Data Bases Data Mining

Foundations of databases: from functional dependencies to normal forms

Database Group

http://liris.cnrs.fr/ecoquery/dokuwiki/doku.php?id=enseignement:dbdm:start

March 1, 2017

Exemple

Let $\mathcal{U} = \{id, name, address, cnum, desc, grade\}$ a set of attributes to model students and courses. We consider the following database schemas :

- $R1 = \{Data\}$ with $schema(Data) = \mathcal{U}$[1].
- $R2 = \{Student, Course, Enrollment\}$ avec
  - $schema(Student) = \{id, name, address\}$
  - $schema(Course) = \{cnum, desc\}$
  - $schema(Enrollment) = \{id, cnum, grade\}$

How to compare these schemas?

- Which one is the "best"?
- Why?

[1] Similar to a spreadsheet.

Exemple

Data	*id*	*name*	*address*	*cnum*	*desc*	*grade*
	124	Jean	Paris	F234	Philo I	A
	456	Emma	Lyon	F234	Philo I	B
	789	Paul	Marseille	M321	Analyse I	C
	124	Jean	Paris	M321	Analyse I	A
	789	Paul	Marseille	CS24	BD I	B

Is there any problem here?

Exemple

Data	*id*	*name*	*address*	*cnum*	*desc*	*grade*
	124	Jean	Paris	F234	Philo I	A
	456	Emma	Lyon	F234	Philo I	B
	789	Paul	Marseille	M321	Analyse I	C
	124	Jean	Paris	M321	Analyse I	A
	789	Paul	Marseille	CS24	BD I	B

Is there any problem here?

Redundancies!

Redundancies

Data	*id*	*name*	*address*	*cnum*	*desc*	*grade*
	124	Jean	Paris	F234	Philo I	A
	456	Emma	Lyon	F234	Philo I	B
	789	Paul	Marseille	M321	Analyse I	C
	124	Jean	Paris	M321	Analyse I	A
	789	Paul	Marseille	CS24	BD I	B

Intuition on functional dependencies

- A student' *id* gives her/his name and address, so for each new enrollment, his/her name and address are duplicated!
- $\pi_{id,name,address}(Data)$ is the graph of a (partial) function $f : id \to name \times address$, similarly for $\pi_{cnum,desc}(Data)$
- $R2 = \{Student, Course, Enrollment\}$ is better than $R1 = \{Data\}$ because it avoids redundancies by keeping unrelated information (e.g., a student's name and a course' description) unrelated...

Functional is a theoretical tool to capture and reason on this phenomenon.

Functional dependencies: definition

Syntax

A *Functional Dependency (FD)* over a relation schema R is a formal expression of the form[2], with $X, Y \subseteq R$:

$$R : X \rightarrow Y$$

- $X \rightarrow Y$ is read "X functionally determines Y" or "X gives Y"
- A FD $X \rightarrow Y$ is trivial when $Y \subseteq X$
- A FD is standard when $X \neq \emptyset$.
- A set of attributes X is a key when $R : X \rightarrow R$

Semantics

Let r be a relation (a.k.a. *instance*) over R. The FD $R : X \rightarrow Y$ is *satisfied* by r, written $r \models R : X \rightarrow Y$, *iff*

$$\forall t_1, t_2 \in r . t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

[2] We write $X \rightarrow Y$ when R is clear from the context.

What constraint is implied by a *non-standard* FD?

What constraint is implied by a *non-standard* FD?

Why a *trivial* FD is said to be *trivial*?

Example

r	**A**	**B**	**C**	**D**
t_1	a_1	b_1	c_1	d_1
t_2	a_1	b_1	c_1	d_2
t_3	a_1	b_2	c_2	d_3
t_4	a_2	b_2	c_3	d_4

- $r \models AB \rightarrow C$ (no counter-example)
- $r \models D \rightarrow ABCD$ (no counter-example)
- $r \not\models AB \rightarrow D$ (e.g., $t_1[AB] = t_2[AB]$ but $t_1[D] \neq t_2[D]$)
- $r \not\models A \rightarrow C$ (e.g., $t_2[A] = t_3[A]$ but $t_2[C] \neq t_3[C]$)

Checking if a FD $R : X \to A$ holds in an instance

Using SQL (of course), with $X = \{A_1, \ldots, A_n\}$

```
SELECT A1, ..., An COUNT(DISTINCT A) AS NB
FROM R
GROUP BY A1, ..., An
HAVING COUNT(DISTINCT A) > 1;
```

Logical implication

Definition

Let F be a set of FDs on a relation schema R and let f be a single FD on R. We overload $\models$ for a set of FDs:

$$r \models F \text{ iff } \forall f \in F.r \models f$$

F logical (semantically) implies f, written

$$F \models f \text{ iff } \forall r.r \models F \Rightarrow r \models f$$

Example

With $F = \{A \rightarrow BCD, BC \rightarrow E\}$ and $r \models F$, the following hold as well:

- $r \models A \rightarrow CD$
- $r \models A \rightarrow E$

It can be proved using the definition of $\models$ and basic reasoning on projection of tuples.

Armstrong's System for FD

Armstrong's System

The following rules constitute the so call *Armstrong's system* for FDs:

- Reflexivity

$$\frac{Y \subseteq X}{X \rightarrow Y}$$

- Augmentation

$$\frac{X \rightarrow Y}{WX \rightarrow WY}$$

- Transitivity

$$\frac{X \rightarrow Y \qquad Y \rightarrow Z}{X \rightarrow Z}$$

Proof using Armstrong's system

Example

Let $\Sigma = \{A \to B, B \to C, CD \to E\}$ be a set of FDs on $\{A, B, C, D, E\}$.
We show that $\Sigma \vdash AD \to E$

$$\dfrac{\dfrac{\dfrac{A \to B \qquad B \to C}{A \to C}}{AD \to CD} \qquad CD \to E}{AD \to E}$$

Properties

Soundness and completeness

- The system is sound if $F \vdash f \Rightarrow F \models f$
  if there is a proof, the proof is valid
- The system is complete if $F \models f \Rightarrow F \vdash f$
  if it's valid, there is a proof

$$F \models \alpha \Leftrightarrow F \vdash \alpha$$

Soundness

Prove for every rule that, if its hypothesis are valid then its conclusion is valid as well.

Example: transitivity

Let r be ans instance on R s.t. $r \models X \to Y$ et $r \models Y \to Z$. Let $t_1, t_2 \in r$ be two tuples in r s.t. $t_1[X] = t_2[X]$, we have to show that $t_1[Z] = t_2[Z]$. Using $r \models X \to Y$ we deduce that $t_1[Y] = t_2[Y]$, then using $r \models Y \to Z$ we deduce that $t_1[Z] = t_2[Z]$. So the transitivity of FDs amounts to the transitivity of equality...

Additional rules

- Decomposition

$$\frac{X \to YZ}{X \to Y}$$

- Composition

$$\frac{X \to Y \qquad X \to Z}{X \to YZ}$$

- Pseudo-transitivity

$$\frac{X \to Y \qquad WY \to Z}{WX \to Z}$$

This rules are sound and can be (safely) added to Armstrong's system

Règles admissibles (1/3)

- Decomposition

$$\frac{X \to YZ}{X \to Y}$$

Autrement dit, si $\sum \vdash X \to YZ$,
alors $\sum \vdash X \to Y$

En effet,

— par Reflexivity, $YZ \to Y$
— par Transitivity avec $X \to YZ$,
on obtient $X \to Y$

- Reflexivity

$$\frac{Y \subseteq X}{X \to Y}$$

- Augmentation

$$\frac{X \to Y}{WX \to WY}$$

- Transitivity

$$\frac{X \to Y \quad Y \to Z}{X \to Z}$$

Règles admissibles (2/3)

- Composition

$$\frac{X \to Y \qquad X \to Z}{X \to YZ}$$

Autrement dit, si $\sum \vdash X \to Y$ et $\sum \vdash X \to Z$, alors $\sum \vdash X \to YZ$

- Reflexivity $\frac{Y \subseteq X}{X \to Y}$
- Augmentation $\frac{X \to Y}{WX \to WY}$
- Transitivity $\frac{X \to Y \qquad Y \to Z}{X \to Z}$

En effet,

— par Augmentation sur $X \to Y$, $XZ \to YZ$ (1)

— par Augmentation sur $X \to Z$, $XX \to XZ$

— or X=XX (concaténation notation pour union),
donc $X \to XZ$ (2)

— par Transitivity entre (1) et (2), on obtient $X \to YZ$

Règles admissibles (3/3)

- Pseudo-transitivity

$$\frac{X \to Y \qquad WY \to Z}{WX \to Z}$$

Autrement dit, si $\sum \vdash X \to Y$ et $\sum \vdash WY \to Z$,
alors $\sum \vdash WX \to Z$

En effet,

— par Augmentation sur $X \to Y$, $WX \to WY$ (1)

— par Transitivity avec $WY \to Z$, on obtient $WX \to Z$

- Reflexivity

$$\frac{Y \subseteq X}{X \to Y}$$

- Augmentation

$$\frac{X \to Y}{WX \to WY}$$

- Transitivity

$$\frac{X \to Y \qquad Y \to Z}{X \to Z}$$

Completeness

Formal proofs

A (formal) proof of f from Σ using Armstrong' system written $\Sigma \vdash f$ is a *sequence* $\langle f_0, \ldots, f_n \rangle$ of FDs s.t. $f_n = f$ et $\forall i \in [0..n]$:

- either $f_i \in \Sigma$;
- or f_i is the *conclusion* of a rule of which all its *antecedents* $f_0 \ldots f_p$ appear before f_i in the sequence.

Completeness: $\Sigma \models X \rightarrow Y \Rightarrow \Sigma \vdash X \rightarrow Y$

We need a clear distinction between

- the semantic closure of X: $X^+ = \{A \mid \Sigma \models X \rightarrow A\}$
- the syntactic closure of X: $X^\star = \{A \mid \Sigma \vdash X \rightarrow A\}$

Lemma: $\Sigma \vdash X \rightarrow Y \Leftrightarrow Y \subseteq X^\star$

Preuve du lemme

D’abord, dans ces définitions
A est un **attribut** (pas un ensemble)

(⇒) Si $\sum \vdash X \to Y$, pour tout $A \in Y$,

on peut démontrer $\sum \vdash X \to A$:

par la règle admissible Decomposition.

Lemma: $\Sigma \vdash X \to Y \Leftrightarrow Y \subseteq X^\star$

- the semantic closure of X: $X^+ = \{A \mid \Sigma \models X \to A\}$
- the syntactic closure of X: $X^\star = \{A \mid \Sigma \vdash X \to A\}$

- Reflexivity $\dfrac{Y \subseteq X}{X \to Y}$
- Augmentation $\dfrac{X \to Y}{WX \to WY}$
- Transitivity $\dfrac{X \to Y \quad Y \to Z}{X \to Z}$

- Decomposition $\dfrac{X \to YZ}{X \to Y}$
- Composition $\dfrac{X \to Y \quad X \to Z}{X \to YZ}$
- Pseudo-transitivity $\dfrac{X \to Y \quad WY \to Z}{WX \to Z}$

Preuve du lemme

Lemma: $\Sigma \vdash X \rightarrow Y \Leftrightarrow Y \subseteq X^{\star}$

- the semantic closure of X: $X^{+} = \{A \mid \Sigma \models X \rightarrow A\}$
- the syntactic closure of X: $X^{\star} = \{A \mid \Sigma \vdash X \rightarrow A\}$

D'abord, dans ces définitions
A est un **attribut** (pas un ensemble)

($\Leftarrow$) Soit Y=$A_1 \ldots A_n$ (attributs) $\subseteq X^*$.

I.e., on a une preuve de $\sum \vdash X \rightarrow A_i$ pour tout i.

On démontre par récurrence sur n que $\sum \vdash X \rightarrow Y$.

— Si n=0, c'est par Reflexivity.

— Sinon, on a $\sum \vdash X \rightarrow A_1 \ldots A_{n-1}$ par hypothèse de récurrence

et $\sum \vdash X \rightarrow A_n$ par hypothèse.

On conclut par la règle admissible Composition.

- Reflexivity $\dfrac{Y \subseteq X}{X \rightarrow Y}$
- Augmentation $\dfrac{X \rightarrow Y}{WX \rightarrow WY}$
- Transitivity $\dfrac{X \rightarrow Y \quad Y \rightarrow Z}{X \rightarrow Z}$
- Decomposition $\dfrac{X \rightarrow YZ}{X \rightarrow Y}$
- Composition $\dfrac{X \rightarrow Y \quad X \rightarrow Z}{X \rightarrow YZ}$
- Pseudo-transitivity $\dfrac{X \rightarrow Y \quad WY \rightarrow Z}{WX \rightarrow Z}$

Un autre lemme

De même, on peut démontrer:

Lemme'. $\Sigma \models X \to Y$ ssi $Y \subseteq X^+$.

La preuve est similaire, il suffit de remarquer que les règles du système d'Armstrong sont **correctes**, et peuvent donc être appliquées pour déduire des conséquences **sémantiques**.

Lemma: $\Sigma \vdash X \to Y \Leftrightarrow Y \subseteq X^\star$

- the semantic closure of X: $X^+ = \{A \mid \Sigma \models X \to A\}$
- the syntactic closure of X: $X^\star = \{A \mid \Sigma \vdash X \to A\}$

(Note: X^+ est noté X^* dans le Alice…)

- Reflexivity $\dfrac{Y \subseteq X}{X \to Y}$
- Augmentation $\dfrac{X \to Y}{WX \to WY}$
- Transitivity $\dfrac{X \to Y \quad Y \to Z}{X \to Z}$

- Decomposition $\dfrac{X \to YZ}{X \to Y}$
- Composition $\dfrac{X \to Y \quad X \to Z}{X \to YZ}$
- Pseudo-transitivity $\dfrac{X \to Y \quad WY \to Z}{WX \to Z}$

Completeness

$$\Sigma \models X \to Y \Rightarrow \Sigma \vdash X \to Y$$
$$\equiv \Sigma \nvdash X \to Y \Rightarrow \Sigma \not\models X \to Y$$
$$\equiv \Sigma \nvdash X \to Y \Rightarrow \exists r.(r \models \Sigma \wedge r \not\models X \to Y)$$

The crux is to find an instance r,
with $X^\star = X_1 \ldots X_n$ et $Z_1 \ldots Z_p = R \setminus X^\star$

r	X_1	$\ldots$	X_n	Z_1	$\ldots$	Z_p
s	x_1	$\ldots$	x_n	z_1	$\ldots$	z_p
t	x_1	$\ldots$	x_n	y_1	$\ldots$	y_p

$r \models \Sigma$ but $r \not\models X \to Y$

Complétude (1/5)

On commence par caractériser X^+ comme un plus petit point fixe.

- the semantic closure of X: $X^+ = \{A \mid \Sigma \models X \to A\}$
- the syntactic closure of X: $X^\star = \{A \mid \Sigma \vdash X \to A\}$

(Note: X^+ est noté X^* dans le Alice…)

Soit L le treillis des ensembles d'attributs contenant X.

Soit $F : L \to L$, $\quad F(W) \stackrel{\text{def}}{=} X \cup \bigcup\{Z \mid Y \to Z \text{ dans } \sum \text{ tq. } Y \subseteq W\}$

Lemme 1. X^+ est le plus petit point fixe de F.

Note: F est croissante sur poset fini avec plus petit élément X,
donc $X^+ = F^n(X)$ pour n assez grand.

Ceci servira à la preuve, et mènera aussi à un algorithme.

Complétude (2/5)

Soit $F : L \to L$,
$F(W) \stackrel{\text{def}}{=} X \cup \bigcup\{Z \mid Y \to Z \text{ dans } \sum \text{ tq. } Y \subseteq W\}$

- the semantic closure of X: $X^+ = \{A \mid \Sigma \models X \to A\}$
- the syntactic closure of X: $X^\star = \{A \mid \Sigma \vdash X \to A\}$

Lemme 1. X^+ est le plus petit point fixe lfp(F) de F.
Preuve (1/2). Pour tout $A \in F(X^+)$, soit $A \in X \subseteq X^+$, soit
il existe une fd $Y \to Z$ dans $\sum$ tq. $Y \subseteq X^+$, et $A \in Z$.
Donc $\sum \models X \to Y$ (Lemme')
Donc $\sum \models X \to A$ par la correction de Reflexivity et Transitivity.
I.e., $A \in X^+$.
Ceci montre que $F(X^+) \subseteq X^+$.
Or lfp(F)= $F^n(X)$ pour un certain n;
une récurrence sur n + la croissance de F donnent: lfp(F) $\subseteq X^+$.

Complétude (3/5)

C'est le point central

Soit $F : L \to L$,
$F(W) \stackrel{\text{déf}}{=} X \cup \cup\{Z \mid Y \to Z \text{ dans } \sum \text{ tq. } Y \subseteq W\}$

- semantic closure of X: $X^+ = \{A \mid \Sigma \models X \to A\}$
- the syntactic closure of X: $X^\star = \{A \mid \Sigma \vdash X \to A\}$

Lemme 1. X^+ est le plus petit point fixe lfp(F) de F.
Preuve (2/2). Supposons $X^+ \subsetneq$ lfp(F). On construit une BD D:

r	X_1	...	X_n	Z_1	...	Z_p
s	x_1	...	x_n	z_1	...	z_p
t	x_1	...	x_n	y_1	...	y_p

où $\{X_1, \ldots, X_n\}$=lfp(F) et $Z_1, \ldots, Z_p$ sont les autres attributs (X^+ contient un Z_i);
aussi, $y_j \neq z_j$ pour tout j entre 1 et p.

D satisfait toutes les fd $Y \to Z$ de $\sum$:
— si $Y \subseteq$ lfp(F)=$\{X_1, \ldots, X_n\}$ alors Z aussi (déf. d'un point fixe)
— sinon, Y contient un attribut Z_j, et il n'y a pas de couple de rangées distinctes ayant les mêmes valeurs pour cet attribut ($y_j \neq z_j$)

D donne les mêmes valeurs aux attributs de X des deux rangées.
Comme $Z_i \in X^+$, par déf. de X^+, on a $y_i = z_i$: contradiction. □

Complétude (4/5)

Soit $F : L \to L$,

$F(W) \overset{\text{def}}{=} X \cup \bigcup\{Z \mid Y \to Z \text{ dans } \Sigma \text{ tq. } Y \subseteq W\}$

- the semantic closure of X: $X^+ = \{A \mid \Sigma \models X \to A\}$
- the syntactic closure of X: $X^\star = \{A \mid \Sigma \vdash X \to A\}$

Lemme 2. *Pour tout n*, $F^n(X) \subseteq X^*$.

Preuve. Récurrence sur n.

— Pour n=0, $X \subseteq X^*$ par Reflexivity.

— Pour n≥1, pour tout $A \in F^n(X)$, soit $A \in X$ (Reflexivity)
soit il existe $Y \to Z$ dans Σ tq. $A \in Z$ et $Y \subseteq F^{n-1}(X)$.
Par hyp. réc., $Y \subseteq X^*$ par hyp. réc.,
donc $\Sigma \vdash X \to Y$ par: Lemma: $\Sigma \vdash X \to Y \Leftrightarrow Y \subseteq X^\star$
Donc $\Sigma \vdash X \to Z$ par Transitivity.
Donc $\Sigma \vdash X \to A$ par la règle admissible Decomposition.
Donc $A \in X^*$ par Lemma. □

- Reflexivity $\dfrac{Y \subseteq X}{X \to Y}$
- Augmentation $\dfrac{X \to Y}{WX \to WY}$
- Transitivity $\dfrac{X \to Y \quad Y \to Z}{X \to Z}$
- Decomposition $\dfrac{X \to YZ}{X \to Y}$
- Composition $\dfrac{X \to Y \quad X \to Z}{X \to YZ}$
- Pseudo-transitivity $\dfrac{X \to Y \quad WY \to Z}{WX \to Z}$

Complétude (5/5)

Soit $F : L \to L$,
$F(W) \stackrel{\text{def}}{=} X \cup \bigcup\{Z \mid Y \to Z \text{ dans } \Sigma \text{ tq. } Y \subseteq W\}$

Lemme 1. X^+ est le plus petit point fixe lfp(F) de F.
Lemme 2. *Pour tout n*, $F^n(X) \subseteq X^*$.

Or lfp(F) = $F^n(X)$ pour un certain *n*. Donc $X^+ \subseteq X^*$.
L'inclusion réciproque est la correction. Donc:

Théorème (complétude). $X^+ = X^*$.

- the semantic closure of X: $X^+ = \{A \mid \Sigma \models X \to A\}$
- the syntactic closure of X: $X^\star = \{A \mid \Sigma \vdash X \to A\}$

- Reflexivity $\dfrac{Y \subseteq X}{X \to Y}$
- Augmentation $\dfrac{X \to Y}{WX \to WY}$
- Transitivity $\dfrac{X \to Y \quad Y \to Z}{X \to Z}$

- Decomposition $\dfrac{X \to YZ}{X \to Y}$
- Composition $\dfrac{X \to Y \quad X \to Z}{X \to YZ}$
- Pseudo-transitivity $\dfrac{X \to Y \quad WY \to Z}{WX \to Z}$

Inference problem for FDs

Armstrong's system leads to a (inefficient) decision procedure for the *inference problem*.

Inference problem for FDs

Let F be a set of FDs and f a single FD, does $F \models f$ hold true?

Lemma: $F \models X \rightarrow Y$ iff $Y \subseteq X^+$

Thus, if we have an (efficient) algorithm to compute X^+, we can (efficiently) solve the inference problem:

1. Given Σ and $X \rightarrow Y$, compute X^+ w.r.t. Σ
2. Return $Y \subseteq X^+$

Closure algorithm: $Closure(\Sigma, X)$

Data: Σ a set of FDs, X a set of d'attributes.
Result: X^+, the closure of X w.r.t. Σ

$Cl := X$
$done := false$
while $(\neg done)$ **do**
  $done := true$
  forall $W \rightarrow Z \in \Sigma$ **do**
    if $W \subseteq Cl \wedge Z \not\subseteq Cl$ **then**
      $Cl := Cl \cup Z$
      $done := false$
return Cl

Algorithm 1: $Closure(\Sigma, X)$

Cet algorithme calcule les itérés de la fonction F utilisée dans la preuve de complétude du système d'Armstrong, partant du plus petit élément de L, à savoir X.
On en déduit la correction facilement.

How many times[3] do we compute $W \subseteq Cl \wedge Z \not\subseteq Cl$ w.r.t. $|\Sigma| = n$?

Cet algorithme est très proche de l'algorithme de calcul du plus petit modèle d'un ensemble de clauses de Horn, qui est en temps linéaire; on peut optimiser de même celui-ci pour qu'il tourne en temps linéaire

[3] at worst, using a bad strategy at line 5.

Second algorithm

Data: Σ a set of FDs, X a set of d'attributes.
Result: X^+, the closure of X w.r.t. Σ

```
for W → Z ∈ F do
    count[W → Z] := |W|
    for A ∈ W do
        list[A] := list[A] ∪ W → Z
closure := X, update := X
while update ≠ ∅ do
    Choose A ∈ update
    update := update \ {A}
    for W → Z ∈ list[A] do
        count[W → Z] := count[W → Z] − 1
        if count[W → Z] = 0 then
            update := update ∪ (Z \ closure)
            closure := closure ∪ Z
return closure
```

Algorithm 2: $Closure'(\Sigma, X)$

Example : AE^+

$$\Sigma = \{A \to I; AB \to E; BI \to E; CD \to I; E \to C\}$$

Initialization

$$\begin{array}{ll} List[A] = \{A \to D; AB \to E\} & count[A \to D] = 1 \\ List[B] = \{AB \to E; BI \to E\} & count[AB \to E] = 2 \\ List[C] = \{CD \to I\} & count[BI \to E] = 2 \\ List[D] = \{CD \to I\} & count[CD \to I] = 2 \\ List[E] = \{E \to C\} & count[E \to C] = 1 \\ List[I] = \{BI \to E\} & \end{array}$$

Cover

Cover of a set of FDs

With $F^+ = \{f \mid F \models f\}$, let Σ et Γ be two sets of FDs,
Γ is a cover of Σ *iff* $\Gamma^+ = \Sigma^+$

Data: F a set of FDs
Result: G a *minimal* (in cardinality) cover of F

$G := \emptyset$
for $X \to Y \in F$ **do**
  $G := G \cup \{X \to X^+\}$;
for $X \to X^+ \in G$ **do**
  if $G \setminus \{X \to X^+\} \vdash X \to X^+$ **then**
    $G := G \setminus \{X \to X^+\}$;
return G;

Algorithm 3: *Minimize*(F)

Normal forms

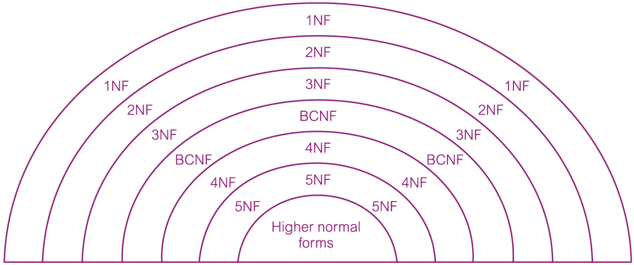

Figure 13.7 Diagrammatic illustration of the relationship between the normal forms.

Application of FD: Normalization

We write $\langle R, \Sigma \rangle$ with R a relation schema and Σ a set of FDs on R. A set of attribute X is a *minimal key* of $\langle R, \Sigma \rangle$ iff:

- X is a key of R (i.e., $X \to R$ holds)
- X is *minimal* w.r.t. set inclusion: $\forall . X' \subsetneq X \Rightarrow X' \not\to R$

Third Normal Form (3NF)

$\langle R, \Sigma \rangle$ is in 3NF iff, for all *non-trivial* FD $X \to A$ of Σ^+, one of the following conditions holds:

- X is a key of R
- A is a member of *at least* one minimal key of R[4]

Boyce-Codd Normal Form (BCNF)

$\langle R, \Sigma \rangle$ is in BCNF iff, for all *non-trivial* $X \to A$ of Σ^+, X is a key of R.

Informally, $\langle R, \Sigma \rangle$ is good when Σ is nothing but the key!

[4] An attribute that appears in *at least* one minimal key is said to be a *prime attribute*.

Example

3NF captures most of redundancies

- $\langle ABC, \{A \to B, B \to C\}\rangle$ is *not* in 3NF
  A is the unique *minimal* key. Considering $B \to C$, C is *not* prime and B is *not* a key. Clearly, ABC should be divided into AB and BC
- $\langle ABC, \{AB \to C, C \to B\}\rangle$ *is* in 3NF
  There are two *minimal* keys: AB and AC. Every attribute is prime so the 3NF condition holds. Unfortunately, some redundancies still hold but there is no way to decompose ABC into smaller relation without loss of FD!

BCNF captures all redundancies (expressed by FD)

- $\langle ABC, \{AB \to C, C \to B\}\rangle$ is *not* in BCNF
  Considering $C \to B$, C alone is not a key.

Synthesis algorithm

Data: R the set of all attributes
Data: Σ a set of FDs on R
Result: A decomposition $\mathbf{R}$ of R according to Σ

$F := \mathit{Reduce}(\mathit{Minimize}(\Sigma))$
for $X \to Y \in F$ **do**
    $\mathbf{R} := \mathbf{R} \cup \{XY\}$
for $R \in \mathbf{R}$ **do**
    if $\exists R'.R \subsetneq R'$ **then** $\mathbf{R} := \mathbf{R} \setminus \{R\}$;
$\mathit{Keys} := \{X \mid X \to U \wedge \forall Z.Z \subsetneq X \Rightarrow Z \not\to U\}$
if $\forall R \in \mathbf{R}.\ \nexists K \in \mathit{Cle}.K \subseteq R$ **then**
    $\mathit{pick}\ K \in \mathit{Cle}$
    $\mathbf{R} := \mathbf{R} \cup \{K\}$
return R

Algorithm 4: $\mathit{Synthesis}(\Sigma, U)$

End.