Decomposition Algorithms

- It is always possible to find a dependency-preserving, lossless-join decomposition of a relation such that all relations in the decomposition are in 3NF.
- It is always possible to find a lossless-join decomposition of a relation into BCNF relations.
- It is **not** always possible to find a dependency-preserving, lossless-join decomposition of a relation into BCNF relations.

Algorithm for finding a dependency-preserving, lossless-join decomposition into 3NF relations:

1. Group together all FDs that have the same left-hand side (e.g. if we have the FDs X→Y1, X→Y2, X→Y3, we group them together as X→Y1Y2Y3)
2. For each FD X→Y in step 1, form the relation (XY) in the decomposition.
3. If any X'Y' is a subset of any XY, then remove the relation (X'Y') from the decomposition.
4. If none of the relations obtained after step 3 contains a candidate key of the original relation, form a relation for K in the decomposition, where K is one of the candidate keys for the original relation.

Example:

Given the relation R(A, B, C, D, E) and the FDs F= {A→B, A→C, C→A, BD→E}, find a decomposition of R into 3NF relations that is lossless-join and dependency-preserving.

Solution:

Step 1: F'={A→BC, C→A, BD→E}
Step 2: R1(A, B, C); R2(A, C); R3(B, D, E)
Step 3: We remove R2, as it is a subset of R1 -> R1(A, B, C); R3(B, D, E)
Step 4: R has two candidate keys: AD and CD, none of them is contained in R1 or R3, so we form R4(A, D)

The final decomposition is: R1(A, B, C); R3(B, D, E); R4(A, D)

Algorithm for finding a lossless-join decomposition into BCNF relations:

1. Group together all FDs that have the same left-hand side (e.g. if we have the FDs $X \rightarrow Y1$, $X \rightarrow Y2$, $X \rightarrow Y3$, we group them together as $X \rightarrow Y1Y2Y3$)
2. Compute F^+: The set of all functional dependencies that can be inferred from F
3. while there are relations R_i, which are not in BCNF:
Let $X \rightarrow Y$ be a non-trivial FD in F^+ that holds on R_i such that $X \rightarrow R_i$ is not in F^+
Decompose R_i into two relations R_i-Y and XY

Example:

Given R(A, B, C, D, E, G, H) and F={B→E, B→H, E→A, E→D, AH→C}, find a lossless-join decomposition of R into BCNF relations.

Solution:

F: {AH→C, E→AD, B→EH}

The only candidate key for R is BG, hence R is not in BCNF as all FDs fail the check for BCNF.

For AH→C, AH→ABCDEGH is not in F^+, therefore we break R into R1(A, H, C) and R2(A, B, D, E, G, H)

R1 is in BCNF, but R2 is not (the only candidate key for R2 is BG and E→AD violates BCNF). For E→AD, E→ABDEGH is not in F^+, therefore we break R2 into R3(E, A, D) and R4(B, E, H, G).

R4 is not in BCNF (the only candidate key for R4 is BG and B→EH violates BCNF). For B→EH, B→BEGH is not in F^+, therefore we break R4 into R5(B, E, H) and R6(B, G).

The final decomposition is: R1(A, H, C); R3(E, A, D); R5(B, E, H); R6(B, G).