CS 353 Fall 2022
Homework 5 Solutions

Q.1 [10 pts]
a, c, e, f

Q.2 [20 pts, 10 pts each]

(a)
The decomposition is lossless. AB ∩ AC = A is the key of R1.

(b)
The decomposition is not lossless (it is lossy). AB ∩ BC = B is not a key for R1 or R2. This can be shown through an example instance of R:

R:

A	B	C
a1	b1	c1
a2	b1	c2

R1:

A	B
a1	b1
a2	b1

R2:

B	C
b1	c1
b1	c2

R1 $\bowtie$ R2:

A	B	C
a1	b1	c1
a1	b1	c2
a2	b1	c1
a2	b1	c2

So, R ≠ R1 $\bowtie$ R2

Q.3 [24 pts, 8 pts each]

(a)
CA and CD
Since attribute C is not determined by any other attribute it must be a part of any candidate keys. First check whether C is a candidate key.
C+ = C, it is not a super key, thus it cannot be a candidate key.
Add one more attribute to C and check whether it is a candidate key.
CA+ = ABCDE
CB+ = BCE
CD+ = ABCDE
CE+ = CE
CA and CD are super keys (unique) and they are also minimal. Therefore, CA and CD are candidate keys.

(b)
It is not in BCNF. For A → D, A is not a super key.

(c)
It is not in 3NF. For BC → E, BC is not a super key and E is not part of any candidate key.

Q.4 [22 pts]
(a) [10 pts]
For A → B in F, we compute A+ in G.
A+ in G is ABC which includes B. So, A → B is inferred from G.
For AB → C in F, we compute AB+ in G.
AB+ in G is ABC which includes C. So, AB → C is inferred from G.
For D → AC in F, we compute D+ in G.
D+ in G is ABCDE which includes AC. So, D → AC is inferred from G.
For D → E in F, we compute D+ in G.
D+ in G is ABCDE which includes E. So, D → E is inferred from G.

(b) [6 pts]
No. E → B is not inferred from F. E+ in F is E which does not include B.

(c) [6 pts]
No. In order to be equivalent, each set must cover the other one. Since F does not cover G, these two functional dependency sets are not equivalent.

Q.5 [24 pts, 12 pts each]
(a)
D is extraneous in CD → B, since B is in $(C)^+$, we replace CD → B by C → B:
{A → BD, C → B, C→D, B → D}
C →B and C →D are combined into C → BD: {A → BD, C → BD, B → D}
D is extraneous in A → BD, since A+ under {A → B, C → BD, B → D} includes D.
We replace A → BD by A → B: {A → B, C → BD, B → D}
D is extraneous in C → BD, since C+ under {A → B, C → B, B → D} includes D.
We replace C → BD by C → B: {A → B, C → B, B → D}
No other extraneous attributes. As a result, Fc = {A → B, C → B, B→D}

(b)
We first find the candidate key(s) of R.
AC must be part of any candidate key since A and C do not appear on the right hand side of any FD.
AC+ = ABCD. AC is both unique and minimal. Therefore, AC is the only candidate key.

We now check if R is in 3NF.
For A → B, A is not a super key and B 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: AB, CB, BD.
Since the candidate key AC is not included any of these 3 relations, we add relation AC as well.
There is no redundant relations.
As a result, R is decomposed into 4 3NF relations: AB, CB, BD and AC.