CS-353 Section 2 HW5

Ömer Oktay Gültekin 21901413

Q.1 [12 pts, 6 pts each] Given relation R(A, B, C, D, E), determine whether the decomposition of R into ABC and ADE is lossless with the following set of functional dependencies:

(a) F = {A → C, A → D, E → D}.

Test: If $R_1 \cap R_2 \rightarrow R_1$ or $R_1 \cap R_2 \rightarrow R_2$, the decomposition is lossless.

⇒ $R_1 = ABC$, $R_2 = ADE$ ⇒ $ABC \cap ADE = \{A\}$. $A \rightarrow ADE$? No, $A \rightarrow ABC$? No ⇒

$A \rightarrow ACD$, the decomposition is not lossless.

(b) F = {A → B, A → D, D → E}.

$R_1 = ABC$, $R_2 = ADE$ ⇒ $ABC \cap ADE = \{A\}$. Since $A \rightarrow ABDE$, $A \rightarrow \underbrace{ADE}_{R2}$ is also true. Therefore, the decomposition is lossless.

Q.2 [20 pts] Given a relation R (A, B, C, D, E) with FDs A → BC, B → D, C → E.

(a) [5 pts] Determine if A → E holds on R.

Using decomposition rule, $A \rightarrow BC \Rightarrow A \rightarrow B$ and $A \rightarrow C$ holds

using transivity rule, $A \rightarrow C$ and $C \rightarrow E \Rightarrow A \rightarrow E$ holds.

$A \rightarrow E$ holds on R.

(b) [5 pts] Determine if B → E holds on R.

We only have $B \rightarrow BD$, $B \rightarrow E$ does not holds on R.

(c) [10 pts] Determine if R in BCNF. If not, decompose it into BCNF relations using the BCNF decomposition algorithm discussed in the class. Indicate which FD violates BCNF in each step of decomposition.

$A \rightarrow ABCDE = R \Rightarrow A$ is a superkey for R.

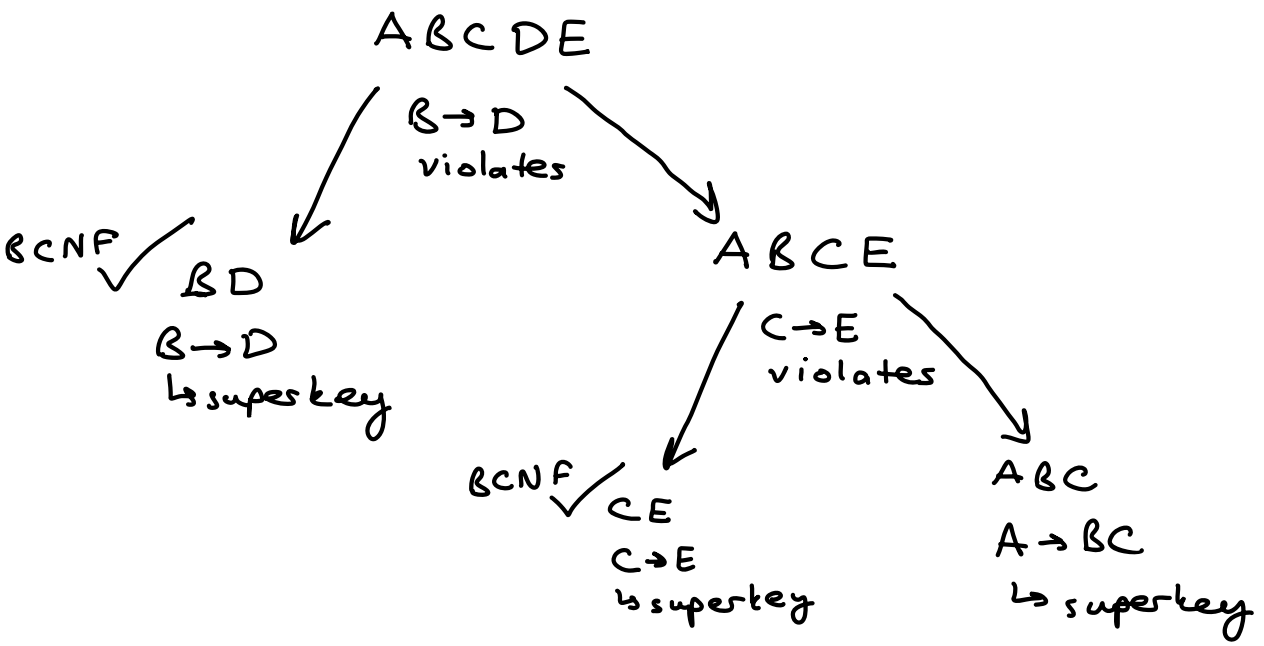
⇒ $A \rightarrow BC$ ✓

- $B \rightarrow D$, is B superkey? $B \rightarrow BD$ ⇒ not superkey
- $C \rightarrow E$, is C superkey? $C \rightarrow E$ ⇒ not superkey

→ violates BCNF.

Decompose R into: $-(\alpha \cup \beta)$ and

$-(R-(\beta-\alpha))$ for each violating FD



R is replaced by BD, CE, and ABC which are all in BCNF.

Q.3 [20 pts] Given a relation R(A, B, C, D, E, F, G) and its functional dependencies:
AD → F, AE → G, DF → BC, E → C, G → E
(a) [10 pts] Find the candidate key(s) of R. Show how you derived the key(s).

from the given FDs, it is seen that A and D are not appear in right side of the FDs, meaning that they do not determined by any attribute. Therefore, they should appear in all candidate keys.
we can start by adding trivial dependencies to FDs:

$AD \to F \Rightarrow AD \to ADF$

$AE \to G \Rightarrow AE \to AEG$

$DF \to BC \Rightarrow DF \to DFBC$

$E \to C \Rightarrow E \to EC$

$G \to E \Rightarrow G \to GE$

First check AD, since they are minimal.
$AD \to ADF$ and $DF \to DFBC \Rightarrow AD \to ABCDF$ by transivity rule. E and G can't be added.

since $AD \rightarrow R$ is not holds, AD is not a candidate key.

If we add G to both sides using augmentation rule:

$ADG \rightarrow ABCDFG$ and $G \rightarrow E \Rightarrow ADG \rightarrow ABCDEFG = R$

$\Rightarrow$ ADG is a candidate key.

If we add E to AD:

$ADE \rightarrow ABCDEF$ and $AE \rightarrow G \Rightarrow ADE \rightarrow ABCDEFG = R$

$\Rightarrow$ ADE is a candidate key.

There is no other possibility of candidate key containing A and D. Therefore, candidate keys are ADG and ADE. //

(b) [5 pts] Check if R is in BCNF. Why or why not?

None of the given FDs are trivial and none of them has at least ADG and ADE in their left hand side. Therefore, all FDs violates BCNF and R is not in BCNF. //

(c) [5 pts] Check if R is in 3NF. Why or why not?

R is not in BCNF. Therefore, we only need to check if any attribute A in right side is part of a cand. key (i.e right side should only consists of A, D, E, G or left side)

$AD \rightarrow F$, $DF \rightarrow BC$, $E \rightarrow C$ FDs violates 3NF. Therefore, R is not in 3NF. //

Q.4 [24 pts] Given the relation schema R(A, B, C, D, E) with the functional dependency set
F = {AB → C, A → E, C → DE, D → BE}
(a) [10 pts] Find a Canonical Cover Fc of F.
Show all your work.

Step 1: Replace $\alpha_1 \to \beta_1$ and $\alpha_1 \to \beta_2$ in F with $\alpha_1 \to \beta_1\beta_2$

Not applicable to F //

Step 2: Eliminate extraneous attributes at left hand sides of FDs

- For $AB \to C$ check if A is extraneous.

  B^+ (under F) = B not contains C ⇒ A is not extraneous in $AB \to C$
- For $AB \to C$ check if B is extraneous.

  A^+ (under F) = AE not contains C ⇒ B is not extraneous.
- $A \to E$, $C \to DE$, $D \to BE$ not applicable (1 element at left sides)

Step 3: Eliminate extraneous attributes at right sides of FDs

- $AB \to C$, $A \to E$ not applicable.
- for $C \to DE$ check if E is extraneous.

  C^+ (under $\{AB \to C, A \to E, C \to D, D \to BE\}$) = CDBE contains E ⇒ E is extraneous in $C \to DE$

  ⇒ $F' = \{AB \to C, A \to E, C \to D, D \to BE\}$
- For $D \to BE$ check if E is extraneous

  D^+ (under $\{AB \to C, A \to E, C \to D, D \to B\}$) = DB not contains E ⇒ E is not extraneous in $D \to BE$
- For $D \to BE$ check if B is extraneous

  D^+ (under $\{AB \to C, A \to E, C \to D, D \to E\}$) = DE not contains B ⇒ B is not extraneous in $D \to BE$

No other eliminations.

$F_c = F' = \{AB \to C, A \to E, C \to D, D \to BE\}$ //

Show all your work.

(b) [14 pts] Check if R is in 3NF. If not, decompose it into 3NF relations using the lossless and dependency preserving decomposition algorithm that makes use of the canonical cover you found in part (a).

Find candidate keys:

$R(A, B, C, D, E)$, $f_c = \{AB \to C, A \to E, C \to D, D \to BE\}$

A never appears in right side, A must be in candidate keys. Check A is the cand. key:

$A \to E \Rightarrow A \to AE \Rightarrow$ not cand. key since B, C, D missing.

Check AB: $AB \to C \Rightarrow AB \to ABC$ and $C \to D \Rightarrow AB \to ABCD$ and $D \to BE \Rightarrow AB \to ABCDE \Rightarrow AB$ is a cand. key

Check AC: $A \to E \Rightarrow AC \to EC$ and $C \to D \Rightarrow AC \to CDE$ and $D \to BE \Rightarrow AC \to BCDE \Rightarrow AC \to ABCDE \Rightarrow AC$ is a cand. key. (+trivial)

Check AD: $A \to E \Rightarrow AD \to ED$ and $D \to BE \Rightarrow AD \to BDE$ (+trivial) $\Rightarrow AD \to ABDE$ and $AB \to C \Rightarrow AD \to ABCDE \Rightarrow AD$ is a cand.

candidate keys: AB, AC, AD.

Check 3NF:

$AB \to C \Rightarrow AB$ is cand. key ✓

- $A \to E \Rightarrow$ A is not cand. key and E is not in any cand. keys ✗
- $C \to D \Rightarrow$ D is in cand. key AD ✓
- $D \to BE \Rightarrow$ D is not cand. key and E is not in any cand. keys ✗

→ Not in 3NF form.

Decomposition Algorithm:

For loop decomposes following relations:

$R1 = ABC$, $R2 = AE$, $R3 = CD$, $R4 = DBE$

At least one relation contains candidate key (R1 contains AB) ✓

There is no schema to delete that is the subset of other schema.

Thus, decomposed 3NF relations:

(A, B, C), (A, E), (C, D), (D, B, E) //

Q.5 [24 pts, 12 pts each] The following relation is used to maintain some information about the employees of a company. In addition to the years each employee worked in each particular section, various emails and addresses used by the employee are also stored in the relation.

Employee(id, name, from, to, section, email, address)

The following functional dependencies and multivalued dependencies are given for the Employee relation:

id → name

id, from, to → section

id →>> email

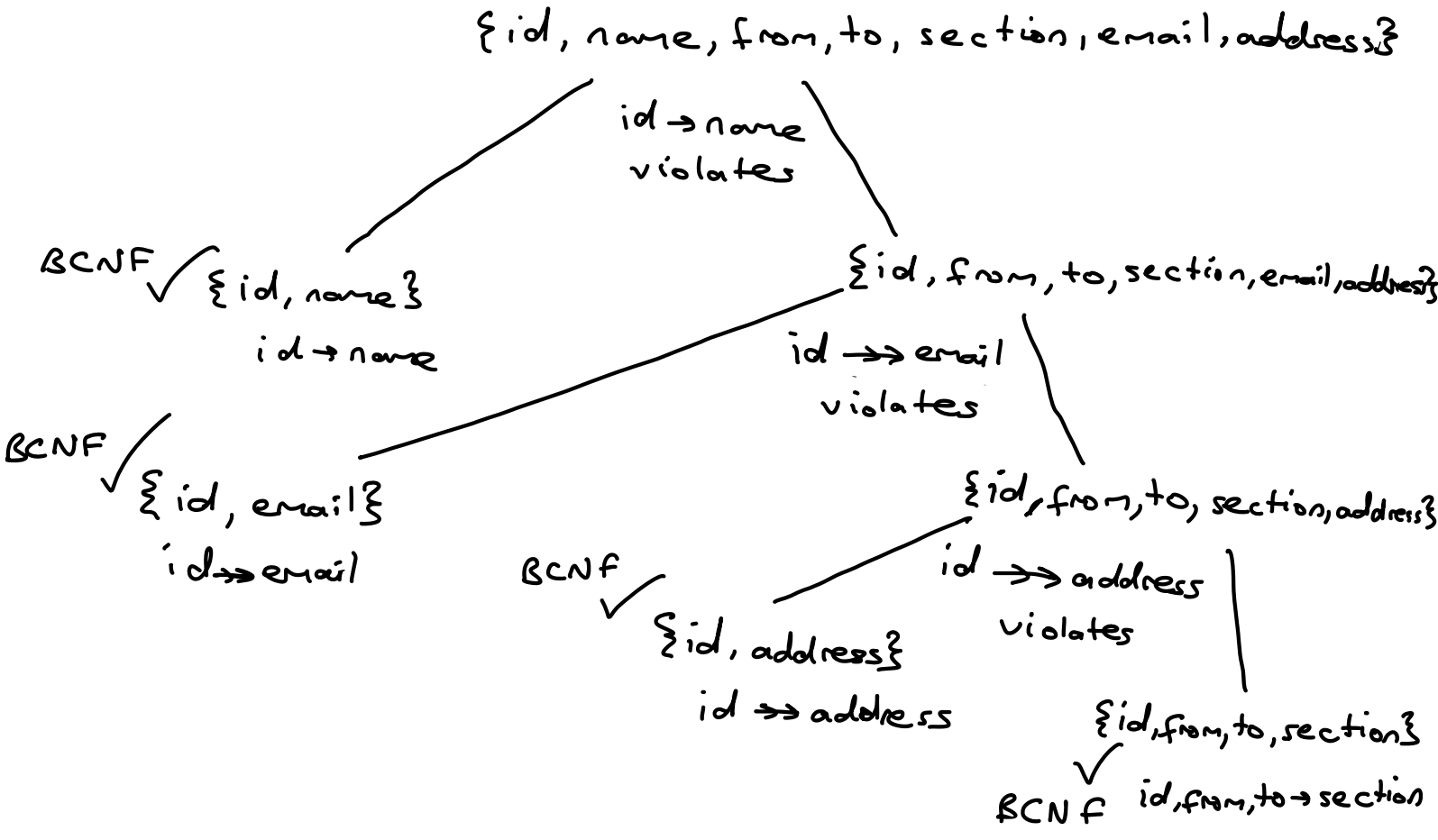
id →>> address

(a) Check if Employee is in BCNF. If not, decompose it into BCNF relations.

id → name ⇒ id → id, name ⇒ id is not a super key

⇒ Employee is not in BCNF.

BCNF Decomposition:



BCNF relations : {id, name}, {id, email}, {id, address}, {id, from, to, section} //

(b) Using the multivalued dependencies, check if each relation you found in part (a) is in 4NF. If not, decompose it into 4NF relations.

From part A:

$R_1 = \{id, name\}$

$R_2 = \{id, email\}$

$R_3 = \{id, address\}$

$R_4 = \{id, from, to, section\}$

using $\alpha \rightarrow \beta \Rightarrow \alpha \twoheadrightarrow \beta$:

Check R1:

id $\twoheadrightarrow$ name and $id^+ = id, name \Rightarrow$ id is a super key $\Rightarrow$ R1 is in 4NF.

Check R2:

id $\twoheadrightarrow$ email and $id^+ = id, email \Rightarrow$ id is a super key $\Rightarrow$ R2 is in 4NF.

Check R3:

id $\twoheadrightarrow$ address and $id^+ = id, address \Rightarrow$ id is a super key $\Rightarrow$ R3 is in 4NF.

Check R4:

id, from, to $\twoheadrightarrow$ section and $\{id, from, to\}^+ = id, from, to, section \Rightarrow \{id, from, to\}$ is a super key $\Rightarrow$ R4 is in 4NF.