CS1555 Recitation 11 Solution

Objective: to practice normalization, finding canonical forms, checking for lossless decompositions, and decomposing relations into BCNF.

**Part 1:** For each of the following relations R and sets of functional dependencies F, do the following:

1. Find the canonical cover (minimal cover) of F.
2. Using the canonical cover, find the keys of the R.

**1.** Consider the following set of functional dependencies F on a relation R (A, B, C, D, E):

A → BC

A → D

B → C

C → D

DE → C

BC → D

Finding the canonical form:

* Transform all FDs to canonical form (i.e., one attributes on the right):

A → B

A → C

A → D

B → C

C → D

DE → C

BC → D

* Drop extraneous attributes:

B in BC → D is extraneous, since we already have C → D. The set of FDs becomes:

A → B

A → C

A → D

B → C

C → D

DE → C

* Drop redundant FDs:

A → B and B → C implies A → C, so we drop A → C.

A → B, B → C and C → D implies A → D, so we drop A → D.

The set of FDs becomes:

A → B

B → C

C → D

DE → C

which is the canonical cover of F.

Finding the keys of R:

Observations:

* A and E do not appear in the right hand side of any FDs, so they have to appear in all keys of R.
* AE+ : AE → AEB (since A → B ) → AEBC (since B → C) → AEBCD (since C → D). So AE is a key of R.

In this case, we do not need to consider any other combination, because any other combination containing AE (e.g., AEB) is a super key and not minimal.

**2.** Consider the following set of functional dependencies F on relation R (A, B, C, D, E, H):

A → C

AC → D

E → AD

E → H

A → CD

E → AH

Finding the canonical form:

* Transform all FDs to canonical form (i.e., one attribute on the right):

A → C

AC →D

E → AD becomes E →A and E→D

E → H

A→ CD becomes A→C and A→D

E → AH becomes E→A and E→H

* Remove redundant dependencies:

A → C

AC → D

E → A

E → D

E → H

A → D

* Drop extraneous attributes:

AC→D can be removed because we have A→D so C is redundant:

A → C

E → A

E → D

E → H

A → D

* Drop redundant FDs: Try removing some dependencies in F and still have a set of dependencies equivalent to F.

E→D can be deduced from E→A and A→D so we can remove E→D.

The set of FDs becomes:

A→C

E→A

E→H

A→D

which is the canonical cover of F.

Finding the keys of R:

Observations:

* B and E do not appear in the right hand side of any FDs, so they have to appear in all keys of R.
* BE+: BE → AEB (because E → A) → AEBC (because A → C) → AEBCD (because A → D) → AEBCDH (because E → H). So BE is a key of R.

In this case, we do not need to consider any other combination, because any other combination containing BE (e.g., AEB) is a super key and not minimal.

**Part 2:**

1. Consider the following set of functional dependencies F on relation R (A, B, C, D, E, H):

A → C

AC → D

E → AD

E → H

A → CD

E → AH

The key for R is *EB* and the following set of functional dependencies constitutes the canonical cover:

A → C, E → A, E → H, A → D

1. Using Synthesis Method, construct a set of 3NF relations.

We are starting from a canonical cover. 🡪 We can skip steps 1, 2 and 3.

Step 4: Group FDs with same determinant:

A → CD

E → AH

Step 5: Construct a relation for each group:

R1(A,C,D)

R2(E,A,H)

Step 6: If none of the relations contain the key for the original relation, add a relation with the key.

R3(E,B)

R1, R2 and R3 are in 3NF and in BCNF.

1. Using Universal Method, decompose R into a set of BCNF relations.

Using A → C to decompose R, we get:

R1(A,B,D,E,H) in 1NF

R2(A,C) is already in BCNF form

Using A → D to decompose R1, we get:

R11(A,B,E,H) in 1NF

R12(A,D) is already in BCNF form

Using E → A to decompose R11, we get:

R111(B,E,H) in 1NF

R112(E,A) is already in BCNF form

Using E → H to decompose R111, we get:

R1111(B,E) in BCNF form

R1112(E,H) is already in BCNF form

Group the relations with the same key:

R1(A,C,D)

R2(E,A,H)

R3(E,B)

R1, R2 and R3 are in BCNF form.

1. Consider the following set of functional dependencies F on relation R (A, B, C, D, E):

A → BC

A → D

B → C

C → D

DE → C

BC → D

The key for R is *AE* and the following set of functional dependencies constitutes the canonical cover:

A → B, B → C, C → D, DE → C

1. Using Synthesis Method, construct a set of 3NF relations.

We are starting from a canonical cover. 🡪 We can skip steps 1, 2 and 3.

We can skip Step 4: no FDs with same determinant

Step 5: Construct a relation for each group:

R1(A,B)

R2(B,C)

R3(C,D)

R4(D, E,C)

Step 6: If none of the relations contain the key for the original relation, add a relation with the key.

R5(A,E)

R1, R2, R3, R4, and R5 are in 3NF and in BCNF.

1. Using Universal Method, decompose R into a set of BCNF relations.

Using B → C to decompose R, we get:

R1(A,B,C,D,E) in 1NF

R2(B,C) is already in BCNF form

Using C → D to decompose R1, we get:

R11(A,B,C,D,E) in 1NF

R12(C,D) is already in BCNF form

Using A → B to decompose R11, we get:

R111(A,C,D,E) in 1NF

R112(A,B) is already in BCNF form

Using DE → C to decompose R111, we get:

R1111(A,E) in BCNF form

R1112(D,E,C) is already in BCNF form

Group the relations with the same key:

R1(B,C)

R2(C,D)

R3(A,B)

R4(D,E,C)

R5(A,E)

R1, R2, R3, R4, and R5 are in 3NF and in BCNF.

**Part 3:** Assume that R is decomposed into:

R1 (A, B), F1 = {A → B}

R2 (B, C), F2 = {B → C}

R3 (C, D, E), F3 = {C → D, DE → C)

Is this decomposition a lossless-join decomposition? Use the table method.

Checking for lossless-join:

Initially the Table looks like this:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1(A,B) | K | K | U | U | U |
| R2(B,C) | U | K | K | U | U |
| R3(C,D,E) | U | U | K | K | K |

Using B → C: we can replace U13 by K

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1(A,B) | K | K | **K** | U | U |
| R2(B,C) | U | K | K | U | U |
| R3(C,D,E) | U | U | K | K | K |

Using C → D: we can replace U14 and U24 by K

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1(A,B) | K | K | **K** | **K** | U |
| R2(B,C) | U | K | K | **K** | U |
| R3(C,D,E) | U | U | K | K | K |

We cannot proceed and there is no row of all known values 🡪 the decomposition is lossy.