Functional Dependencies and Boyce-Codd Normal Form (BCNF) Decomposition

Let X and Y be two sets of attributes. A functional dependency X→Y holds if rows that have the same values of X also have the same values of Y.

Example: Consider a table with addresses and suppose that postal_code→city. What does that mean? What about city→postal_code?

Note: X is a key if X→all the attributes of a table

By the decomposition rule, if we know that A→BC, then A→B and A→C are both true. Also, A→A (by the reflexivity rule), so instead of saying A→ABC, we can say A→BC.

Recall: an FD X→Y is trivial if Y is a subset of X. For example, A→A or AB→A are trivial.

Boyce-Codd Normal Form (BCNF)

A table T is in BCNF if for all *nontrivial* functional dependencies X→A that hold on T, X contains a key of T

Algorithm for BCNF decomposition

Input: Table T, Set F of all non-trivial functional dependencies that hold on T

While there remains at a least one FD in F, call it X→Y, where X does not contain a key
- Let B be the table containing X and Y
- Decompose B into two tables:
  - one with just X and Y
  - one with all the columns of B except Y

Example

T(ABCDE)
F = { AE->BCD, A->B, B->A, DE->C }
Decompose T into BCNF. Identify the candidate keys of the decomposed tables and prove that your decomposed schema is in BCNF.

Solution:

First, note that the AE is a key of T. So, AE->BCD does not break BCNF.
Next, note that A->B breaks BCNF.
Decompose T into T1(AB), with keys A and B, and T2(ACDE) with key AE
In the second iteration of the algorithm, note that DE->C breaks BCNF
Decompose T2 into T2a(CDE), with key DE, and T2b(ADE), with key AE

Final answer: T(AB), keys A or B, T2A(CDE) key DE, T2B(ADE) key AE
In BCNF because all the dependencies in F now contain a key on the left-hand side.

Practice questions

1. Let T be a table with four attributes: ABCD. Assume F = { B→ACD, C→D, C→A }. Decompose T into BCNF, identify the keys of the decomposed tables, and prove that your decomposed schema is in BCNF.

2. Now assume F = { BD→AC, B→C, D→A }. Decompose T into BCNF, identify the keys of the decomposed tables, and prove that your decomposed schema is in BCNF.