(15 pts) Exercise 11.16 on Page 408 in Elmasri and Navathe.

Show that, if the matrix $S$ resulting from Algorithm 11.1 does not have a row that is all $a$ symbols, projecting $S$ on the decomposition and joining it back will always produce at least one spurious tuple.

We will prove this using direct technique.

Given a relation $R$ that is decomposed into a number of relations $R_1, R_2, ..., R_m$, Algorithm 11.1 begins the matrix $S$ that we consider to be some relation state $r$ of $R$. Row $i$ in $S$ represents a tuple $t_i$ (corresponding to relation $R_i$) that has $a$ symbols in the columns that correspond to the attributes of $R_i$ and $b$ symbols in the remaining columns. We all know that during the loop of step 4, the algorithm transforms the row of matrix so that they represent tuples that satisfy all the functional dependencies in $F$. Since we never change an $a$ symbol into a $b$ symbol during the application of the algorithm, then projecting $S$ on each $R_i$ at the end of applying the algorithm will produce one row consisting of all $a$ symbols in each $R_i$.

On the other hand, for every relation $R_i$, there is at least some attribute of $R_i$ that also appears in to other $R_j$ ($j<>i$) so that they can naturally join with each other. In this case, if the result from algorithm 11.1 produces no row of all $a$'s, natural join of the projection of this table state onto individual tables will still produce an additional row of all $a$'s. This is because there is one row consisting of all $a$ symbols in each $R_i$. That all $a$ row in the joined table is a spurious tuple (since it did not exist in $S$ but will exist after projecting and joining over the $R_i$'s).

To clarify the proof, let’s consider the example from the textbook (Figure 11.1a).

The matrix $S$ resulting from Algorithm 11.1 is as below:

<table>
<thead>
<tr>
<th>Ssn</th>
<th>Ename</th>
<th>Pnumber</th>
<th>Pname</th>
<th>Plocation</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>b11</td>
<td>a2</td>
<td>b13</td>
<td>b14</td>
<td>a5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>a1</td>
<td>b22</td>
<td>a3</td>
<td>a4</td>
<td>a5</td>
</tr>
</tbody>
</table>

This matrix $S$ does not have a row that is all $a$ symbols.

Projecting $S$ on the decomposition returns:
### R1

<table>
<thead>
<tr>
<th>Ename</th>
<th>Plocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>a2</td>
<td>a5</td>
</tr>
<tr>
<td>b22</td>
<td>a5</td>
</tr>
</tbody>
</table>

### R2

<table>
<thead>
<tr>
<th>Ssn</th>
<th>Pnumber</th>
<th>Pname</th>
<th>Pname</th>
<th>Plocation</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>b11</td>
<td>b13</td>
<td>b14</td>
<td>a5</td>
<td>b16</td>
<td></td>
</tr>
<tr>
<td>a1</td>
<td>a3</td>
<td>a4</td>
<td>a5</td>
<td>a6</td>
<td></td>
</tr>
</tbody>
</table>

Joining R1’s back returns:

<table>
<thead>
<tr>
<th>Ssn</th>
<th>Ename</th>
<th>Pnumber</th>
<th>Pname</th>
<th>Pname</th>
<th>Plocation</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>b11</td>
<td>a2</td>
<td>b13</td>
<td>b14</td>
<td>a5</td>
<td>b16</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>a1</td>
<td>a2</td>
<td>a3</td>
<td>a4</td>
<td>a5</td>
<td>a6</td>
</tr>
<tr>
<td>*</td>
<td>b11</td>
<td>b22</td>
<td>b13</td>
<td>b14</td>
<td>a5</td>
<td>b16</td>
</tr>
<tr>
<td>a1</td>
<td>b22</td>
<td>a3</td>
<td>a4</td>
<td>a5</td>
<td>a6</td>
<td></td>
</tr>
</tbody>
</table>

The spurious tuples are **bold** and marked with by asterisks (*). One of them is all *a* symbols. The point is this one does not exist in the original R.
(15 pts) Exercise 11.17 on Page 408 in Elmasri and Navathe.

Show that the relation schemas produced by Algorithm 11.3 are in BCNF.

We will prove this using Contradiction technique.

Suppose the decomposition $D = \{R_1, R_2, ..., R_m\}$ produced by Algorithm 11.3 are NOT in BCNF. It implies that some $R_i$ contains some functional dependency $X \rightarrow Y$ that violates BCNF. However, according to the algorithm, that $R_i$ should be decomposed into two relation schemas $(R_i - Y)$ and $(X \cup Y)$. In other words, $R_i$ does not exist, which is a contradiction.
(30 pts) Exercise 11.26 on Page 409 in Elmasri and Navathe.

Apply the decomposition algorithm (Algorithm 11.3) to the relation R and the set of dependencies F in Exercise 10.26. Repeat for the dependencies G in Exercise 10.27.

Given from Exercise 10.26:

Universal relation
- \( R = \{A, B, C, D, E, F, G, H, I, J\} \)

Set of functional dependencies F:
- \( \{A, B\} \rightarrow \{C\} \)
- \( \{A\} \rightarrow \{D, E\} \)
- \( \{B\} \rightarrow \{F\} \)
- \( \{F\} \rightarrow \{G, H\} \)
- \( \{D\} \rightarrow \{I, J\} \)

**Step 1:** Set \( D = \{R\} = \{Q1 \{A, B, C, D, E, F, G, H, I, J\}\} \)

**Step 2:** Q1 is not in BCNF because for the FD \( \{A\} \rightarrow \{D, E\} \), \( A^+ = \{A, D, E, I, J\} \) does not include all the attributes in Q1. In other words, A is not a super key in Q1. Therefore, Q1 should be decomposed into 2 relations Q1 \{A, B, C, F, G, H, I, J\} and Q2 \{A, D, E\}.

\( D = \{Q1 \{A, B, C, F, G, H, I, J\}, Q2 \{A, D, E\}\} \)

**Repeat Step 2:** Q1 is not in BCNF because for the FD \( \{B\} \rightarrow \{F\} \), \( B^+ = \{B, F, G, H\} \) does not include all the attributes in Q1. In other words, B is not a super key in Q1. Therefore, Q1 should decomposed into 2 relations Q1 \{A, B, C, G, H, I, J\} and Q3 \{B, F\}.

\( D = \{Q1 \{A, B, C, G, H, I, J\}, Q2 \{A, D, E\}, Q3 \{B, F\}\} \)

Relations in D now are in BCNF.

Given from Exercise 10.27:

Universal relation
- \( R = \{A, B, C, D, E, F, G, H, I, J\} \)

Set of functional dependencies G:
• \{A, B\} \rightarrow \{C\}
• \{B, D\} \rightarrow \{E, F\}
• \{A, D\} \rightarrow \{G, H\}
• \{A\} \rightarrow \{I\}
• \{H\} \rightarrow \{J\}

**Step 1:** Set \(D = \{R\} = \{Q1\{A, B, C, D, E, F, G, H, I, J\}\}\)

**Step 2:** \(Q1\) is not in BCNF because for the FD \(\{A, B\} \rightarrow \{C\}\), \(\{A, B\}^+ = \{A, B, C, I\}\) does not include all the attributes in \(Q1\). In other words, \(\{A, B\}\) is not a super key in \(Q1\). Therefore, \(Q1\) should be decomposed into 2 relations \(Q1\{A, B, D, E, F, G, H, I, J\}\) and \(Q2\{A, B, C\}\).
\(D = \{Q1\{A, B, D, E, F, G, H, I, J\}, Q2\{A, B, C\}\}\)

**Repeat Step 2:** \(Q1\) is not in BCNF because for the FD \(\{B, D\} \rightarrow \{E, F\}\), \(\{B, D\}^+ = \{B, D, E, F\}\) does not include all the attributes in \(Q1\). In other words, \(\{B, D\}\) is not a super key in \(Q1\). Therefore, \(Q1\) should be decomposed into 2 relations \(Q1\{A, B, D, G, H, I, J\}\) and \(Q3\{B, D, E, F\}\).
\(D = \{Q1\{A, B, D, G, H, I, J\}, Q2\{A, B, C\}, Q3\{B, D, E, F\}\}\)

**Repeat Step 2:** \(Q1\) is not in BCNF because for the FD \(\{A, D\} \rightarrow \{G, H\}\), \(\{A, D\}^+ = \{A, D, G, H, I, J\}\) does not include all the attributes in \(Q1\). In other words, \(\{A, D\}\) is not a super key in \(Q1\). Therefore, \(Q1\) should be decomposed into 2 relations \(Q1\{A, B, D, I, J\}\) and \(Q4\{A, D, G, H\}\).
\(D = \{Q1\{A, B, D, I, J\}, Q2\{A, B, C\}, Q3\{B, D, E, F\}, Q4\{A, D, G, H\}\}\)

**Repeat Step 2:** \(Q1\) is not in BCNF because for the FD \(\{A\} \rightarrow \{I\}\), \(\{A\}^+ = \{A, I\}\) does not include all the attributes in \(Q1\). In other words, \(\{A\}\) is not a super key in \(Q1\). Therefore, \(Q1\) should be decomposed into 2 relations \(Q1\{A, B, D, J\}\) and \(Q5\{A, I\}\).
\(D = \{Q1\{A, B, D, J\}, Q2\{A, B, C\}, Q3\{B, D, E, F\}, Q4\{A, D, G, H\}, Q5\{A, I\}\}\)

**Relations in D now are in BCNF.**
(15 pts) Exercise 11.29 (c) on Page 409 in Elmasri and Navathe.

Given from Exercise 10.26:

Universal relation

- $R = \{A, B, C, D, E, F, G, H, I, J\}$

Set of functional dependencies $F$:

- $\{A, B\} \rightarrow \{C\}$
- $\{A\} \rightarrow \{D, E\}$
- $\{B\} \rightarrow \{F\}$
- $\{F\} \rightarrow \{G, H\}$
- $\{D\} \rightarrow \{I, J\}$

Given from 11.29 (c): $D_3 = \{R_1, R_2, R_3, R_4, R_5\}$

- $R_1 = \{A, B, C, D\}$
- $R_2 = \{D, E\}$
- $R_3 = \{B, F\}$
- $R_4 = \{F, G, H\}$
- $R_5 = \{D, I, J\}$

(1) Check the dependency preservation property

By definition of dependency preservation property, we will check whether:

\[ ((\pi_{R_1}(F)) \cup (\pi_{R_2}(F)) \cup (\pi_{R_3}(F)) \cup (\pi_{R_4}(F)) \cup (\pi_{R_5}(F)))^+ = F^+ \]

We calculate:

- $\pi_{R_1}(F) = \{\{A, B\} \rightarrow \{C\}, \{A\} \rightarrow \{D\}\}$
- $\pi_{R_2}(F) = \emptyset$
- $\pi_{R_3}(F) = \{\{B\} \rightarrow \{F\}\}$
- $\pi_{R_4}(F) = \{\{F\} \rightarrow \{G, H\}\}$
- $\pi_{R_5}(F) = \{\{D\} \rightarrow \{I, J\}\}$

And $(\pi_{R_1}(F)) \cup (\pi_{R_2}(F)) \cup (\pi_{R_3}(F)) \cup (\pi_{R_4}(F)) \cup (\pi_{R_5}(F)) = \{\{A, B\} \rightarrow \{C\}, \{A\} \rightarrow \{D\}, \{B\} \rightarrow \{F\}, \{F\} \rightarrow \{G, H\}, \{D\} \rightarrow \{I, J\}\}$
Note that \{A\} \rightarrow \{E\} is missing in \((\pi_{R1}(F)) \cup (\pi_{R2}(F)) \cup (\pi_{R3}(F)) \cup (\pi_{R4}(F)) \cup (\pi_{R5}(F))\) and we cannot derive it from other dependencies. Therefore, \(((\pi_{R1}(F)) \cup (\pi_{R2}(F)) \cup (\pi_{R3}(F)) \cup (\pi_{R4}(F)) \cup (\pi_{R5}(F)))^+ \neq F^+

**In conclusion, decomposition D3 does NOT have the dependency preservation property.**
(3) Determine which normal form each relation in the decomposition is in.

* Consider R1 = \{A, B, C, D\}

We calculated \(\pi_{R1}(F) = \\{\{A, B\} \rightarrow \{C\}, \{A\} \rightarrow \{D\}\}\) from the answer (1).

By applying Algorithm 11.4(a), we can find the key for R1: \(K1 = \{A, B\}\)

This relation R1 is not 2NF because there is a partial dependency \(\{A\} \rightarrow \{D\}\).

* Consider R2 = \{D, E\}

We calculated \(\pi_{R2}(F) = \emptyset\) from the answer (1).

By applying Algorithm 11.4(a), we can find the key for R2: \(K2 = \{D, E\}\)

This relation R2 is BCNF because it has no functional dependency.

* Consider R3 = \{B, F\}

We calculated \(\pi_{R3}(F) = \\{\{B\} \rightarrow \{F\}\}\) from the answer (1).

By applying Algorithm 11.4(a), we can find the key for R3: \(K3 = \{B\}\)

This relation R3 is 2NF because non-prime attribute F is fully dependent on the key B.

This relation R3 is 3NF because non-prime attribute F is nontransitively dependent on the key B.

This relation R3 is BCNF because in \(\{B\} \rightarrow \{F\}\), B is the superkey of R3.

* Consider R4 = \{F, G, H\}

We calculated \(\pi_{R4}(F) = \\{\{F\} \rightarrow \{G, H\}\}\) from the answer (1).

By applying Algorithm 11.4(a), we can find the key for R4: \(K4 = \{F\}\)

This relation R4 is 2NF because non-prime attributes G and H are fully dependent on the key F.

This relation R4 is 3NF because non-prime attributes G and H are nontransitively dependent on the key F.

This relation R4 is BCNF because in \(\{F\} \rightarrow \{G, H\}\), F is the superkey of R4.

* Consider R5 = \{D, I, J\}

We calculated \(\pi_{R5}(F) = \\{\{D\} \rightarrow \{I, J\}\}\) from the answer (1).

By applying Algorithm 11.4(a), we can find the key for R5: \(K5 = \{D\}\)

This relation R5 is 2NF because non-prime attributes I and J are fully dependent on the key D.

This relation R5 is 3NF because non-prime attributes I and J are nontransitively dependent on the key D.

This relation R5 is BCNF because in \(\{D\} \rightarrow \{I, J\}\), D is the superkey of R5.
(25 pts) Exercise 11.30 on Page 409 in Elmasri and Navathe.

Given:

REFRIG(Model#, Year, Price, Manuf_plant, Color)

which is abbreviated REFRIG(M, Y, P, MP, C)

Set of functional dependencies F:

- \( M \rightarrow MP \)
- \( \{M, Y\} \rightarrow P \)
- \( MP \rightarrow C \)

a. Evaluate the candidate keys:

* Consider \( \{M\} \)

\( \{M\} \) is not a key because M cannot determine P.

* Consider \( \{M, Y\} \)

- \( \{M, Y\} \rightarrow P \) (given)
- \( \{M, Y\} \rightarrow MP \) (using IR1 and \( \{M, Y\} \) is the superset of M)
- \( \{M, Y\} \rightarrow C \) (using IR3 on \( \{M, Y\} \rightarrow MP \) and \( MP \rightarrow C \))

\( \{M, Y\} \) determines all the attributes of REFRIG. Thus, \( \{M, Y\} \) is a key.

* Consider \( \{M, C\} \)

\( \{M, C\} \) is not a key because M and C cannot determine Y and P.

b. State whether REFRIG is in 3NF and in BCNF

REFRIG is not 2NF because there is \( M \rightarrow MP \) in which M is a part of the key \( \{M, Y\} \). Thus, REFRIG is not 3NF either.

REFRIG is not BCNF either because in the functional dependency \( M \rightarrow MP \), M is not the superkey.

c. Check the lossless property

Given the decomposition of REFRIG into D:

- R1 (M, Y, P)
- R2 (M, MP, C)

Using the test for Binary Decomposition, we calculate:

\( (R1 \cap R2) = \{M\} \)
\( (R1 - R2) = \{Y, P\} \)
(R2 – R1) = {MP, C}

Then, we have:

{M} → {MP} (given)

{M} → {C} (using IR3 on {M} → {MP} and {MP} → {C})

{M} → {MP, C}

So, (R1 ∩ R2) → (R2 – R1) or the decomposition D is lossless.