Chapter 15.1

CS-6360 Database Design

Chris Irwin Davis, Ph.D.

Email: chrisirwindavis@utdallas.edu
Phone: (972) 883-3574
Office: ECSS 4.705

Notation

- In the following discussion, we use an abbreviated notation when discussing functional dependencies.
- We concatenate attribute variables and drop the commas for convenience.
- Hence,
  - the FD $\{X,Y\} \rightarrow \{Z\}$ is abbreviated to $XY \rightarrow Z$, and
  - the FD $\{X, Y, Z\} \rightarrow \{U, V\}$ is abbreviated to $XYZ \rightarrow UV$.

Inference Rules

- IR1 (reflexive rule):
  - If $X \supseteq Y$, then $X \rightarrow Y$.
- IR2 (augmentation rule):
  - $\{X \rightarrow Y\} \models XZ \rightarrow YZ$.
- IR3 (transitive rule):
  - $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$.

Inference Rules

- IR4 (decomposition, or projective, rule):
  - $\{X \rightarrow YZ\} \models X \rightarrow Y \; \underline{and} \; X \rightarrow Z$
- IR5 (union, or additive, rule):
  - $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ.$
- IR6 (pseudo-transitive rule):
  - $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z.$

Closure

- **Definition**. For each such set of attributes X, we determine the set $X+$ of attributes that are functionally determined by X based on F;
  - $X+$ is called the *closure* of X under F.
  - Algorithm 16.1 can be used to calculate $X+$.

Algorithm 15.1

Input: A set F of FDs on a relation schema R, and a set of attributes X, which is a subset of R.

$X^+ := X$;
repeat
    old$X^+ := X^+$;
    for each functional dependency $Y \rightarrow Z$ in F do
        if $X^+ \supseteq Y$ then $X^+ := X^+ \cup Z$;
until ($X^+ =$ oldX^+);

Coverage

- **Definition**. A set of functional dependencies F is said to **cover** another set of functional dependencies E if every FD in E is also in $F+$
  - That is, if every dependency in E can be inferred from F;
  - Alternatively, we can say that E is covered by F.
- **Definition**. Two sets of functional dependencies E and F are **equivalent** if $E+ = F+$. Therefore, equivalence means that every FD in E can be inferred from F, and every FD in F can be inferred from E
  - that is, E is equivalent to F if both the conditions—E covers F and F covers E—hold.

Minimal Set of Functional Dependencies

- Informally, a **minimal cover** of a set of functional dependencies E is a set of functional dependencies F that satisfies the property that every dependency in E is in the closure $F+$ of F.
- In addition, this property is lost if any dependency from the set F is removed
  - F must have no redundancies in it, and the dependencies in F are in a standard form.

Minimal Cover

- To satisfy these properties, we can formally define a set of functional dependencies F to be **minimal** if it satisfies the following conditions:
  - Every dependency in F has a single attribute for its right-hand side.
  - We cannot replace any dependency $X \longrightarrow A$ in F with a dependency $Y \longrightarrow A$, where Y is a proper subset of X, and still have a set of dependencies that is equivalent to F.
  - We cannot remove any dependency from F and still have a set of dependencies that is equivalent to F.

Minimal Cover (cont'd)

- We can think of a minimal set of dependencies as being a set of dependencies in a standard or canonical form and with no redundancies.
- Condition 1 just represents every dependency in a canonical form with a single attribute on the right-hand side.
- Conditions 2 and 3 ensure that there are no redundancies in the dependencies either by:
  - having redundant attributes on the left-hand side of a dependency (Condition 2), or
  - having a dependency that can be inferred from the remaining FDs in F (Condition 3).

Minimal Cover

- **Definition**. A minimal cover of a set of functional dependencies *E* is a minimal set of dependencies (in the standard canonical form and without redundancy) that is equivalent to *E*. We can always find at least one minimal cover *F* for any set of dependencies *E* using Algorithm 15.2.

Input: A set of functional dependencies E.

1. Set $F := E$.
2. Replace each functional dependency $X \to \{A_1, A_2, \ldots, A_n\}$ in F by the n functional dependencies $X \to A_1, X \to A_2, \ldots, X \to A_n$.
3. For each functional dependency $X \to A$ in F
   for each attribute B that is an element of X
   if $\{ \{F - \{X \to A\} \} \cup \{ (X - \{B\}) \to A\} \}$ is equivalent to F
   then replace $X \to A$ with $(X - \{B\}) \to A$ in F.
4. For each remaining functional dependency $X \to A$ in F
   if $\{F - \{X \to A\} \}$ is equivalent to F,
   then remove $X \to A$ from F.

Algorithm 15.2

- Let the given set of FDs be $E : \{B \longrightarrow A, D \longrightarrow A, AB \longrightarrow D\}$.
- We have to find the **minimal cover** of E.
  - All above dependencies are in canonical form (that is, they have only one attribute on the right-hand side), so we have completed step 1 of Algorithm 16.2 and can proceed to step 2.
  - In step 2 we need to determine if $AB \longrightarrow D$ has any redundant attribute on the left-hand side; that is, can it be replaced by $B \longrightarrow D$ or $A \longrightarrow D$?

- Since $B \longrightarrow A$, by augmenting with B on both sides (IR2), we have $BB \longrightarrow AB$, or $B \longrightarrow AB$ (i). However, $AB \longrightarrow D$ as given (ii).
- Hence by the transitive rule (IR3), we get from (i) and (ii), $B \longrightarrow D$. Thus $AB \longrightarrow D$ may be replaced by $B \longrightarrow D$.
- We now have a set equivalent to original E, say E': $\{B \longrightarrow A, D \longrightarrow A, B \longrightarrow D\}$. No further reduction is possible in step 2 since all FDs have a single attribute on the left-hand side.
- In step 3 we look for a redundant FD in E′. By using the transitive rule on $B \longrightarrow D$ and $D \longrightarrow A$, we derive $B \longrightarrow A$. Hence $B \longrightarrow A$ is redundant in E' and can be eliminated.
- Therefore, the minimal cover of E is $\{B \longrightarrow D, D \longrightarrow A\}$.