G53RDB: Theory of Relational Databases

Lecture 10

Natasha Alechina
School of Computer Science & IT
nza@cs.nott.ac.uk

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 $X \rightarrow Y$ implies $Z \rightarrow W$ and we know that $Z \rightarrow W$ does not hold, then we don't need to check whether $X \rightarrow Y$ holds.

Logical consequence (implication)

- Let Σ 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 ($X,Y \subseteq U$).
- $X \rightarrow Y$ is a ***logical consequence*** of Σ, or Σ ***logically implies*** $X \rightarrow Y$, if any relation over attributes in U which satisfies functional dependencies in Σ, also satisfies $X \rightarrow Y$.
- In symbols: `Σ logically implies $X \rightarrow Y$ ' is denoted as $\Sigma \models X \rightarrow Y$.

Logical consequence: example

- Let
  - U = {Name, Age, CanVote},
  - Σ = {Name $\rightarrow$ Age, Age $\rightarrow$ CanVote}
  - $X \rightarrow Y$ = Name $\rightarrow$ CanVote

then $\Sigma \models X \rightarrow Y$.

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 $\rightarrow$ Age)
- if s(Age) = t(Age), then s(CanVote)=t(CanVote)
  (because R satisfies Age $\rightarrow$ 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 $\rightarrow$ CanVote)

Transitivity

- In exactly the same way, we can prove that in general,

$\{X \to Y, Y \to Z\} \models X \to Z$

- That is, for any relation R whose schema includes X, Y and Z, it holds that if R satisfies $X \to Y$ and $Y \to Z$, then R satisfies $X \to Z$.

Armstrong closure

- Given a set of functional dependencies Σ over a set of attributes U, ***fd closure*** X^* of $X \subseteq U$ is defined as follows:

$$X^* = \{A \in U: \Sigma \models X \to A\}$$

- Why is it a useful concept? By definition,

$$\Sigma \models X \to Y \text{ if, and only if, } Y \subseteq X^*$$

- So we can test whether $\Sigma \models X \to Y$ 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 Σ and m is the size of X.

Algorithm to compute fd closure

- Input: a set of functional dependencies Σ over some set of attributes U and a set $X \subseteq U$ of attributes.
- Output: the closure X^* of X with respect to Σ.

unused = Σ ;
closure = X;
repeat until no further change:
  if $W \to Z \in$ unused and $W \subseteq$ closure then
    unused = unused – $\{W \to Z\}$;
    closure = closure $\cup$ Z
output closure

Correctness of the closure algorithm

$\Sigma \models X \to Y$ if, and only if, $Y \subseteq X^*$ (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 $\Sigma \models X \to Y$.
  - another is that that it does not fail to discover dependencies: if $\Sigma \models X \to Y$, then $Y \subseteq X^*$.

Correctness of the closure algorithm

- To prove: if $Y \subseteq X^*$, then $\Sigma \models X \to Y$.
- Proof by induction:
  - Basis: at step 0, closure only contains attributes in X. for any subset Y of X, it trivially holds that $\Sigma \models X \to Y$.
  - Inductive step: suppose that for all W added at the previous step, $\Sigma \models X \to W$. Now we add a new set of attributes Z because W is in Σ and $\Sigma \models W \to Z$. But then $\Sigma \models X \to Z$ by transitivity.

Correctness of the closure algorithm

- To prove: if $\Sigma \models X \to Y$, then $Y \subseteq X^*$.
- If Z is not a subset of X^*, we can construct a relation which satisfies Σ and does not satisfy $X \to Z$. This will show that Σ does not logically imply $X \to Z$.
- The relation R contains just two tuples, s and t. They agree on the values of all attributes in X^* and disagree on all attributes which are not in X^*, including at least some in Z. So R satisfies Σ and falsifies $X \to Z$. (A bit more details on the next slide.)

Counterexample

R

	$X_1,\ldots,X_n$	$Y_1,\ldots,Y_k$	$Z_1,\ldots,Z_m$
s	$a_1,\ldots,a_n$	$b_1,\ldots,b_k$	$c_1,\ldots,c_m$
t	$a_1,\ldots,a_n$	$b_1,\ldots,b_k$	$d_1,\ldots,d_m$

The idea is that $X = \{X_1,\ldots,X_n\}$, $X^*=\{X_1,\ldots,X_n, Y_1,\ldots Y_k\}$, two tuples s and t agree on X* and disagree on Zs (c_i is not equal to d_i for every i in 1,…,m). Clearly R does not satisfy $X \to Z$ if some of Z's attributes are among Z_i.

Let us show that R satisfies Σ. Assume otherwise: there is $W \to V$ in Σ 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 $W \to V$ in Σ, 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 $X \to Y$

FD2 (augmentation): if $X \to Y$ then $XZ \to YZ$

FD3 (transitivity): if $X \to Y$ and $Y \to Z$ then $X \to Z$

- Here, XZ is short for $X \cup Z$.

Derivability

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

Example

- Let A,B,C,D,E be attributes and

  $\Sigma = \{A \to B, B \to C, CD \to E\}$
- Then $\Sigma \vdash AD \to E$:
  - $A \to C$ from $A \to B$ and $B \to C$ by FD3
  - $AD \to CD$ from $A \to C$ by FD2
  - $AD \to E$ from $AD \to CD$ and $CD \to E$ by FD3

Soundness and completeness

- Armstrong's rules are ***sound:*** we can never derive dependencies which do not hold:

  if $\Sigma \vdash X \to Y$ then $\Sigma \models X \to Y$.
- Armstrong's rules are ***complete:*** if a dependency $X \to Y$ is a logical consequence of Σ, then we can derive $X \to Y$ from Σ.

  if $\Sigma \models X \to Y$ then $\Sigma \vdash X \to Y$.

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 $X \to Y$ from Σ, by applying FD1-FD3 we only obtain logical consequences.
- FD1 (reflexivity): if $Y \subseteq X$ then $X \to Y$.
- 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 $X \to Y$ then $XZ \to YZ$.
- Take any relation R which satisfies $X \to Y$. 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 $X \to Y$, 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 $X \to Y$ and $Y \to Z$ then $X \to Z$.
- Really the same as our example of logical consequence.
- Take any relation R which satisfies $X \to Y$ and $Y \to Z$.
- 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 $X \to Y$, s(Y)=t(Y).
- Then by $Y \to Z$, s(Z)=t(Z).

Proof of completeness

- To show that if a dependency $X \to Y$ follows from Σ, then it is also derivable from Σ using the axioms; in other words, $\Sigma \models X \to Y$ implies $\Sigma \vdash X \to Y$:

The proof consists of two parts:

- $\Sigma \vdash X \to X^*$ (Σ implies a functional dependency between X and the set of all attributes in the closure of X with respect to Σ).
- If $\Sigma \models X \to Y$, then $Y \subseteq X^*$. From $X \to X^*$ and $Y \subseteq X^*$ we can derive $X \to Y$ by FD1 and FD3 (the proof of this property, called decomposition, is given later in the lecture).

$\Sigma \vdash X \to X^*$

- By induction, we show that at every step i in construction of X^*, $\Sigma \vdash X \to closure_i$, where $closure_i$ is the set of attributes in the closure at step i.
- Basis of induction: at step 0, $closure_0 = \{X\}$, and $\Sigma \vdash X \to X$ by FD1.
- Inductive step: suppose $\Sigma \vdash X \to closure_i$, prove $\Sigma \vdash X \to closure_{i+1}$.

$\Sigma \vdash X \to X^*$

- Inductive step:
  - (1) assume $\Sigma \vdash X \to closure_i$ inductive hypothesis
  - (2) $\Sigma \vdash closure_i \to W$ by fd1
  - (3) $\Sigma \vdash W \to Z$ because $W \to Z$ is in Σ
  - (4) $\Sigma \vdash closure_i \to Z$ by fd3 from (2), (3)
  - (5) $\Sigma \vdash closure_i \to closure_i \cup Z$ by fd2 from (4)

  (note that $closure_i \cup closure_i = closure_i$)
  - (6) $\Sigma \vdash closure_i \to closure_{i+1}$ from (5)
  - (7) $\Sigma \vdash X \to closure_{i+1}$ by fd3 from the inductive hypothesis (1) and (6).

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 $X \to YZ$, then $X \to Y$.

- From $Y \subseteq YZ$ we get $YZ \to Y$ by FD1.
- From $X \to YZ$ and $YZ \to Y$ we get $X \to Y$ by FD3.

Derivable rules

Union: if $X \to Y$ and $X \to Z$, then $X \to YZ$.

- From $X \to Y$ we get $XX \to XY$ by FD2, and since $XX=X$ we get $X \to XY$.
- From $X \to Z$ we get $XY \to YZ$ by FD2.
- From $X \to XY$ and $XY \to YZ$ we get $X \to YZ$ by FD3.

Reading

- Ullman, Widom, chapter 3.5
- Stanczyk et al, Chapter 7.3 (Armstrong axiomatization of functional dependencies, 3NF).
- Abiteboul, Hull, Vianu, *Foundations of Databases*, chapter 8 for axiomatization of multivalued dependencies and functional dependencies.

Informal coursework

- Show that FD1 - FD3 imply (9)

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