CS 353 Fall 2024
Homework 5 Solutions

Q.1 [10 pts, 5 pts each]
(a) Decomposition is not lossless, because the intersection of ABCD and BE (i.e., B) is not a superkey for any of the relations R1 and R2 (B^+ = BD).

(b) Decomposition is lossless, because the intersection of ABD and ACE (i.e., A) is a superkey for the relation R1 (A^+ = ABD).

Q.2 [15 pts, 3 pts each]
(a) B^+ = BG. So, B $\rightarrow$ A does not hold on R.

(b) A^+ = ABFG. So, A $\rightarrow$ D does not hold on R.

(c) AD^+ = ABDFG. So, AD $\rightarrow$ FG holds on R.

(d) AC^+ = ABCDEFG. So, AC $\rightarrow$ D holds on R.

(e) BC^+ = ABCDEFG. So, BC $\rightarrow$ F holds on R.

Q.3 [15 pts]
E → C using E → A, A → C (transitivity)
ED → CD using E → C (augmentation)
BE → DE using B $\rightarrow$ D (augmentation)
BE → CD using BE → DE, ED → CD (transitivity)

Q.4 [15 pts] F1 and F2 are not equivalent. To be equivalent we must have $F1^+ = F2^+$ (i.e., functional dependencies in F1 are implied by F2, and functional dependencies in F2 are implied by F1.) However, we have a functional dependency in F2 violating this: A $\rightarrow$ B which is not implied by F1 (A^+ under F1 is AC which does not include B).

Q.5 [20 pts] For A → B, A is not a superkey (A^+ = AB). This is a violation of BCNF.
(C → E and E → G also violate BCNF.)
Using the violation A → B, R is decomposed into AB and ACDEG.
C → E violates BCNF for ACDEG. We decompose ACDEG into CE and ACDG.
C → E and E → G imply C → G which violates BCNF for ACDG.
ACDG is decomposed into CG and ACD which are both in BCNF.
As a result, R is replaced by AB, CE, CG and ACD which are all in BCNF.

Q.6 [25 pts]
(a) [10 pts] B→C and B → D are combined into B → CD: {A → C, BCD → A, C→D, B→CD}
C is extraneous in BCD → A, since A is in $(BD)^+$, we replace BCD → A by BD → A:
{A → C, BD → A, C→D, B→CD}
D is extraneous in BD → A, since A is in $(B)^+$, we replace BD → A by B → A:
{A → C, B → A, C→D, B→CD}

$B \rightarrow A$ and $B \rightarrow CD$ are combined into $B \rightarrow ACD$: $\{A \rightarrow C, B \rightarrow ACD, C \rightarrow D\}$
Check if A is extraneous in $B \rightarrow ACD$:
B^+ under $\{A \rightarrow C, B \rightarrow CD, C \rightarrow D\}$ is BCD which doesn't include A, so A is not extraneous.
Check if C is extraneous in $B \rightarrow ACD$:
B^+ under $\{A \rightarrow C, B \rightarrow AD, C \rightarrow D\}$ is ABCD which includes C, so C is extraneous.
We are left with $\{A \rightarrow C, B \rightarrow AD, C \rightarrow D\}$.
Check if D is extraneous in $B \rightarrow AD$:
B^+ under $\{A \rightarrow C, B \rightarrow A, C \rightarrow D\}$ is ABCD which includes D, so D is extraneous.
As a result, Fc = $\{A \rightarrow C, B \rightarrow A, C \rightarrow D\}$

(b) [15 pts] We first find the candidate key(s) of R.
B must be part of any candidate key since it does not appear on the right hand side of any FD.
B^+ = ABCD. B is both unique and minimal. Therefore, B is the only candidate key.

We now check if R is in 3NF.
For $A \rightarrow C$, A is not a super key (A^+ = ACD) and C is not part of a candidate key. Therefore, R is not in 3NF.

Using the lossless and dependency preserving 3NF decomposition algorithm, we add one relation for each FD in Fc which was computed in part (a): AC, BA, CD.
Since the candidate key B is included in one of these 3 relations, we do not add any more relation.
There is no redundant relations.
As a result, R is decomposed into three 3NF relations: AC, BA, CD.