# HW4 Solution [1]

April 1, 2018

[1] CS 5530 Database Systems; Spring 2018 Instructor: Feifei Li, University of Utah

# Problem 1

1) Proof: By definition, when $A \to B$, we have: for any valid relational instance $r$ conforming to the schema $R$, and any two records $t_1$ and $t_2$ from $r$, if $\pi_A(t_1) = \pi_A(t_2)$, then $\pi_B(t_1) = \pi_B(t_2)$. Similarly, when $B \to C$, we have: for any valid relational instance $r$ conforming to the schema $R$, and any two records $t_1$ and $t_2$ from $r$, if $\pi_B(t_1) = \pi_B(t_2)$, then $\pi_C(t_1) = \pi_C(t_2)$. Thus, we have: for any two records $t_1$ and $t_2$ from $r$, if $\pi_A(t_1) = \pi_A(t_2)$, then $\pi_C(t_1) = \pi_C(t_2)$, which by definition implies that $A \to C$.

2) Proof: Given $A \to B$ and using augmentation rule, we have $AC \to BC$. Given $C \to D$ and using augmentation rule, we have $BC \to BD$. Now using transitivity, we have $AC \to BD$.

3) Proof: We prove this by contradiction. Assume that $AB \to C$ does imply $A \to C$. Consider the following example. Assume $R = KABC$, where $K$ is the primary key; a valid instance $r$ that conforms to $R$ is as follows:

| K | A | B | C |
|---|---|---|---|
| 1 | 10 | 5 | 7 |
| 2 | 10 | 5 | 7 |
| 3 | 10 | 4 | 8 |
| 4 | 10 | 4 | 8 |

Clearly, this instance satisfies that $AB \to C$, but it does satisfy $A \to C$, which has vioated our assumption.

# Problem 2

1) An employee may choose how many hours he/she'd like to work on a project on any given date. True or False. Why?

False. E→J means that an employee works only on one project, and EJ→H means that an employee works the same number of hours on the same project no matter on which date.

2) An employee may work on multiple different projects. True or False. Why?

False. We have E→J which implies that an employee can only work on one project.

3) An employee receives only one rating for a project. True or False. Why?

False. We have EJD→R which implies that an employee may receive multiple different ratings on different dates on the same project.

4) Multiple employees may work on the same project. True or False. Why?

True. We do not have J→E.

5) What's the key of $W$? Show your steps.

E→J implies that E→EJ, given EJ→H, this means that E→EJH. Now, E→EJ implies that ED→EJD. Hence, given EJD→R, we have ED→R. Now, given E→EJH and ED→R, we have ED→EJHRD. Hence, the key of $W$ is ED. That's the only key for $W$.

6) Is W in BCNF? Why or why not?

W is NOT in BCNF. Consider just E→J, the LHS is not a superkey (and clearly, it is not a trivial FD). This has violated the BCNF requirement.

7) Decompose $W$ into BCNF. Makes sure your decomposition is lossless, and show your steps. Is your decomposition dependency preserving?

Given E→J, we can get E→H, and ED→R. So $F^{-}$={E→H, E→J, ED→R}. Hence, we decompose P into R'=EJRD, and R2=EH, and then R' is further decomposed into R0=ERD, and R1=EJ. It's easy to check that R0=ERD, R1=EJ, and R2=EH are dependency-preserving and lossless-join.

8) Write the FD for the following constraint with respect to W=EJHRD: All employees work on the same project on one day must work the same number of hours on that date. Briefly justify your answer.

JD→H, a unique combination of pid and date determines the number of hours anyone needs to work for that project on that date.

## Problem 3

1) Suppose R={W, Z, A}. Hence, X=WZ, and Y=R-Z=WA. As a result, X∩Y=W (given that W∩Z=∅). Now, given that W→Z, and W→W, we have W→WZ (union rule). Hence, X∩Y→X, which means that the decomposition is lossless.

2) Without loss of generality, suppose that W∩Z=A, then schema R could be written as {X, (W-A), A, (Z-A)} for some X and A. Clearly, given that W→Z, we have W→(Z-A). Also note that W∩(Z-A)= ∅. Hence, we can decmpose R into W(Z-A), and R-(Z-A)=XW and such a decomposition is lossless-join decmposition.

## Problem 4

1) R→W has violated the BCNF requirement, as R is not a superkey. R→W has also violated the 3NF requirement, as R is not a superkey and R is also not a subset of any key.

2) $R_1$∩$R_2$=R, and R→RW=$R_2$; hence, this is lossless.

$F_1$ = {S→SNLRH}; $F_2$ = {R→W}. S is the key of $R_1$, and R is the key of $R_2$. Hence, both relations are now in BCNF.

3) Note that $\{F_1 \cup F_2\}^+ = F^+$; in particular, S→R from $F_1$ and R→W from $F_2$ provides S→W that is missing after the decomposition.

## Problem 5

1) $F_X$ = {A→B, B→A}, and $F_Y$ = {B→C, C→B}; hence, $\{F_X \cup F_y\}^+ = F^+$; this is dependency preserving. Note that when deriving $F_X$ and $F_Y$, we need to use $F^+$ instead of simply using F.

This decomposition is also lossless, since $X \cap Y = C$ and C→BC= $Y$.

2) $F_X = \emptyset$, and $F_Y$ = { C→B}; $\{F_X \cup F_y\}^+ \neq F^+$ as A→B has gone missing. Hence, this is not dependency preserving.

This decomposition is still lossless though, since we will have $X \cap Y = C$ and C→BC= $Y$.