Lab3

Johanna Gerhardsen (johge201), Sofia Bertmar (sofbe892)

Task 1

a) **{C} → {B}**

We use decomposition on {C} → {A,D}, which entails that if {C} → {A,D} then {C} → {A} and {C} → {D}.
We use decomposition again on {A} → {B,C} which gives {A} → {B} and {A} → {C}.
Finally transitivity is used on {C} → {A} and {A} → {B} which results in {C} → {B}.

b) **{A,E} → {F}**

We use decomposition on {C} → {A,D}, which entails that if {C} → {A,D} then {C} → {A} and {C} → {D}.
We use decomposition again on {A} → {B,C} which gives {A} → {B} and {A} → {C}.
Since {A} → {C} and {C} → {D} then decomposition gives {A} → {D}.
Now we use pseudo-transitivity:
If {A} → {D} and {D,E} → F then {A,E} → {F}.

Task 2

a)

X+ = {A} = {A, B, C, D} because A points to B and C and C points to A and D which gives us X+ without E and F.

b)

X+ = {C, E} = {A, B, C, D, E, F}.

Task 3

a) **Determine the candidate key(s) for R.**

We want to find the attributes which are not present on the right hand side of the FDs. In our case only A is not present on the right hand side, meaning that it can not be determined by other attributes. Therefore A will definitely be part of the candidate keys. A+ is also equal to only A (since we do not have B), therefore we can not determine further candidate keys, meaning that there must be more. If we then choose A and B, and find the closure of (A,B)+, we can see that it is one candidate key because it gives all the attributes from the given relation R. The same result comes from finding the closure of (A,D)+, since it returns all of the attributes from R. No further minimum candidate keys can be found.

b) **Note that R is not in BCNF. Which FD(s) violate the BCNF condition?**

FD2 and FD3 violate the BCNF condition.

Input relation is not in BCNF: it is not in 3NF and not all functional dependencies satisfy at least one of the following conditions: (1) The right-hand side is a subset of the left hand side, or (2) the left-hand side is a superkey (or minimum key) of the relation. The functional dependencies that failed are: D → B; E → F.

c) **Decompose R into a set of BCNF relations, and describe the process step by step (don't forget to determine the FDs and the candidate key(s) for all of the relation schemas along the way).**

The candidate key (A,B) is a superkey for R and therefore doesn't violate BCNF. FD2 violates BCNF since E is not a key. FD3 also violates BCNF, since D is not a key. Choosing to start with FD2 and therefore we decompose the original attributes into two relations - R1(EF) and R2(ABCDE), R2 we get by subtracting the right handside of the violating FD from the original attributes.

- For R1(EF) the candidate key is E and the only FD that applies to this is FD2 E → F. Therefore it's in the BCNF.
- For R2(ABCDE) candidate keys are (A,B) and (A,D). Now we check if it is in BCNF. The third FD's LHS is not a superkey and the relation R2 is decomposed into R3(DB) and R4(ACDE).
- For R3(DB) the candidate key is D, the only FD applicable here is FD3 so it is in the BCNF.
- For R4(ACDE) we check if it is in the BCNF and can tell that the candidate key is (A,D) the functional dependency applicable here will be AD → EC.

The relations we get are therefore: AD → EC, E → F and D → B.

Task 4

a) **Show that R is not in BCNF.**

We want to find the attributes which are not present on the right hand side of the FDs. In our case B and C is not present on the right hand side, meaning that it can not be determined by other attributes. If we have B and C, and find the closure of (B,C)+, we can see that it is the only candidate key because it gives all the attributes from the given relation R.

b) **Decompose R into a set of BCNF relations (describe the process step by step).**

We start by finding the minimal cover of our FDs. Step 1 is to rewrite the FDs so that the right hand side only contains one attribute. FD3 will remain as is (C → D). FD2 will be decomposed into BCD → A and BCD → E. FD1 will be decomposed into ABC → D and ABC→ E. Due to FD3 we can now rewrite the left hand side of some FDs, which will give us: C→ D, BC → E and BC→ A. This gives us our minimal cover: FDX C → D & FDY BC→ AE.

Now we want to determine if FDs are in BCNF. FDX is not in BCNF since C is not a superkey and therefore we decompose into two relations: R1(CD) and R2(ABCE).

- For R1(CD) the candidate key is C and the only FD that applies to this is FDX C → D. Therefore it's in the BCNF.

- For R2(ABCE) candidate keys are (B,C). Now we check if it’s in the BCNF. Since FDY BC->AE is applicable here, this means that it’s already in the BCNF.

The relations we get are therefore C→ D and BC → AE.