Functional Dependency Problems
1. For relation $R\{A,B,C,D,E,F,G\}$
   \{ $A \to B$, $BC \to DE$, $AEF \to G$ \} $\vdash ACF \to DG$

Source of the RHS

So we need to show that $ACF \to BC$ and $AEF$
Notice $A \to B$ can be augmented to $AC \to BC$
that transitively yields $DE$.

By the ‘shootover rule’ (augmentation),
$AC \to DE$ yields $AC \to ACDE$

And with more augmentation we get
$ACF \to ACDEF$
1. For relation R\{A,B,C,D,E,F,G\}
   \{ A \rightarrow B, BC \rightarrow DE, AEF \rightarrow G \} \models ACF \rightarrow DG

1. A \rightarrow B \quad \text{(given)}
2. BC \rightarrow DE \quad \text{(given)}
3. AEF \rightarrow G \quad \text{(given)}
4. AC \rightarrow BC \quad \text{(1, aug.)}
5. AC \rightarrow DE \quad \text{(4,2 trans)}
6. ACF \rightarrow DEF \quad \text{(5, aug.)}
7. ACF \rightarrow ACDEF \quad \text{(6, aug.)}
8. ACF \rightarrow AEF \quad \text{(7, decompl)}
9. ACF \rightarrow G \quad \text{(8,3 trans)}
10. ACF \rightarrow D \quad \text{(6, decompl)}
11. ACF \rightarrow DG \quad \text{(9,10 union)}
2. For R \{ABCDEF\}

\{A \rightarrow BC, B \rightarrow E, CD \rightarrow EF \} \models AD \rightarrow F

1. A \rightarrow BC \quad \text{(given)}
2. B \rightarrow E \quad \text{(given)}
3. CD \rightarrow EF \quad \text{(given)}
4. AD \rightarrow BCD \quad \text{(1, aug)}
5. AD \rightarrow CD \quad \text{(4, decomp)}
6. AD \rightarrow EF \quad \text{(5,3 trans)}
7. AD \rightarrow F \quad \text{(6, decomp)}
NOTE: Proof Strategies

• Suppose you have to prove AB → CD
• Try deducing a functional dependency with CD on the RHS (right hand side)
  – Augmentation, Union or Decomposition can modify RHS of FDs.
  – result: x → CD
Strategies (cont.)

- Try deducing a functional dependency with AB on the LHS.
  - Augmentation, Reflexivity, Pseudo-transitivity can affect LHS.
  - result: \( AB \rightarrow y \)
- Now try to deduce \( y \rightarrow x \)
- Try disproof first: it is mechanical.
Are 2 sets of FDs Equivalent?

First method:
- Compute the closure of F
- Compute the closure of G
- See if they are equal

Second method
- Show every FD in F can be proven from G
- Show every FD in G can be proven from F
3. Equivalent sets of FDs?

\[ F = \{ B \rightarrow CD, AD \rightarrow E, B \rightarrow A \} \]

\[ G = \{ B \rightarrow CDE, B \rightarrow ABC, AD \rightarrow E \} \]

- \( F \subseteq G \) ? (AD → E in both.)
- 1. B → CD (given)
- 2. B → A (given)
- 3. AD → E* (given)
- 4. B → ACD (1,2, union)
- 5. B → AD (4, decomp)
- 6. B → E (5,3, trans)
- 7. B → ACDE (4,6,union)
- 8. B → CDE* (7,decomp)
- 9. B → AC (4, decomp)
- 10. B → ABC* (9,aug)

*FD in G to be Proven
3. Continued

\[ F = \{ B \rightarrow CD, \ AD \rightarrow E, \ B \rightarrow A \} \]
\[ G = \{ B \rightarrow CDE, \ B \rightarrow ABC, \ AD \rightarrow E \} \]

- \[ G \not\models F \] (prove \( F \) from \( G \)?)
- Obviously
4. What is the key for R?

• \{KEY\} \rightarrow \{REST\} because key must determine all fields

• **KEY must include all NOT on RHS of ANY functional dependency**
  – Only fields on RHS are determined.
  – An undetermined field must be in the key.

• **REST must include all NOT on LHS**
  – They don't determine anything
  – so they can not be part of key.

• **If field is not in ANY FD, it must be part of the key**
4. What is the key for R?

- \( R=(A,B,C,D,E,F,G,H,I,J) \)
  - FDs: \( AB \rightarrow C, \ BD \rightarrow EF, \ AD \rightarrow GH, \ A \rightarrow I, \ H \rightarrow AJ \)
- NOT on RHS: Key must include BD
- NOT on LHS: Rest must include CEGFIJ
- Unknown: A and H
- Is BD a Key? No: it only determines EF which don't determine anything else
- If BD were a key, we would stop here.
- So Key might include A or H
  - But not both, Why not?
FDs: $AB \rightarrow C$, $BD \rightarrow EF$, $AD \rightarrow GH$, $A \rightarrow I$, $H \rightarrow AJ$

- Try adding $A$: is $ABD$ a key?
  - $ABD \rightarrow EF$ because $BD \rightarrow EF$ (aug)
  - $ABD \rightarrow C$ because $AB \rightarrow C$ (aug)
  - $ABD \rightarrow GH$ because $AD \rightarrow GH$ (aug)
  - $ABD \rightarrow I$ because $A \rightarrow I$ (aug)
  - $ABD \rightarrow J$ because $AD \rightarrow GH$ & $H \rightarrow AJ$ (aug, decomp, trans)
FDs: AB → C, BD → EF, AD → GH, A → I, H → AJ

- Try adding H: is BDH a key?
- Since H → AJ, H → A.
- So BDH → BDA, which is a key.
  - So BDH determines all that ABD determines.
- BDH is another key
- 2 overlapping keys: ABD and BDH.
5. Counter example:

\{ XY \rightarrow Z, Z \rightarrow X \} \models Y \rightarrow XZ ?

- Method: Set up a database in which LHS is NOT Violated but RHS IS violated

- Why? Because \models is a form of implication and implication is only false when LHS is True and RHS is false

- When is RHS (Y \rightarrow XZ )false?
  - It too is a kind of implication!
  - When 2 tuples agree in Y and disagree in XZ
counter: \{ XY \rightarrow Z, Z \rightarrow X \} \not\models Y \rightarrow XZ

• Requirements for counter:
  • two tuples
    – agree in Y, disagree in X and/or Z
    – do not violate LHS.
  • Important Note: given A \rightarrow B
    – two tuples which disagree in A cannot violate A \rightarrow B.
    – Why? Because implication is false only when A is true and B is false.
counter: \{ XY \rightarrow Z, Z \rightarrow X \} \not\models Y \rightarrow XZ

- X \quad Y \quad Z \quad \text{(attributes)}
  a \quad b \quad c \quad \text{(first tuple)}
  ? \quad b \quad ? \quad \text{(y must be the same)}

- What about X? If X is the same:
  X \quad Y \quad Z
  a \quad b \quad c
  a \quad b \quad ?

- Problem: Cannot violate LHS (XY \rightarrow Z)
  - so Z must be the same
  - but cannot have 2 identical tuples.
  - Therefore, make X different
counter: \{ XY \to Z, Z \to X \} \not\models Y \to XZ

- X Y Z
  a b c
  H b ? (y the same & x different)
- What about Z? Suppose Z is the same.
  - Maybe OK since Y \to XZ still violated.
  - X Y Z
    a b c
    H b c
- But this would violate the LHS. Why?
counter: \( \{ XY \rightarrow Z, Z \rightarrow X \} \not\models Y \rightarrow XZ \)

- Solution:
  \[
  \begin{array}{ccc}
  X & Y & Z \\
  a & b & c \\
  H & b & K \\
  \end{array}
  \]

- Check:
  - RHS violated?
    - Yes: Same in \( Y \), different in \( X \) and/or \( Z \)
  - LHS not violated?
    - \( XY \rightarrow Z \): no two the same in \( XY \)
    - \( Z \rightarrow X \): no two the same in \( Z \)