1. **[10 points] Exercise 8.26 page 373.**

**Use Armstrong’s axioms to prove the soundness of the decomposition rule.**

Armstrong’s Axioms:

1. (Reflexivity)

1. (Augmentation)
2. (Transitivity)

Decomposition Rule

We know (trivially) that and , then by reflexivity . Given then by transitivity, By the same reasoning, we can say , and reflexively Using transitivity again, we can see that and , then .

Therefore, the decomposition of is sound. ⏹

1. **[90 points; each part is 15 points] Exercise 8.29 page 373.**

**Consider the following set *F* of functional dependencies on the relation schema**

***r* (*A, B, C, D, E, F*): F = { *A*→*BCD, BC*→*DE, B*→*D, D*→*A}***

1. **Compute *B*+.**

B+ = B = BD = ABD = ABCD = ABCDE

1. **Prove (using Armstrong’s axioms) that *AF* is a superkey.**

(AF)+ = AF = ABCDF = ABCDEF → r, therefore AF is a superkey.

1. **Compute a canonical cover for the above set of functional dependencies *F*; give each step of your derivation with an explanation.**

*Fc = F = {A→BCD, BC→DE, B→D, D→A}*

Is D extraneous is A→BCD?

A+ = A = ABC = ABCDE which contains D. D is extraneous.

*Fc = {A→BC, BC→DE, B→D, D→A}*

Is C extraneous in BC→DE?

B+ = B = BDE = ABCDE, which contains C. C is extraneous.

We can eliminate B→D because it is a redundant functional dependency since B→DE.

Therefore the canonical cover is *Fc = {A→BC, B→DE, D→A}.*

1. **Give a 3NF decomposition of *r* based on the canonical cover.**

3NF iff for every nontrivial FD is a super key for R OR every attribute in is a member of a candidate key (K→R and no st →R) for R.

R1 = ABC

R2 = BDE

R3 = AD

No schemas contain a candidate key for R, therefore we need an R4, which can be any candidate key. Note: None of our functional dependencies generate F, so it must be in every candidate key. F+ = F. Consider immediate supersets. FA+ = R, FB+ = R, and FD+ = R, while FC+ and FE+ do not produce all attributes. Let R4 be the first of these minimal supersets. Then d = {ABC, BDE, AD, FA}.

1. **Give a BCNF decomposition of *r* using the original set of functional dependencies.**

R

BC→DE / \

BCDE R’ = ABCF

Since AF is a superkey for r, we know AF → BC holds on R’ and is a superkey there as well. d = {BCDE, ABCF}

1. **Can you get the same BCNF decomposition of *r* as above, using the canonical cover?**

*F = {* *A*→*BCD, BC*→*DE, B*→*D, D*→*A} Fc = {A→BC, B→DE, D→A}.*

The only hope of getting the same decomposition using F and Fc is to start with the same initial decomposition. Our only option is D→A. The similarities stop there, so unless the first decomposition offers us a complete BCNF decomp, then we can see that the next branch in the decomp will vary regardless of the choice.

R R

D→A / \ D→A / \

DA R’ = BCDEF DA R’ = BCDEF

BC→DE / \ B→DE / \

BCDE R”= BCF BDE R” = BCF

d = {AD, BCDE, BCF} d = {AD, BDE, BCF}