COMP3311 Week 9 Tutorial Sample Solutions

Kenneth Li

Closures

The closure of a set of attributes X, X^+ are all the attributes X can determine given a set of FDs.

Question 2

Consider the relation $R(A, B, C, D, E, F, G)$ and the set of functional dependencies

$$F = \{A \to B, BC \to F, BD \to EG, AD \to C, D \to F, BEG \to FA\},$$

compute the following:

(a) A^+

Initially $A^+ = A$. Keep going through functional dependencies, and add attributes on the RHS to the closure if all the LHS is in A^+.

$$A \to B \text{ (add B)}.$$

There are no more attributes we can add to the closure. Therefore, $A^+ = AB$.

(b) $ACEG^+$
Initially $ACEG^+ = ACEG$. Keep going through functional dependencies, and add attributes on the RHS to the closure if all the LHS is in A^+.

$$A \to B \text{ (add B)}$$
$$BC \to F \text{ (add F)}$$

We can use $BEG \to FA$, but we don't get any new attributes to the closure. We can't add D, so therefore, $ACEG^+ = ABCEFG$.

(c) BD^+
Initially $BD^+ = BD$. Keep going through functional dependencies, and add attributes on the RHS to the closure if all the LHS is in $ACEG^+$.

$$BD \to EG \text{ (add E and G to closure)}$$
$$BEG \to FA \text{ (add F and A)}$$
$$AD \to C \text{ (add C)}$$

We have all the attributes of R so we can't add anymore. So $BD^+ = ABCDEFG$, and so, BD is a superkey (and later a candidate key), as it can determine all the attributes.

Candidate Keys

Smallest set(s) of attributes that can determine all the attributes in R, so their closure is all of R.

Remember that attributes that never appear in the RHS of any FDs must be in the candidate keys, as they can never be determined otherwise.

Question 4a

Consider a relation $R(A, B, C, D)$. For each of the following sets of functional dependencies, assuming that those are the only dependencies that hold for R, do the following:

(a) List all of the candidate key(s) for R. (i) $C \to D, C \to A, B \to C$

For this set, you can see that B never appears in the RHS of the FDs, so B must be part of the candidate key(s).
Try smallest set B^+.

$$\begin{aligned}B &\to C \text{ (add C)}\\ C &\to D \text{ (add D)}\\ C &\to A \text{ (add A)}\end{aligned}$$

So $B^+ = ABCD = R$, and so B is the candidate key.

(ii) $B \to C, D \to A$

Note that B and D don't appear on any RHS. Try BD^+.

$$\begin{aligned}B &\to C \text{ (add C)}\\ D &\to A \text{ (add A)}\end{aligned}$$

So BD is the candidate key.

(iii) $ABC \to D, D \to A$

Note that B and C don't appear on any RHS. Try BC^+ first.
But we can't use $ABC \to D$ as we don't have A. So $BC^+ = BC \neq ABCD$.

Try next biggest size, 3-attribute closures with BC.

First try ABC^+. $ABC \to D$ so $ABC^+ = ABCD$.

Then, also try BCD^+. $D \to A$ so $BCD^+ = ABCD$.

Therefore, ABC and BCD are the candidate keys.

(iv) $A \to B, BC \to D, A \to C$

A doesn't appear in any RHS. First, try A^+.

$$\begin{aligned}A &\to B \text{ (add B)}\\ A &\to C \text{ (add C)}\\ BC &\to D \text{ (add D)}\end{aligned}$$

So, A is the candidate key.

(v) $AB \to C, AB \to D, C \to A, D \to B$

Can't use RHS trick for this question sadly. First consider single attribute closures.

$$\begin{aligned}&A^+ = A\\ &B^+ = B\\ &C^+ = AC \text{ (from } C \to A)\\ &D^+ = BD \text{ (from } D \to B)\end{aligned}$$

So the candidate keys aren't a single attribute only. Try two-attribute keys.

$$AB^+ = ABCD \text{ (from } AB \to C \text{ and } AB \to D)$$
$$AC^+ = AC \text{ (from } C \to A)$$
$$AD^+ = ABCD \text{ (from } D \to B \text{ and } AB \to C)$$
$$BC^+ = ABCD \text{ (from } C \to A \text{ and } AB \to D)$$
$$BD^+ = BD \text{ (from } D \to B)$$
$$CD^+ \geq C^+ + D^+ = AC + BD = ABCD$$

So the candidate keys are AB, AD, BC and CD.

(vi) $A \to BCD$

A is obviously the candidate key.

Identifying BCNF and 3NF

Always find candidate key(s) first to help out. Check BCNF first. Every FDs $LHS \to RHS$ must satisfy at least one condition from:

1. Trivial FD, where RHS is a subset of the LHS (e.g. $ABC \to B$, $AB \to AB$)

2. The LHS must be a superkey, which can determine all other attributes. This means that since candidate keys are the smallest-sized superkeys, the LHS must contain a candidate key inside.

If it is BCNF, it is also 3NF. If not, add the third condition of 3NF, as 3NF is more lenient.

3. Is the RHS a single attribute and part of a candidate key?

Mainly check if FD satisfies the second condition, as first is trivial and easy to identify.

Question 4b and 4c

(b) Show whether R is in BCNF?

(c) Show whether R is in 3NF?

(i) $C \to D, C \to A, B \to C$

B was the candidate key. Check BCNF first

$C \to D$ (not BCNF, LHS = C is not a super-key as doesn't contain candidate key B)
$C \to A$
$B \to C$

Note that all the other FDs would also not be BCNF, but one failing is enough to say R is not BCNF. Try 3NF

$C \to D$ (not 3NF, as although RHS = D is a single att, it is not in a candidate key)
$C \to A$
$B \to C$

The other FDs also fail 3NF too, but one failed FD is enough to conclude that R isn't 3NF.

(ii) $B \to C, D \to A$

BD was candidate key.

$B \to C$ (not BCNF, LHS = B is not a super-key as doesn't contain candidate key BD)
$D \to A$

Therefore R isn't BCNF. For 3NF

$$\begin{aligned} &B \to C \text{ (not 3NF, RHS = C is a single attribute, but } RHS \not\subseteq BD\text{)} \\ &D \to A \end{aligned}$$

So R isn't 3NF either.

(iii) $ABC \to D, D \to A$

ABC and BCD are candidate keys. For BCNF

$$\begin{aligned} ABC \to D &\text{ (BCNF, ABC is a candidate key and so, is a super-key)} \\ D \to A &\text{ (not BCNF, D not super-key as doesn't have candidate key)} \end{aligned}$$

So R is not BCNF. For 3NF

$$\begin{aligned} ABC \to D &\text{ (BCNF, so 3NF too)} \\ D \to A &\text{ (3NF, is single attribute and part of candidate key BCD)} \end{aligned}$$

So R is 3NF.

(iv) $A \to B, BC \to D, A \to C$

A is candidate key.

$$\begin{aligned} A \to B &\text{ (BCNF, A is a candidate key and so, is a super-key)} \\ BC \to D &\text{ (not BCNF, BC doesn't contain any candidate key)} \\ A \to C & \end{aligned}$$

Not BCNF. Check 3NF

$$\begin{aligned} A \to B &\text{ (BCNF, so 3NF)} \\ BC \to D &\text{ (not 3NF as D is not in candidate key A)} \\ A \to C & \end{aligned}$$

So R isn't 3NF either.

(v) $AB \to C, AB \to D, C \to A, D \to B$

Candidate keys were AB, AD, BC and CD.

$$\begin{aligned} AB \to C &\text{ (BCNF, as AB is super-key)} \\ AB \to D &\text{ (BCNF, as AB is super-key)} \\ C \to A &\text{ (not BCNF, as C not super-key)} \\ D \to B & \end{aligned}$$

R not BCNF.

$$\begin{aligned} AB \to C &\text{ (BCNF)} \\ AB \to D &\text{ (BCNF)} \\ C \to A &\text{ (3NF, A is single att and part of candidate key AB)} \\ D \to B &\text{ (3NF, B is single att and part of candidate key AB)} \end{aligned}$$

Therefore R is 3NF.

(vi) $A \to BCD$

A is the candidate key.

$$A \to BCD \text{ (BCNF, as A is super-key)}$$

So R is BCNF. R is also then, 3NF as well.

Minimal Cover

Mainly for 3NF decomposition. A minimal cover F_c is the more optimal version of a set of FDs F. It removes redundant functional dependencies, and redundant functional attributes in the functional dependencies. If you try to remove an FD or an attribute from an FD, F_c would represent a different set of FDs for R.

To compute the minimal cover:

1. Make every FD into canonical form $X \to A$, where A is a single attribute.
2. Remove redundant attributes from the LHS. Candidate keys may be helpful.
3. Remove redundant functional dependencies, usually using the transitive property.

Question 13

Compute a minimal cover for:

$$F = \{B \to A, D \to A, AB \to D\}$$

Question doesn't say what R is, so assume R = ABCD (although doesn't affect answer if ABD or ABCDE...).

If R = ABCD, the candidate key would be BC.

Step 1: Convert all FDs to canonical form. All FDs are already in canonical form.

$$F_c = \{B \to A, D \to A, AB \to D\}$$

Step 2: Remove redundant attributes from the LHS of each FD (don't change what the FD means).

We can't remove anything from $B \to A$ or $D \to A$. So we only worry about $AB \to D$. Since B is part of the candidate key, it probably holds more importance in F, so we prove that we can remove A from $AB \to D$ by proving $B \to D$.

Consider the other FD in F_c, $B \to A$. Then

$$\begin{aligned} B \to A &= BB \to AB \text{ (augmentation)} \\ &= B \to AB. \end{aligned}$$

Since $B \to AB$ and $AB \to D$, then $B \to D$ by transitivity.

$$F_c = \{B \to A, D \to A, B \to D\}$$

Step 3: Remove redundant FDs.

$$F_c = \{B \to A, D \to A, B \to D\}$$

Take note that $B \to A$ and $B \to D$ have the same LHS, and $D \to A$ has the RHSes of those two. So we could use transitivity to prove one is redundant.

Since $B \to D$ and $D \to A$, then this implies by transitivity that $B \to A$. So we can remove $B \to A$ as it is already implied by the other two (we don't lose any meaning).

So finally,

$$F_c = \{D \to A, B \to D\}$$

Question 17's Minimal Cover

Not a tut question, just more minimal cover practice. Note that the provided minimal cover is wrong. Consider the schema R and set of fds F

$$
\begin{aligned}
R &= ABCDEFGH \\
F &= \{ABH \to C, A \to D, C \to E, BGH \to F, F \to AD, E \to F, BH \to E\}
\end{aligned}
$$

We will find the minimal cover. From the RHS trick, BGH must be in the candidate keys. We can prove that BGH is the candidate key.

Step 1: Convert into canonical form $X \to A$, where RHS is a single att

$$F_c = \{ABH \to C, A \to D, C \to E, BGH \to F, F \to A, F \to D, E \to F, BH \to E\}.$$

Step 2: Remove redundant attributes from LHS.

For $ABH \to C$, we probably could remove A as BH is in the candidate key and $A^+ = AD$, which doesn't have C. Prove this by proving $BH \to C$.

$$
\begin{aligned}
BH &\to E \\
&\to F \text{ (from } E \to F) \\
&\to A \text{ (from } F \to A)
\end{aligned}
$$

Since $BH \to A$, then $BH \to ABH$ by augmentation. So since $ABH \to C$ as well, by transitivity, $BH \to C$, so we can remove A from $ABH \to C$.

We can't really remove any more attributes from $BH \to C$, as there are no FDs with LHS B or LHS H. For $BGH \to F$, consider removing G, as it never appears in any FD at all (especially the LHS). Prove $BH \to F$.

$$
\begin{aligned}
BH &\to E \\
&\to F \text{ (from } E \to F)
\end{aligned}
$$

So we can remove G from $BH \to F$. For $BH \to E$, there doesn't seem to be any FD with B or H in the LHS alone, so impossible to prove that $B \to E$ or $H \to E$.

$$F_c = \{BH \to C, A \to D, C \to E, BH \to F, F \to A, F \to D, E \to F, BH \to E\}.$$

Step 3: Remove redundant functional dependencies.

Note that $BH \to C$ and $BH \to E$ have the same LHS BH, and $C \to E$ has both of those RHSes. By transitivity, $BH \to C$ and $C \to E$ implies $BH \to E$. So $BH \to E$ is redundant.

For $BH \to C$ and $BH \to F$, we'll see if any of these are redundant. C never appears in the RHS in the other FDs, so it is impossible to prove it to be redundant. Try prove that $BH \to F$ through $BH \to C$.

$$
\begin{aligned}
BH &\to C \\
&\to E \text{ (from } C \to E) \\
&\to F \text{ (from } E \to F)
\end{aligned}
$$

So $BH \to F$ is redundant, as it can be derived from other FDs. Also take note of $F \to A, F \to D$ and $A \to D$. Since $F \to A$ and $A \to D$ implies $F \to D$, then $F \to D$ is redundant.

So the final result is

$$F_c = \{BH \to C, A \to D, C \to E, F \to A, E \to F\}.$$

3NF Decomposition

To decompose a schema into 3NF:

1. Compute the minimal cover.

2. Flatten the functional dependencies to make relations/tables.

3. Add the candidate key as a relation if the existing relations aren't connected enough.

Question 17

The minimal cover was

$$F_c = \{BH \to C, A \to D, C \to E, F \to A, E \to F\}$$

So flattening the FDs gives relations

BHC **A**D **C**E **F**A **E**F

where the keys are in bold. There isn't enough connectivity, as G isn't even represented here, and so we add the candidate key BGH as a relation.

BHC **A**D **C**E **F**A **E**F **BGH**

BCNF Decomposition

To decompose a schema into BCNF:

1. Calculate the candidate keys of the FDs.

2. Go through each FD for a relation. If the FD $LHS \to RHS$ is not in BCNF, then split the relation R into two:

$S = LHS^+$, with the FDs used to find that closure.

$T =$ R - LHS^+ + LHS, with all remaining FDs that don't completely change from the removed attributes.

Note that affecting the LHS of an FD would change entirely what the FD means, but removing attributes from the RHS of an FD is ok.

Question 15

Consider the schema R and set of ids F

$$R = ABCDEFGH$$
$$F = \{ABH \to C, A \to DE, BGH \to F, F \to ADH, BH \to GE\}$$

Produce a BCNF decomposition of R.

First, find the candidate key. Note that B doesn't appear in the RHS, so it has to be part of the candidate key.

B alone cannot use any functional dependencies to determine any other attributes, so use two-attribute keys instead.

BH is the only two-set attribute set with an FD in the LHS, so consider BH^+.

$$BH \to GE \text{ (add EG)}$$
$$BGH \to F \text{ (add F)}$$
$$F \to ADH \text{ (add AD)}$$
$$ABH \to C \text{ (add C)}$$

So BH is the candidate key, since $BH^+ = ABCDEFGH$.

Now go through each FD. Working out may differ, but for this decomposition, we go in order.

For $ABH \to C$, this FD is in BCNF, as the LHS ABH has candidate key BH and so is a super-key.

However, for $A \to DE$, the LHS A isn't a super-key, and so we split R into two. Note that $A^+ = ADE$, so

$R_1 : A^+ = ADE,$ with FDs $= \{A \to DE\}$ and key A
$R_2 : ABCDEFG - DE = ABCFGH$ with FDs $= \{ABH \to C, BGH \to F, F \to AH, BH \to G\}$ and key BH

Note that FDs can decompose and combine the RHS, and so $F \to ADH$ can become $F \to A, F \to D, F \to H$. Since D was removed, we only remove $F \to D$, and recombine to get $F \to AH$. Similar logic for $BH \to GE$.

R_1 is in BCNF. For R_2:

$$\begin{aligned} ABH &\to C \text{ (is BCNF as has candidate key BH in LHS)} \\ BGH &\to F \text{ (is BCNF as has candidate key BH in LHS)} \\ F &\to AH \text{ (is not BCNF)} \\ BH &\to G \text{ (is BCNF as has candidate key BH in LHS)} \end{aligned}$$

We split R_2 into two tables by the FD $F \to AH$:

$R_3 : F^+ = FAH,$ with FDs $= \{F \to AH\}$ and key F
$R_4 : ABCFGH - AH = BCFG$ with FDs $= \{\}$ and key of these FDs $BCFG$

There are no FDs in R_4 mainly due to the removal of H. Affecting the LHS affects what the dependency means, so we delete these functional dependencies (reminder BCNF doesn't preserve all FDs).

$XY \to Z$ doesn't imply that $X \to Z$ or $Y \to Z$.

R_3 is in BCNF, and R_4 is definitely BCNF (no functional dependencies to check conditions on).

So the BCNF decomposition result is R_1 R_3 R_4, or

ADE **F**AH **BCFG**.

Question 7 cont

(a) $C \to D, C \to A, B \to C$
Remember CK is B. So $C \to D$ violates BCNF. $C^+ = ACD$ from $C \to D$ and $C \to A$
$R_1 : C^+ = ACD$ with FDs $C \to D$ and $C \to A$, and key C
$R_2 : ABCD - ACD + C = BC$ with FD $B \to C$, and key B

Both tables are in BCNF now.

(b) $B \to C, D \to A$
CK is BD. $B \to C$ violates BCNF. $B^+ = BC$ so
R_1: $B^+ = BC$ with FD $B \to C$, key B
R_2: $ABCD - BC + B = ABD$ with FD $D \to A$, key BD

R_1 is BCNF, R_2 isn't still and becomes BD and AD from the FD $D \to A$. So final result is BC, BD, AD.

(c) $ABC \to D, D \to A$
CKs were ABC and BCD. $D \to A$ violates and $D^+ = DA$.

$R_1 : DA$ with FD $D \to A$ and key D
$R_2 : ABCD - DA + D = BCD$ with no FDs (so key would be BCD).

Both tables are now BCNF, although we had to lose $ABC \to D$.

(d) $A \to B$, $BC \to D$, $A \to C$
A was the CK so the FD $BC \to D$ violates BCNF. $BC^+ = BCD$
$R_1 : BCD$ with key BC, and FD $BC \to D$
$R_2 : ABCD - BCD + BC = ABC$ with FDs $A \to B, A \to C$ and key A
Both are now in BCNF.

(e) $AB \to C$, $AB \to D$, $C \to A$, $D \to B$
CKs were AB, AD, BC, CD.
First two FDs are good, but $C \to A$ violates BCNF so split on it.
$R_1 : C^+ = AC$ with key C and FD $C \to A$
$R_2 : ABCD - AC + C = BCD$ with keys BC, CD, and the FDs being $D \to B$
Note that $AB \to C$ and $AB \to D$ weren't preserved. R_2 still isn't BCNF so split R_2 by $D \to B$ so that
$R_3 : DB$ with FD $D \to B$, key D
$R_4 : BCD - DB + D = CD$ with no FDs, key = CD

So the final result is AC DB CD.

(f) $A \to BCD$
R is already in BCNF as the key is A.