Closure of a Set of Attributes (II)

- Algorithm for calculating A^+ is based on the repeated use of the transitivity rule and has a runtime complexity that is quadratic in the size of F

  A^+ *CalculateAttributeClosure*(F, A)

  // Input: A set F of FDs over a relation schema R and a set $A \subseteq R$ of attributes

  // Output: The closure A^+ of attributes for which $A \rightarrow A^+$ holds

  $A^+ = A$; // due to reflexivity rule

  repeat

  $\quad OldA^+ = A^+$

  $\quad$ **for each** FD $B \rightarrow C \in F$ **do**

  $\quad\quad$ **if** $B \subseteq A^+$ **then** $A^+ := A^+ \cup \mathrm{C}$

  until $A^+ = OldA^+$

  return A^+

- Basic idea: $B \subseteq A^+$ means that $A \rightarrow B$. Using $B \rightarrow C$ and applying the transitivity rule gives us $A \rightarrow C$. Therefore, $C \subseteq A^+$ must hold.

Closure of a Set of Attributes (III)

- Example 1
  - Let $R(A, B, C, G, H, I)$ be a relation schema, and let $F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CI \rightarrow G\}$ be a set of FDs
  - Task: Compute AG^+
  - We use the algorithm *CalculateAttributeClosure* and set $AG^+ := AG$
  - In the loop we set $Old\_AG^+ := AG$ and check all FDs whether they can contribute to AG^+.
  - First we take $A \rightarrow B$ and check whether $A \subseteq AG^+$ holds. This is the case. Therefore, we set $AG^+ := AG^+ \cup B = ABG$ (due to transitivity)
  - Next, we take $A \rightarrow C$, and using the same argument as before, we obtain $AG^+ := AG^+ \cup C = ABCG$
  - Next, we take $CG \rightarrow H$ and find that $CG \subseteq AG^+$ holds so that we get $AG^+ := AG^+ \cup H = ABCGH$
  - Next we take $CI \rightarrow G$ and find that $CI \not\subseteq AG^+$ holds

Closure of a Set of Attributes (IV)

- Example 1 (*continued*)
  - Since $Old\_AG^+ \neq AG^+$ holds, we perform a second loop and set Old_AG^+ to AG^+, that is, $Old\_AG^+ := ABCGH$
  - We see soon that no FD from F can increase AG^+
  - Since $Old\_AG^+ = AG^+$ holds, the algorithm terminates, and we get $AG^+ := ABCGH$

- Easy method to check whether $A \subseteq R$ is a superkey
  - A is a superkey if $A \rightarrow R$
  - Therefore
    - Compute A^+
    - Check whether $A^+ = R$ holds
    - If yes, then A is a superkey; otherwise, it is not
  - In the example above, AG is not a superkey since the attribute I cannot be reached

Closure of a Set of Attributes (V)

- Example 2
  - We look at an earlier example again: Given the schema $R(A, B, C)$ and the set $F = \{A \rightarrow B, B \rightarrow C\}$ on R, determine the closure F^+
  - Determine the *power set* of ABC, i.e. , the set of all sets that are subsets of ABC: $\{\emptyset, A, B, C, AB, AC, BC, ABC\}$
  - Compute the attribute closures of all subsets except $\emptyset$
  - $A^+ = ABC$

    $\Rightarrow A \rightarrow A, A \rightarrow B, A \rightarrow C, A \rightarrow AB, A \rightarrow AC, A \rightarrow BC, A \rightarrow ABC$ [7 FDs]
  - $B^+ = BC$

    $\Rightarrow B \rightarrow B, B \rightarrow C, B \rightarrow BC$ [3 FDs]
  - $C^+ = C$

    $\Rightarrow C \rightarrow C$ [1 FD]
  - $AB^+ = ABC$

    $\Rightarrow AB \rightarrow A, AB \rightarrow B, AB \rightarrow C, AB \rightarrow AB, AB \rightarrow BC, AB \rightarrow AC,$
    $AB \rightarrow ABC$ [7 FDs]

Closure of a Set of Attributes (VI)

- Example 2 (*continued*)
  - $AC^+ = ABC$
    $\Rightarrow AC \rightarrow A$, $AC \rightarrow B$, $AC \rightarrow C$, $AC \rightarrow AB$, $AC \rightarrow AC$, $AC \rightarrow BC$,
    $AC \rightarrow ABC$ [7 FDs]
  - $BC^+ = BC$
    $\Rightarrow BC \rightarrow B$, $BC \rightarrow C$, $BC \rightarrow BC$ [3 FDs]
  - $ABC^+ = ABC$
    $\Rightarrow ABC \rightarrow A$, $ABC \rightarrow B$, $ABC \rightarrow C$, $ABC \rightarrow AB$, $ABC \rightarrow AC$, $ABC \rightarrow BC$,
    $ABC \rightarrow ABC$ [7 FDs]
  - Algorithm finds the same 35 FDs of F^+ as the exponential algorithm for computing F^+ before
  - Comments: No valid FD is found more than once, $2^{|R|} - 1$ sets to explore; for each set, $|F|$ subset tests are needed (cost of a subset test is not constant and implementation dependent); the decomposition of $2^{|R|} - 1$ attribute closures are needed; algorithm is expensive too but by far not as expensive as the previous one

Equivalence of Sets of Functional Dependencies (I)

- Assuming that two students independently determine the sets *F* and *G* of FDs respectively for the same schema *R*
- Question: How can they find out that both sets have the same meaning and are thus equivalent?
- Answer:

$$
\begin{tikzcd}
F \arrow[d] \arrow[r, phantom, "\equiv"] & G \arrow[d] \\
F^+ \arrow[r, phantom, "="] & G^+
\end{tikzcd}
$$

('≡' means "equivalent")

- Two sets *F* and *G* of FDs are equivalent if, and only if, $F^+ = G^+$ holds
- The definition of equivalence is convincing because the equality of the closures of *F* and *G* implies that the same FDs can be inferred from *F* and *G*

Equivalence of Sets of Functional Dependencies (II)

- Equivalence means that every FD in F can be inferred from G, and every FD in G can be inferred from F
- Example
  - Show that the two sets $F = \{A \to C, AC \to D, E \to AD, E \to H\}$ and $G = \{A \to CD, E \to AH\}$ are equivalent
  - We show first: Every FD in F can be inferred from G ("G covers F"), i.e., for each FD $X \to Y \in F$ we calculate X^+ with respect to G and then check whether $Y \subseteq X^+$ holds
    - F has the left-hand sides A, AC, and E
    - With respect to G we calculate A^+, AC^+, and E^+ and obtain $A^+ = ACD$, $AC^+ = ACD$, and $E^+ = ACDEH$
    - We check whether the right-hand sides of the FDs in F are in the respective attribute closures just computed for their left-hand sides:

      $A \to C : C \subseteq A^+$ holds $\qquad AC \to D : D \subseteq AC^+$ holds

      $E \to AD : AD \subseteq E^+$ holds $\qquad E \to H : H \subseteq E^+$ holds

Equivalence of Sets of Functional Dependencies (III)

- Example (*continued*)
  - We show second: Every FD in G can be inferred from F ("F covers G"), i.e., for each FD $X \rightarrow Y \in G$ we calculate X^+ with respect to F and then check whether $Y \subseteq X^+$ holds
    - G has the left-hand sides A and E
    - With respect to F we calculate A^+ and E^+ and obtain $A^+ = ACD$ and $E^+ = ACDEH$
    - We check whether the right-hand sides of the FDs in G are in the respective attribute closures just computed for their left-hand sides:

      $A \rightarrow CD : CD \subseteq A^+$ holds $\qquad E \rightarrow AH : AH \subseteq E^+$ holds
  - We obtain that F covers G and G covers F, i.e., F and G are equivalent
- Two sets F and G of FDs are equivalent if, and only if, F covers G and G covers F

Minimal Cover (I)

- Synonym: canonical cover
- Motivation
  - FDs in F are integrity constraints that a DBMS has to check with each insertion, update, or deletion for possible violations
  - Goal for performance reasons: Computation of a *minimal set* of FDs that are equivalent to F
  - Such a minimal set is called a minimal cover
- A minimal cover of a set F of FDs is a set F_c of FDs such that
  - F and F_c are equivalent
  - every FD in F_c has a single attribute on its right-hand side (standard form)
  - it is not possible to replace any FD $X \to A$ in F_c by an FD $Y \to A$ with $Y \subset X$ and still have a set of FDs that is equivalent to F_c
  - it is not possible to remove any FD from F_c and still have a set of FDs that is equivalent to F_c
- This representation of a minimal cover is the standard form or canonical form and without redundancies

Minimal Cover (II)

- The nonstandard form of a minimum cover makes use of the union rule and combines the FDs with the same left-hand side into a single FD
- Alternative, equivalent definition: F_c is called a minimal cover (nonstandard form) of a given set F of FDs if holds:
  - $F_c \equiv F$, i.e., $F_c^+ = F^+$
  - In F_c there are no FDs $A \rightarrow B$ where A or B contain extraneous attributes, i.e., they are reduced as much as possible.
    - We cannot omit any attribute on the *left* side of any FD; otherwise, we would change the semantics:

      $\forall\, a \in A : (F_c - \{A \rightarrow B\} \cup \{(A - \{a\}) \rightarrow B\})^+ \neq F_c^+$
    - We cannot omit any attribute on the *right* side of any FD, otherwise we would change the semantics:

      $\forall\, b \in B : (F_c - \{A \rightarrow B\} \cup \{A \rightarrow (B - \{b\})\})^+ \neq F_c^+$
  - Each left side of the FDs in F_c occurs only once, i.e.,

    $\forall\, f_1 = A \rightarrow B \in F_c\ \forall\, f_2 = C \rightarrow D \in F_c, f_1 \neq f_2 : A \neq C$ (nonstandard form)

Minimal Cover (III)

- Algorithm for computing a minimal cover

  F_c *CalculateMinimalCover*(F)
  // Input: A set F of FDs
  // Output: A minimal cover F_c

  // Step 1: Initialize F_c
  $F_c := F$

  // Step 2: Perform a left reduction of the FDs in F_c, i.e., identify and remove
  all attributes on the left-hand sides of FDs in F_c that are extraneous
  for each $A \rightarrow B \in F_c$ **do**
  for each $a \in A$ **do**
  if $A - \{a\} \neq \emptyset$ **and** $B \subseteq$ *CalculateAttributeClosure*(F_c, $A - \{a\}$) **then**
  $F_c := F_c - \{A \rightarrow B\} \cup \{(A - \{a\}) \rightarrow B\}$

Minimal Cover (IV)

- Algorithm for computing a minimal cover (*continued*)

// Step 3: Perform a right reduction of the remaining FDs in F_c, i.e., identify and remove all attributes on the right-hand sides of FDs in F_c that are extraneous

for each $A \rightarrow B \in F_c$ **do**

    for each $b \in B$ **do**

        if $b \in CalculateAttributeClosure(F_c - \{A \rightarrow B\} \cup \{A \rightarrow (B - \{b\})\}, A)$ **then**

            $F_c := F_c - \{A \rightarrow B\} \cup \{A \rightarrow (B - \{b\})\}$

// Step 4: Remove all FDs of the form $A \rightarrow \emptyset$ from F_c, which have perhaps been produced in the previous step, since they are meaningless

for each $A \rightarrow B \in F_c$ **do**

    if $B = \emptyset$ **then** $F_c := F_c - \{A \rightarrow \emptyset\}$

Minimal Cover (V)

- Algorithm for computing a minimal cover (*continued*)

  // Step 5a: If the goal is to obtain a minimal cover in *standard form*,
  decompose the right-hand sides of all FDs in F_c such that each
  FD in F_c has a single attribute on its right-hand side

  for each $A \rightarrow B \in F_c$ **do**

  if $B = \{b_1, \ldots, b_n\}$ **and** $n > 1$ **then**

  $F_c := F_c - \{A \rightarrow B\} \cup \{A \rightarrow \{b_1\}, \ldots, A \rightarrow \{b_n\}\}$

  return F_c

Minimal Cover (VI)

- Algorithm for computing a minimal cover (*continued*)

  // Step 5b: If the goal is to obtain a minimal cover in *nonstandard form*,
  apply the union rule to all FDs with equal left-hand sides

  $H := F_c$

  $F_c := \emptyset$

  for each $A \rightarrow B \in H$ **do**

  $G := \emptyset$ // FDs that have been processed and that have to be deleted from
  // H at the end of each loop

  $X := \emptyset$ // Union of all right-hand sides of FDs with A on their left-hand side

  for each $C \rightarrow D \in H$ **do**

  if $A = C$ **then**

  $G := G \cup \{C \rightarrow D\}$

  $X := X \cup D$

  $H := H - G$

  $F_c := F_c \cup \{A \rightarrow X\}$

  return F_c

Minimal Cover (VII)

- Example 1
  - Compute a minimum cover for the set $F = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$ of FDs on $R(A, B, D)$
  - Step 1
    - $F_c := \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$
  - Step 2
    - Only $AB \rightarrow D$ has more than one attribute on its left-hand side
    - To check whether B can be removed, we compute whether $D \subseteq \mathit{CalculateAttributeClosure}(F_c, A)$ holds
    - This is not the case since $A^+ = A$ and $D \not\subseteq A$ holds
    - To check whether A can be removed, we compute whether $D \subseteq \mathit{CalculateAttributeClosure}(F_c, B)$ holds
    - This is the case since $B^+ = ABD$ and $D \subseteq ABD$
    - Hence, A can be removed, and we obtain $F_c := \{B \rightarrow A, D \rightarrow A, B \rightarrow D\}$

Minimal Cover (VIII)

- Example 1 (*continued*)
  - Step 3
    - To check whether A can be removed from $B \rightarrow A$, we check whether $A \subseteq CalculateAttributeClosure(\{B \rightarrow \emptyset, D \rightarrow A, B \rightarrow D\}, B)$ holds
    - This is the case since $B^+ = ABD$ and $A \subseteq ABD$
    - Hence, A can be removed, and we obtain $F_c := \{B \rightarrow \emptyset, D \rightarrow A, B \rightarrow D\}$
    - To check whether A can be removed from $D \rightarrow A$, we check whether $A \subseteq CalculateAttributeClosure(\{B \rightarrow \emptyset, D \rightarrow \emptyset, B \rightarrow D\}, D)$ holds
    - This is not the case since $D^+ = D$ and $A \not\subset D$ holds
    - To check whether D can be removed from $B \rightarrow D$, we check whether $D \subseteq CalculateAttributeClosure(\{B \rightarrow \emptyset, D \rightarrow A, B \rightarrow \emptyset\}, B)$ holds
    - This is not the case since $B^+ = B$ and $D \not\subset B$ holds
    - After this step we have: $F_c := \{B \rightarrow \emptyset, D \rightarrow A, B \rightarrow D\}$
  - Step 4: We obtain $F_c := \{D \rightarrow A, B \rightarrow D\}$
  - Step 5a/5b: $F_c := \{D \rightarrow A, B \rightarrow D\}$ is in both forms

Minimal Cover (IX)

- Example 2
  - Compute a minimum cover for the set $F = \{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C\}$ of FDs on $R(A, B, C)$
  - Step 1
    - $F_c := \{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C\}$
  - Step 2
    - Only $AB \rightarrow C$ has more than one attribute on its left-hand side
    - To check whether A can be removed, we compute whether $C \subseteq CalculateAttributeClosure(F_c, B)$ holds
    - This is the case since $B^+ = BC$ and $C \subseteq BC$
    - Hence, A can be removed, and we obtain $F_c := \{A \rightarrow BC, B \rightarrow C, A \rightarrow B\}$
    - This also means that the number of FDs in F_c has been reduced by 1

Minimal Cover (X)

- Example 2 (*continued*)
  - We have so far: $F_c := \{A \rightarrow BC, B \rightarrow C, A \rightarrow B\}$
  - Step 3
    - To check whether C can be removed from $A \rightarrow BC$, we check whether $C \subseteq \mathit{CalculateAttributeClosure}(\{A \rightarrow B, B \rightarrow C\}, A)$ holds
    - This is the case since $A^+ = ABC$ and $C \subseteq ABC$
    - Hence, C can be removed, and we obtain $F_c := \{A \rightarrow B, B \rightarrow C\}$
    - To check whether B can be removed from $A \rightarrow B$, we check whether $B \subseteq \mathit{CalculateAttributeClosure}(\{A \rightarrow \emptyset, B \rightarrow C\}, A)$ holds
    - This is not the case since $A^+ = A$ and $B \not\subseteq A$ holds
    - To check whether C can be removed from $B \rightarrow C$, we check whether $C \subseteq \mathit{CalculateAttributeClosure}(\{A \rightarrow B, B \rightarrow \emptyset\}, B)$ holds
    - This is not the case since $B^+ = B$ and $C \not\subseteq B$ holds
    - After this step we have: $F_c := \{A \rightarrow B, B \rightarrow C\}$
  - Step 4: Nothing to do since there is no FD with an $\emptyset$ on its right-hand side
  - Step 5a/5b: $F_c := \{A \rightarrow B, B \rightarrow C\}$ is in both forms

Minimal Cover (XI)

- Example 3
  - This example shows that more than one minimal cover can exist for the *same* set F of FDs
    - The minimal covers computed for the same F of FDs depend on the order in which the FDs are processed
    - Different orders can lead to different minimal covers
    - However, the algorithm computes exactly one of them; they are all equivalent
  - Compute a minimum cover for the set $F = \{A \rightarrow BC, C \rightarrow AB, B \rightarrow AC\}$ of FDs on $R(A, B, C, D)$
  - Step 1
    - $F_c := \{A \rightarrow BC, C \rightarrow AB, B \rightarrow AC\}$
  - Step 2
    - There is no FD that has more than one attribute on its left-hand side
    - Therefore, nothing has to be done

Minimal Cover (XII)

- Example 3 (*continued*)
  - We have so far: $F_c := \{A \rightarrow BC, C \rightarrow AB, B \rightarrow AC\}$
  - Step 3
    - In $A \rightarrow BC$ both B and C are extraneous under F_c
      - C can be removed since $C \subseteq CalculateAttributeClosure(\{A \rightarrow B, C \rightarrow AB, B \rightarrow AC\}, A)$ holds: $A^+ = ABC$ and $C \subseteq ABC$
      - B can be removed since $B \subseteq CalculateAttributeClosure(\{A \rightarrow C, C \rightarrow AB, B \rightarrow AC\}, A)$ holds: $A^+ = ABC$ and $B \subseteq ABC$
    - We are not allowed to remove B and C at the same time since the algorithm picks one of the two and deletes it
    - Case 1: C is removed; we get $F_c^1 = \{A \rightarrow B, C \rightarrow AB, B \rightarrow AC\}$
      - B is now not extraneous in $A \rightarrow B$ since $A^+ = A$ under $\{A \rightarrow \emptyset, C \rightarrow AB, B \rightarrow AC\}$ holds and $B \not\subset A$ holds
      - Continuing the algorithm, we find that A and B are extraneous in the right-hand side of $C \rightarrow AB$ under F_c^1
      - B can be removed since $B \subseteq CalculateAttributeClosure(\{A \rightarrow B, C \rightarrow A, B \rightarrow AC\}, C)$ holds: $C^+ = ABC$ and $B \subseteq ABC$

Minimal Cover (XIII)

- Example 3 (*continued*)
  - Step 3 (*continued*)
    - A can be removed since $B \subseteq CalculateAttributeClosure(\{A \to B, C \to B, B \to AC\}, C)$ holds: $C^+ = ABC$ and $A \subseteq ABC$
    - Case 1.1: B is removed; we get $F_c^2 = \{A \to B, C \to A, B \to AC\}$
      - A is now not extraneous in $C \to A$ since $C^+ = C$ under $\{A \to B, C \to \emptyset, B \to AC\}$ holds and $A \not\subset C$ holds
      - C is not extraneous in $B \to AC$ since $B^+ = AB$ under $\{A \to B, C \to A, B \to A\}$ holds and $C \not\subset AB$ holds
      - A is extraneous in $B \to AC$ since $B^+ = ABC$ under $\{A \to B, C \to A, B \to C\}$ holds and $A \subseteq ABC$ holds
      - We get $F_c^3 = \{A \to B, C \to A, B \to C\}$
      - C is not extraneous in $B \to C$ since $B^+ = B$ under $\{A \to B, C \to A, B \to \emptyset\}$ holds and $C \not\subset B$ holds
      - The algorithm terminates, and we obtain the first minimal cover $F_{c1} = \{A \to B, C \to A, B \to C\}$

Minimal Cover (XIV)

- Example 3 (*continued*)
  - Step 3 (*continued*)
    - Case 1.2: A is removed; we get $F_c^4 = \{A \to B, C \to B, B \to AC\}$
      - B is now not extraneous in $C \to B$ since $C^+ = C$ under $\{A \to B, C \to \varnothing, B \to AC\}$ holds and $B \not\subset C$ holds
      - C is not extraneous in $B \to AC$ since $B^+ = AB$ under $\{A \to B, C \to B, B \to A\}$ holds and $C \not\subset AB$ holds
      - A is not extraneous in $B \to AC$ since $B^+ = BC$ under $\{A \to B, C \to B, B \to C\}$ holds and $A \not\subset BC$ holds
      - The algorithm terminates, and we obtain the second minimal cover $F_{c2} = \{A \to B, C \to B, B \to AC\}$
  - Case 2: B is removed; we get $F_c^5 = \{A \to C, C \to AB, B \to AC\}$
    - Similarly to case 1, we obtain two further minimal covers:
      - $F_{c3} = \{A \to C, C \to B, B \to A\}$
      - $F_{c4} = \{A \to C, C \to AB, B \to C\}$

Minimal Cover (XV)

- Example 3 (*continued*)
  - Step 3 (continued)
    - For $F = \{A \to BC, C \to AB, B \to AC\}$ we have detected the following four minimal covers:
      - $F_{c1} = \{A \to B, C \to A, B \to C\}$
      - $F_{c2} = \{A \to B, C \to B, B \to AC\}$
      - $F_{c3} = \{A \to C, C \to B, B \to A\}$
      - $F_{c4} = \{A \to C, C \to AB, B \to C\}$
    - Note that more minimal covers can be found for F
  - Step 4
    - There is no FD with an $\varnothing$ on its right-hand side
  - Step 5a
    - We have to modify F_{c2} and F_{c4} and obtain

      $F_{c2}' = \{A \to B, C \to B, B \to A, B \to C\}$

      $F_{c4}' = \{A \to C, C \to A, C \to B, B \to C\}$
  - Step 5b: F_{c1}, F_{c2}, F_{c3}, and F_{c4} are already in nonstandard form