Functional Dependencies and Normalization

Definition 1 Let R be a relation scheme and let $X \subseteq R$ and $Y \subseteq R$. We say that a relation instance $r(R)$ *satisfies* a functional dependency $X \to Y$ if for every pair of tuples $t_1 \in r$ and $t_2 \in r$, if $t_1[X] = t_2[X]$ then $t_1[Y] = t_2[Y]$.

Definition 2 An instance r of relation scheme R is called a *legal* instance if it is a true reflection of the mini-world facts it represents (i.e. it satisfies all constraints imposed on it in the real world).

Definition 3 Let R be a relation scheme and let $X \subseteq R$ and $Y \subseteq R$. Then $X \to Y$, a *functional dependency* on scheme R, is *valid* if every legal instance $r(R)$ satisfies $X \to Y$.

Mathematical Properties of Functional Dependencies

Definition 4 Let F be a set of FDs on scheme R and f be another FD on R. Then, F implies f, denoted by $F \models f$, if every relation instance $r(R)$ that satisfies all FDs in F also satisfies f.

Definition 5 Let F be a set of FDs on scheme R. Then, the *closure* of F, denoted by F^+, is the set of all FDs implied by F.

Definition 6 Let F and G be sets of FDs on scheme R. Then, F and G are equivalent, denoted by $F \equiv G$, if $F \models G$ and $G \models F$.

Inference rules for FDs (Armstrong's Axioms)

Reflexivity: If $Y \subseteq X$ then, $X \to Y$. Such FDs are called *trivial* FDs.

Augmentation: If $X \to Y$, then $XZ \to YZ$.

Transitivity: If $X \to Y$ and $Y \to Z$, then $X \to Z$.

Definition 7 Let F be a set of FDs on scheme R and f be another FD on R. Then, a derivation for f from F is a sequence of FDs with f as the last FD in the sequence such that any element in the sequence is either an element of F or is an FD produced by the application of one of the Armstrong's axioms to FDs earlier in the sequence.

Definition 8 Let F be a set of FDs on scheme R and f be another FD on R. Then, F derives f, denoted by $F \vdash f$, if there is a derivation for f using only Armstrong's axioms.

Theorem 1 *Armstrong's axioms are sound and complete, i.e.* $F \models f$ *if and only if* $F \vdash f$*.*

Additional Rules of Inference:

Union: if $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$.
Proof: Using Armstrong's Axioms:

1. $X \rightarrow Y$, Given
2. $X \rightarrow Z$, Given
3. $X \rightarrow XZ$, Augment 2 by X
4. $XZ \rightarrow YZ$, Augment 1 by Z
5. $X \rightarrow YZ$, Transitivity using 3 and 4.

Decomposition: if $X \rightarrow YZ$ then $X \rightarrow Y$ and $X \rightarrow Z$.
Proof: Using Armstrong's Axioms:

1. $X \rightarrow YZ$, Given
2. $YZ \rightarrow Y$, Reflexivity
3. $X \rightarrow Y$, Transitivity on 1 and 2.

Similar proof for $X \rightarrow Z$.

Definition 9 Let $X \subseteq R$ be a set of attributes and F be a set of FDs that hold on R. Then

$$X_F^+ = \{A | F \models X \rightarrow A\}$$

Algorithm to calculate X_F^+:

```
Xplus := X;
repeat
  oldXplus := Xplus;
  for each FD Y --> Z in F
    if Y is a subset of Xplus then
      Xplus := Xplus union Z
until (Xplus = oldXplus)
output Xplus
```

Testing implication and equivalence of FDs

The X_F^+ algorithm can be used to check if F implies $X \rightarrow Y$. If $Y \subseteq X_F^+$ then F implies $X \rightarrow Y$.

The above test can be extended to check if $F \vdash G$. For each FD $X \rightarrow Y \in G$, if $Y \subseteq X_F^+$, then $F \vdash G$.

Equivalence of F and G can also be checked similarly. For each FD $X \rightarrow Y \in G$, if $Y \subseteq X_F^+$ and for each FD $X \rightarrow Y \in F$, if $Y \subseteq X_G^+$ then F and G are equivalent.

Given R and F, $K \subseteq R$ is a superkey for R if $K \rightarrow R$ holds.

Given, R and F, one can compute all candidate keys for R by exhaustively checking for all subsets of R starting from single attribute subsets. As soon as a candidate key is identifed, all its supersets need not be checked for key property as they will be superkeys. Note: If some of the attributes of R are absent from the right hand sides of FDs in F, these must be part of candidate keys. In this case, the search for candidate keys must start with the missing attributes from the right hand side of FDs. If the missing set of attributes forms a candidate key then this must be the only candidate key.

Definition 10 Let F be a set of FDs. A *Minimal Cover* of F is a set of FDs G that has the following properties:

1. G is equivalent to F.
2. All FDs in G have the form $X \rightarrow A$, where A is a single attribute.
3. It is not possible to make G "smaller" (and still satisfy the above two properties) by
   (a) Deleting a FD. i.e. $G - \{X \rightarrow A\} \not\equiv G$, for any FD $X \rightarrow A \in G$.
   (b) Deleting an attribute from the left hand side of a FD. i.e. $G - \{XA \rightarrow B\} + \{X \rightarrow B\} \not\equiv G$, for any FD $XA \rightarrow B \in G$.

Algorithm to find minimal cover for a set of FDs F

```
Step 1: Let G be the set of FDs obtained from F by decomposing the
right hand sides of each FD to a single attribute.

Step 2: Remove all redundant attributes from the left hand sides of FDs in G.

Step 3: From the resulting set of FDs, remove all redundant FDs.

Output the resulting set of FDs.
```

Example: Consider R = ABCDEFGH and the following set of FDs, F:
$ABH \rightarrow C$
$A \rightarrow D$
$C \rightarrow E$
$BGH \rightarrow F$
$F \rightarrow AD$
$E \rightarrow F$
$BH \rightarrow E$

Converting right hand sides to single attributes, we get:

$ABH \rightarrow C$
$A \rightarrow D$

$C \rightarrow E$
$BGH \rightarrow F$
$F \rightarrow A$
$F \rightarrow D$
$E \rightarrow F$
$BH \rightarrow E$

Perform steps 2 and 3....

Definition 11 A relation scheme R, F is said to be in 3NF (3rd Normal Form) if for every FD $X \rightarrow A$ in F^+ one of the following holds:

1. $A \in X$, i.e. the FD is trivial, or
2. X is a super key for R, or
3. A belongs to some candidate key for R.

Definition 12 A relation scheme R, F is said to be in BCNF (Boyce Codd Normal Form) if for every FD $X \rightarrow A$ in F^+ one of the following holds:

1. $A \in X$, i.e. the FD is trivial, or
2. X is a super key for R.

Definition 13 Let R be a relation scheme. Then, $(R_1, \ldots, R_n)$ is a *decomposition* of R if each R_i is a subset of R and $R_1 \cup \ldots \cup R_n = R$.

Definition 14 A decomposition $\rho = \{R_1, \ldots, R_n\}$ of scheme R with FDs F is a *Lossless Join Decomposition* if for every $r(R)$,

$$r = \pi_{R_1}(r) \bowtie \cdots \bowtie \pi_{R_n}(r).$$

Definition 15 A decomposition $\rho = \{R_1, \ldots, R_n\}$ of scheme R with FDs F is a *FD-Preserving Decomposition* if

$$F \equiv F_1 \cup \ldots \cup F_n$$

where $F_i = \{X \rightarrow A | X \rightarrow A \in F^+ \text{ and } XA \subseteq R_i\}$.

3NF synthesis algorithm
Input: R and F
Output: A lossless join and FD-preserving 3NF decomposition of R
Method:

1. Calculate the minimal cover of F; call it G (Combine all FDs with same LHS into one FD using "union" rule of inference). Also compute the candidate keys for R.

2. For each FD $X \rightarrow Y$ in G, generate a relation scheme XY in the decomposition.

3. If there are some attributes, say Z, of R that do not appear in any decomposed scheme, then create a separate scheme in the decomposition for Z.

4. If none of the decomposed schemes contain a candidate key, create a separate scheme in the decomposition for one of the candidate keys K.

BCNF Decomposition algorithm; call the function bcnf
Input: R and F
Output: A lossless join BCNF decomposition of R
Method:

1. Identify a functional dependency $X \rightarrow Y$ in F that violates the BCNF condition; return R if none found.

2. return $\{XY\} \cup bcnf(\text{R - Y}, \pi_{R-Y}(F))$

where $\pi_{R-Y}(F) = \{X \rightarrow Y | X \rightarrow Y \in F^+ \text{ and } XY \subseteq R - Y\}$.