Covers for Functional Dependencies

CIS-611. Victor Matos


Agenda

- 1. Covers and Equivalence
- 2. Non-redundant covers
- 3. Extraneous attributes
- 4. Canonical covers
- 5. Minimum covers
Introduction

- In this section we will explore ways of representing FDs in a concise mode
- Reducing the size of a set of FDs benefits the computation of closures and other related algorithms
- Reduced sets of FDs help ensure database consistency and correctness

Overview

- The overall strategy is to minimize the set of FDs for a given relational schema in order to
  - (1) save storage and,
  - (2) reduce the number of compliance-tests performed when the database records are modified
- Optimization is done by throwing away unwanted rules and unneeded attributes
Example

Consider the database schema $R(ABC)$ under the set $F_1 = \{A \rightarrow B, B \rightarrow C, A \rightarrow C, AB \rightarrow C, A \rightarrow BC\}$

Again, the same database schema $R(ABC)$ but the set $F_2 = \{A \rightarrow B, B \rightarrow C\}$

It is easy to see that $F_1$ and $F_2$ are equivalent.

Using the more compact set $F_2$ will provide a better representation of the semantic of data in $R(ABC)$ as well as better performance/storage.

1. Covers and Equivalences

- Two sets of FDs $F_1$ and $F_2$ over scheme $R$ are equivalent, written $F_1 \equiv F_2$ if $F_1^+ = F_2^+$
- If $F_1 \equiv F_2$ then $F_1$ is a cover for $F_2$

Example: The sets

$F_1 = \{AB \rightarrow C, A \rightarrow D, CD \rightarrow E\}$ and
$F_2 = \{A \rightarrow BCE, A \rightarrow ABC, CD \rightarrow E\}$

are NOT equivalent
Example

Consider the schema R(ABCDE) and the sets of FDs $F_1$, $F_2$ and $F_3$ listed below

$F_1 = \{ A \rightarrow BC, A \rightarrow D, CD \rightarrow E \}$

$F_2 = \{ A \rightarrow BCE, A \rightarrow ABD, CD \rightarrow E \}$

$F_3 = \{ A \rightarrow BCDE \}$

By observation of the sets we conclude that:

- $F_1$ and $F_2$ are equivalent ($A^+ = ABCDE$, $CD^+ = CDE$ in both sets)
- $F_1$ and $F_3$ are not equivalent ($F_1 \geq F_3$, but $F_3$ cannot imply $CD \rightarrow E$)

Observation

A mechanical way of testing if $F_1 \equiv F_2$ is to simply compare $F_1^+$ and $F_2^+$

*However:* Computing the closure $F^+$ of a set of FDs is time consuming.

*We need a ‘better’ way of working (other than the brute-force generation of all the possible derivation sequences)*
**Member Function**

Determine if a single FD $X \rightarrow Y$ is derived from a set $F$ of FDs

**Input:** A set of FDs and an FD $X \rightarrow Y$

**Output:** true if $F \Rightarrow X \rightarrow Y$, false otherwise

```
Member(F, X → Y)
begin
  if ($X \subseteq F^+$) then return (true);
  else return (false);
end
```

**Lemma**

Given sets $F_1$ and $F_2$ over schema $R$, $F_1 \equiv F_2$ if and only if $F_1 \Rightarrow F_2$ and $F_2 \Rightarrow F_1$

**Function** DERIVES ( $F_1$, $F_2$)

**Input:** Two sets of FDs $F_1$ and $F_2$

**Output:** true if $F_1 \Rightarrow F_2$, false otherwise

```
begin
  v = true;
  for each FD $X \rightarrow Y$ in $F_2$ do
    v = v and Member ( $X \rightarrow Y$, $F_1$ );
  return ( v );
end
```
How to check that two sets of dependencies are equivalent?

Input: Two sets of FDs $F_1$ and $F_2$
Output: true if $F_1 \equiv F_2$, false otherwise

Function EQUIV ($F_1$, $F_2$)

$$v = \text{DERIVES (} F_1, F_2 \text{) and DERIVES (} F_2, F_1 \text{);}$$

return ($v$);

end

---

How to check that two sets of dependencies are equivalent?

Example: Consider the schema $R(ABCDE)$ and the sets of FDs $F_1$ and $F_2$ listed below

$F_1 = \{A \rightarrow B, B \rightarrow C, A \rightarrow C, AB \rightarrow C, A \rightarrow BC\}$
$F_2 = \{A \rightarrow B, B \rightarrow C\}$

Observations:

1. Each dependency in $F_2$ is in $F_1$, therefore $F_1$ DERIVES $F_2$.
2.1 Rule $A \rightarrow C$ in $F_1$ is a member of $F_2$.
2.2 Rule $AB \rightarrow C$ in $F_1$ is a member of $F_2$.
2.3 Rule $A \rightarrow BC$ in $F_1$ is a member of $F_2$.
2.4 Therefore all dependencies in $F_1$ are DERIVED from $F_2$.
3. According to the EQUIV function $F_1$ and $F_2$ are equivalent.
A set $F_1$ of FDs is non-redundant if there is no proper sub-set $F'$ of $F_1$ such that $F_1 \equiv F'$.

If such an $F'$ exists then $F_1$ is redundant.

$F_1$ is a non-redundant cover for $F_2$ if
1. $F_1$ is a cover for $F_2$ ($F_1 \equiv F_2$), and
2. $F_1$ is a non-redundant set of FDs.

**Example.** Universe is $\{ A, B, C \}$. There are three set of FDs

- $F_1 = \{ AB \rightarrow C, A \rightarrow B, B \rightarrow C, A \rightarrow C \}$
- $F_2 = \{ AB \rightarrow C, A \rightarrow B, B \rightarrow C \}$
- $F_3 = \{ A \rightarrow B, B \rightarrow C \}$

**Observations**
1. $F_1$ is a redundant cover for $F_2$
2. $F_3$ is a non redundant cover for $F_2$
1. Testing Redundancy
How to eliminate a rule X→Y from a set of FDs?

EXAMPLE:
Let R(ABC) and F= { A → B, B → A, B → C, A → C }

Observe that rule A → C is redundant. It could be obtained by applying transitivity on A → B and B → C.

Therefore F2 = { A → B, B → A, B → C } is a cover for F (it also happens to be non-redundant).

2. Testing Redundancy
How to eliminate a rule X→Y from a set of FDs?

ALGORITHM NonRedundant (F_1)
Input: An arbitrary set F_1 of FDs
Output: A non-redundant cover F_2 for F_1
begin
  F_2 = F_1;
  for each FD X→Y in F_1 do
    if Member(X → Y, F_2 - (X → Y) )
      then F_2 = F_2 - ( X → Y );
  return ( F_2 );
end;
3. Testing Redundancy
How to eliminate a rule $X \rightarrow Y$ from a set of FDs?

**EXAMPLE:**
Let $R(ABC)$ and $F = \{ A \rightarrow B, B \rightarrow A, B \rightarrow C, A \rightarrow C \}$

After applying the previous function we obtain:

$$F_2 = F - \{ A \rightarrow C \} = \{ A \rightarrow B, B \rightarrow A, B \rightarrow C \}$$

is a non-redundant cover for $F$.

---

Extraneous Attributes
Our intention is to discard from $F$ *unnecessary* attributes appearing in individual FDs without changing the closure of $F$.

Let $F$ be a set of FDs over schema $R$, and let $X \rightarrow Y$ be an FD in $F$. Attribute $A$ in $R$ is extraneous in the rule $X \rightarrow Y$ if either

**LEFT.** Assume $X = A Z$, $X \neq Z$ and

$$F \equiv ( F - \{ X \rightarrow Y \} ) \cup \{ Z \rightarrow Y \}$$

**RIGHT.** Assume $Y = AW$, $Y \neq W$ and

$$F \equiv ( F - \{ X \rightarrow Y \} ) \cup \{ X \rightarrow W \}$$
Extraneous Attributes

Example
Consider schema R = {A, B, C, D} and FDs
F = {A → BC, B → C, AB → D}

Left Reduction. We look for FDs having more than one attribute on the LHS. The only candidate is: AB → D
(1). Extract A from AB → D
(2). If (1) didn’t work try taking B from AB → D

Right Reduction. Look for FDs with more than one attribute on the RHS. The only candidate is: A → BC
(3). Try removing B from A → BC
(4). If (3) didn’t work try deleting C from A → BC

Algorithm: LeftReduction (G)

Input: A set G of FDs
Output: A left-reduced cover F for G

begin
    F=G;
    for each X → Y in G do
        for each attribute A in X do
            if Member (X-A → Y, F) then {
                Remove A from X in X → Y;
                Update F adding new rule X-A → Y;
            }
        return( F );
end;
Algorithm: RightReduction (G)

Input: A set G of FDs
Output: A right-reduced cover F for G

begin
  F=G;
  for each X → Y in G do
    for each attribute A in Y do
      if Member(X → Y, F-{X → Y} U {X → Y-A} ) {  
        Remove A from Y in X → Y;
        Update F adding new rule X → Y-A  
      }
    return( F );
  end;
end;

Canonical Cover
Constructing a ‘better’ representation of F

Consider the relation schema R subject to F

1. Create set F’ by removing extraneous attributes in the following order: first do left then right reductions

2. Remove redundant functional dependencies.

3. The new set (R,F’ ) is called a Canonical Cover for (R, F)
**Canonical Covers**

**Example** Let $r(ABCDIJM)$ be subject to the FDs

$F = \{ A \rightarrow C, AC \rightarrow J, AB \rightarrow DE, AB \rightarrow CDI, C \rightarrow M, A \rightarrow M \}$

(a) **Left reduction**

1. Test rule $AC \rightarrow J$. Drop $A$, to make $C \rightarrow J$. New set $F_2$ is not equivalent to $F$ (Member $(AC \rightarrow J, F_2) = false$). Therefore $A$ can not be eliminated.

2. Try dropping $C$ from $AC \rightarrow J$. Make $F_2 = F - \{AC \rightarrow J\} + \{A \rightarrow C\}$. $F_2$ is equivalent to $F$ (Member $(AC+ J, F_2) = true$). Consequently $C$ can be eliminated from $AC \rightarrow J$.

(b) **Right reduction.**

1. Attribute $D$ in $AB \rightarrow DE$ is redundant, and

2. Attribute $C$ in $AB \rightarrow CDI$ is redundant, eliminate both.

(c) **Redundant Rules.** Eliminate rule $A \rightarrow M$

(d) **Result:** $F = \{ A \rightarrow CI, AB \rightarrow EDI, C \rightarrow M \}$

**Canonical Covers**

*Your turn ...*

**Example.** Let $r(ABC)$ be subject to the FDs

$F = \{ A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C \}$

(a) **Left reduction.**

(b) **Right reduction.**

(c) **Redundant Rules.**

(d) **Result ?**
The Structure of Non-Redundant Covers

Observation: A schema R could have more than one non-redundant cover. Our next goal is to explore the similarities between them.

Definition

Two sets of attributes X and Y are equivalent under FDs F, written \( X \leftrightarrow Y \), if \( F \Rightarrow X \Rightarrow Y \) and \( F \Rightarrow Y \Rightarrow X \).

Lemma. Given equivalent, non-redundant covers F and G, for every left side X of an FD in F there is an equivalent left side V of an FD in G.

Example:

Let \( R(ABCDE) \) and FD sets

\[
F = \{ A \rightarrow BC, B \rightarrow A, AD \rightarrow E \} \quad \text{and} \quad G = \{ A \rightarrow ABC, B \rightarrow A, BD \rightarrow E \}
\]

Notice that

\[
E_F(A) = \{ A \rightarrow BC, B \rightarrow A \} \quad \text{(observe that } A_F^+ = A_G^+ = ABC) \\
E_F(B) = \{ A \rightarrow BC, B \rightarrow A \} \\
E_F(AD) = \{ AD \rightarrow E \} \quad \text{(observe } AD_F^+ = ABCDE) \\
E_G(A) = \{ A \rightarrow BC, B \rightarrow A \} \quad \text{(observe that } A_G^+ = A_G^+ = ABC) \\
E_G(B) = \{ A \rightarrow BC, B \rightarrow A \} \\
E_G(BD) = \{ BD \rightarrow E \} \quad \text{(observe } BD_G^+ = ABCDE)
\]

F and G are non-redundant and equivalent to each other (non redundant in the sense that rules can not be removed, however there are superfluous attributes in G).

Observe that: \( A_F \leftrightarrow A_G, B_F \leftrightarrow B_G \) and \( AD_F \leftrightarrow BD_G \)
The Structure of Non-Redundant Covers

Definition. Equivalent Classes

For a set of FDs $F$ over scheme $R$ and a set $X \subseteq R$, let $E_F(X)$ be the set of FDs in $F$ with left sides equivalent to $X$.

Let $E_F$ be the set $\{E_F(X) / X \subseteq R \text{ and } E_F(X) \neq \emptyset\}$

$E_F(X)$ is empty when no left side of any FD in $F$ is equivalent to $X$. $E_F$ is always a partition of $F$.

Example: Consider the previous example. $F = \{A \rightarrow BC, B \rightarrow A, AD \rightarrow E\}$ and $G = \{A \rightarrow ABC, B \rightarrow A, BD \rightarrow E\}$

<table>
<thead>
<tr>
<th>$E_F$</th>
<th>$E_G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_F(A) = {A \rightarrow BC, B \rightarrow A}$</td>
<td>$E_G(A) = {A \rightarrow ABC, B \rightarrow A}$</td>
</tr>
<tr>
<td>$E_F(B) = E_F(A)$</td>
<td>$E_G(B) = E_G(A)$</td>
</tr>
<tr>
<td>$E_F(AD) = {AD \rightarrow E}$</td>
<td>$E_G(BD) = {BD \rightarrow E}$</td>
</tr>
</tbody>
</table>

The Structure of Non-Redundant Covers

Example: Consider the non-redundant set of FDs $F = \{A \rightarrow BC, B \rightarrow A, AD \rightarrow E, BD \rightarrow I\}$

Let’s find all the equivalent classes of $F$.

Find closures of left-hand-sides:

- $A^+ = ABC$
- $B^+ = ABC \quad \text{same, therefore } E_F(A) = E_F(B) = \{A \rightarrow BC, B \rightarrow A\}$
- $(AD)^+ = ABCDEI$
- $(BD)^+ = ABCDEI \quad \text{same } E_F(AD) = E_F(BD) = \{AD \rightarrow E, BD \rightarrow I\}$

Therefore $E_F = \{E_F(A), E_F(AD)\}$
Observation. A non-redundant cover of a set G of FDs does not necessarily have as few FDs as any cover for G.

Definition. A set of FDs F is minimum if F has as few FDs as any equivalent set of FDs.

A minimum set of FDs is also non-redundant.

Example.
The set G is non-redundant but not minimum, since F is equivalent to G but has fewer FDs.

F is a minimum cover for G.
(observe that G is already a Canonical cover)

G = \{ A→BC, B→A, AD→E, BD→I \}

F = \{ A→BC, B→A, AD→E I \}
Direct Determination

Unlike non-redundant covers, the definition of minimum covers provides no guide for finding minimum covers or even for testing minimality.

The next concept on *direct functional determination* gives us the means to compute minimum covers.

**Definition.** Given a set of FDs $G$, $X$ directly determines $Y$ under $G$, written $X \rightarrow\leftarrow Y$, if there is a non-redundant cover $F$ for $G$ with an $F$-based DDAG $H$ for $X \rightarrow Y$ such that $U(H) \cap E_F(X) = \emptyset$.

In other words, we can find a non-redundant cover $F$ for $G$ in which $X \rightarrow Y$ can be derived using only FDs in $\{ F - E_F(X) \}$.

Observe that $X \rightarrow X$ always holds, that $X \rightarrow Y$ implies $X \rightarrow Y$, and that $E_F(X)$ can be empty.

---

Re-phrasing

**Definition.** Given a set of FDs $G$, $X$ directly determines $Y$ under $G$, written $X \leftrightarrow Y$, if there is a non-redundant cover $F$ for $G$ with an $F$-based DDAG $H$ for $X \rightarrow Y$ such that $U(H) \cap E_F(X) = \emptyset$.

Equivalent to:

\[ X \leftrightarrow Y \text{ if } \{ F - E_F(X) \} \Rightarrow X \rightarrow Y \]
Example. Let G = F = { A→CD, AB→E, B → I, DI → J }. Then AB→J under G, as the DDAG H in next figure shows.

Here the ‘use set’ for the DDAG H drawn for AB→J is:
U(H) = { A→D, B → I, DI → J } and
Observe that
E_{F}(AB) = { AB → E }

Their intersection is empty, therefore AB → J

---

Computing Minimum Covers

Example. Let F = { A → BC, BC → A, AD→E, AD → I, E→B }.

Observe that BCD ↔ A D ( observe that BCD^{*}=AD^{*}=ABCDEI )
Figure below shows an F-based DDAG for BCD→AD that uses no FDs from E_{F}(AD) and no FDs in E_{F}(BCD).

U(H) = { BC → A }
E_{F}(BCD) = Ø
E_{F}(AD) = { AD→E, AD→I }

observe that U(H) ∩ E_{F}(BCD) = U(H) ∩ E_{F}(AD) =Ø
therefore BCD → AD
similarly AD → BCD
**Computing Minimum Covers**

**Theorem.** Let $F$ be a non-redundant set of FDs that is *not minimum.* There is some $E_F(X)$ containing distinct FDs $Y \rightarrow U$ and $Z \rightarrow V$ such that $Y \rightarrow Z$.

**Observation**
We know the set $F = \{ A \rightarrow BC, B \rightarrow A, AD \rightarrow E, BD \rightarrow I \}$ is non-redundant, and we have computed its equivalence classes already.

$$E_F(A) = E_F(B) = \{ A \rightarrow BC, B \rightarrow A \} \quad \text{and} \quad E_F(AD) = E_F(BD) = \{ AD \rightarrow E, BD \rightarrow I \}$$

Consider the set $E_F(BD)$. According to previous theorem $AD \rightarrow BD$ as well as $BD \rightarrow AD$. See the DDAG and its use set for $BD \rightarrow AD$

Use set $U(H) = \{ B \rightarrow A \}$ notice that $U(H) \cap E_F(BD) = \emptyset$
therefore $BD \rightarrow AD$

---

**Computing Minimum Covers**

**Observation** (*continuation...*)

Therefore the non-redundant cover

$$F = \{ A \rightarrow BC, B \rightarrow A, AD \rightarrow E, BD \rightarrow I \}$$

is *not* a minimum set.

As we will see it could be reduced to the smaller *minimum* cover

$$F_2 = \{ A \rightarrow BC, B \rightarrow A, BD \rightarrow EI \}$$
or equivalently

$$F_3 = \{ A \rightarrow BC, B \rightarrow A, AD \rightarrow EI \}$$
Computing Minimum Covers

Algorithm. MINIMIZE
Input: A set of FDs G.
Output: A minimum cover for G.

MINIMIZE(G)
begin
  F := NONREDUN(G);
  for each $E_F(X)$ do
    for each $Y \rightarrow M$ in $E_F(X)$ do
      for each $Z \rightarrow V \neq Y \rightarrow M$ in $E_F(X)$ do
        if DDERIVES($F$, $Y \rightarrow Z$) then
          replace $Y \rightarrow M$ and $Z \rightarrow V$ by $Z \rightarrow V$ in $F$,
      end.
  return($F$)
end.

MINIMIZE can be implemented to have time complexity $O(np)$ on inputs of length $n$ with $p$ FDs.

Example
Set $F = \{ A \rightarrow BC, B \rightarrow A, AD \rightarrow E, BD \rightarrow I \}$ from previous examples is known to be non-redundant, we also have computed its equivalence classes:

$E_F(A) = E_F(B) = \{ A \rightarrow BC, B \rightarrow A \}$ and $E_F(AD) = E_F(BD) = \{ AD \rightarrow E, BD \rightarrow I \}$

MINIMIZE algorithm will consider the set $E_F(A)$ and conclude that neither $A \rightarrow B$ nor $B \rightarrow A$ holds.

However, when working with the set $E_F(AD)$ the MINIMIZE algorithm will discover that $BD \rightarrow AD$ and $AD \rightarrow BD$. Therefore the rules $\{ AD \rightarrow E, BD \rightarrow I \}$ will be re-written as $\{ AD \rightarrow EI \}$.

This contraction will reduce the set $F$ to a minimum cover $F_2 = \{ A \rightarrow BC, B \rightarrow A, AD \rightarrow EI \}$
# Where to get more information?


