CSC343H1 – Assignment 3 Decompositions

Danny Chen, Luka Djukic

March 18, 2020

1. a) Find the minimal basis for S_p.

First, we'll eliminate any duplicate FDs in S_p. Then we'll break up all of the remaining FDs in S_p into singleton RHSs and sort them at the end. We'll call this set S_1:

Original FD	Keep?	Singleton RHS FDs (S_1)
$IJ \to K$	Yes	$IJ \to K$
$J \to LI$	Yes	$J \to L$ $J \to I$
$JN \to KM$	Yes	$JN \to K$ $JN \to M$
$K \to IJL$	Yes	$K \to I$ $K \to J$ $K \to L$
$KLN \to M$	Yes	$KLN \to M$
$M \to J$	No	removed duplicate FD $M \to J$ in S_p
$M \to IJL$	Yes	$M \to I$ $M \to J$ $M \to L$

Next, we'll try to simplify the LHSs of all of the FDs in S_1 that have at least 2 attributes. We'll call the simplified set of FDs S_2.

Singleton RHS FDs (S_1)	Closures	Simplify?	Simplified FDs (S_2)
$IJ \to K$	$I^+ = I$ $J^+ = IJKL$	Yes	$J \to K$
$JN \to K$	$J^+ = IJKL$	Yes	$J \to K$
$JN \to M$	$J^+ = IJKL$ $N^+ = N$ $JN^+ = IJKLMN$	No	$JN \to M$
$KLN \to M$	$K^+ = IJKL$ $L^+ = L$ $N^+ = N$ $KL^+ = IJKL$ $KN^+ = IJKLMN$	Yes	$KN \to M$

The third column contains all FDs in the second column, but sorted and renumbered.

FD	Simplified FDs (S_2)	Unique FDs (S_2)	Sorted, Renumbered FDs (S_2)
1	$J \rightarrow K$	$J \rightarrow K$	$J \rightarrow I$
2	$J \rightarrow L$	$J \rightarrow L$	$J \rightarrow K$
3	$J \rightarrow I$	$J \rightarrow I$	$J \rightarrow L$
4	$J \rightarrow K$	removed duplicate FD $J \rightarrow K$	$JN \rightarrow M$
5	$JN \rightarrow M$	$JN \rightarrow M$	$K \rightarrow I$
6	$K \rightarrow I$	$K \rightarrow I$	$K \rightarrow J$
7	$K \rightarrow J$	$K \rightarrow J$	$K \rightarrow L$
8	$K \rightarrow L$	$K \rightarrow L$	$KN \rightarrow M$
9	$KN \rightarrow M$	$KN \rightarrow M$	$M \rightarrow I$
10	$M \rightarrow I$	$M \rightarrow I$	$M \rightarrow J$
11	$M \rightarrow J$	$M \rightarrow J$	$M \rightarrow L$
12	$M \rightarrow L$	$M \rightarrow L$	-

Next, I'll try to eliminate each FD in S_2 using the closure test:

Sorted FDs (S_2)	Closure	Eliminate?	FDs Left (S_2)
$J \rightarrow I$	$J^+_{S_2-\{(1)\}} = \underline{I}JKL$	Yes	
$J \rightarrow K$	$J^+_{S_2-\{(1),(2)\}} = JL$	No	$J \rightarrow K$
$J \rightarrow L$	$J^+_{S_2-\{(1),(3)\}} = IJK\underline{L}$	Yes	
$JN \rightarrow M$	$JN^+_{S_2-\{(1),(3),(4)\}} = IJKL\underline{M}N$	Yes	
$K \rightarrow I$	$K^+_{S_2-\{(1),(3),(4),(5)\}} = JKL$	No	$K \rightarrow I$
$K \rightarrow J$	$K^+_{S_2-\{(1),(3),(4),(6)\}} = IKL$	No	$K \rightarrow J$
$K \rightarrow L$	$K^+_{S_2-\{(1),(3),(4),(7)\}} = IJK$	No	$K \rightarrow L$
$KN \rightarrow M$	$KN^+_{S_2-\{(1),(3),(4),(8)\}} = IJKLN$	No	$KN \rightarrow M$
$M \rightarrow I$	$M^+_{S_2-\{(1),(3),(4),(9)\}} = \underline{I}JKLM$	Yes	
$M \rightarrow J$	$M^+_{S_2-\{(1),(3),(4),(9),(10)\}} = LM$	No	$M \rightarrow J$
$M \rightarrow L$	$M^+_{S_2-\{(1),(3),(4),(9),(11)\}} = IJK\underline{L}M$	Yes	

Therefore, our minimal basis consist of the FDs

$$S_3 = \{J \rightarrow K, K \rightarrow I, K \rightarrow J, K \rightarrow L, KN \rightarrow M, M \rightarrow J\}.$$

b) Find all keys in R.

Recall from 1a) that our new FDs are:

$$S_3 = \{J \to K, K \to I, K \to J, K \to L, KN \to M, M \to J\}.$$

To significantly reduce the number of closures that we'll need to compute, we'll note sets of attributes that either must appear in every key, or ones that cannot appear in any key. Then we'll verify whether the remaining attributes can appear in each candidate key.

Taken from the "Finding all keys" worksheet, we'll use these speedup rules applicable to all relations R, including this question where R has attributes $IJKLMNOP$.

1. If an attribute A doesn't appear on the LHS or RHS of any FD $X \to Y$, A must be in every key.
2. If an attribute A doesn't appear on the RHS of any FD $X \to Y$, A must be in every key.
3. If an attribute A doesn't appear on the LHS, but instead appears on the RHS of every FD $X \to Y$, A cannot be in any key.

We'll summarize our findings in this table, which is also inspired by the same worksheet:

Attribute in R	On LHS?	On RHS?	Rule Applied	Conclusion
I	-	Yes	3	cannot be in any key
J	Yes	Yes	none	check
K	Yes	Yes	none	check
L	-	Yes	3	cannot be in any key
M	Yes	Yes	none	check
N	Yes	-	2	must be in every key
O	-	-	1	must be in every key
P	-	-	1	must be in every key

Therefore, we only need to check all combinations of J, K, and M. Every combination must include N, O, and P since they are in every key, and exclude I and L since they cannot appear in any key.

Candidate Key	Closure	Is Key?	Result Keys
$JNOP$	$JNOP^+ = IJKLMNOP$	Yes	$JNOP$
$KNOP$	$KNOP^+ = IJKLMNOP$	Yes	$KNOP$
$MNOP$	$MNOP^+ = IJKLMNOP$	Yes	$MNOP$

All other possibilities include supersets of $JNOP$, $KNOP$, and $MNOP$, but none of them can be keys since they are not minimal.
Hence, R has keys $JNOP$, $KNOP$, and $MNOP$.

c) Use the 3NF synthesis algorithm to find a lossless, dependency-preserving decomposition of R in 3NF form.

From 1a), the minimal basis is the set:

$$S_3 = \{J \to K, K \to I, K \to J, K \to L, KN \to M, M \to J\}.$$

Moreover, 1b) tells us that R has keys $JNOP$, $KNOP$, and $MNOP$.

Before performing the 3NF synthesis algorithm, we'll first merge the RHSs of all FDs. This gives us a smaller set of relations that will still form a lossless and dependency-preserving decomposition of R in 3NF form. We'll call this set S_4:

$$S_4 = \{J \to K, K \to IJL, KN \to M, M \to J\}.$$

Now when we perform the 3NF synthesis algorithm, we'll get one relation per FD in S_4. These are the resulting set of relations we'll get:

$$R_1(J, K), R_2(I, J, K, L), R_3(K, M, N), R_4(J, M).$$

Since the attributes JK in R_1 occur within R_2, we can remove relation R_1. This leaves us with these relations:

$$R_2(I, J, K, L), R_3(K, M, N), R_4(J, M).$$

Since none of J, K, KN, and M are superkeys of our candidate keys for R, we'll need to define a new relation R_5 that contains all of the attributes of one of our candidate superkeys in R. We'll choose the superkey $JNOP$ for R_5:

$$R_5(J, N, O, P).$$

Thus, a lossless, dependency-preserving decomposition of R could be

$$R_2(I, J, K, L), R_3(K, M, N), R_4(J, M), R_5(J, N, O, P).$$

d) Does your schema allow redundancy? Justify your answer.

A schema allows redundancy if and only if one of the relations R_2, R_3, R_4 violates BCNF, or when one of the LHSs of all FDs in R is not a superkey for R_2, R_3, R_4. This is the case when at least one nontrivial FD of R has a LHS that doesn't form a superkey of R.

From 1a), the minimal basis is the set:

$$S_3 = \{J \to K, K \to I, K \to J, K \to L, KN \to M, M \to J\}.$$

Because the FDs $K \to IJL$, $KN \to M$, and $M \to J$ all come from relations R_2, R_3, and R_4, these FDs are in BCNF. Now we'll need to check if the FD $J \to K$ violates BCNF.

Notice that we can project the FD $J \to K$ into R_2. However, $J^+ = JK$ which shows that J is not a superkey of R_2.
Hence, the FD $J \to K$ violates BCNF and relation R_2 violates BCNF. So our schema allows redundancy.

2. a) Which of the FDs in S_T violate BCNF?

To determine whether the FDs in S_T violate BCNF, we will determine which LHSs of each FD in S_T do not form a superkey of T.

FD	Closure of LHS	LHS is a Superkey of T?	Satisfies BCNF?
$C \rightarrow EH$	$C^+ = CDEFGHIJ$	Yes	Yes
$DEI \rightarrow F$	$DEI^+ = DEFI$	No	No
$F \rightarrow D$	$F^+ = DF$	No	No
$EH \rightarrow CJ$	$EH^+ = CDEFGHIJ$	Yes	Yes
$J \rightarrow FGI$	$J^+ = DFGIJ$	No	No

Therefore, these FDs violate BCNF:

$$\{DEI \rightarrow F, F \rightarrow D, J \rightarrow FGI\}.$$

b) Use BCNF decomposition to find a lossless, dependency-preventing decomposition of T.

Recall that the FDs in S_T are

$$S_T = \{C \to EH, DEI \to F, F \to D, EH \to CJ, J \to FGI\}.$$

Also, from 2a) we found that the FDs

$$DEI \to F, F \to D, J \to FGI$$

in S_T violate BCNF. We'll create new relations for each BCNF-violating FD.

1. Decompose T using FD $DEI \to F$.
Notice that $DEI^+ = DEIF$ and $T - (DEI^+ - DEI) = CDEGHIJ$.
This yields two relations: $R_1(D, E, I, F)$ and $R_2(D, E, I, C, G, H, J)$.
We'll project all of the FDs in S_T onto R_1:

Attributes	Closure	New FDs for R_1?	Conclusions
D	$D^+ = D$	nothing	
E	$E^+ = E$	nothing	
I	$I^+ = I$	nothing	
F	$F^+ = FD$	$F \to D$	this FD violates BCNF

Therefore R_1 violates BCNF and we need to decompose it further.
2. Decompose R_1 using FD $F \to D$.
Notice that $F^+ = FD$ and $R_1 - (F^+ - F) = EIF$.
This yield two more relations: $R_3(F, D)$ and $R_4(E, I, F)$.
We'll project all of the FDs in S_T onto R_3:

Attributes	Closure	New FDs for R_3?	Conclusions
F	$F^+ = FD$	$F \to D$	F is a superkey
D	$D^+ = D$	nothing	
Supersets of F	irrelevant	only generates weaker FDs	

Therefore, R_3 has FDs $F \to D$ and is now in BCNF form.
We'll project all of the FDs in S_T onto R_4:

Attributes	Closure	New FDs for R_4?	Conclusions
E	$E^+ = E$	nothing	
I	$I^+ = I$	nothing	
F	$F^+ = FD$	nothing	
EI	$EI^+ = EI$	nothing	
EF	$EF^+ = EF$	nothing	
IF	$IF^+ = IF$	nothing	
EIF	$EIF^+ = EIF$	nothing	EIF is a superkey

Therefore, R_4 has FDs $E \to F$, $I \to F$, and $EI \to F$, and is now in BCNF form.

Now back to R_2. We'll project all of the FDs in S_T onto R_2:

Attributes	Closure	New FDs for R_4?	Conclusions
D	$D^+ = D$	nothing	
E	$E^+ = E$	nothing	
I	$I^+ = I$	nothing	
C	$C^+ = CEFGHIJ$	$C \to EH$	this FD violates BCNF

Therefore R_2 violates BCNF and we need to decompose it further.

3. Decompose R_2 using the FD $J \to FGI$.

Notice that $J^+ = JGID$ and $R_2 - (J^+ - J) = ECHJ$.

This yields two more relations: $R_5(J, G, I, D)$ and $R_6(E, C, H, J)$.

We'll project all of the FDs in S_T onto R_5:

Attributes	Closure	New FDs for R_4?	Conclusions
J	$J^+ = JGID$	$J \to DIG$	J is a superkey
G	$G^+ = G$	nothing	
I	$G^+ = G$	nothing	
D	$D^+ = D$	nothing	

Therefore, R_5 has the FD $J \to DIG$ and is now in BCNF form.

We'll project all of the FDs in S_T onto R_6:

Attributes	Closure	New FDs for R_4?	Conclusions
E	$E^+ = E$	nothing	
C	$C^+ = CEH$	$C \to EH$	
H	$H^+ = H$	nothing	
J	$J^+ = J$	nothing	
EH	$EH^+ = ECHJ$	$EH \to CJ$	EH is a superkey

Therefore, R_6 has the FD $CEH \to J$ and is now in BCNF form.

To summarize, here are all of our decomposed relations and their projected FDs:

Relations	FDs
$R_3(D, F)$	$F \to D$
$R_4(E, F, I)$	none
$R_5(D, G, I, J)$	$J \to DIG$
$R_6(E, C, H, J)$	$C \to EH$, $EH \to CJ$