Introduction to Databases
Tutorial 6

Dr Paolo Guagliardo

Fall 2020 (week 8)

Requirement for tutorial marks: Attempt all parts of at least one of the two problems below.

Problem 1. Consider a relation schema over attributes A, B, C, D, E, F with the following set of FDs:

$$EF \to BC\ , \qquad A \to D\ , \qquad B \to AE\ , \qquad BD \to C$$

(a) Find all candidate keys and prime attributes of the schema.

(b) List all of the FDs that violate the requirements of BCNF.

(c) Apply the BCNF decomposition algorithm. Is the resulting schema dependency-preserving?

Solution. Call Σ the given set of FDs and U the given set of attributes (that is, $ABCDEF$).

(a) We do not want to explicitly build the DAG of the powerset of U (which would be very big in this case), so we take a lazy approach that within that graph looks only at what is needed on the fly. The first step is to try and prune the graph as much as possible by looking for attributes that do not appear in the r.h.s. of any of the given FDs. These attributes must be given, as they cannot be derived using the FDs, and therefore they must be part of every key. F does not appear in the r.h.s. of any of the given FDs, hence every key must contain it. This means that, in the graph, we can remove every node that does not include F. We then proceed normally, but without explicitly writing down the graph.

$C_\Sigma(F) = F$, so F is not a key (so F is removed from the graph). We then try to add one attribute at a time (that is, we check the parents of F in the graph):

- $C_\Sigma(AF) = AFD$, so AF is not a key. We need to further expand this node.
- $C_\Sigma(BF) = BFAEDC$, so BF is a candidate key. We do not need to explore any node that contains BF (because BF and all of its ancestors are removed from the graph).
- $C_\Sigma(CF) = CF$, so CF is not a key. We need to further expand this node.
- $C_\Sigma(DF) = DF$, so DF is not a key. We need to further expand this node.
- $C_\Sigma(EF) = EFBCAD$, so EF is a candidate key. We do not need to explore any node that contains EF (because EF and all of its ancestors are removed from the graph).

We now try and extend AF, CF and DF to be keys; in particular, we need to check each of these nodes' parents that are still in the graph. If we added B or E we would get a key, but it would not be minimal, as it would contain BF or EF, which are minimal (indeed, all nodes containing BF and EF have been removed from the graph for this reason). So we have:

- $C_\Sigma(ACF) = ACFD$, so ACF is not a key.
- $C_\Sigma(ADF) = ADF$, so ADF is not a key.

- $C_\Sigma(CDF) = CDF$, so CDF is not a key.

We are still not done. We need to try whether we can extend ACF, ADF and CDF to be keys. If we added B or E we would get a key, but it would not be minimal, as it would contain BF or EF, which are minimal. So the only combination that remains to be checked is $ACDF$. But $C_\Sigma(ACDF) = ACDF$, so $ACDF$ is not a key. Thus the only candidate keys are BF and EF. In turn, the prime attributes of the schema are B, E and F.

(b) All the FDs in Σ are non-trivial (the r.h.s. is not a subset of the l.h.s.), so their l.h.s. would be required to be a key in order to satisfy the BCNF conditions. We know from the previous points that any set of attributes that does not include BF or EF (which are the candidate keys of the schema) is not a key. So the FDs $A \to D$, $B \to AE$ and $BD \to C$ violate BCNF.

(c) Let $S = (ABCDEF, \Sigma)$, where Σ is the given set of FDs. To decompose into BCNF we choose one violation: $B \to AE$. We have $C_\Sigma(B) = BAEDC$, and so we split S into S_1 and S_2 with attributes $BAEDC$ and BF, respectively.

The FDs for S_1 are given by the projection of Σ onto $BAEDC$: $\Sigma_1 = \{A \to D, B \to AE, BD \to C\}$ (no other FD on attributes $BAEDC$ can be derived from Σ but not from Σ_1). There are no FDs for S_2, as Σ does not imply any non-trivial FD on attributes BF.

We have $S_1 = (BAEDC, \Sigma_1)$ and $S_2 = (BF, \varnothing)$. The latter is trivially in BCNF, but the former is not because A is not a key for S_1 so $A \to D$ violates BCNF. Since $C_{\Sigma_1}(A) = AD$, we split S_1 into S_{11} and S_{12} with attributes AD and $ABEC$, respectively.

The only FD for S_{11} is $A \to D$. The FDs for S_{12} are given by projecting Σ_1 onto $ABEC$; we get $\Sigma_{12} = \{B \to AE, B \to C\}$. Observe that $B \to C$ was not in Σ_1, but it is implied by it and so we must include it when projecting the FDs; no other FD on attributes $ABEC$ can be derived from Σ_1 that cannot be derived also from Σ_{12}. Both S_{11} and S_{12} are now in BCNF.

The decomposition process can be represented as the following tree:

$$S = \{(ABCDEF, \Sigma)\}$$

$$S_1 = (BAEDC, \underbrace{\{A \to D, B \to AE, BD \to C\}}_{\Sigma_1}) \qquad S_2 = (BF, \varnothing)$$

$$\underbrace{(AD, \{A \to D\})}_{S_{11}} \qquad S_{12} = (ABEC, \underbrace{\{B \to AE, B \to C\}}_{\Sigma_{12}})$$

The final lossless BCNF decomposition of S is given by the leaves of the above tree. The union of all the FDs in the decomposed schema is $\Sigma' = \{A \to D, B \to AE, B \to C\}$; the decomposition is not dependency preserving because Σ' and Σ do not have the same closure. Observe that while we lost $EF \to BC$ because it cannot be derived from Σ', we did not lose $BD \to C$ because $\Sigma' \models BD \to C$.

Problem 2. Consider a relation schema over attributes A, B, C, D, E, F with the following set of FDs:

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

(a) Find the prime attributes and candidate keys of the schema.

(b) List all of the FDs that violate the requirements of BCNF.

(c) Apply the BCNF decomposition algorithm. Is the resulting schema dependency-preserving?

Solution. Call Σ the given set of FDs.

(a) The attributes D and F do not appear in the r.h.s. of any of the given FDs, hence every key must contain them (since they cannot be derived in any way). The closure of DF w.r.t. Σ is $ABCDEF$, so DF is the only candidate key. In turn, the prime attributes of the schema are D and F.

(b) All of the FDs in Σ are non-trivial (the r.h.s. is not a subset of the l.h.s.), so their l.h.s. would be required to be a key in order to satisfy the BCNF conditions. Let us see whether that's the case:

- $C_\Sigma(D) = DA$, so D is not a key and the FD $D \to A$ violates BCNF. At this point, we can already conclude that the given schema is not in BNCF (one violation is enough).
- $C_\Sigma(F) = FBC$, so F is not a key and the FD $F \to B$ violates BCNF.
- $C_\Sigma(DF) = DFEABC$, so DF is a key and the FD $DF \to E$ does not violate BCNF.
- $C_\Sigma(B) = BC$, so B is not a key and the FD $B \to C$ violates BCNF.

(c) Let $S = \{(ABCDEF, \Sigma)\}$. From the previous point, the FDs $D \to A$, $F \to B$ and $B \to C$ in Σ violate BCNF. To decompose S 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 is 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:

$$S = \{(ABCDEF, \Sigma)\}$$

$$S_1 = (FBC, \{\underbrace{F \to B, B \to C}_{\Sigma_1}\}) \qquad S_2 = (FADE, \{\underbrace{D \to A, DF \to E}_{\Sigma_2}\})$$

$$\underbrace{(BC, \{B \to C\})}_{S_{11}} \qquad \underbrace{(BF, \{F \to B\})}_{S_{12}} \qquad \underbrace{(DA, \{D \to A\})}_{S_{21}} \qquad \underbrace{(DEF, \{DF \to E\})}_{S_{22}}$$

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 will 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.