3. Functional Dependencies

Stéphane Bressan

January 22, 2015

This lecture is based on material by Professor Ling Tok Wang.

CS 4221: Database Design

The Relational Model

Ling Tok Wang
National University of Singapore

1

CS4221: The Relational Model

https://www.comp.nus.edu.sg/~lingtw/cs4221/rm.pdf

Content

"[...] one can say that the relational model is almost devoid of semantics. [...] One approach to remedy this deficiency is to devise means to specify the missing semantics. These semantic specifications are called semantic or integrity constraints [...]. Of particular interest are the constraints called data dependencies or depencies for short."
Fundementals of Dependency Theory [TTC 1987],
by Moshe Y. Vardi

Author	
First_name	Last_name
Peter	Chen
Jeffrey	Ullman
David	Maier
Serge	Abiteboul
Albert	Maier
Edgar	Codd

$$\{First\_name\} \rightarrow \{Last\_name\}$$

This example is not about semantics!?

Author	
First_name	Last_name
Peter	Chen
Jeffrey	Ullman
David	Maier
Serge	Abiteboul
Albert	Maier
Peter	Buneman
Edgar	Codd

Author		
Email	First_name	Last_name
pchen@lsu.edu	Peter	Chen
ullman@gmail.com	Jeffrey	Ullman
maier@pdx.edu	David	Maier
serge.abiteboul@irnria.fr	Serge	Abiteboul
amaier@de.ibm.com	Albert	Maier

$$\{Email\} \rightarrow \{Fist\_name, Last\_name\}$$

$$\{Email\} \rightarrow \{Fist\_name, Last\_name\}$$

```
CHECK(NOT EXISTS
        (SELECT *
        FROM Authors A1, Authors A2
        WHERE A1.Email=A2.Email
        AND (A1.Fisrt_name <> A2. First_name
           OR A1.Last_name <> A2. Last_name)

CREATE TABLE Author
...
Email ... PRIMARY KEY,
...)
```

Author		
Email	First_name	Last_name
pchen@lsu.edu	Peter	Chen
ullman@gmail.com	Jeffrey	Ullman
maier@pdx.edu	David	Maier
serge.abiteboul@irnria.fr	Serge	Abiteboul
amaier@de.ibm.com	Albert	Maier
zc@fudan.edu.cn	Zhang	Chen
zhang.chen@ibm.com	Zhang	Chen
	Edgar	Codd

"In dblp, [...] Different authors are assigned a unique key and their names are distinguished in our data stock by a unique numerical suffix to their name."
`http://www.informatik.uni-trier.de/~ley`

Readings

- Abiteboul S.,Hull R. and Vianu V. "Foundations of Databases", Chapter 8 (`http://webdam.inria.fr/Alice/pdfs/all.pdf`).

Definition

An instance r of a relation schema R satisfies a dependency σ is noted as follows.

$$r \models \sigma$$

Definition

An instance r of a relation schema R satisfies a set of dependencies Σ if and only if it satisfies all the dependencies in the set.

$$(r \models \Sigma) \Leftrightarrow \ (\forall \sigma \in \Sigma\ (r \models \sigma))$$

Definition

An instance r of a relation schema R is a valid instance of R with Σ if and only if it satisfies Σ.

Definition

An instance r of a relation schema R satisfies the functional dependency σ: $X \rightarrow Y$, X functionally determines Y or Y is functionally dependent on X, with $X \subset R$ and $Y \subset R$, if and only if tuples of r agree on their X-values, then they agree on their Y-values.

$$(r \models \sigma)$$

$$\Leftrightarrow$$

$$(\forall t_1 \in r \ \forall t_2 \in r \ (t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]))$$

each X-value in r has associated with exactly one Y-value in r

Author	
First_name	Last_name
Peter	Chen
Jeffrey	Ullman
David	Maier
Serge	Abiteboul
Albert	Maier
Edgar	Codd

$$\{First\_name\} \rightarrow \{Last\_name\}$$

When we talk about R with Σ, we talk about all the valid instances of R with respect to the functional depenencies in R.

Definition

A set of attributes Y of a relation schema R is said to be **functionally dependent** on a set of attributes X of R if at any time any valid instance r of R satisfies the functional dependency σ.

What is the meaning of $\emptyset \rightarrow \{A\}$?

Definition

A functional dependency $X \rightarrow Y$ is trivial if and only if $Y \subset X$.

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

$\{A\} \rightarrow \{A\}$ is trivial.
$\{A, B\} \rightarrow \{A\}$ is trivial.
$\{A, B\} \rightarrow \emptyset$ is trivial.

Definition

A functional dependency $X \rightarrow Y$ is non-trivial if and only if $Y \not\subset X$.

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

$\{A\} \rightarrow \{B\}$ is non-trivial.
$\{A, C\} \rightarrow \{B, C\}$ is non-trivial.

Definition

A functional dependency $X \rightarrow Y$ is completely non-trivial if and only if $Y \cap X = \emptyset$.

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

$\{A\} \rightarrow \{B\}$ is completely non-trivial.
$\{A, C\} \rightarrow \{B, C\}$ is not completely non-trivial.

Definition

Let Σ be a set of functional dependencies of a relation schema R. The closure of Σ, noted Σ^+, is the set of all functional dependencies logically entailed by the functional dependencies in Σ.

$$\Sigma^+ = \{\sigma \mid \forall r \textit{ of } R\ ((\forall \gamma \in \Sigma\ (r \models \gamma)) \Rightarrow r \models \sigma\}$$

$$\Sigma^+ = \{\sigma \mid (R \textit{ with } \Sigma) \models \sigma\}$$

$$\Sigma^+ = \{X \rightarrow Y \mid X \subset R \wedge Y \subset R \wedge \Sigma \models X \rightarrow Y\}$$

$$\Sigma^+ = \text{ all the functional dependencies holding on } R \text{ with } \Sigma!$$

Example

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

$\Sigma^+ = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}, \{A\} \rightarrow \{A\}, \{D\} \rightarrow \{D\}, \{A, B\} \rightarrow \{A\}, \{A, C\} \rightarrow \{B, C\}, \{A, D\} \rightarrow \{B\}, \{C\} \rightarrow \{B\}, \cdots\}$

Find

- a trival functional dependency in Σ^+.
- a non-trival but not completely non-trivial functional dependency in Σ^+.
- a completely non-trivial functional dependency in Σ^+.

Example

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

$\{A, D\} \rightarrow \{B, C\} \in \Sigma^{+}$?

Armed with only the definition of functional dependency, the problems of computing Σ^{+} and of testing membership to Σ^{+} are daunting tasks.

Definition

Let Σ be a set of functional dependencies of a relation schema R. The closure of a set of attributes $S \subset R$, noted S^+, is the following set of all attributes that are functionally dependent on S.

$$\{a \in R \mid \exists(S \rightarrow \{a\}) \in \Sigma^+\}$$

Example

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

$\{C\}^+ = \{A, B, C\}$
$\{A, D\}^+ = \{A, B, D\}$
$\{C, D\}^+ = \{A, B, C, D\}$

```
input  : S, Σ
output : S⁺
begin
    Ω := Σ ; // Ω stands for``unused''
    Γ := S ; // Γ stands for ``closure''
    while X → Y ∈ Ω and X ∈ Γ do
        Ω := Ω − {X → Y};
        Γ := Γ ∪ Y;
    end
    return Γ;
end
```

Algorithm 1: Attribute Closure Algorithm

Example

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

Compute $\{C\}^+$ using Algorithm 1.

Example

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

1. $\Omega = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$
   $\Gamma = \{C\}^+$
2. use $\{C\} \rightarrow \{A\}$ $(C \in \Gamma)$
   $\Omega = \{\{A\} \rightarrow \{B\}\}$
   $\Gamma = \{C\} \cup \{A\} = \{C, A\}$
3. use $\{A\} \rightarrow \{B\}$ $(A \in \Gamma)$
   $\Omega = \emptyset$
   $\Gamma = \{C, A\} \cup \{B\} = \{C, A, B\}$
4. return $\Gamma = \{C, A, B\}$

Theorem

Algorithm 1 computes S^+ *for* R *with* Σ.

Proof.

1. We prove that Algorithm 1 is sound

   We will do that formally later when we are equipped with adequate tools, but it is easy to convince oneself that each step of the algorithm is logically sound. Namely, $\Gamma \subset S^+$.

2. We prove that Algorithm 1 is complete. Namely, we prove that $S^+ \subset \Gamma$. Namely, if $A \in R - \Gamma$, then (R with Σ) $\not\models \Gamma \to \{A\}$.

   1. We construct an instance r of R with two t-uples such that $r \models \Sigma$ and $r \not\models \Gamma \to \{A\}$ for $A \in R - \Gamma$. This is always possible because by Algorithm 1 if $X \to Y \in \Sigma$, if $X \in \Gamma$, then $Y \in \Gamma$.

3. Q.E.D

Example

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

$\{C\}^+ = \{A, B, C\}$

R			
A	B	C	D

Example

$R = \{A, B, C, D\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{C\} \rightarrow \{A\}\}$

R			
A	B	C	D
1	1	1	1
1	1	1	2

The instance satisfies Σ but does not satisfy $\{A, B, C\} \rightarrow \{D\}$. To show that a functional depenency does not hold, it suffices to show a valid instance that does not satisfy the functional dependency.

Definition

Let Σ_1 and Σ_2 be sets of functional dependencies on a relation schema R. Σ_1 and Σ_2 are equivalent, $\Sigma_1 \equiv \Sigma_2$, if and only if they have the same closure.

$$\Sigma_1^+ = \Sigma_2^+$$

Example

$\Sigma_1 = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$
$\Sigma_2 = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{A, C\}, \{A\} \rightarrow \{A\}\}$

Σ_1 and Σ_2 are equivalent

$$\Sigma_1 \equiv \Sigma_2$$

Theorem

Let Σ_1 and Σ_2 be sets of functional dependencies on a relation schema R. Σ_1 and Σ_2 are equivalent if and only if each functional dependency in one are logical consequence of those in the other and vice versa.

$$(\forall \sigma \in \Sigma_1(\sigma \in \Sigma_2)) \wedge (\forall \sigma \in \Sigma_2(\sigma \in \Sigma_1))$$

Example

$\Sigma_1 = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{A\}\}$
$\Sigma_2 = \{\{A\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{B\} \rightarrow \{A\}\}$

Σ_1 and Σ_2 are equivalent

$$\Sigma_1 \equiv \Sigma_2$$

Definition

Let Σ be a set of functional dependencies on a relation schema R. Let $X \rightarrow Y \in \Sigma^+$. The functional dependency $X \rightarrow Y$ is a full dependency if and only if it is non-trivial and there exists no proper subset X' of X ($X' \subset X$ and $X' \neq X$) such that $X' \rightarrow Y \in \Sigma^+$. We say that Y is fully dependent on X.

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

$\{B\}$ is fully dependent on $\{A\}$. $\{A\} \rightarrow \{B\}$ is a full dependency. $\{B, C\}$ is not fully dependent on $\{A, C\}$. $\{A, C\} \rightarrow \{B, C\}$ is not a full dependency.

Definition

Let Σ be a set of functional dependencies on a relation schema R. Two sets of attributes $S_1 \subset R$ and $S_2 \subset R$ are functionally equivalent if and only if they are functionally dependent on each other.

$$S_1 \rightarrow S_2 \in \Sigma^+ \wedge S_2 \rightarrow S_1 \in \Sigma^+$$

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

$\{A\}$ and $\{B\}$ are functionally equivalent.
$\{A, C\}$ and $\{B, C\}$ are functionally equivalent.

Definition

Let Σ be a set of functional dependencies on a relation schema R. Two sets of attributes $S_1 \subset R$ and $S_2 \subset R$ are properly functionally equivalent if and only if they are functionally equivalent and there exists no proper subset $S_1' \subset S_1$, $S_1' \neq S_1$ or $S_2' \subset S_2$, $S_2' \neq S_2$, such that:

$$S_1' \rightarrow S_2 \in \Sigma^+ \vee S_2' \rightarrow S_1 \in \Sigma^+$$

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

$\{A\}$ and $\{B\}$ are properly functionally equivalent.
$\{A, C\}$ and $\{B, C\}$ are not properly functionally equivalent.

Definition

Let Σ be a set of functional dependencies on a relation schema R. An attribute A is transitively dependent on a set S_1 of attributes if and only if there exists a set of attributes S_2 such that :

$$\{A\} \not\subseteq S_1 \wedge \{A\} \not\subseteq S_2 \wedge S_2 \rightarrow S_1 \notin \Sigma^+$$

$$S_1 \rightarrow S_2 \in \Sigma^+ \wedge S_2 \rightarrow \{A\} \in \Sigma^+$$

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\}$

$\{C\}$ is transitively dependent on A.

Definition

Let Σ be a set of functional dependencies on a relation schema R. Let $S \to R \in \Sigma^+$. S is a superkey of R with Σ.

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \to \{B, C\}, \{B\} \to \{A\}\}$

$\{A, B\} \to \{A, B, C\}$.
$\{A, B\}$ is a superkey of R with Σ.

Definition

Let Σ be a set of functional dependencies on a relation schema R. Let S be a superkey of R with Σ. S is a candidate key (sometimes called key) of R with Σ if and only if there no proper subset S' of S that is also a superkey.

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

$\{A, B\} \rightarrow \{A, B, C\}$.
$\{A, B\}$ is not a candidate key of R with Σ.
$\{A\} \rightarrow \{A, B, C\}$.
$\{A\}$ is a candidate key of R with Σ.

Definition

Let Σ be a set of functional dependencies on a relation schema R. Let S be a candidate key of R with Σ. S is a primary key if the designer decides so.

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

$\{A\}$ is a candidate key of R with Σ.
$\{B\}$ is a candidate key of R with Σ.
The designer decides that $\{A\}$ is a primary key.

Definition

Let Σ be a set of functional dependencies on a relation schema R. A prime attribute is an attribute that is appears in some candidate key of R with Σ (otherwise it is called a non-prime attribute).

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

A is a prime attribute of R with Σ.
B is a prime attribute of R with Σ.
C is a non-prime attribute of R with Σ.

Theorem

Let Σ be a set of functional dependencies on a relation schema R. The number of candidate keys can be factorial in the size of Σ and exponential in the size of R. [S. Osborn, "Normal Forms for Relational Data Bases", PhD thesis of the University of Waterloo, 1977].

Theorem

Let Σ be a set of functional dependencies on a relation schema R. The problem of deciding whether an attribute is prime or not is NP-Complete. [S. Osborn and L. Lucchesi, "Candidate Keys for relations", Journal of Computer and System Sciences 17, 270-279, 1978].

Example

$R =$
$\{\text{STUDENT\#}, \text{COURSE\#}, \text{S-NAME}, \text{C-DESCRIPTION}, \text{MARK}\}$.
$\Sigma =$
$\{\{\text{STUDENT\#}\} \rightarrow \{\text{S-NAME}\}$,
$\{\text{COURSE\#}\} \rightarrow \{\text{C-DESCRIPTION}\}$,
$\{\text{STUDENT\#}, \text{COURSE\#}\} \rightarrow \{\text{MARK}\}\}$

- Find Σ^+.
- Find $\{\text{STUDENT\#}, \text{MARK}\}^+$ with Σ.
- Find the candidate keys of R with Σ.
- Find the prime attributes of R with Σ.

Σ^+

$\{\{STUDENT\#, COURSE\#\} \rightarrow \{S\text{-}NAME\},$
$\{STUDENT\#, COURSE\#\} \rightarrow \{C\text{-}DESCRIPTION\},$
$\{STUDENT\#, S\text{-}NAME, COURSE\#\} \rightarrow \{MARK\},$
$\ldots\}$

$\{STUDENT\#, MARK\}^+$

$\{STUDENT\#, S\text{-}NAME, MARK\}$

Candidate Keys

Only one: $\{STUDENT\#, COURSE\#\}$

Prime Attributes

STUDENT#, COURSE#

Definition

Let R be a set of attributes. The following inference rules are the Armstrong Axioms.

- Reflexivity (Inclusion, *Projectivity*)
  $\forall X \subset R\ \forall Y \subset R$
  $((Y \subset X) \Rightarrow (X \rightarrow Y))$.
- Augmentation
  $\forall X \subset R\ \forall Y \subset R\ \forall Z \subset R$
  $((X \rightarrow Y) \Rightarrow (X \cup Z \rightarrow Y \cup Z))$.
- Transitivity
  $\forall X \subset R\ \forall Y \subset R\ \forall Z \subset R$
  $((X \rightarrow Y \wedge Y \rightarrow Z) \Rightarrow (X \rightarrow Z))$.

Technically, the Armstrong Axioms are not axioms but inference rules.

Theorem

The Reflexivity inference rule is sound (correct, valid).

Theorem

The Augmentation inference rule is sound.

Theorem

The Transitivity inference rule is sound.

Proof of Soundness for Transitivity.

1. Let Σ be a set of functional dependencies on a relation schema R. Let $X \rightarrow Y$ and $Y \rightarrow Z$ be in Σ.
2. We know that for all valid instance r of R with Σ $(\forall t_1 \in r\ \forall t_2 \in r\ (t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]))$ by definition of a functional dependency.
3. We know that for all valid instance r of R with Σ $(\forall t_1 \in r\ \forall t_2 \in r\ (t_1[Y] = t_2[Y] \Rightarrow t_1[Z] = t_2[Z]))$ by definition of a functional dependency.
4. Therefore for all valid instance r of R with Σ $(\forall t_1 \in r\ \forall t_2 \in r\ (t_1[X] = t_2[X] \Rightarrow t_1[Z] = t_2[Z]))$ by definition of a functional dependency.
5. Therefore $X \rightarrow Z \in \Sigma^+$.
6. Q.E.D.

Theorem

The Armstrong Axioms are complete.

Proof Sketch

1. We prove that for any set of attribute $S \in R$, then $S \rightarrow S^{+}$ can be derived from the Armstrong Axioms.
    1. This is recursively true because every step of the attribute closure algorithm is of the form: $S \rightarrow S^{i}$ and $X \rightarrow Y$ with $X \subset S^{i}$.
    2. Therefore $S^{i} \rightarrow X$ by Reflexivity.
    3. Therefore $S^{i} \rightarrow X \cup S^{i}$ by Augmentation of (2) with S^{i}.
    4. Therefore $S \rightarrow S^{i+1}$, where $S^{i+1} = S_{i} \cup Y$, by Transitivity of (1) and (3).
    5. Q.E.D

...

Proof Sketch

1. We prove that if $X \to Y \in \Sigma^+$, then it can be derived from $X \to X^+$.
    1. We know that $Y \subset X^+$ by property of the attribute closure.
    2. Therefore $X^+ \to Y$ by Reflexivity.
    3. Therefore $X \to Y$ by Transitivity of $X \to X^+$ and $X^+ \to Y$.
    4. Q.E.D
2. Q.E.D.

Theorem

Let Σ be a set of functional dependencies on a relation schema R. Σ^+ can be computed by the iterative fixpoint application of the Armstrong Axioms.

Example

$R = \{A, B, C\}$
$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$

to be done...

Theorem

The Weak Reflexivity axiom is sound.
$\forall X \subset R$
$(X \rightarrow \emptyset)$.

This one is an axiom.

Proof.

1. Let R be a relation schema.
2. Let $X \subset R$.
3. We know that $\emptyset \subset X$.
4. Therefore$X \rightarrow \emptyset$ by Reflexivity.
5. Q.E.D

Theorem

The Weak Augmentation inference rule is sound.
$\forall X \subset R\ \forall Y \subset R\ \forall Z \subset R$
$((X \rightarrow Y) \Rightarrow (X \cup Z \rightarrow Y))$.

Proof.

1. Let R be a relation schema.
2. Let $X \subset R$.
3. Let $Y \subset R$.
4. Let $Z \subset R$.
5. Let $X \to Y$
6. We know that $X \subset X \cup Z$.
7. Therefore $X \cup Z \to X$ by Reflexivity.
8. Therefore $X \cup Z \to Y$ by Transitivity of (7) and (5).
9. Q.E.D

Theorem

The Strict Reflexivity (Reflexivity) axiom is sound.
$\forall X \subset R$
$(X \rightarrow X)$.

Theorem

The Pseudo-Transitivity inference rule is sound.
$\forall X \subset R \ \forall Y \subset R \ \forall Z \subset R \ \forall W \subset R$
$((X \rightarrow Y \wedge W \rightarrow Z \wedge W \subset Y) \Rightarrow (X \rightarrow Z))$.

Theorem

The Union inference rule is sound.

$\forall X \subset R\ \forall Y \subset R\ \forall Z \subset R$
$((X \rightarrow Y \wedge X \rightarrow Z) \Rightarrow (X \rightarrow Y \cup Z))$.

Theorem

The Decomposition (Projectivity) inference rule is sound.

$\forall X \subset R\ \forall Y \subset R\ \forall Z \subset R$
$((X \rightarrow Y \cup Z) \Rightarrow (X \rightarrow Y))$.

Theorem

The Composition inference rule is sound.

$\forall X \subset R\ \forall Y \subset R\ \forall Z \subset R\ \forall W \subset R$
$((X \rightarrow Y \wedge Z \rightarrow W) \Rightarrow (X \cup Z \rightarrow Y \cup W))$.

Theorem

Reflexivity, Union and Transitivity form a complete set of inference rules for functional dependencies.

Theorem

Weak Reflexivity, Weak Augmentation and Pseudo-Transitivity form a complete set of inference rules for functional dependencies.

How to prove it?

How to prove it?

Example

$R =$
$\{\text{STUDENT\#}, \text{COURSE\#}, \text{S-NAME}, \text{C-DESCRIPTION}, \text{MARK}\}$.
$\Sigma =$
$\{\{\text{STUDENT\#}\} \rightarrow \{\text{S-NAME}\}$,
$\{\text{COURSE\#}\} \rightarrow \{\text{C-DESCRIPTION}\}$,
$\{\text{STUDENT\#}, \text{COURSE\#}\} \rightarrow \{\text{MARK}\}\}$

Prove that
$\{\text{STUDENT\#}, \text{MARK}\} \rightarrow \{\text{STUDENT\#}, \text{S-NAME}, \text{MARK}\}$,
using the Armstrong Axioms.

Proof.

1. We know that $\{\text{STUDENT\#}\} \rightarrow \{\text{S-NAME}\}$.
2. Therefore
$\{\text{STUDENT\#}, \text{MARK}\} \rightarrow \{\text{STUDENT\#}, \text{S-NAME}, \text{MARK}\}$
by Augmentation of (1) with $\{\text{STUDENT\#}, \text{MARK}\}$.
3. Q.E.D.

Minimal Cover

Example

$$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C, D\},$$

$$\{D\} \rightarrow \{B\}, \{A, B, E\} \rightarrow \{F\}\}$$

Σ contains redundancies: some functional dependencies have redundant attributes (extraneous attributes), some functional dependencies are redundant.

Example

$$\Sigma = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C, D\},$$

$$\{D\} \rightarrow \{B\}, \{A, B, E\} \rightarrow \{F\}\}$$

We begin by simplifying the right-hand-sides of the functional dependencies to singletons.

$$\Sigma_1 = \{\{A\} \rightarrow \{B\}, \{A\} \rightarrow \{C\}, \{B\} \rightarrow \{C\},$$

$$\{B\} \rightarrow \{D\}, \{D\} \rightarrow \{B\}, \{A, B, E\} \rightarrow \{F\}\}$$

The two sets are equivalent.

$$\Sigma \equiv \Sigma_1$$

Proof.

We can transform one into the other using Union and Decomposition.

Example

$$\Sigma_1 = \{\{A\} \rightarrow \{B\}, \{A\} \rightarrow \{C\}, \{B\} \rightarrow \{C\},$$

$$\{B\} \rightarrow \{D\}, \{D\} \rightarrow \{B\}, \{A, B, E\} \rightarrow \{F\}\}$$

Σ_1 contains a functional dependency with a redundant attribute: $\{A, B, E\} \rightarrow \{B\}$ can be replaced by $\{A, E\} \rightarrow \{B\}$.

$$\Sigma_2 = \Sigma - \{\{A, B, E\} \rightarrow \{F\}\} \cup \{\{A, E\} \rightarrow \{F\}\}$$

$$\Sigma_2 = \{\{A\} \rightarrow \{B\}, \{A\} \rightarrow \{C\}, \{B\} \rightarrow \{C\},$$

$$\{B\} \rightarrow \{D\}, \{D\} \rightarrow \{B\}, \{A, E\} \rightarrow \{F\}\}$$

The two sets are equivalent.

$$\Sigma_1 \equiv \Sigma_2$$

Proof.

We prove that both sets of functional dependencies are equivalent by proving that functional dependencies from one can be derived from those of the other and vice versa.

- We prove that $\Sigma_1 \models \{A, E\} \rightarrow \{F\}$.
- We prove that $\Sigma_2 \models \{A, B, E\} \rightarrow \{F\}$.

Proof.

We prove that $\Sigma_1 \models \{A, E\} \rightarrow \{F\}$.

- We know that $\{A, B, E\} \rightarrow \{F\}$.
- We know that $\{A\} \rightarrow \{B\}$.
- Therefore $\{A, E\} \rightarrow \{A, B, E\}$ by Augmentation of (2) with $\{A, E\}$.
- Therefore $\{A, E\} \rightarrow \{F\}$ By transitivity of (3) and (1).

  Q.E.D.

Proof.

We prove that $\Sigma_2 \models \{A, B, E\} \rightarrow \{F\}$.

- We know that $\{A, E\} \rightarrow \{F\}$.
- We know that $\{A, E\} \subset \{A, B, E\}$.
- Therefore $\{A, B, E\} \rightarrow \{A, E\}$ by Reflexivity with (2).
- Therefore $\{A, B, E\} \rightarrow \{F\}$ by Transitivity of (3) and (1).

  Q.E.D.

Example

$$\Sigma_2 = \{\{A\} \to \{B\}, \{A\} \to \{C\}, \{B\} \to \{C\},$$

$$\{B\} \to \{D\}, \{D\} \to \{B\}, \{A, E\} \to \{F\}\}$$

Σ_2 contains a redundant functional dependency: $\{A\} \to \{C\}$ is not needed. We can remove it.

$$\Sigma_3 = \Sigma_2 - \{\{A\} \to \{C\}\}$$

$$\Sigma_3 = \{\{A\} \to \{B\}, \{B\} \to \{C\},$$

$$\{B\} \to \{D\}, \{D\} \to \{B\}, \{A, E\} \to \{F\}\}$$

The two sets are equivalent.

$$\Sigma_2 \equiv \Sigma_3$$

Proof.

We prove that both sets of functional dependencies are equivalent by proving that functional dependencies from one can be derived from those of the other and vice versa.
$\{A\} \rightarrow \{C\}$ can be obtained by Transitivity of $\{A\} \rightarrow \{B\}$ and $\{B\} \rightarrow \{C\}$

Example

$$\Sigma_3 = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\},$$

$$\{B\} \rightarrow \{D\}, \{D\} \rightarrow \{B\}, \{A, E\} \rightarrow \{F\}\}$$

We can regroup functional dependencies with the same left-hand-sides.

$$\Sigma_4 = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C, D\},$$

$$\{D\} \rightarrow \{B\}, \{A, E\} \rightarrow \{F\}\}$$

The two sets are equivalent.

$$\Sigma_3 \equiv \Sigma_4$$

Proof.

We can transform one into the other using Union and Decomposition.

Definition

A minimal cover (or covering) of a set of functional dependencies Σ is an equivalent set $\Sigma 3$ of functional depencies such that:

1. The right-hand-sides of the functional dependencies in $\Sigma 3$ are singletons.
2. The left-hand-sides of the functional dependencies in $\Sigma 3$ are minimal, namely, there are no redundant (extraneous) attributes.
3. $\Sigma 3$ is minimal, namely, there are no redundant functional dependency.

Theorem

The minimal cover of a set of functional dependencies Σ always exists but is not unique.

Theorem

A minimal cover of a set of functional dependencies Σ can be computed by the following steps in this order:

1. *Normalize the right-hand-sides of the functional dependencies to singletons.*
2. *Simplify the left-hand-sides of the functional dependencies by eliminating redundant (extraneous) attributes.*
3. *Remove the redundant functional dependencies.*

Proof.

It is quite obvious that the minimal cover is equivalent to Σ (see the example for outline of the proofs) and minimal (by definition) except for the fact that the three simplification steps must be done in the given sequence. This is to avoid interference.

Theorem

The above algorithm is non deterministic.

Theorem

Some minimal covers of a set of functional dependencies Σ cannot be found by the above algorithm.

Theorem

All minimal cover can be found by the algorithm applied on Σ^+.

Example

$\Sigma_1 = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{C\} \rightarrow \{A\}\}$
$\Sigma_2 = \{\{A\} \rightarrow \{C\}, \{C\} \rightarrow \{B\}, \{B\} \rightarrow \{A\}\}$

Definition

An extended minimal cover (or partitioned covering) of a set of functional dependencies Σ is an equivalent set $\Sigma 4$ obtained by regrouping the functional dependencies in a minimal cover that have the same right-hand-side.