University of Toronto
csc343, Fall 2020

Solutions for Assignment 3

IMPORTANT: In both BCNF decomposition and 3NF synthesis, there are choice points. This means different solutions are possible, depending on which choices are made.

BCNF Problem

Although one can often skip ahead to some of the conclusions or combine steps, these solutions are very systematic, so that you can see the full pattern.

1. Consider a relation schema R with attributes $ABCDEFGHI$ with functional dependencies S:

$$S = \{AEG \to F, \quad B \to AD, \quad AG \to HI, \quad BG \to D\}$$

   (a) State which of the given FDs violate BCNF.

   BCNF requires that the LHS of an FD be a superkey.

   - $\times$ $AEG^+ = AEGFHI$, so AEG is not a superkey and $AEG \to F$ violates BCNF.
   - $\times$ $B^+ = BAD$, so B is not a superkey and $B \to AD$ also violates BCNF.
   - $\times$ $AG^+ = AGHI$, so AG is not a superkey and $AG \to HI$ also violates BCNF.
   - $\times$ $BG^+ = BGDAHI$, so BG is not a superkey and $BG \to D$ also violates BCNF.

   (b) Employ the BCNF decomposition algorithm to obtain a lossless and redundancy-preventing decomposition of relation R into a collection of relations that are in BCNF. Make sure it is clear which relations are in the final decomposition, and don't forget to project the dependencies onto each relation in that final decomposition. Because there are choice points in the algorithm, there may be more than one correct answer. List the final relations in alphabetical order (order the attributes alphabetically within a relation, and order the relations alphabetically).

   - Decompose R using FD $AEG \to F$. $AEG^+ = AEGFHI$, so this yields two relations: $R1 = AEFGHI$ and $R2 = ABCDEG$.
   - Project the FDs onto $R1 = AEFGHI$.

A	E	F	G	H	I	closure	FDs
✓						$A^+ = A$	nothing
	✓					$E^+ = E$	nothing
		✓				$F^+ = F$	nothing
			✓			$G^+ = G$	nothing
				✓		$H^+ = H$	nothing
					✓	$I^+ = I$	nothing
✓	✓					$AE^+ = AE$	nothing
✓		✓				$AF^+ = AF$	nothing
✓			✓			$AG^+ = AGHI$	$AG \to HI$: violates BCNF; abort the projection

   We must decompose $R1$ further.

- Decompose $R1$ using FD $AG \to HI$. $AG^+ = AGHI$, so this yields two relations: $R3 = AGHI$ and $R4 = AEFG$.
- Project the FDs onto $R3 = AGHI$.

A	G	H	I	closure	FDs
✓				$A^+ = A$	nothing
	✓			$G^+ = G$	nothing
		✓		$H^+ = H$	nothing
			✓	$I^+ = I$	nothing
✓	✓			$AG^+ = AGHI$	$AG \to HI$; AG is a superkey of $R3$
supersets of AG				irrelevant	can only generate weaker FDs than what we already have
✓		✓		$AH^+ = AH$	nothing
✓			✓	$AI^+ = AI$	nothing
	✓	✓		$GH^+ = GH$	nothing
	✓		✓	$GI^+ = GI$	nothing
		✓	✓	$HI^+ = HI$	nothing
✓		✓	✓	$AHI^+ = AHI$	nothing
	✓	✓	✓	$GHI^+ = GHI$	nothing

  This relation satisfies BCNF.
- Project the FDs onto $R4 = AEFG$.

A	E	F	G	closure	FDs
✓				$A^+ = A$	nothing
	✓			$E^+ = E$	nothing
		✓		$F^+ = F$	nothing
			✓	$G^+ = G$	nothing
✓	✓			$AE^+ = AE$	nothing
✓		✓		$AF^+ = AF$	nothing
✓			✓	$AG^+ = AGHI$	nothing
	✓	✓		$EF^+ = EF$	nothing
	✓		✓	$EG^+ = EG$	nothing
		✓	✓	$FG^+ = FG$	nothing
✓	✓	✓		$AEF^+ = AEF$	nothing.
✓	✓		✓	$AEG^+ = AEGF$	$AEG \to F$; AEG is a superkey of $R4$.
✓		✓	✓	$AFG^+ = AFGHI$	nothing.
	✓	✓	✓	$EFG^+ = EFG$	nothing.

  This relation satisfies BCNF.
- Now return and project the FDs onto $R2 = ABCDEG$.

A	B	C	D	E	G	closure	FDs
✓						$A^+ = A$	nothing
	✓					$B^+ = BAD$	$B \to AD$: violates BCNF; abort the projection

  We must decompose $R2$ further.
- Decompose $R2$ using FD $B \to AD$. $B^+ = BAD$, so this yields two relations: $R5 = ABD$ and $R6 = BCEG$.
- Project the FDs onto $R5 = ABD$.

A	B	D	closure	FDs
✓			$A^+ = A$	nothing
	✓		$B^+ = BAD$	$B \rightarrow AD$; B is a superkey of $R5$
supersets of B			irrelevant	can only generate weaker FDs than what we already have
		✓	$D^+ = D$	nothing
✓		✓	$AD^+ = AD$	nothing

This relation satisfies BCNF.

- Project the FDs onto $R6 = BCEG$.

B	C	E	G	closure	FDs
✓				$B^+ = BAD$	nothing
	✓			$C^+ = C$	nothing
		✓		$E^+ = E$	nothing
			✓	$G^+ = G$	nothing
✓	✓			$BC^+ = BCAD$	nothing
✓		✓		$BE^+ = BEAD$	nothing
✓			✓	$BG^+ = BGADHI$	nothing
	✓	✓		$CE^+ = CE$	nothing
	✓		✓	$CG^+ = CG$	nothing
		✓	✓	$EG^+ = EG$	nothing
✓	✓	✓		$BCE^+ = BCEAD$	nothing
✓	✓		✓	$BCG^+ = BCGADHI$	nothing
✓		✓	✓	$BEG^+ = BEGADHIF$	nothing
	✓	✓	✓	$CEG^+ = CEG$	nothing

This relation satisfies BCNF.

- Final decomposition (order the attributes alphabetically within a relation, and order the relations alphabetically):
  i. $R5 = ABD$ with FD $B \rightarrow AD$,
  ii. $R4 = AEFG$ with FD $AEG \rightarrow F$,
  iii. $R3 = AGHI$ with FD $AG \rightarrow HI$,
  iv. $R6 = BCEG$ with no FDs.

(c) Does your schema preserve dependencies? Explain how you know that it does or does not?

Yes, it does. For each of the first three of the original FDs in S, there is a relation that includes all of the FD's attributes. This ensures that each of them will project onto a relation and therefore be preserved. There is no relation in the schema, however, that includes all the attributes of the final FD, $BG \rightarrow D$. However, we note that $BG \rightarrow D$ is implied by the other FDs (that do project) because it is a weaker version of $B \rightarrow D$ (that projects onto $R5$). This means the $BG \rightarrow D$ is enforced. Therefore, our decomposition is dependency preserving.

(d) Use the Chase Test to show that your schema is a lossless-join decomposition.

The chase test demonstrates that it is a lossless-join decomposition:

A	B	C	D	E	F	G	H	I
a	**b**	1	**d**	2	3	4	5	6
a	1	2	3	**e**	**f**	**g**	h	i
a	1	2	3	4	5	**g**	**h**	**i**
a	**b**	**c**	d	**e**	f	**g**	h	i

3NF Problem

Although one can often skip ahead to some of the conclusions or combine steps, these solutions are very systematic, so that you can see the full pattern.

2. Consider a relation A with attributes $LMNOPQRS$ and functional dependencies B:

$$B = \{LNOP \to M, \quad M \to NQ, \quad NO \to LQ, \quad MNQ \to LO, \quad LMQ \to NOS\}$$

(a) Compute a minimal basis for B. In your final answer, put the FDs into alphabetical order.

- To find a minimal basis, we'll first simplify FDs to singleton right-hand sides. We'll also number the resulting FDs for easy reference, and call this set $B1$:

1	$LNOP \to M$
2	$M \to N$
3	$M \to Q$
4	$NO \to L$
5	$NO \to Q$
6	$MNQ \to L$
7	$MNQ \to O$
8	$LMQ \to N$
9	$LMQ \to O$
10	$LMQ \to S$

- The next step is to try reducing the LHS of FDs with multiple attributes on the LHS. For these closures, we will close over the full set $B1$, including even the FD being considered for simplification; remember that we are not considering removing the FD, just strengthening it. Note: The order in which we consider attributes to reduce may affect the results we get, but we will always get *a* minimal basis.

1 $LNOP \to M$

$L^+ = L$ so we can't reduce the LHS to L.
$N^+ = N$ so we can't reduce the LHS to N.
$O^+ = O$ so we can't reduce the LHS to O.
$P^+ = P$ so we can't reduce the LHS to P.
$LN^+ = LN$ so we can't reduce the LHS to LN.
$LO^+ = LO$ so we can't reduce the LHS to LO.
$LP^+ = LP$ so we can't reduce the LHS to LP.
$NO^+ = NOLQ$ so we can't reduce the LHS to NO.
$NP^+ = NP$ so we can't reduce the LHS to NP.
$OP^+ = OP$ so we can't reduce the LHS to OP.
$LNO^+ = LNOQ$ so we can't reduce the LHS to LNO.
$LOP^+ = LOP$ so we can't reduce the LHS to LOP.
$LNP^+ = LNP$ so we can't reduce the LHS to LNP.
$NOP^+ = NOPLQMOS$ so we can reduce the LHS to NOP. ✓

4 $NO \to L$

$N^+ = N$ so we can't reduce the LHS to N.
$O^+ = O$ so we can't reduce the LHS to O.

5 $NO \to Q$

$N^+ = N$ so we can't reduce the LHS to N.
$O^+ = O$ so we can't reduce the LHS to O.

6 $MNQ \to L$

$M^+ = MNQOLS$ so we can reduce the LHS to M. ✓

7 $MNQ \to O$

$M^+ = MNQOLS$ so we can reduce the LHS to M. ✓

8 $LMQ \to N$

We know from (2) that $M \to N$, so we can reduce the LHS to M. ✓

9 $LMQ \to O$

We saw in (7) that $M^+ = MNQOLS$, so we can reduce the LHS to M. ✓

10 $LMQ \to S$

$L^+ = L$ so we can't reduce the LHS to L.

$M^+ = MNQOLS$ so we can reduce the LHS to M. ✓

- Let's call the set of FDs that we have after reducing left-hand sides $B2$:

1	$NOP \to M$
2	$M \to N$
3	$M \to Q$
4	$NO \to L$
5	$NO \to Q$
6	$M \to L$
7	$M \to O$
8	$M \to N$
9	$M \to O$
10	$M \to S$

- Now we'll look for redundant FDs to eliminate. Each row in the table below indicates which of the 10 FDs we still have on hand as we consider removing the next one. Of course, as we do the closure test to see whether we can remove $X \to Y$, we can't use $X \to Y$ itself, so an FD is never included in its own row. Again, note that there might be more than one correct result depending on the order of elimination.

FD	Exclude these from $B2$ when computing closure	Closure	Decision
1	1	There's no way to get M without this FD	keep
2	2	Duplicate FD to (8)	discard
3	2, 3	$M^+ = MLONSQ$	discard
4	2, 3, 4	$NO^+ = NOQ$	Keep
5	2, 3, 5	$NO^+ = NOL$	Keep
6	2, 3, 6	$M^+ = MONSLQ$	discard
7	2, 3, 6, 7	Duplicate FD to (9)	discard
8	2, 3, 6, 7, 8	There's no way to get N without this FD	keep
9	2, 3, 6, 7, 9	There's no way to get O without this FD	keep
10	2, 3, 6, 7, 10	There's no way to get S without this FD	keep

- No further simplifications are possible.
- So the following set $B3$ is a minimal basis (ordered alphabetically):

8	$M \to N$
9	$M \to O$
10	$M \to S$
4	$NO \to L$
5	$NO \to Q$
1	$NOP \to M$

(b) Using your minimal basis from the last subquestion, compute all keys for A.

Examing all subsets of the attributes would be very time-consuming because there are 2^8 of them.

Some careful reasoning can spare us having to compute many closures.

Notice that R is not anywhere in the Minimal Basis FDs. That means it has to be in every key — there is no other way to get it. In fact, even if an attribute appears in an FD, if it never appears on a RHS, it will have to be in every key. P is an example of this.

If, on the other hand, an attribute appears only on the RHS of the FDs, never the left, it is of no help to us in computing closures. It cannot be in any key, because you could always remove it and still get the same closure. Q is an example of this.

To summarize, here is what we know so far:

	Appears on		
Attribute	LHS	RHS	Conclusion
Q, L, S	–	✓	is not in any key
M, N, O	✓	✓	must check
P	✓	–	must be in every key
R	–	–	must be in every key

This means that we only have to consider all combinations of M, N, and O: M, N, O, MN, MO, NO. For each, we must add in P and R, since they are in every key. So the list of possible keys to consider is: $MPR, NPR, OPR, MNPR, MOPR, NOPR$.

- $MPR^+ = MPRNOSLQ$. So MPR is a key, and any superset of MPR cannnot be a key, since it would not be minimal.
- $NPR^+ = NPR$. This is not a key.
- $OPR^+ = OPR$. This is not a key.
- $NOPR^+ = NOPRMSLQ$. This is a key. If we remove any attribute, the remaining set of attributes cannot be a superkey for relation A.

We achieved a great speed-up using our method; instead of 64 closures, we only had to compute 4!

In conclusion, there are 2 keys for P: MPR and $NOPR$.

(c) Employ the 3NF synthesis algorithm to obtain a lossless and dependency-preserving decomposition of relation A into a collection of relations that are in 3NF. Do not "over normalize", this means that you should combine all FDs with the same left-hand side into a single relation. If your schema includes one relation that is a subset of another, remove the smaller one.

- Following the 3NF synthesis algorithm, we would get one relation for each FD of the minimal basis from the previous step. However, we should merge the right-hand sides before doing so. This will yield a smaller set of relations and they will still form a lossless and dependency-preserving decomposition of relation R into a collection of relations that are in 3NF.
- Let's call the revised FDs $B4$:
  $M \rightarrow NOS$
  $NO \rightarrow LQ$
  $NOP \rightarrow M$
- The set of relations that would result would have these attributes:

$$R1(M, N, O, S), \quad R2(L, N, O, Q), \quad R3(M, N, O, P)$$

- No relation is a subset of another relation, otherwise we would need to throw the redundant subset relation away. Thus we keep all 3 relations.
- R is not included in any of the relations, thus no relation above is a superkey for A. So we must add another relation whose schema is some key. We will choose a key we computed in the previous step: MPR.

- So the final set of relations is:

$$R1(M, N, O, S), \quad R2(L, N, O, Q), \quad R3(M, N, O, P), \quad R4(M, P, R)$$

(d) Does your schema allow redundancy? Explain how you know that it does or does not.

- Because we formed each relation from an FD, the LHS of those FDs are indeed superkeys for their relations. However, there may be other FDs that violate BCNF and therefore allow redundancy. The only way to find out is to project the FDs onto each relation. If an FD is obtained with a LHS that is not a superkey for the relation, we have discovered the possibility for redundancy.
- Projecting $B4$ onto $R3$, we obtain, among other things, $M \rightarrow NO$. $M^+ = MNOSLQ$, so M is not is a superkey of relation $R3$.
- Therefore, our schema admits redundancy.