Functional Dependencies- Examples

Dr. Bassam Hammo

Exercise #1: FD's From DB Instances

- Below is an instance of R(A1,A2,A3,A4). Choose the FD which may hold on R

1. A4 ➔ A1
2. A2A3 ➔ A4
3. A2A3 ➔ A1

A1	A2	A3	A4
1	2	3	4
1	2	3	5
6	7	8	2
2	1	3	4

Solution #1: FD's From DB Instances

1. A4 → A1 ???

- **Incorrect:** The 1st and 4th tuple violates it

2. A2A3 → A4 ???

- **Incorrect:** The1st and 2nd tuple violates it.

3. A2A3 → A1 ???

- Correct!

A1	A2	A3	A4
1	2	3	4
1	2	3	5
6	7	8	2
2	1	3	4

Uses of Attribute Closure

There are several uses of the attribute closure algorithm:

- Testing for superkey:
  - To test if α is a superkey, we compute α^+, and check if α^+ contains all attributes of *R*.
- Testing functional dependencies
  - To check if a functional dependency $\alpha \rightarrow \beta$ holds (or, in other words, is in F^+), just check if $\beta \subseteq \alpha^+$.
  - That is, we compute α^+ by using attribute closure, and then check if it contains β.
  - Is a simple and cheap test, and very useful

Uses of Attribute Closure

There are several uses of the attribute closure algorithm:

- Computing closure of F
  - For each $\gamma \subseteq R$, we find the closure γ^+, and for each $S \subseteq \gamma^+$, we output a functional dependency $\gamma \rightarrow S$.

Exercise #2: Checking if an FD Holds on F Using the Closure

- Let R(ABCDEFGH) satisfy the following functional dependencies: {A->B, CH->A, B->E, BD->C, EG->H, DE->F}
- Which of the following FD is also guaranteed to be satisfied by R?

1. BFG → AE
2. ACG → DH
3. CEG → AB

Hint: Compute the closure of the LHS of each FD that you get as a choice. If the RHS of the candidate FD is contained in the closure, then the candidate follows from the given FDs, otherwise not.

Solution #2: Checking if an FD Holds on F Using the Closure

- FDs: {A->B, CH->A, B->E, BD->C, EG->H, DE->F}

1. BFG → AE ???
   - **Incorrect:** BFG+ = BFGEH, which includes E, but not A
2. ACG → DH ???
   - **Incorrect:** ACG+ = ACGBE, which includes neither D nor H.
3. CEG → AB ???
   - **Correct:** CEG+ = CEGHAB, which contains AB

Exercise #3: Checking for Keys Using the Closure

- Which of the following could be a key for R(A,B,C,D,E,F,G) with functional dependencies {AB➔C, CD➔E, EF➔G, FG➔E, DE➔C, and BC➔A}

1. BDF
2. ACDF
3. ABDFG
4. BDFG

Solution #3: Checking for Keys Using the Closure

- {AB->C, CD->E, EF->G, FG->E, DE->C, and BC->A}

1. BDF ???

- No. $BDF^+ = BDF$

2. ACDF ???

- No. $ACDF^+ = ACDFEG$ (The closure does not include B)

3. ABDFG ???

- No. This choice is a superkey, but it has proper subsets that are also keys (e.g. $BDFG^+ = BDFGECA$)

Solution #3: Checking for Keys Using the Closure

- {AB->C, CD->E, EF->G, FG->E, DE->C, and BC->A}

4. BDFG ???

- $BDFG^+$ = ABCDEFG
- Check if any subset of BDFG is a key:
  - Since B, D, F never appear on the RHS of the FDs, they must form part of the key.
  - BDF^+ = BDF ← Not key
  - So, BDFG is the minimal key, hence the candidate key

Finding Keys using FDs

- **Tricks for finding the key:**
- If an attribute never appears on the *RHS* of any FD, it *must be part of the key*
- If an attribute never appears on the *LHS* of any FD, but appears on the *RHS* of any FD, it *must not be part of any key*

Exercise #4: Checking for Keys Using the Closure

Consider $R = \{A, B, C, D, E, F, G, H\}$ with a set of FDs

$F = \{CD \rightarrow A, EC \rightarrow H, GHB \rightarrow AB, C \rightarrow D, EG \rightarrow A,$

$H \rightarrow B, BE \rightarrow CD, EC \rightarrow B\}$

Find all the candidate keys of R

Solution #4: Checking for Keys Using the Closure

F = {CD→A, EC→H, GHB→AB, C→D, EG→A, H→B, BE→CD, EC→B}

- First, we notice that:
  - **EFG** never appear on RHS of any FD. So, **EFG** must be part of ANY key of R
  - **A** never appears on LHS of any FD, but appears on RHS of some FD. So, **A** is not part of ANY key of R
  - We now see if EFG is itself a key…
  - EFG+ = EFGA ≠ R; So, **EFG** alone is not key

Solution #4: Checking for Keys Using the Closure

- Checking by adding single attribute with **EFG** (except **A**):
- BEFG+ = ABCDEFGH = R; it's a key [BE→CD, EG→A, EC→H]
- CEFG+ = ABCDEFGH = R; it's a key [EG→A, EC→H, H→B, BE→CD]
- DEFG+ = ADEFG ≠ R; it's not a key [EG→A]
- EFGH+ = ABCDEFGH = R; it's a key [EG→A, H→B, BE→CD]
- If we add any further attribute(s), they will form the superkey. Therefore, we can stop here searching for candidate key(s).
- Therefore, candidate keys are: {BEFG, CEFG, EFGH}

Exercise #5: Checking for Keys Using the Closure

Consider R = {A, B, C, D, E, F, G} with a set of FDs

F = {ABC→DE, AB→D, DE→ABCF, E→C}

Find all the candidate keys of R

Solution #5: Checking for Keys Using the Closure

F = {ABC→DE, AB→D, DE→ABCF, E→C}

- First, we notice that:
  - **G** never appears on RHS of any FD. So, **G** must be part of ANY key of R.
  - **F** never appears on LHS of any FD, but appears on RHS of some FD. So, **F** is not part of ANY key of R
  - G+ = G ≠ R   So, G alone is not a key!

Solution #5: Checking for Keys Using the Closure

- Now we try to find keys by adding more attributes (except F) to G
  - Add LHS of FDs that have only one attribute (E in E→C):
  - GE+ = GEC ≠ R
  - Add LHS of FDs that have two attributes (AB in AB→D and DE in DE→ABCF):
  - GAB+ = GABD
  - GDE+ = ABCDEFG = R; [DE→ABCF] It's a key!
  - Add LHS of FDs that have three attributes (ABC in ABC→DE), but not taking super set of GDE:
  - GABC+ = ABCDEFG = R; [ABC→DE, DE→ABCF] It's a key!
  - GABE+ = ABCDEFG = R; [AB→D, DE→ABCF] It's a key!
  - If we add any further attribute(s), they will form the superkey. Therefore, we can stop here.
  - The candidate key(s) are {GDE, GABC, GABE}

Canonical Cover

- Sets of functional dependencies may have redundant dependencies that can be inferred from the others
  - Eg: $A \rightarrow C$ is redundant in: $\{A \rightarrow B,\ \ B \rightarrow C,\ \ A \rightarrow C\}$
  - Parts of a functional dependency may be redundant
    - E.g. on RHS: $\{A \rightarrow B,\ \ B \rightarrow C,\ \ A \rightarrow CD\}$ can be simplified to $\{A \rightarrow B,\ \ B \rightarrow C,\ \ A \rightarrow D\}$
    - E.g. on LHS: $\{A \rightarrow B,\ \ B \rightarrow C,\ \ AC \rightarrow D\}$ can be simplified to $\{A \rightarrow B,\ \ B \rightarrow C,\ \ A \rightarrow D\}$
- Intuitively, a canonical cover of F is a "minimal" set of functional dependencies equivalent to F, having no redundant dependencies or redundant parts of dependencies