Plan of the lecture

- Logical implication for functional dependencies
- Armstrong closure. Algorithm for computing the closure
- Inference rules for functional dependencies
- Soundness and completeness of the inference rules
- Inference rules for functional and multivalued dependencies
- Examples of derivations

Implication problem for fds

- In order to normalise a relation, we need to check lots of dependencies (for example, find all functional dependencies and check if in any of them determinant is not a key).
- If we had a way to generate all dependencies implied by certain other dependencies, we could save ourselves some work.
- For example, if we know that X → Y implies Z → W and we know that Z → W does not hold, then we don’t need to check whether X → Y holds.

Logical consequence (implication)

- Let Σ be a set of functional dependencies over a set of attributes U, and X → Y a functional dependency involving attributes from the same set (X,Y ⊆ U).
- X → Y is a logical consequence of Σ, or Σ logically implies X → Y, if any relation over attributes in U which satisfies functional dependencies in Σ, also satisfies X → Y.
- In symbols: ‘Σ logically implies X → Y’ is denoted as Σ |= X → Y.

Logical consequence: example

- Let
  - U = {Name, Age, CanVote},
  - Σ = {Name → Age, Age → CanVote}
  - X → Y = Name → CanVote

then Σ |= X → Y.

Logical consequence: example

- Proof: consider any relation R over U. Assume that it satisfies both dependencies in Σ. So, for any two tuples s and t in R,
  - if s(Name)=t(Name), then s(Age) = t(Age)
    (because R satisfies Name → Age)
  - if s(Age) = t(Age), then s(CanVote)=t(CanVote)
    (because R satisfies Age → CanVote)
- so for any two tuples s and t in R,
  - if s(Name)=t(Name), then s(CanVote)=t(CanVote)
    (hence R satisfies Name → CanVote)
Transitivity

• In exactly the same way, we can prove that in general, 
  \( \{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z \)
• That is, for any relation \( R \) whose schema includes \( X, Y \) and \( Z \), it holds that if \( R \) satisfies \( X \rightarrow Y \) and \( Y \rightarrow Z \), then \( R \) satisfies \( X \rightarrow Z \).

Armstrong closure

• Given a set of functional dependencies \( \Sigma \) over a set of attributes \( U \), \textit{fd closure} \( X^* \) of \( X \subseteq U \) is defined as follows:
  \[ X^* = \{ A \in U: \Sigma \models X \rightarrow A \} \]
• Why is it a useful concept? By definition,
  \[ \Sigma \models X \rightarrow Y \text{ if, and only if, } Y \subseteq X^* \]
• So we can test whether \( \Sigma \models X \rightarrow Y \) by computing \( X^* \).
• There is a linear time algorithm to compute \( X^* \); here is a simpler, less efficient version – it is \( O(n \times m) \), where \( n \) is the size of \( \Sigma \) and \( m \) is the size of \( X \).

Algorithm to compute fd closure

• Input: a set of functional dependencies \( \Sigma \) over some set of attributes \( U \) and a set \( X \subseteq U \) of attributes.
• Output: the closure \( X^* \) of \( X \) with respect to \( \Sigma \).

\[
\begin{align*}
\text{unused} & = \Sigma ; \\
\text{closure} & = X ; \\
\text{repeat until no further change:} \\
\quad \text{if } W \rightarrow Z \in \text{unused and } W \subseteq \text{closure} \text{ then} \\
\quad \quad \text{unused} = \text{unused} \setminus \{W \rightarrow Z\}; \\
\quad \text{closure} = \text{closure} \cup Z \\
\text{output closure}
\end{align*}
\]

Correctness of the closure algorithm

• To prove: if \( \Sigma \models X \rightarrow Y \), then \( Y \subseteq X^* \) (as computed by the algorithm).
• Proof: there are two things to prove.
  – one is that the closure algorithm does not claim too much, that is, if \( Y \subseteq X^* \) then \( \Sigma \models X \rightarrow Y \).
  – another is that it does not fail to discover dependencies: if \( \Sigma \models X \rightarrow Y \), then \( Y \subseteq X^* \).
Counterexample

<table>
<thead>
<tr>
<th>R</th>
<th>X*</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>X_{i1},...,X_{in}, Y_{i1},...,Y_{ik}, Z_{i1},...,Z_{im}</code></td>
<td><code>a_{i1},...,a_{in}, b_{i1},...,b_{ik}, c_{i1},...,c_{im}, d_{i1},...,d_{im}</code></td>
</tr>
</tbody>
</table>

The idea is that \( X = \{X_{1},...,X_{n}\} \), \( X^{*} = \{X_{1},...,X_{n}, Y_{1},...,Y_{k}\} \), two tuples \( s \) and \( t \) agree on \( X^{*} \) and disagree on \( Z \). Clearly \( R \) does not satisfy \( X \rightarrow Z \) if some of \( Z \)'s attributes are among \( Z_{i} \). Let us show that \( R \) satisfies \( \Sigma \). Assume otherwise: there is \( W \rightarrow V \) in \( \Sigma \) such that \( R \) does not satisfy it. So \( s \) and \( t \) agree on \( W \) (so \( W \subseteq X^{*} \)) and disagree on \( V \) (so \( V \) is not a subset of \( X^{*} \)). But if \( W \subseteq X^{*} \) and \( W \rightarrow V \) in \( \Sigma \), then by construction of \( X^{*} \), also \( V \) should be in \( X^{*} \), a contradiction.

Inference rules for fds

- The following rules are due to Armstrong (1974):
  - Let \( X,Y,Z \) be sets of attributes of some relation \( R \). Then
    - FD1 (reflexivity): if \( Y \subseteq X \) then \( X \rightarrow Y \)
    - FD2 (augmentation): if \( X \rightarrow Y \) then \( XZ \rightarrow YZ \)
    - FD3 (transitivity): if \( X \rightarrow Y \) and \( Y \rightarrow Z \) then \( X \rightarrow Z \)
  - Here, \( XZ \) is short for \( X \cup Z \).

Derivability

- Let \( \Sigma \) be a set of functional dependencies over a set of attributes \( U \), and \( X \rightarrow Y \) a functional dependency involving attributes from the same set.
- \( X \rightarrow Y \) is derivable from \( \Sigma \) (by inference rules FD1-FD3) if we can obtain \( X \rightarrow Y \) by applying inference rules to dependencies in \( \Sigma \) (or, if there is a finite sequence of dependencies, each of which is either in \( \Sigma \), or obtained from the previous dependencies by FD1-FD3).
- In symbols: \( \Sigma \models X \rightarrow Y \) is denoted as \( \Sigma \models X \rightarrow Y \).

Example

- Let \( A,B,C,D,E \) be attributes and
  \[ \Sigma = \{ A \rightarrow B, B \rightarrow C, CD \rightarrow E \} \]
- Then \( \Sigma \models AD \rightarrow E \):
  - \( A \rightarrow C \) from \( A \rightarrow B \) and \( B \rightarrow C \) by FD3
  - \( AD \rightarrow CD \) from \( A \rightarrow C \) by FD2
  - \( AD \rightarrow E \) from \( AD \rightarrow CD \) and \( CD \rightarrow E \) by FD3

Soundness and completeness

- Armstrong’s rules are sound: we can never derive dependencies which do not hold:
  - if \( \Sigma \models X \rightarrow Y \) then \( \Sigma \models X \rightarrow Y \).
- Armstrong’s rules are complete: if a dependency \( X \rightarrow Y \) is a logical consequence of \( \Sigma \), then we can derive \( X \rightarrow Y \) from \( \Sigma \):
  - if \( \Sigma \models X \rightarrow Y \) then \( \Sigma \models X \rightarrow Y \).

Proof of soundness

- The proof is by induction on the length of the derivation (the number of rule applications). We show that at each step in deriving \( X \rightarrow Y \) from \( \Sigma \), by applying FD1-FD3 we only obtain logical consequences.
  - FD1 (reflexivity): if \( Y \subseteq X \) then \( X \rightarrow Y \).
  - Take any relation \( R \). We want to show that for any two tuples \( s \) and \( t \) in \( R \), if \( s(X) = t(X) \), then \( s(Y) = t(Y) \).
  - Suppose \( s(X) = t(X) \). We know that \( Y \subseteq X \), so if \( s \) and \( t \) agree on all attributes in \( X \), then they agree on all attributes in \( Y \). So \( s(Y) = t(Y) \).
Proof of soundness

• FD2 (augmentation): if \( X \rightarrow Y \) then \( XZ \rightarrow YZ \).
• Take any relation \( R \) which satisfies \( X \rightarrow Y \). We want to show that for any two tuples \( s \) and \( t \) in \( R \), if \( s(XZ)=t(XZ) \), then \( s(YZ)=t(YZ) \).
• Assume \( s(XZ)=t(XZ) \). This is the same as \( s(X)=t(X) \) and \( s(Z)=t(Z) \).
• Since \( R \) satisfies \( X \rightarrow Y \), from \( s(X)=t(X) \) we get \( s(Y)=t(Y) \).
• We know that \( s(Y)=t(Y) \) and \( s(Z)=t(Z) \), so \( s(YZ)=t(YZ) \).

Proof of soundness

• FD3 (transitivity): if \( X \rightarrow Y \) and \( Y \rightarrow Z \) then \( X \rightarrow Z \).
• Really the same as our example of logical consequence.
• Take any relation \( R \) which satisfies \( X \rightarrow Y \) and \( Y \rightarrow Z \).
• We want to show that for any two tuples \( s \) and \( t \) in \( R \), if \( s(X)=t(X) \), then \( s(Z)=t(Z) \).
• Assume \( s(X)=t(X) \). Then by \( X \rightarrow Y \), \( s(Y)=t(Y) \).
• Then by \( Y \rightarrow Z \), \( s(Z)=t(Z) \).

Proof of completeness

• To show that if a dependency \( X \rightarrow Y \) follows from \( \Sigma \), then it is also derivable from \( \Sigma \) using the axioms; in other words, \( \Sigma \models X \rightarrow Y \) implies \( \Sigma \models X \rightarrow Y \): The proof consists of two parts:
• \( \Sigma \models X \rightarrow X^* (\Sigma \text{ implies a functional dependency between } X \text{ and the set of all attributes in the closure of } X \text{ with respect to } \Sigma) \).
• If \( \Sigma \models X \rightarrow Y \), then \( Y \subseteq X^* \). From \( X \rightarrow X^* \) and \( Y \subseteq X^* \) we can derive \( X \rightarrow Y \) by FD1 and FD3 (the proof of this property, called decomposition, is given later in the lecture).

\( \Sigma \models X \rightarrow X^* \)

• By induction, we show that at every step \( i \) in construction of \( X^* \), \( \Sigma \models X \rightarrow \text{closure}_i \), where \( \text{closure}_i \) is the set of attributes in the closure at step \( i \).
• Basis of induction: at step 0, \( \text{closure}_0 = \{X\} \), and \( \Sigma \models X \rightarrow X \) by FD1.
• Inductive step: suppose \( \Sigma \models X \rightarrow \text{closure}_i \), prove \( \Sigma \models X \rightarrow \text{closure}_{i+1} \).

Derivable rules

• Some other rules are also sound, but we do not need them for completeness because they follow from FD1-FD3.
• For example, Decomposition: if \( X \rightarrow YZ \), then \( X \rightarrow Y \).
• From \( Y \subseteq YZ \) we get \( Y \rightarrow Y \) by FD1.
• From \( X \rightarrow YZ \) and \( YZ \rightarrow Y \) we get \( X \rightarrow Y \) by FD3.
Derivable rules

Union: if $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$.
- From $X \rightarrow Y$ we get $XX \rightarrow XY$ by FD2, and since $XX = X$ we get $X \rightarrow XY$.
- From $X \rightarrow Z$ we get $XY \rightarrow YZ$ by FD2.
- From $X \rightarrow XY$ and $XY \rightarrow YZ$ we get $X \rightarrow YZ$ by FD3.

Reading

- Ullman, Widom, chapter 3.5
- Stanczyk et al, Chapter 7.3 (Armstrong axiomatization of functional dependencies, 3NF).

Informal coursework

- Show that FD1 - FD3 imply

Pseudo-transitivity: if $X \rightarrow Y$ and $TY \rightarrow Z$, then $TX \rightarrow Z$. 