CSC343 Worksheet 13 Solution

July 5, 2020

1. a)

A	B	C	D	E
a	b	c	d_1	e_1
a_1	b	c	d	e_2
a	b_1	c	d_2	e

Step 1 ($B \to E$):

A	B	C	D	E
a	b	c	d_1	e_1
a_1	b	c	d	e_1
a	b_1	c	d_2	e

Step 2 ($CE \to A$):

A	B	C	D	E
a	b	c	d_1	e_1
a	b	c	d	e_1
a	b_1	c	d_2	e

So in this case, an example of an instance of R that is not lossless is:

Title	Studio Name	President	Year	President Address
Toy Story	Pixar	Steve Jobs	2000	123 ABC Street
Star Wars	Fox	Lachlan Murdoch	1977	Hollywood
Return of the Jedi	Fox	Lachlan Murdoch	1983	Hollywood

- $S_1 = \{A, B, C\}$

Title	Studio Name	President
Toy Story	Pixar	Steve Jobs
Star Wars	Fox	Lachlan Murdoch
Return of the Jedi	Fox	Lachlan Murdoch

- $S_2 = \{C, D, E\}$

President	Year	President Address
Steve Jobs	2000	123 ABC Street
Lachlan Murdoch	1977	Hollywood
Lachlan Murdoch	1983	Hollywood

- $S_3 = \{C, E, A\}$

Title	President	President Address
Toy Story	Steve Jobs	123 ABC Street
Star Wars	Lachlan Murdoch	Hollywood
Return of the Jedi	Lachlan Murdoch	Hollywood

- $S_1 \bowtie S_2$

Title	Studio Name	President	Year	President Address
Toy Story	Pixar	Steve Jobs	2000	123 ABC Street
Star Wars	Fox	Lachlan Murdoch	1977	Hollywood
Star Wars	Fox	Lachlan Murdoch	1983	Hollywood
Return of the Jedi	Fox	Lachlan Murdoch	1977	Hollywood
Return of the Jedi	Fox	Lachlan Murdoch	1983	Hollywood

- $S_1 \bowtie S_2 \bowtie S_3$

Title	Studio Name	President	Year	President Address
Toy Story	Pixar	Steve Jobs	2000	123 ABC Street
Star Wars	Fox	Lachlan Murdoch	1977	Hollywood
Star Wars	Fox	Lachlan Murdoch	1983	Hollywood
Return of the Jedi	Fox	Lachlan Murdoch	1977	Hollywood
Return of the Jedi	Fox	Lachlan Murdoch	1983	Hollywood

Notes:

- Decomposition: The good bad and ugly
  1) **Elimination of Anomalies** by decomposition as in Section 3
  2) **Recoverability of Information** Can we recover the original relation from the tuples in its decomposition?
  3) **Preservation of Dependences (lossless join):** Can we be sure that after reconstructing the original relation from the decompositions, the original FD's satisfy?

  BCNF: $\rightarrow$ satisfies 1) and 2) Not good. NONO
- The Chase Test for Lossless Join
  - Tests whether the decomposition is lossless

  Input:
  - A relation R

- A decomposition of R
- A set of functional dependencies

Output:

- Whether the decomposition is loseless or not
- $\Pi_{S_1}(R) \bowtie \Pi_{S_2}(R) \bowtie \cdots \Pi_{S_i}(R) = R$

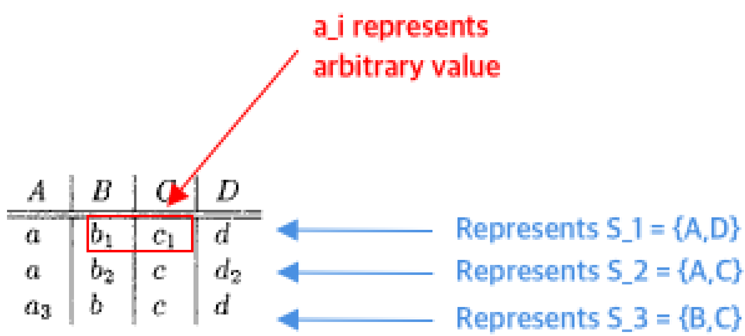
Three things to remember:

1. The natural join is associate and commutative
2. Any tuple t in R is surely in $\pi_{S_1}(R) \bowtie \pi_{S_2}(R) \bowtie \cdots \bowtie \pi_{S_k}(R)$.
3. We have to check to see any tuple in the $\pi_{S_1}(R) \bowtie \pi_{S_2}(R) \bowtie \cdots \bowtie \pi_{S_k}(R)$.

Example:

$S_1 = \{A, D\}$, $S_2 = \{B, C\}$, $S_3 = \{A, C\}$

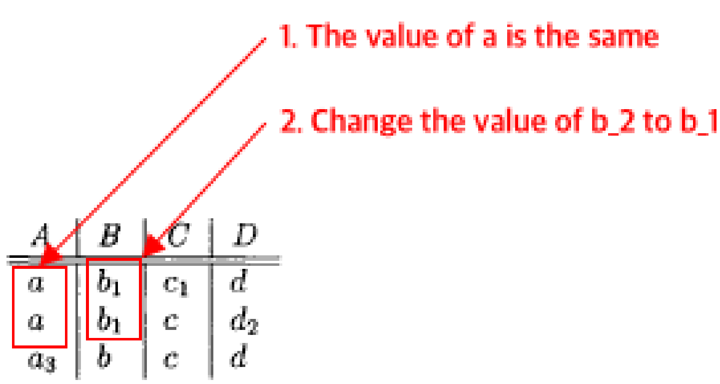
$A \to B$, $B \to C$, $CD \to A$



A	B	C	D
a	b_1	c_1	d
a	b_2	c	d_2
a_3	b	c	d

Step 1: $A \to B$

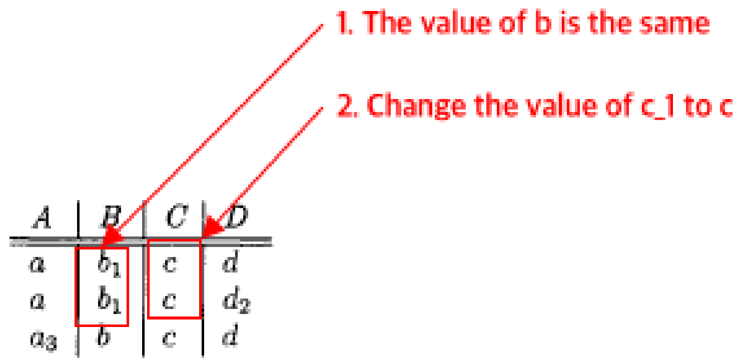
Set the value b with the same value of a to be the same. (e.g. $b_2 \to b_1$)



A	B	C	D
a	b_1	c_1	d
a	b_1	c	d_2
a_3	b	c	d

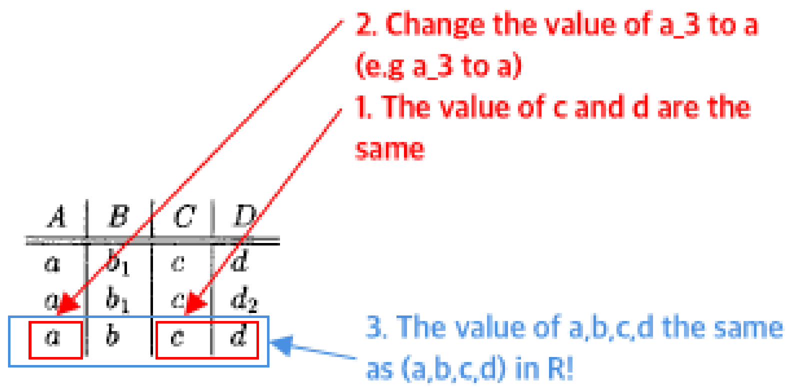
Step 2: $B \to C$

Set the value c with the same value of b to be the same. (e.g. $b_2 \to b_1$)



Step 3: $CD \to A$

Set the value a with the same value of c and d to be the same. (e.g. $a_3 \to a$)



So, we can conclude the join is lossless.

b)

A	B	C	D	E
a	b	c	d_1	e_1
a_1	b	c	d	e_2
a	b_1	c	d_2	e

Step 1 ($AC \to E$**):**

A	B	C	D	E
a	b	c	d_1	e
a_1	b	c	d	e_2
a	b_1	c	d_2	e

Step 2 ($BC \to D$**):**

A	B	C	D	E
a	b	c	d	e
a_1	b	c	d	e_2
a	b_1	c	d_2	e

a, b, c, d, e exists. So by the Chast test, the decomposition of $R(A, B, C, D, E) : AC \rightarrow E, BC \rightarrow D$ into $\{A, B, C\}$, $\{B, C, D\}$, $\{A, C, E\}$ is lossless.

c)

A	B	C	D	E
a	b	c	d_1	e_1
a_1	b	c	d	e_2
a	b_1	c	d_2	e

Step 1 ($A \rightarrow D$):

A	B	C	D	E
a	b	c	d_1	e_1
a_1	b	c	d	e_2
a	b_1	c	d_1	e

Step 2 ($D \rightarrow E$):

A	B	C	D	E
a	b	c	d_1	e
a_1	b	c	d	e_2
a	b_1	c	d_1	e

Step 3 ($B \rightarrow D$):

A	B	C	D	E
a	b	c	d	e
a_1	b	c	d	e_2
a	b_1	c	d	e

a, b, c, d, e exists. So by the Chast test, the decomposition of $R(A, B, C, D, E) : A \rightarrow D, D \rightarrow E, B \rightarrow D$ into $\{A, B, C\}$, $\{B, C, D\}$, $\{A, C, E\}$ is lossless.

d)

A	B	C	D	E
a	b	c	d_1	e_1
a_1	b	c	d	e_2
a	b_1	c	d_2	e

Step 1 ($A \rightarrow D$):

A	B	C	D	E
a	b	c	d_1	e_1
a_1	b	c	d	e_2
a	b_1	c	d_1	e

Step 2 ($CD \to E$):

A	B	C	D	E
a	b	c	d_1	e
a_1	b	c	d	e_2
a	b_1	c	d_1	e

Step 3 ($E \to D$):

A	B	C	D	E
a	b	c	d_1	e
a_1	b	c	d	e_2
a	b_1	c	d_1	e

So in this case, the relation is not lossless.

An example of an instance of R that is not lossless is:

Phone ID	Grade	Student Name	Phone #	Physical Address
1	89	John Doe	111-222-3333	123 ABC Street
2	89	John Doe	222-222-3333	123 ABC Street
1	62	Josh Doe	111-222-3333	123 ABC Street
3	94	Frank McKay	444-555-6666	234 ABC Street

- $S_1 = \{A, B, C\}$

Phone ID	Grade	Student Name
1	89	John Doe
2	89	John Doe
1	62	Josh Doe
3	94	Frank McKay

- $S_2 = \{C, D, E\}$

Student Name	Phone #	Physical Address
John Doe	111-222-3333	123 ABC Street
John Doe	222-222-3333	123 ABC Street
Josh Doe	111-222-3333	123 ABC Street
Frank McKay	444-555-6666	234 ABC Street

- $S_3 = \{A, C, E\}$

Phone ID	Student Name	Physical Address
1	John Doe	123 ABC Street
2	John Doe	123 ABC Street
1	Josh Doe	123 ABC Street
3	Frank McKay	234 ABC Street

- $S_1 \bowtie S_2$

Phone ID	Grade	Student Name	Phone #	Physical Address
1	89	John Doe	111-222-3333	123 ABC Street
2	89	John Doe	111-222-3333	123 ABC Street
1	89	John Doe	222-222-3333	123 ABC Street
2	89	John Doe	222-222-3333	123 ABC Street
1	62	Josh Doe	111-222-3333	123 ABC Street
3	94	Frank McKay	444-555-6666	234 ABC Street

- $S_1 \bowtie S_2 \bowtie S_3$

Phone ID	Grade	Student Name	Phone #	Physical Address
1	89	John Doe	111-222-3333	123 ABC Street
2	89	John Doe	111-222-3333	123 ABC Street
1	89	John Doe	222-222-3333	123 ABC Street
2	89	John Doe	222-222-3333	123 ABC Street
1	62	Josh Doe	111-222-3333	123 ABC Street
3	94	Frank McKay	444-555-6666	234 ABC Street

2. The sets of FDs in 1.b) and 1.c) are the ones where the dependencies are preserved from decomposition.

3. a) i) Find 3NF Violations

- $\{A, B\}^+ = \{A, B\}$
  - Violates 3NF
  - Doesn't have C required for $AB \to C$
  - Second and third don't imply first
- $\{C\}^+ = \{C\}$
  - Violates 3NF
  - Doesn't have D required for $C \to D$
  - First and third don't imply second
- $\{D\}^+ = \{D\}$
  - Violates 3NF
  - Doesn't have A required for $D \to A$
  - First and second don't imply third

ii) Decompose relations, as necessary, into collections as a part of 3NF

1. Find a minimal basis for F, say G

A possible FD to simplify is $AB \to C$. But since $\{B\}^+ = \{B\}$, and $\{A\}^+ = \{A\}$, and both has C missing, the FD cannot be simplified further.

so the minimal basis for F is

$AB \to C$, $C \to D$, $D \to A$

2. For each functional dependency $X \to A$ in G, use XA as the schema of one of the relations in the decomposition

   $S_1(A, B, C)$, $S_2(C, D)$, $S_3(D, A)$

3. If none of the relation schemas from Step 2 is a superkey for R, add another relation whose schema is a key for R. And drop redundant relations.

   The keys for this relation are $\{A, B, C\}, \{B, C, D\}, \{D, A, B\}$

   The attributes in $S_1(A, B, C)$ is a key. So, this step can be skipped.

Notes:

- 3NF
  - Definition
    - A elation R is in 3NF if

      For each nontrivial FD, the left side is a superkey (BCNF), or the right side consists of prime attributes only.
  - Our expectation after decomposing are:
  1. Elimination of Anomalies
  2. Recoverability of Information (Recovering original relation after decomposition)
  3. Preservation of Information (Recovering original tuples after decomposition)

  Key: 3NF guarentees 2) and 3) but not 1)
- Violations
  - $X \to A$ violates 3NF if and only if X is not a superkey and also A is not prime.

  Prime Attributes are attributes that are part of a key.

- Synthesis algorithm for 3NF Schemas
  1. Check if the FD's are minimal
     - To verify, we should show that we cannot eliminate any of FD's. That is, using algorithm 3.7, no two of the FD's imply the third
  2. Find a minimal basis for F, say G
  3. For each functional dependency $X \to A$ in G, use XA as the schema of one of the relations in the decomposition
  4. If none of the relation schemas from Step 3 is a superky for R, add another relation whose schema is a key for R. And drop redundant relations.

  Example:

  $R(A, B, C, D, E) : AB \to C, C \to B, A \to D$

1. Check if the FD's are minimal
   - To verify, we should show that we cannot eliminate any of FD's. That is, using algorithm 3.7, no two of the FD's imply the third

   1) $\{A, B\}^+$

      Using the FDs $C \to B$, $A \to D$, we have $\{A, B\}^+ = \{A, B, D\}$.

      Since C is required for $AB \to C$, we can conclude second and third doesn't imply the first

   2) $\{C\}^+$

      Using the FDs $AB \to C$, $A \to D$, we have $\{C\}^+ = \{C\}$.

      Since B is required for $C \to B$ but not in $\{C\}^+ = \{C\}$, we can conclude first and third doesn't imply the second

   3) $\{A\}^+$

      Using the FDs $AB \to C$, $C \to D$, we have $\{A\}^+ = \{A\}$.

      Since D is required for $C \to D$ but not in $\{A\}^+ = \{A\}$, we can conclude first and second doesn't imply the third

2. Find a minimal basis for F, say G

   $AB \to C$, $C \to B$ and $A \to D$

3. For each functional dependency $X \to A$ in G, use XA as the schema of one of the relations in the decomposition

   $S_1(A, B, C)$, $S_2(C, B)$, $S_3(A, D)$

4. If none of the relation schemas from Step 3 is a superky for R, add another relation whose schema is a key for R. And drop redundant relations.

   Take all combinations of attributes A, B, C, D, E. We have $\{A, B, C\}$ and $\{A, B, E\}$ as keys.

   Thus, adding one, the extra relation we have is $S_4(A, B, E)$.

   And since B, C in $S_2(B, C)$ is also in $S_1(A, B, C)$, S_2 needs to be dropped.

   So, we have $S_1(A, B, C)$, $S_3(A, D)$, $S_4(A, B, E)$

b) i) Find 3NF Violations

- $\{B\}^+ = \{B, C, D\}$
  - Violates 3NF

  - Doesn't have C required for $B \to C$
  - Doesn't have C required for $B \to D$
  - Second and third don't imply first

ii) Decompose relations, as necessary, into collections as a part of 3NF

1. Find a minimal basis for F, say G

   $B \to D$ and $B \to C$

2. For each functional dependency $X \to A$ in G, use XA as the schema of one of the relations in the decomposition

   $S_1(B, D)$, $S_2(B, C)$

3. If none of the relation schemas from Step 2 is a superkey for R, add another relation whose schema is a key for R. And drop redundant relations.

   The keys for this relation are $\{A, B\}$

   So, the new sets of relation are $S_1(B, D)$, $S_2(B, C)$, $S_3(A, B)$

c) i) Find 3NF Violations

- $\{A, B\}^+ = \{A, B\}$
  - Violates 3NF
  - Doesn't have C required for $AB \to C$
  - Second, third and fourth don't imply first
- $\{B, C\}^+ = \{B, C\}$
  - Violates 3NF
  - Doesn't have D required for $BC \to D$
  - first, third and fourth don't imply second
- $\{C, D\}^+ = \{C, D\}$
  - Violates 3NF
  - Doesn't have A required for $CD \to A$
  - first, second and fourth don't imply third
- $\{A, D\}^+ = \{A, D\}$
  - Violates 3NF
  - Doesn't have B required for $AD \to B$
  - first, second and third don't imply fourth

ii) Decompose relations, as necessary, into collections as a part of 3NF

1. Find a minimal basis for F, say G

   $AB \rightarrow C$, $BC \rightarrow D$, $CD \rightarrow A$ and $AD \rightarrow B$

2. For each functional dependency $X \rightarrow A$ in G, use XA as the schema of one of the relations in the decomposition

   $S_1(A, B, C)$, $S_2(B, C, D)$, $S_3(C, D, A)$, $S_4(A, D, B)$

3. If none of the relation schemas from Step 2 is a superkey for R, add another relation whose schema is a key for R. And drop redundant relations.

   The keys for this relation are $\{A, B, C\}$, $\{B, C, D\}$, $\{C, D, A\}$,$\{A, D, B\}$

   The attributes in $S_1(A, B, C)$ is a key. So, this step can be skipped.