF, DIG, FDHG \]

is lossless or not.

Write down each change of the matrix in the 4th step.