Data Modelling/Data Base Systems

VU 184.685/VU 184.686, WS 2020

Relational Design Theory – Functional Dependencies

Anela Lolić

Institute of Logic and Computation, TU Wien

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Acknowledgements

The slides are based on the slides (in German) of Sebastian Skritek.

The content is based on Chapter 6 of
(Kemper, Eickler: Datenbanksysteme – Eine Einführung).

For related literature in English see Chapter 19 of
(Ramakrishnan, Gehrke: Database Management Systems).

Overview

Aims

- fine-tuning of the relational schema
- quality of a relational schema:
  - satisfying consistency conditions
  - avoidance of redundancies
- modelling with data dependencies
  - functional dependencies
  - inclusion dependencies
  - compound dependencies

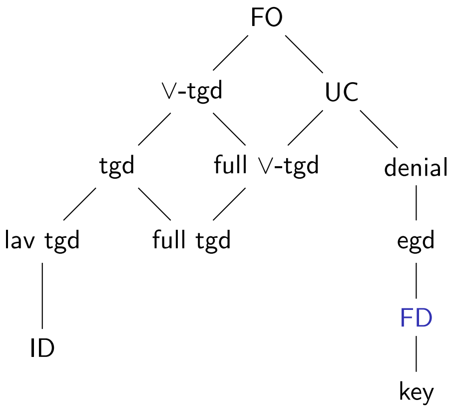
Aims

- basis: functional dependencies (FDs)
  - motivation
  - definition
  - determination
  - closure
  - canonical cover
  - key
- normalforms as quality criterion
- possible improvement of the relational schema
  - synthesis algorithm
  - decomposition

Functional Dependencies

- motivation
- definition
- determination
- closure
- equivalence
- canonical cover
- key

Integrity Constraints: an Essential DB-Tool



Application of Functional Dependencies

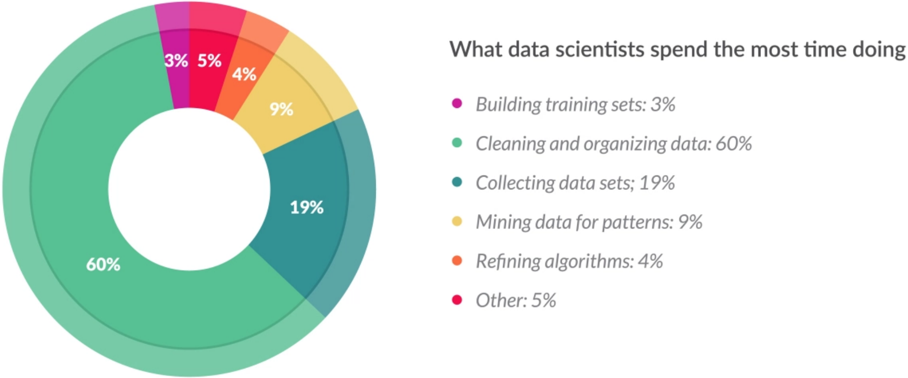
advantages of functional dependencies:

- simple semantics
- many problems are efficiently computable

use of functional dependencies for/in:

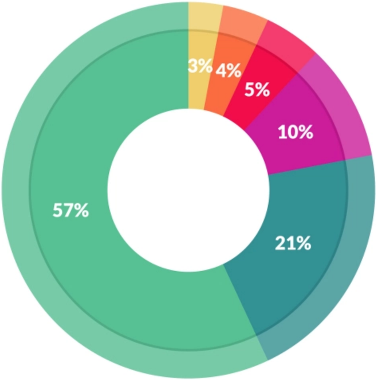
- consistency conditions for data bases
- determination of the quality of relational schemata
- "data exchange" and "data integration"
- "data cleaning"
- ...

Application: Data Cleaning



(Forbes, May 2016)
(Thanks to Emanuel Sallinger)

Application: Data Cleaning



What's the least enjoyable part of data science?

- *Building training sets: 10%*
- *Cleaning and organizing data: 57%*
- *Collecting data sets: 21%*
- *Mining data for patterns: 3%*
- *Refining algorithms: 4%*
- *Other: 5%*

(Forbes, May 2016)
(Thanks to Emanuel Sallinger)

Overview

- motivation
- definition
- determination
- closure
- equivalence
- canonical cover
- key

Functional Dependencies – Notation

Notation:

- schema $\mathcal{R} = \{A, B, C, D, \ldots, H\}$
- attribute: $A, B, C, \ldots$ attribute sets: $\alpha, \beta, \ldots$
- relation: R, tuple: $r, s, t, \ldots$ projection: $r.\alpha, t.\beta, \ldots$

- set difference (for $a \in \mathcal{A}$): $\mathcal{A} - a$ instead of $\mathcal{A} \setminus \{a\}$

Functional Dependencies

Definition (functional dependencies)

Let $\mathcal{R}$ be e relational schema and $\alpha \subseteq \mathcal{R}, \beta \subseteq \mathcal{R}$.
A functional dependency (FD) is a relationship $\alpha \rightarrow \beta$.

"α determines β"

Functional Dependencies

Definition (functional dependencies)

Let $\mathcal{R}$ be e relational schema and $\alpha \subseteq \mathcal{R}, \beta \subseteq \mathcal{R}$.
A functional dependency (FD) is a relationship $\alpha \rightarrow \beta$.

"α determines β"

Definition (semantics of functional dependencies)

A relation R satisfies a functional dependency (FD) $\alpha \rightarrow \beta$ if and only if for all tuples $r, t \in R$ with $r.\alpha = t.\alpha$ it holds that: $r.\beta = t.\beta$.

Functional Dependencies

Definition (functional dependencies)

Let $\mathcal{R}$ be e relational schema and $\alpha \subseteq \mathcal{R}, \beta \subseteq \mathcal{R}$.
A functional dependency (FD) is a relationship $\alpha \rightarrow \beta$.

"α determines β"

Definition (semantics of functional dependencies)

A relation R satisfies a functional dependency (FD) $\alpha \rightarrow \beta$ if and only if for all tuples $r, t \in R$ with $r.\alpha = t.\alpha$ it holds that: $r.\beta = t.\beta$.

$\alpha \rightarrow \beta$: if two tuples have the same value for all attributes in α, then they have the same values for all attributes in β.

Functional Dependencies

Definition (functional dependencies)

Let $\mathcal{R}$ be e relational schema and $\alpha \subseteq \mathcal{R}, \beta \subseteq \mathcal{R}$.
A functional dependency (FD) is a relationship $\alpha \rightarrow \beta$.

"α determines β"

Definition (semantics of functional dependencies)

A relation R satisfies a functional dependency (FD) $\alpha \rightarrow \beta$ if and only if for all tuples $r, t \in R$ with $r.\alpha = t.\alpha$ it holds that: $r.\beta = t.\beta$.

$\alpha \rightarrow \beta$: if two tuples have the same value for all attributes in α, then they have the same values for all attributes in β.

"the α-values uniquely (functionally) determine the β-values"

Functional Dependencies

Example

- a family tree implies the following FDs:
  - child $\rightarrow$ father, mother

family tree

child	mother	father	grandmother	grandfather
Sofie	Sabine	Alfons	Linde	Lothar
Sofie	Sabine	Alfons	Lisa	Hubert
Niklas	Sabine	Alfons	Linde	Lothar
Niklas	Sabine	Alfons	Lisa	Hubert

Functional Dependencies

Example

- a family tree implies the following FDs:
  - child $\rightarrow$ father, mother
  - child, grandmother $\rightarrow$ grandfather

family tree

child	mother	father	grandmother	grandfather
Sofie	Sabine	Alfons	Linde	Lothar
Sofie	Sabine	Alfons	Lisa	Hubert
Niklas	Sabine	Alfons	Linde	Lothar
Niklas	Sabine	Alfons	Lisa	Hubert

Functional Dependencies

Example

- a family tree implies the following FDs:
  - child $\rightarrow$ father, mother
  - child, grandmother $\rightarrow$ grandfather
  - child, grandfather $\rightarrow$ grandmother

family tree

child	mother	father	grandmother	grandfather
Sofie	Sabine	Alfons	Linde	Lothar
Sofie	Sabine	Alfons	Lisa	Hubert
Niklas	Sabine	Alfons	Linde	Lothar
Niklas	Sabine	Alfons	Lisa	Hubert

Functional Dependencies

Example

- a family tree implies the following FDs:
  - child $\rightarrow$ father, mother
  - child, grandmother $\rightarrow$ grandfather
  - child, grandfather $\rightarrow$ grandmother

family tree

child	mother	father	grandmother	grandfather
Sofie	Sabine	Alfons	Linde	Lothar
Sofie	Sabine	Alfons	Lisa	Hubert
Niklas	Sabine	Alfons	Linde	Lothar
Niklas	Sabine	Alfons	Lisa	Hubert
Sofie	...	...	Linde	Willi

Functional Dependencies

Example

- schema $\mathcal{R} = \{A, B, C, D\}$ of the relation R
- question: are the following FDs on R valid or not:

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

R

A	*B*	*C*	*D*
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

Functional Dependencies

Example

- schema $\mathcal{R} = \{A, B, C, D\}$ of the relation R
- question: are the following FDs on R valid or not:

$\{A\} \to \{B\}$ yes
$\{C, D\} \to \{B\}$
$\{B\} \to \{C\}$
$\{A, B\} \to \{C\}$
$\{B, C\} \to \{A\}$
$\{B\} \to \{A\}$

R			
A	*B*	*C*	*D*
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

Functional Dependencies

Example

- schema $\mathcal{R} = \{A, B, C, D\}$ of the relation R
- question: are the following FDs on R valid or not:

FD	
$\{A\} \to \{B\}$	yes
$\{C, D\} \to \{B\}$	yes
$\{B\} \to \{C\}$	
$\{A, B\} \to \{C\}$	
$\{B, C\} \to \{A\}$	
$\{B\} \to \{A\}$	

R			
A	*B*	*C*	*D*
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

Functional Dependencies

Example

- schema $\mathcal{R} = \{A, B, C, D\}$ of the relation R
- question: are the following FDs on R valid or not:

FD	
$\{A\} \to \{B\}$	yes
$\{C, D\} \to \{B\}$	yes
$\{B\} \to \{C\}$	no
$\{A, B\} \to \{C\}$	
$\{B, C\} \to \{A\}$	
$\{B\} \to \{A\}$	

R			
A	*B*	*C*	*D*
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

Functional Dependencies

Example

- schema $\mathcal{R} = \{A, B, C, D\}$ of the relation R
- question: are the following FDs on R valid or not:

$\{A\} \to \{B\}$	yes
$\{C, D\} \to \{B\}$	yes
$\{B\} \to \{C\}$	no
$\{A, B\} \to \{C\}$	yes
$\{B, C\} \to \{A\}$	
$\{B\} \to \{A\}$	

R			
A	*B*	*C*	*D*
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

Functional Dependencies

Example

- schema $\mathcal{R} = \{A, B, C, D\}$ of the relation R
- question: are the following FDs on R valid or not:

FD	
$\{A\} \rightarrow \{B\}$	yes
$\{C, D\} \rightarrow \{B\}$	yes
$\{B\} \rightarrow \{C\}$	no
$\{A, B\} \rightarrow \{C\}$	yes
$\{B, C\} \rightarrow \{A\}$	no
$\{B\} \rightarrow \{A\}$	

R			
A	*B*	*C*	*D*
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

Functional Dependencies

Example

- schema $\mathcal{R} = \{A, B, C, D\}$ of the relation R
- question: are the following FDs on R valid or not:

$\{A\} \rightarrow \{B\}$	yes
$\{C, D\} \rightarrow \{B\}$	yes
$\{B\} \rightarrow \{C\}$	no
$\{A, B\} \rightarrow \{C\}$	yes
$\{B, C\} \rightarrow \{A\}$	no
$\{B\} \rightarrow \{A\}$	no

R			
A	*B*	*C*	*D*
a4	b2	c4	d3
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c3	d2
a3	b2	c4	d3

Satisfying a FD

find: a way to check whether a given relation R satisfies a FD $\alpha \rightarrow \beta$:

Satisfying a FD

find: a way to check whether a given relation R satisfies a FD $\alpha \to \beta$:

- the following SQL query has an empty result

```
select * from R r1, R r2
where r1.α = r2.α and r1.β ≠ r2.β;
```

Satisfying a FD

find: a way to check whether a given relation R satisfies a FD $\alpha \rightarrow \beta$:

- the following SQL query has an empty result

```
select * from R r1, R r2
where r1.α = r2.α and r1.β ≠ r2.β;
```

```
select * from R r1 where exists(
  select * from R r2 where r1.α = r2.α
                     and r1.β ≠ r2.β);
```

Satisfying a FD

find: a way to check whether a given relation R satisfies a FD $\alpha \rightarrow \beta$:

- the following SQL query has an empty result

```
select * from R r1, R r2
where r1.α = r2.α and r1.β ≠ r2.β;
```

```
select * from R r1 where exists(
  select * from R r2 where r1.α = r2.α
                      and r1.β ≠ r2.β);
```

- for all possible values c the result of the query

$$\pi_\beta(\sigma_{\alpha=c}(R))$$

contains at most one tuple

Satisfying a FD

algorithm to check whether a given
relation R satisfies the FD $\alpha \rightarrow \beta$:

input: $(R, \alpha \rightarrow \beta)$: relation R and a FD $\alpha \rightarrow \beta$

output: yes if FD is satisfied, no otherwise

satisfiability $(R, \alpha \rightarrow \beta)$

- sort R by the values of α
- in case all groups of tuples with same values of α have the same values for β: output(yes), output(no) otherwise

Satisfying a FD

algorithm to check whether a given
relation R satisfies the FD $\alpha \to \beta$:

input: $(R, \alpha \to \beta)$: relation R and a FD $\alpha \to \beta$

output: yes if FD is satisfied, no otherwise

satisfiability $(R, \alpha \to \beta)$

- sort R by the values of α
- in case all groups of tuples with same values of α have the same values for β: output(yes), output(no) otherwise

running time of satisfiability is determined by the expense of the sorting - $O(nlogn)$

Overview

- motivation
- definition
- determination
- closure
- equivalence
- canonical cover
- key

Determination of FDs

Example

given: information about professors based on the following attributes
professor:{[persNr, name, rank, room, city, street, zipcode, area code, state, population, state government]}
question: Which functional dependencies can be determined based on the semantics of the world to be modelled?

Determination of FDs

Example

given: information about professors based on the following attributes
professor:$\{$[persNr, name, rank, room, city, street, zipcode, area code, state, population, state government]$\}$
question: Which functional dependencies can be determined based on the semantics of the world to be modelled?

- persNr is a candidate key:
  $\{$persNr$\}$ $\rightarrow$ $\{$persNr, name, rank, room, city, street, zipcode, area code, state, population, state government$\}$

Determination of FDs

Example

given: information about professors based on the following attributes professor:{[persNr, name, rank, room, city, street, zipcode, area code, state, population, state government]}
question: Which functional dependencies can be determined based on the semantics of the world to be modelled?

- persNr is a candidate key:
  {persNr} → {persNr, name, rank, room, city, street, zipcode, area code, state, population, state government}
- cities are unique within a state:
  {city, state} → {population, area code}

Determination of FDs

Example

- the zipcode identifies city, state and population:
  $\{\text{zipcode}\} \rightarrow \{\text{state, city, population}\}$

Determination of FDs

Example

- the zipcode identifies city, state and population:
  $\{\text{zipcode}\} \rightarrow \{\text{state, city, population}\}$
- the zipcode does not change within the street of a city:
  $\{\text{state, city, street}\} \rightarrow \{\text{zipcode}\}$

Determination of FDs

Example

- the zipcode identifies city, state and population:
  $\{\text{zipcode}\} \rightarrow \{\text{state, city, population}\}$
- the zipcode does not change within the street of a city:
  $\{\text{state, city, street}\} \rightarrow \{\text{zipcode}\}$
- state government stores the governor's party:
  $\{\text{state}\} \rightarrow \{\text{state government}\}$

Determination of FDs

Example

- the zipcode identifies city, state and population:
  $\{\text{zipcode}\} \rightarrow \{\text{state, city, population}\}$
- the zipcode does not change within the street of a city:
  $\{\text{state, city, street}\} \rightarrow \{\text{zipcode}\}$
- state government stores the governor's party:
  $\{\text{state}\} \rightarrow \{\text{state government}\}$
- there can only be one professor assigned to a room:
  $\{\text{room}\} \rightarrow \{\text{persNr}\}$

Overview

- motivation
- definition
- determination
- closure
- equivalence
- canonical cover
- key

Closure of a Set of Attributes resp. FDs

given: set of FDs F

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name

Closure of a Set of Attributes resp. FDs

given: set of FDs F

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name

question 1: given an additional set of attributes γ, which attributes in γ can be functionally determined by F?

Closure of a Set of Attributes resp. FDs

given: set of FDs F

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name

question 1: given an additional set of attributes γ, which attributes in γ can be functionally determined by F?

question 2: which other FDs can be derived from F?

Closure of a Set of Attributes

given: a set of attributes γ and a set of FDs F

question: which attributes of γ are functionally determined by F?

example: $\{\text{room} \rightarrow \text{persNr}, \text{persNr} \rightarrow \text{name}\}, \{\text{room}\}$
$\Rightarrow \{\text{room}, \text{persNr}, \text{name}\}$

Closure of a Set of Attributes

given: a set of attributes γ and a set of FDs F

question: which attributes of γ are functionally determined by F?

example: $\{\text{room} \to \text{persNr}, \text{persNr} \to \text{name}\}, \{\text{room}\}$
$\Rightarrow \{\text{room}, \text{persNr}, \text{name}\}$

Definition (closure of a set of attributes)

The set of attributes γ^+ which functionally depend on γ are called the closure of the set of attributes γ.

Closure of a Set of Attributes

computation via algorithm attrclosure:

input: (F, γ): set of FDs F and a set of attributes γ

output: set of attributes γ^+.

attrclosure (F, γ)

$\gamma^+ = \gamma$

while $\exists(\alpha \to \beta) \in F$ with $\alpha \subseteq \gamma^+$ and $\beta \nsubseteq \gamma^+$ **do**

$\quad \gamma^+ := \gamma^+ \cup \beta$

return(γ^+)

Closure of a Set of Attributes

Example

let $F = \{RS \rightarrow T, U \rightarrow VX, RX \rightarrow W, T \rightarrow RU\}$

attrclosure$(F, \{T\})$:

attrclosure$(F, \{RS\})$:

Closure of a Set of Attributes

Example

let $F = \{RS \rightarrow T, U \rightarrow VX, RX \rightarrow W, T \rightarrow RU\}$

attrclosure$(F, \{T\})$: $\{R, T, U, V, W, X\}$

attrclosure$(F, \{RS\})$:

Closure of a Set of Attributes

Example

let $F = \{RS \rightarrow T, U \rightarrow VX, RX \rightarrow W, T \rightarrow RU\}$

attrclosure($F, \{T\}$): $\{R, T, U, V, W, X\}$

attrclosure($F, \{RS\}$): $\{R, S, T, U, V, W, X\}$

Closure of FDs

given: set of FDs F

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name

Closure of FDs

given: set of FDs F

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name

question 1: given a set of attributes γ, which attributes of γ are functionally determined by F?

Closure of FDs

given: set of FDs F

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name

question 1: given a set of attributes γ, which attributes of γ are functionally determined by F?

question 2: which other FDs can be derived from F?

Closure of FDs

problem: F set of FDs; which other FDs can be derived?

Closure of FDs

problem: F set of FDs; which other FDs can be derived?

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name $\Rightarrow$ room $\rightarrow$ name

Closure of FDs

problem: F set of FDs; which other FDs can be derived?

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name $\Rightarrow$ room $\rightarrow$ name

Definition ($F_1 \models F_2$)

The set F_2 of FDs can be derived from the set F_1 of FDs, if every relation R which satisfies all FDs in F_1 also satisfies all FDs in F_2.

Closure of FDs

problem: F set of FDs; which other FDs can be derived?

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name $\Rightarrow$ room $\rightarrow$ name

Definition ($F_1 \models F_2$)

The set F_2 of FDs can be derived from the set F_1 of FDs, if every relation R which satisfies all FDs in F_1 also satisfies all FDs in F_2.

Definition (closure of FDs)

The set of all FDs derivable from F is called the closure F^+ of F.

Closure of FDs

problem: F set of FDs; which other FDs can be derived?

example: room $\rightarrow$ persNr, persNr $\rightarrow$ name $\Rightarrow$ room $\rightarrow$ name

Definition ($F_1 \models F_2$)

The set F_2 of FDs can be derived from the set F_1 of FDs, if every relation R which satisfies all FDs in F_1 also satisfies all FDs in F_2.

Definition (closure of FDs)

The set of all FDs derivable from F is called the closure F^+ of F.

compare to mathematics: V a set of vectors. The set of all vectors that can be derived from V with linear combinations is called the linear closure of V.

Deriving FDs through the Attribute Closure

FD $\alpha \rightarrow \beta$: the values for α functionally determine the values for β.

attribute closure: all attributes γ^+, whose values of γ are functionally determined by F

Deriving FDs through the Attribute Closure

FD $\alpha \rightarrow \beta$: the values for α functionally determine the values for β.

attribute closure: all attributes γ^+, whose values of γ are functionally determined by F

Theorem

Given a set of FDs F and a set of attributes γ it holds that:

$$F \models \{\gamma \rightarrow attrclosure(F, \gamma)\}$$

Deriving FDs through the Attribute Closure

FD $\alpha \rightarrow \beta$: the values for α functionally determine the values for β.

attribute closure: all attributes γ^+, whose values of γ are functionally determined by F

Theorem

Given a set of FDs F and a set of attributes γ it holds that:

$$F \models \{\gamma \rightarrow attrclosure(F, \gamma)\}$$

Moreover: $F \models \{\alpha \rightarrow \beta\} \Leftrightarrow \beta \subseteq \text{attrclosure}(F, \alpha)$

Deriving FDs via Armstrong Axioms

Construction of the closure F^+ of F via Armstrong axioms (1974).

Deriving FDs via Armstrong Axioms

Construction of the closure F^+ of F via Armstrong axioms (1974).

Theorem

The Armstrong axioms are complete (construct all implicit FDs) and sound (construct only valid FDs).

The Armstrong Axioms

reflexivity: Let β be a subset of α ($\beta \subseteq \alpha$) then $\alpha \rightarrow \beta$. In particular $\alpha \rightarrow \alpha$.

The Armstrong Axioms

reflexivity: Let β be a subset of α ($\beta \subseteq \alpha$) then $\alpha \to \beta$. In particular $\alpha \to \alpha$.

augmentation: Whenever $\alpha \to \beta$, then $\alpha\gamma \to \beta\gamma$.

The Armstrong Axioms

reflexivity: Let β be a subset of α ($\beta \subseteq \alpha$) then $\alpha \rightarrow \beta$. In particular $\alpha \rightarrow \alpha$.

augmentation: Whenever $\alpha \rightarrow \beta$, then $\alpha\gamma \rightarrow \beta\gamma$.

transitivity: If $\alpha \rightarrow \beta$, $\beta \rightarrow \gamma$ then $\alpha \rightarrow \gamma$.

The Armstrong Axioms

reflexivity: Let β be a subset of α ($\beta \subseteq \alpha$) then $\alpha \to \beta$. In particular $\alpha \to \alpha$.

augmentation: Whenever $\alpha \to \beta$, then $\alpha\gamma \to \beta\gamma$.

transitivity: If $\alpha \to \beta$, $\beta \to \gamma$ then $\alpha \to \gamma$.

additional axioms then simplify the derivation of the closure:

The Armstrong Axioms

reflexivity: Let β be a subset of α ($\beta \subseteq \alpha$) then $\alpha \to \beta$. In particular $\alpha \to \alpha$.

augmentation: Whenever $\alpha \to \beta$, then $\alpha\gamma \to \beta\gamma$.

transitivity: If $\alpha \to \beta$, $\beta \to \gamma$ then $\alpha \to \gamma$.

additional axioms then simplify the derivation of the closure:

union: $\alpha \to \beta$ and $\alpha \to \gamma \Rightarrow \alpha \to \beta\gamma$.

The Armstrong Axioms

reflexivity: Let β be a subset of α ($\beta \subseteq \alpha$) then $\alpha \rightarrow \beta$. In particular $\alpha \rightarrow \alpha$.

augmentation: Whenever $\alpha \rightarrow \beta$, then $\alpha\gamma \rightarrow \beta\gamma$.

transitivity: If $\alpha \rightarrow \beta$, $\beta \rightarrow \gamma$ then $\alpha \rightarrow \gamma$.

additional axioms then simplify the derivation of the closure:

union: $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma \Rightarrow \alpha \rightarrow \beta\gamma$.

decomposition: $\alpha \rightarrow \beta\gamma \Rightarrow \alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$

important: we can always impose only one attribute on the right side.

The Armstrong Axioms

reflexivity: Let β be a subset of α ($\beta \subseteq \alpha$) then $\alpha \rightarrow \beta$. In particular $\alpha \rightarrow \alpha$.

augmentation: Whenever $\alpha \rightarrow \beta$, then $\alpha\gamma \rightarrow \beta\gamma$.

transitivity: If $\alpha \rightarrow \beta$, $\beta \rightarrow \gamma$ then $\alpha \rightarrow \gamma$.

additional axioms then simplify the derivation of the closure:

union: $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma \Rightarrow \alpha \rightarrow \beta\gamma$.

decomposition: $\alpha \rightarrow \beta\gamma \Rightarrow \alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$

important: we can always impose only one attribute on the right side.

pseudo transitivity: $\alpha \rightarrow \beta$ and $\gamma\beta \rightarrow \delta \Rightarrow \alpha\gamma \rightarrow \delta$.

The Armstrong Axioms

Example

deriving FD {zipcode} → {state government} from the remaining FDs in the example schema professors:

we know: {zipcode} → {state, city, population} and
{state} →{state government}

The Armstrong Axioms

Example

deriving FD {zipcode} → {state government} from the remaining FDs in the example schema professors:

we know: {zipcode} → {state, city, population} and
{state} →{state government}

decomposition of {zipcode} → {state, city, population}:

{zipcode} → {state}, {zipcode} → {city}, {zipcode} → {population}

The Armstrong Axioms

Example

deriving FD {zipcode} $\rightarrow$ {state government} from the remaining FDs in the example schema professors:

we know: {zipcode} $\rightarrow$ {state, city, population} and
{state} $\rightarrow$ {state government}

decomposition of {zipcode} $\rightarrow$ {state, city, population}:

{zipcode} $\rightarrow$ {state}, {zipcode} $\rightarrow$ {city}, {zipcode} $\rightarrow$ {population}

transitivity of {zipcode} $\rightarrow$ {state}, {state} $\rightarrow$ {state government}:

{zipcode} $\rightarrow$ {state government}

The Armstrong Axioms

Example

deriving union from reflexivity, augmentation and transitivity:

given: $\alpha \to \beta$, $\alpha \to \gamma$

to show: $\alpha \to \beta\gamma$

The Armstrong Axioms

Example

deriving union from reflexivity, augmentation and transitivity:

given: $\alpha \to \beta$, $\alpha \to \gamma$

to show: $\alpha \to \beta\gamma$

step 1: augmentation of $\alpha \to \beta$: $\alpha \to \alpha\beta$

The Armstrong Axioms

Example

deriving union from reflexivity, augmentation and transitivity:

given: $\alpha \rightarrow \beta$, $\alpha \rightarrow \gamma$

to show: $\alpha \rightarrow \beta\gamma$

step 1: augmentation of $\alpha \rightarrow \beta$: $\alpha \rightarrow \alpha\beta$

step 2: augmentation of $\alpha \rightarrow \gamma$: $\alpha\beta \rightarrow \beta\gamma$

The Armstrong Axioms

Example

deriving union from reflexivity, augmentation and transitivity:

given: $\alpha \to \beta$, $\alpha \to \gamma$

to show: $\alpha \to \beta\gamma$

step 1: augmentation of $\alpha \to \beta$: $\alpha \to \alpha\beta$

step 2: augmentation of $\alpha \to \gamma$: $\alpha\beta \to \beta\gamma$

step 3: transitivity of $\alpha \to \alpha\beta$, $\alpha\beta \to \beta\gamma$: $\alpha \to \beta\gamma$

Overview

- motivation
- definition
- determination
- closure
- equivalence
- canonical cover
- key

Equivalence of Sets of FDs

given: sets F_1, F_2 of FDs

question: Do F_1 and F_2 describe the same set of FDs?

Equivalence of Sets of FDs

Definition (equivalence of FDs)

Two sets F, G of FDs are equivalent ($F \equiv G$), if their closures are equivalent, i.e. $F^+ = G^+$.

mathematics: two sets of vectors are "equivalent", if they span the same vector space.

Equivalence of Sets of FDs

Definition (equivalence of FDs)

Two sets F, G of FDs are equivalent ($F \equiv G$), if their closures are equivalent, i.e. $F^+ = G^+$.

mathematics: two sets of vectors are "equivalent", if they span the same vector space.

obviously: $F \equiv G$ if and only if

- $F \subseteq G^+$ and
- $G \subseteq F^+$

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$ $\qquad$ $F_2 = \{B \rightarrow A, C \rightarrow B, A \rightarrow C\}$

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $\qquad$ $F_2 = \{B \to A, C \to B, A \to C\}$

$$F_1 \equiv F_2?$$

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$ $\qquad$ $F_2 = \{B \rightarrow A, C \rightarrow B, A \rightarrow C\}$

$$F_1 \equiv F_2?$$

$$F_1 \subseteq F_2^+?$$

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2?$

$F_1 \subseteq F_2^+?$

$F_2 \subseteq F_1^+?$

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $\qquad$ $F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2$?

$F_1 \subseteq F_2^+$? $\qquad$ $F_2 \subseteq F_1^+$?

$A \to B$:

$B \to C$:

$C \to A$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$

$F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2$?

$F_1 \subseteq F_2^+$?

$A \to B$:

$B \to C$:

$C \to A$:

$F_2 \subseteq F_1^+$?

$B \to A$:

$C \to B$:

$A \to C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$ $F_2 = \{B \rightarrow A, C \rightarrow B, A \rightarrow C\}$

$F_1 \equiv F_2?$

$F_1 \subseteq F_2^+?$

$A \rightarrow B: \checkmark$

$B \rightarrow C:$

$C \rightarrow A:$

$F_2 \subseteq F_1^+?$

$B \rightarrow A:$

$C \rightarrow B:$

$A \rightarrow C:$

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2$?

$F_1 \subseteq F_2^+$?

$A \to B$: ✓ transitivity of $A \to C$, $C \to B$

$B \to C$:

$C \to A$:

$F_2 \subseteq F_1^+$?

$B \to A$:

$C \to B$:

$A \to C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$ $F_2 = \{B \rightarrow A, C \rightarrow B, A \rightarrow C\}$

$F_1 \equiv F_2$?

$F_1 \subseteq F_2^+$?

$A \rightarrow B$: ✓ transitivity of $A \rightarrow C$, $C \rightarrow B$

$B \rightarrow C$: ✓

$C \rightarrow A$:

$F_2 \subseteq F_1^+$?

$B \rightarrow A$:

$C \rightarrow B$:

$A \rightarrow C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2$?

$F_1 \subseteq F_2^+$?

$A \to B$: ✓ transitivity of $A \to C$, $C \to B$

$B \to C$: ✓ transitivity of $B \to A$, $A \to C$

$C \to A$:

$F_2 \subseteq F_1^+$?

$B \to A$:

$C \to B$:

$A \to C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $\qquad$ $F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2$?

$F_1 \subseteq F_2^+$?

$A \to B$: ✓ transitivity of $A \to C$, $C \to B$

$B \to C$: ✓ transitivity of $B \to A$, $A \to C$

$C \to A$: ✓

$F_2 \subseteq F_1^+$?

$B \to A$:

$C \to B$:

$A \to C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$ $\qquad$ $F_2 = \{B \rightarrow A, C \rightarrow B, A \rightarrow C\}$

$F_1 \equiv F_2?$

$F_1 \subseteq F_2^+?$

$A \rightarrow B$: ✓ transitivity of $A \rightarrow C$, $C \rightarrow B$

$B \rightarrow C$: ✓ transitivity of $B \rightarrow A$, $A \rightarrow C$

$C \rightarrow A$: ✓ transitivity of $C \rightarrow B$, $B \rightarrow A$

$F_2 \subseteq F_1^+?$

$B \rightarrow A$:

$C \rightarrow B$:

$A \rightarrow C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $\quad$ $F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2$?

$F_1 \subseteq F_2^+$?

$A \to B$: ✓ transitivity of $A \to C$, $C \to B$

$B \to C$: ✓ transitivity of $B \to A$, $A \to C$

$C \to A$: ✓ transitivity of $C \to B$, $B \to A$

$F_2 \subseteq F_1^+$?

$B \to A$: ✓

$C \to B$:

$A \to C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$ $\qquad$ $F_2 = \{B \rightarrow A, C \rightarrow B, A \rightarrow C\}$

$$F_1 \equiv F_2?$$

$F_1 \subseteq F_2^+$?

- $A \rightarrow B$: ✓ transitivity of $A \rightarrow C$, $C \rightarrow B$
- $B \rightarrow C$: ✓ transitivity of $B \rightarrow A$, $A \rightarrow C$
- $C \rightarrow A$: ✓ transitivity of $C \rightarrow B$, $B \rightarrow A$

$F_2 \subseteq F_1^+$?

- $B \rightarrow A$: ✓ transitivity of $B \rightarrow C$, $C \rightarrow A$
- $C \rightarrow B$:
- $A \rightarrow C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $\qquad$ $F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2$?

$F_1 \subseteq F_2^+$?

$A \to B$: ✓ transitivity of $A \to C$, $C \to B$

$B \to C$: ✓ transitivity of $B \to A$, $A \to C$

$C \to A$: ✓ transitivity of $C \to B$, $B \to A$

$F_2 \subseteq F_1^+$?

$B \to A$: ✓ transitivity of $B \to C$, $C \to A$

$C \to B$: ✓

$A \to C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$ $\quad$ $F_2 = \{B \rightarrow A, C \rightarrow B, A \rightarrow C\}$

$F_1 \equiv F_2?$

$F_1 \subseteq F_2^+?$

$A \rightarrow B$: ✓ transitivity of $A \rightarrow C$, $C \rightarrow B$

$B \rightarrow C$: ✓ transitivity of $B \rightarrow A$, $A \rightarrow C$

$C \rightarrow A$: ✓ transitivity of $C \rightarrow B$, $B \rightarrow A$

$F_2 \subseteq F_1^+?$

$B \rightarrow A$: ✓ transitivity of $B \rightarrow C$, $C \rightarrow A$

$C \rightarrow B$: ✓ transitivity of $C \rightarrow A$, $A \rightarrow B$

$A \rightarrow C$:

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2?$

$F_1 \subseteq F_2^+?$

$A \to B$: ✓ transitivity of $A \to C$, $C \to B$

$B \to C$: ✓ transitivity of $B \to A$, $A \to C$

$C \to A$: ✓ transitivity of $C \to B$, $B \to A$

$F_2 \subseteq F_1^+?$

$B \to A$: ✓ transitivity of $B \to C$, $C \to A$

$C \to B$: ✓ transitivity of $C \to A$, $A \to B$

$A \to C$: ✓

Equivalence of Sets of FDs

Example (Armstrong axioms)

$F_1 = \{A \to B, B \to C, C \to A\}$ $\quad$ $F_2 = \{B \to A, C \to B, A \to C\}$

$F_1 \equiv F_2$?

$F_1 \subseteq F_2^+$?

$A \to B$: ✓ transitivity of $A \to C$, $C \to B$

$B \to C$: ✓ transitivity of $B \to A$, $A \to C$

$C \to A$: ✓ transitivity of $C \to B$, $B \to A$

$F_2 \subseteq F_1^+$?

$B \to A$: ✓ transitivity of $B \to C$, $C \to A$

$C \to B$: ✓ transitivity of $C \to A$, $A \to B$

$A \to C$: ✓ transitivity of $A \to B$, $B \to C$

Equivalence of Sets of FDs

Example (attribute closure)

$F_1 = \{A \to B, B \to C, C \to A\}$ $F_2 = \{B \to A, C \to B, A \to C\}$

$$F_1 \equiv F_2?$$

$F_1 \subseteq F_2^+?$

$A \to B$:

$B \to C$:

$C \to A$:

$F_2 \subseteq F_1^+?$

$B \to A$:

$C \to B$:

$A \to C$:

Equivalence of Sets of FDs

Example (attribute closure)

$F_1 = \{A \to B, B \to C, C \to A\}$ $F_2 = \{B \to A, C \to B, A \to C\}$

$$F_1 \equiv F_2?$$

$F_1 \subseteq F_2^+?$

$A \to B$: ✓

$B \to C$: ✓

$C \to A$: ✓

$F_2 \subseteq F_1^+?$

$B \to A$: ✓

$C \to B$: ✓

$A \to C$: ✓

Equivalence of Sets of FDs

Example (attribute closure)

$F_1 = \{A \to B, B \to C, C \to A\}$ $F_2 = \{B \to A, C \to B, A \to C\}$

$$F_1 \equiv F_2?$$

$F_1 \subseteq F_2^+?$

$A \to B$: ✓
$B \in \text{attrclosure}(F_2, \{A\})$

$B \to C$: ✓

$C \to A$: ✓

$F_2 \subseteq F_1^+?$

$B \to A$: ✓

$C \to B$: ✓

$A \to C$: ✓

Equivalence of Sets of FDs

Example (attribute closure)

$F_1 = \{A \to B, B \to C, C \to A\}$ $\quad$ $F_2 = \{B \to A, C \to B, A \to C\}$

$$F_1 \equiv F_2?$$

$F_1 \subseteq F_2^+?$

$A \to B$: ✓
$B \in$ attrclosure$(F_2, \{A\})$

$B \to C$: ✓
$C \in$ attrclosure$(F_2, \{B\})$

$C \to A$: ✓

$F_2 \subseteq F_1^+?$

$B \to A$: ✓

$C \to B$: ✓

$A \to C$: ✓

Equivalence of Sets of FDs

Example (attribute closure)

$F_1 = \{A \to B, B \to C, C \to A\}$ $F_2 = \{B \to A, C \to B, A \to C\}$

$$F_1 \equiv F_2?$$

$F_1 \subseteq F_2^+$?

$A \to B$: ✓ $B \in \text{attrclosure}(F_2, \{A\})$

$B \to C$: ✓ $C \in \text{attrclosure}(F_2, \{B\})$

$C \to A$: ✓ $A \in \text{attrclosure}(F_2, \{C\})$

$F_2 \subseteq F_1^+$?

$B \to A$: ✓

$C \to B$: ✓

$A \to C$: ✓

Equivalence of Sets of FDs

Example (attribute closure)

$F_1 = \{A \to B, B \to C, C \to A\}$ $F_2 = \{B \to A, C \to B, A \to C\}$

$$F_1 \equiv F_2?$$

$F_1 \subseteq F_2^+$?

$A \to B$: ✓
$B \in$ attrclosure$(F_2, \{A\})$

$B \to C$: ✓
$C \in$ attrclosure$(F_2, \{B\})$

$C \to A$: ✓
$A \in$ attrclosure$(F_2, \{C\})$

$F_2 \subseteq F_1^+$?

$B \to A$: ✓
$A \in$ attrclosure$(F_1, \{B\})$

$C \to B$: ✓

$A \to C$: ✓

Equivalence of Sets of FDs

Example (attribute closure)

$F_1 = \{A \to B, B \to C, C \to A\}$ $\quad$ $F_2 = \{B \to A, C \to B, A \to C\}$

$$F_1 \equiv F_2?$$

$F_1 \subseteq F_2^+?$

$A \to B$: ✓
$B \in \text{attrclosure}(F_2, \{A\})$

$B \to C$: ✓
$C \in \text{attrclosure}(F_2, \{B\})$

$C \to A$: ✓
$A \in \text{attrclosure}(F_2, \{C\})$

$F_2 \subseteq F_1^+?$

$B \to A$: ✓
$A \in \text{attrclosure}(F_1, \{B\})$

$C \to B$: ✓
$B \in \text{attrclosure}(F_1, \{C\})$

$A \to C$: ✓

Equivalence of Sets of FDs

Example (attribute closure)

$F_1 = \{A \to B, B \to C, C \to A\}$ $\quad$ $F_2 = \{B \to A, C \to B, A \to C\}$

$$F_1 \equiv F_2?$$

$F_1 \subseteq F_2^+?$

$A \to B$: ✓ $B \in \text{attrclosure}(F_2, \{A\})$

$B \to C$: ✓ $C \in \text{attrclosure}(F_2, \{B\})$

$C \to A$: ✓ $A \in \text{attrclosure}(F_2, \{C\})$

$F_2 \subseteq F_1^+?$

$B \to A$: ✓ $A \in \text{attrclosure}(F_1, \{B\})$

$C \to B$: ✓ $B \in \text{attrclosure}(F_1, \{C\})$

$A \to C$: ✓ $C \in \text{attrclosure}(F_1, \{A\})$

Equivalence of Sets of FDs

given: sets of FDs F_1, F_2

to show: $F_1 \equiv F_2$ resp. $F_1 \not\equiv F_2$?

Equivalence of Sets of FDs

given: sets of FDs F_1, F_2

to show: $F_1 \equiv F_2$ resp. $F_1 \not\equiv F_2$?

$F_1 \equiv F_2$: show that $F_1 \subseteq F_2^+$ and $F_2 \subseteq F_1^+$

Equivalence of Sets of FDs

given: sets of FDs F_1, F_2

to show: $F_1 \equiv F_2$ resp. $F_1 \not\equiv F_2$?

$F_1 \equiv F_2$: show that $F_1 \subseteq F_2^+$ and $F_2 \subseteq F_1^+$

$F_1 \not\equiv F_2$: find FD $\alpha \rightarrow \beta \in F_1^+$ such that $\alpha \rightarrow \beta \notin F_2^+$ (or vice versa)

Equivalence of Sets of FDs

given: sets of FDs F_1, F_2

to show: $F_1 \equiv F_2$ resp. $F_1 \not\equiv F_2$?

$F_1 \equiv F_2$: show that $F_1 \subseteq F_2^+$ and $F_2 \subseteq F_1^+$

$F_1 \not\equiv F_2$: find FD $\alpha \to \beta \in F_1^+$ such that $\alpha \to \beta \notin F_2^+$ (or vice versa)

show that $\alpha \to \beta \notin F_2^+$ if and only if $\beta \notin \text{attrclosure}(F_2, \alpha)$

Overview

- motivation
- definition
- determination
- closure
- equivalence
- canonical cover
- key

Canonical Cover

problem: find the shortest possible representation of FDs (“basis”)

Canonical Cover

problem: find the shortest possible representation of FDs ("basis")

solution: the canonical cover

mathematics: basis of a vector space

Canonical Cover

Definition (canonical cover)

F_C is called canonical cover of a set of FDs F if the following criteria are satisfied:

1. $F_C^+ = F^+$ (F_C is equivalent to F)
2. in F_C there are no FDs that contain superfluous attributes
3. each left side of a FD in F_C is unique

Canonical Cover

Definition (canonical cover)

F_C is called canonical cover of a set of FDs F if the following criteria are satisfied:

1. $F_C^+ = F^+$ (F_C is equivalent to F)
2. in F_C there are no FDs that contain superfluous attributes
3. each left side of a FD in F_C is unique

Theorem

To each set F of FDs there is a canonical cover F_C.

mathematics: to each set of vectors that span a vector space there is a base.

Computation of the Canonical Cover (1)

the construction of a canonical cover is obtained directly form the definition:

1. split all FDs using decomposition on the right side (equivalence is guaranteed)

Computation of the Canonical Cover (1)

the construction of a canonical cover is obtained directly form the definition:

1. split all FDs using decomposition on the right side (equivalence is guaranteed)
2. reduce superfluous attributes as follows:

Computation of the Canonical Cover (1)

the construction of a canonical cover is obtained directly form the definition:

1. split all FDs using decomposition on the right side (equivalence is guaranteed)
2. reduce superfluous attributes as follows:
    1. apply left reduction to each FD $\alpha \rightarrow B \in F$: can we delete an attribute from α such that the results remains equivalent to the original set of FDs?

Computation of the Canonical Cover (1)

the construction of a canonical cover is obtained directly form the definition:

1. split all FDs using decomposition on the right side (equivalence is guaranteed)
2. reduce superfluous attributes as follows:
   1. apply left reduction to each FD $\alpha \to B \in F$: can we delete an attribute from α such that the results remains equivalent to the original set of FDs?
   2. apply right reduction to each (remaining) FD $\alpha \to B \in F$: can B be reduced such that the result remains equivalent to the original set of FDs?

Computation of the Canonical Cover (1)

the construction of a canonical cover is obtained directly form the definition:

1. split all FDs using decomposition on the right side (equivalence is guaranteed)
2. reduce superfluous attributes as follows:
    1. apply left reduction to each FD $\alpha \to B \in F$: can we delete an attribute from α such that the results remains equivalent to the original set of FDs?
    2. apply right reduction to each (remaining) FD $\alpha \to B \in F$: can B be reduced such that the result remains equivalent to the original set of FDs?
3. combine FDs using union (equivalence is guaranteed)

Computation of the Canonical Cover (1)

the construction of a canonical cover is obtained directly form the definition:

1. split all FDs using decomposition on the right side (equivalence is guaranteed)
2. reduce superfluous attributes as follows:
    1. apply left reduction to each FD $\alpha \rightarrow B \in F$: can we delete an attribute from α such that the results remains equivalent to the original set of FDs?
    2. apply right reduction to each (remaining) FD $\alpha \rightarrow B \in F$: can B be reduced such that the result remains equivalent to the original set of FDs?
3. combine FDs using union (equivalence is guaranteed)

To ensure obtaining only equivalent results during the reduction steps we will use the algorithm attrclosure.

Computation of the Canonical Cover (2)

1. split all FDs using decomposition on the right side

Computation of the Canonical Cover (2)

1. split all FDs using decomposition on the right side
2. reduce superfluous attributes as follows:
   1. apply left reduction to each FD $\alpha \rightarrow B \in F$:

      $\forall A \in \alpha$: it holds that $B \in \text{attrclosure}(F, \alpha - A)$ (is A superfluous?)
      if yes: replace $\alpha \rightarrow B$ in F by $(\alpha - A) \rightarrow B$

Computation of the Canonical Cover (2)

1. split all FDs using decomposition on the right side
2. reduce superfluous attributes as follows:
   1. apply left reduction to each FD $\alpha \rightarrow B \in F$:

      $\forall A \in \alpha$: it holds that $B \in \text{attrclosure}(F, \alpha - A)$ (is A superfluous?)
      if yes: replace $\alpha \rightarrow B$ in F by $(\alpha - A) \rightarrow B$

   2. apply right reduction to each (remaining) FD $\alpha \rightarrow B \in F$:

      does it hold that $B \in \text{attrclosure}(F - (\alpha \rightarrow B), \alpha)$ (is B resp. $\alpha \rightarrow B$ superfluous?)
      if yes delete $\alpha \rightarrow B$

Computation of the Canonical Cover (2)

1. split all FDs using decomposition on the right side
2. reduce superfluous attributes as follows:
   1. apply left reduction to each FD $\alpha \rightarrow B \in F$:

      $\forall A \in \alpha$: it holds that $B \in \text{attrclosure}(F, \alpha - A)$ (is A superfluous?)
      if yes: replace $\alpha \rightarrow B$ in F by $(\alpha - A) \rightarrow B$

   2. apply right reduction to each (remaining) FD $\alpha \rightarrow B \in F$:

      does it hold that $B \in \text{attrclosure}(F - (\alpha \rightarrow B), \alpha)$ (is B resp. $\alpha \rightarrow B$ superfluous?)
      if yes delete $\alpha \rightarrow B$
3. combine the FDs using union

Explanation for . . .

. . . the left reduction:

$$B \in \text{attrclosure}(F, \alpha - A) \Leftrightarrow F \equiv (F \setminus \{\alpha \to B\}) \cup \{(\alpha - A) \to B\}$$

$$(\text{because } B \in \text{attrclosure}(F, \alpha - A) \Leftrightarrow F \models (\alpha - A) \to B)$$

Explanation for ...

...the left reduction:

$$B \in \text{attrclosure}(F, \alpha - A) \Leftrightarrow F \equiv (F \setminus \{\alpha \to B\}) \cup \{(\alpha - A) \to B\}$$

$$(\text{because } B \in \text{attrclosure}(F, \alpha - A) \Leftrightarrow F \models (\alpha - A) \to B)$$

...the right reduction

$$B \in \text{attrclosure}(F - (\alpha \to B), \alpha) \Leftrightarrow F \equiv F - (\alpha \to B)$$

$$(\text{because } B \in \text{attrclosure}(F - (\alpha \to B), \alpha) \Leftrightarrow F - (\alpha \to B) \models \alpha \to B)$$

Computation of the Canonical Cover (Example)

Example

$F = \{A \to B, B \to C, AB \to C\}$

1. decomposition not necessary

Computation of the Canonical Cover (Example)

Example

$F = \{A \rightarrow B, B \rightarrow C, AB \rightarrow C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

      $A \rightarrow B$: already reduced
      $B \rightarrow C$: already reduced

Computation of the Canonical Cover (Example)

Example

$F = \{A \rightarrow B, B \rightarrow C, AB \rightarrow C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

$$\begin{aligned} A \rightarrow B&: \text{ already reduced} \\ B \rightarrow C&: \text{ already reduced} \\ AB \rightarrow C&: \ C \in \text{attrclosure}(F, A) \end{aligned}$$

Computation of the Canonical Cover (Example)

Example

$F = \{A \to B, B \to C, AB \to C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

$$\begin{aligned} A \to B&: \text{ already reduced} \\ B \to C&: \text{ already reduced} \\ AB \to C&: \; C \in \text{attrclosure}(F, A) \text{ yes} \\ &\Rightarrow F := \{A \to B, B \to C, A \to C\} \end{aligned}$$

Computation of the Canonical Cover (Example)

Example

$F = \{A \to B, B \to C, AB \to C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

      $A \to B$: already reduced

      $B \to C$: already reduced

      $AB \to C$: $C \in \text{attrclosure}(F, A)$ yes

      $\Rightarrow F := \{A \to B, B \to C, A \to C\}$
   2. right reduction:

      $A \to B$: $B \in \text{attrclosure}(F \setminus \{A \to B\}, A)$

Computation of the Canonical Cover (Example)

Example

$F = \{A \to B, B \to C, AB \to C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

      $A \to B$: already reduced

      $B \to C$: already reduced

      $AB \to C$: $C \in \text{attrclosure}(F, A)$ yes

      $\Rightarrow F := \{A \to B, B \to C, A \to C\}$
   2. right reduction:

      $A \to B$: $B \in \text{attrclosure}(F \setminus \{A \to B\}, A)$ no

Computation of the Canonical Cover (Example)

Example

$F = \{A \rightarrow B, B \rightarrow C, AB \rightarrow C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

      $A \rightarrow B$: already reduced
      $B \rightarrow C$: already reduced
      $AB \rightarrow C$: $C \in \text{attrclosure}(F, A)$ yes
      $\Rightarrow F := \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$
   2. right reduction:

      $A \rightarrow B$: $B \in \text{attrclosure}(F \setminus \{A \rightarrow B\}, A)$ no
      $B \rightarrow C$: $C \in \text{attrclosure}(F \setminus \{B \rightarrow C\}, B)$

Computation of the Canonical Cover (Example)

Example

$F = \{A \to B, B \to C, AB \to C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

$$\begin{aligned} A \to B&: \text{ already reduced} \\ B \to C&: \text{ already reduced} \\ AB \to C&: \ C \in \text{attrclosure}(F, A) \text{ yes} \\ &\Rightarrow F := \{A \to B, B \to C, A \to C\} \end{aligned}$$

   2. right reduction:

$$\begin{aligned} A \to B&: \ B \in \text{attrclosure}(F \setminus \{A \to B\}, A) \text{ no} \\ B \to C&: \ C \in \text{attrclosure}(F \setminus \{B \to C\}, B) \text{ no} \end{aligned}$$

Computation of the Canonical Cover (Example)

Example

$F = \{A \to B, B \to C, AB \to C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

      $A \to B$: already reduced
      $B \to C$: already reduced
      $AB \to C$: $C \in \text{attrclosure}(F, A)$ yes
      $\Rightarrow F := \{A \to B, B \to C, A \to C\}$

   2. right reduction:

      $A \to B$: $B \in \text{attrclosure}(F \setminus \{A \to B\}, A)$ no
      $B \to C$: $C \in \text{attrclosure}(F \setminus \{B \to C\}, B)$ no
      $A \to C$: $C \in \text{attrclosure}(F \setminus \{A \to C\}, A)$

Computation of the Canonical Cover (Example)

Example

$F = \{A \to B, B \to C, AB \to C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

      $A \to B$: already reduced
      $B \to C$: already reduced
      $AB \to C$: $C \in \text{attrclosure}(F, A)$ yes
      $\Rightarrow F := \{A \to B, B \to C, A \to C\}$

   2. right reduction:

      $A \to B$: $B \in \text{attrclosure}(F \setminus \{A \to B\}, A)$ no
      $B \to C$: $C \in \text{attrclosure}(F \setminus \{B \to C\}, B)$ no
      $A \to C$: $C \in \text{attrclosure}(F \setminus \{A \to C\}, A)$ yes
      $\Rightarrow F := \{A \to B, B \to C\}$

Computation of the Canonical Cover (Example)

Example

$F = \{A \rightarrow B, B \rightarrow C, AB \rightarrow C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

      $A \rightarrow B$: already reduced
      $B \rightarrow C$: already reduced
      $AB \rightarrow C$: $C \in \mathrm{attrclosure}(F, A)$ yes
      $\Rightarrow F := \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$
   2. right reduction:

      $A \rightarrow B$: $B \in \mathrm{attrclosure}(F \setminus \{A \rightarrow B\}, A)$ no
      $B \rightarrow C$: $C \in \mathrm{attrclosure}(F \setminus \{B \rightarrow C\}, B)$ no
      $A \rightarrow C$: $C \in \mathrm{attrclosure}(F \setminus \{A \rightarrow C\}, A)$ yes
      $\Rightarrow F := \{A \rightarrow B, B \rightarrow C\}$
3. union not applicable

Computation of the Canonical Cover (Example)

Example

$F = \{A \rightarrow B, B \rightarrow C, AB \rightarrow C\}$

1. decomposition not necessary
2. reduction
   1. left reduction:

      $A \rightarrow B$: already reduced
      $B \rightarrow C$: already reduced
      $AB \rightarrow C$: $C \in \text{attrclosure}(F, A)$ yes
      $\Rightarrow F := \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$
   2. right reduction:

      $A \rightarrow B$: $B \in \text{attrclosure}(F \setminus \{A \rightarrow B\}, A)$ no
      $B \rightarrow C$: $C \in \text{attrclosure}(F \setminus \{B \rightarrow C\}, B)$ no
      $A \rightarrow C$: $C \in \text{attrclosure}(F \setminus \{A \rightarrow C\}, A)$ yes
      $\Rightarrow F := \{A \rightarrow B, B \rightarrow C\}$
3. union not applicable $\Rightarrow F_C := \{A \rightarrow B, B \rightarrow C\}$

Canonical Cover

Example

$F = \{A \to BD, AC \to E, CD \to E, E \to A, D \to C\}$

Canonical Cover

Example

$F = \{A \rightarrow BD, AC \rightarrow E, CD \rightarrow E, E \rightarrow A, D \rightarrow C\}$

1. decomposition:
   $F = \{A \rightarrow B, A \rightarrow D, AC \rightarrow E, CD \rightarrow E, E \rightarrow A, D \rightarrow C\}$

Canonical Cover

Example

$F = \{A \rightarrow BD, AC \rightarrow E, CD \rightarrow E, E \rightarrow A, D \rightarrow C\}$

1. decomposition:
   $F = \{A \rightarrow B, A \rightarrow D, AC \rightarrow E, CD \rightarrow E, E \rightarrow A, D \rightarrow C\}$
2. left reduction:

   $A \rightarrow B$: ok

   $A \rightarrow D$: ok

   $E \rightarrow A$: ok

   $D \rightarrow C$: ok

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Canonical Cover

Example

$F = \{A \to BD, AC \to E, CD \to E, E \to A, D \to C\}$

1. decomposition:
   $F = \{A \to B, A \to D, AC \to E, CD \to E, E \to A, D \to C\}$
2. left reduction:

   $A \to B$: ok

   $A \to D$: ok

   $E \to A$: ok

   $D \to C$: ok

   $AC \to E$: $E \in \text{attrclosure}(F, A)$ yes $\Rightarrow$
   $F = \{A \to B, A \to D, A \to E, CD \to E, E \to A, D \to C\}$

FAKULTÄT FÜR INFORMATIK Faculty of Informatics

Canonical Cover

Example

$F = \{A \to BD, AC \to E, CD \to E, E \to A, D \to C\}$

1. decomposition:
   $F = \{A \to B, A \to D, AC \to E, CD \to E, E \to A, D \to C\}$
2. left reduction:

   $A \to B$: ok

   $A \to D$: ok

   $E \to A$: ok

   $D \to C$: ok

   $AC \to E$: $E \in \text{attrclosure}(F, A)$ yes $\Rightarrow$
   $F = \{A \to B, A \to D, A \to E, CD \to E, E \to A, D \to C\}$

   $CD \to E$: $E \in \text{attrclosure}(F, C)$ no
   $E \in \text{attrclosure}(F, D)$ yes $\Rightarrow$
   $F = \{A \to B, A \to D, A \to E, D \to E, E \to A, D \to C\}$

Canonical Cover

Example

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

3. right reduction:

   $A \rightarrow B$: $B \in \text{attrclosure}(F \setminus \{A \rightarrow B\}, A)$ no

Canonical Cover

Example

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

3. right reduction:

$A \to B$: $B \in \text{attrclosure}(F \setminus \{A \to B\}, A)$ no
$A \to D$: $D \in \text{attrclosure}(F \setminus \{A \to D\}, A)$ no

Canonical Cover

Example

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

3. right reduction:

$A \to B$: $B \in \text{attrclosure}(F \setminus \{A \to B\}, A)$ no

$A \to D$: $D \in \text{attrclosure}(F \setminus \{A \to D\}, A)$ no

$A \to E$: $E \in \text{attrclosure}(F \setminus \{A \to E\}, A)$ yes $\Rightarrow$

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

Canonical Cover

Example

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

3. right reduction:

$A \to B$: $B \in \text{attrclosure}(F \setminus \{A \to B\}, A)$ no
$A \to D$: $D \in \text{attrclosure}(F \setminus \{A \to D\}, A)$ no
$A \to E$: $E \in \text{attrclosure}(F \setminus \{A \to E\}, A)$ yes $\Rightarrow$
$F = \{A \to B, A \to D, D \to E, E \to A, D \to C\}$
$D \to E$: $E \in \text{attrclosure}(F \setminus \{D \to E\}, D)$ no

Canonical Cover

Example

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

3. right reduction:

$$
\begin{aligned}
A \to B&: B \in \text{attrclosure}(F \setminus \{A \to B\}, A) \text{ no} \\
A \to D&: D \in \text{attrclosure}(F \setminus \{A \to D\}, A) \text{ no} \\
A \to E&: E \in \text{attrclosure}(F \setminus \{A \to E\}, A) \text{ yes} \Rightarrow \\
&\quad F = \{A \to B, A \to D, D \to E, E \to A, D \to C\} \\
D \to E&: E \in \text{attrclosure}(F \setminus \{D \to E\}, D) \text{ no} \\
E \to A&: A \in \text{attrclosure}(F \setminus \{E \to A\}, E) \text{ no}
\end{aligned}
$$

Canonical Cover

Example

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

3. right reduction:

$A \to B$: $B \in \text{attrclosure}(F \setminus \{A \to B\}, A)$ no
$A \to D$: $D \in \text{attrclosure}(F \setminus \{A \to D\}, A)$ no
$A \to E$: $E \in \text{attrclosure}(F \setminus \{A \to E\}, A)$ yes $\Rightarrow$
$F = \{A \to B, A \to D, D \to E, E \to A, D \to C\}$
$D \to E$: $E \in \text{attrclosure}(F \setminus \{D \to E\}, D)$ no
$E \to A$: $A \in \text{attrclosure}(F \setminus \{E \to A\}, E)$ no
$D \to C$: $C \in \text{attrclosure}(F \setminus \{D \to C\}, D)$ no

Canonical Cover

Example

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

3. right reduction:

$A \to B$: $B \in \text{attrclosure}(F \setminus \{A \to B\}, A)$ no
$A \to D$: $D \in \text{attrclosure}(F \setminus \{A \to D\}, A)$ no
$A \to E$: $E \in \text{attrclosure}(F \setminus \{A \to E\}, A)$ yes $\Rightarrow$
$F = \{A \to B, A \to D, D \to E, E \to A, D \to C\}$
$D \to E$: $E \in \text{attrclosure}(F \setminus \{D \to E\}, D)$ no
$E \to A$: $A \in \text{attrclosure}(F \setminus \{E \to A\}, E)$ no
$D \to C$: $C \in \text{attrclosure}(F \setminus \{D \to C\}, D)$ no

4. combination: $F_C = \{A \to BD, D \to EC, E \to A\}$

Overview

- motivation
- definition
- determination
- closure
- equivalence
- canonical cover
- key

Key

Definition (key)

$\gamma \subseteq \mathcal{R}$ is a key candidate or key, if the following conditions are satisfied:

Key

Definition (key)

$\gamma \subseteq \mathcal{R}$ is a key candidate or key, if the following conditions are satisfied:

1. $\gamma \rightarrow \mathcal{R}$
2. γ is minimal, i.e. for all $A \in \gamma$: $(\gamma - \{A\}) \not\rightarrow \mathcal{R}$

Key

Definition (key)

$\gamma \subseteq \mathcal{R}$ is a key candidate or key, if the following conditions are satisfied:

1. $\gamma \rightarrow \mathcal{R}$
2. γ is minimal, i.e. for all $A \in \gamma$: $(\gamma - \{A\}) \not\rightarrow \mathcal{R}$

in the relational model: primary key for the combination of tables via primary and foreign key

Key

Definition (key)

$\gamma \subseteq \mathcal{R}$ is a key candidate or key, if the following conditions are satisfied:

1. $\gamma \rightarrow \mathcal{R}$
2. γ is minimal, i.e. for all $A \in \gamma$: $(\gamma - \{A\}) \not\rightarrow \mathcal{R}$

in the relational model: primary key for the combination of tables via primary and foreign key

Definition (super key)

$\gamma \subseteq \mathcal{R}$ is a super key, if $\gamma \rightarrow \mathcal{R}$

- no minimality for super keys
- a super key is a superset of a key

Key

Definition (key)

$\gamma \subseteq \mathcal{R}$ is a key candidate or key, if the following conditions are satisfied:

1. $\gamma \to \mathcal{R}$
2. γ is minimal, i.e. for all $A \in \gamma$: $(\gamma - \{A\}) \not\to \mathcal{R}$

in the relational model: primary key for the combination of tables via primary and foreign key

Definition (super key)

$\gamma \subseteq \mathcal{R}$ is a super key, if $\gamma \to \mathcal{R}$

- no minimality for super keys
- a super key is a superset of a key

$\Rightarrow$ every key is also a super key

Classification of Keys

Example

A city is described by its name, the corresponding state, the area code and the population.
question: which FDs hold in this scenario?

Classification of Keys

Example

A city is described by its name, the corresponding state, the area code and the population.

question: which FDs hold in this scenario?

- city names are unique within a state, i.e.
  $\{$name, state$\} \rightarrow \{$area code, population$\}$

Classification of Keys

Example

A city is described by its name, the corresponding state, the area code and the population.
question: which FDs hold in this scenario?

- city names are unique within a state, i.e.
  $\{$name, state$\} \rightarrow \{$area code, population$\}$
- several cities can have the same area code, but only if they have different names, i.e.
  $\{$name, area code$\} \rightarrow \{$state, population$\}$

Classification of Keys

Example

A city is described by its name, the corresponding state, the area code and the population.
question: which FDs hold in this scenario?

- city names are unique within a state, i.e.
  $\{\text{name, state}\} \rightarrow \{\text{area code, population}\}$
- several cities can have the same area code, but only if they have different names, i.e.
  $\{\text{name, area code}\} \rightarrow \{\text{state, population}\}$

question: which key candidates do we obtain based on the FDs?

Classification of Keys

Example

A city is described by its name, the corresponding state, the area code and the population.

question: which FDs hold in this scenario?

- city names are unique within a state, i.e.
  $\{$name, state$\} \rightarrow \{$area code, population$\}$
- several cities can have the same area code, but only if they have different names, i.e.
  $\{$name, area code$\} \rightarrow \{$state, population$\}$

question: which key candidates do we obtain based on the FDs?

- $\{$name, state$\}$

Classification of Keys

Example

A city is described by its name, the corresponding state, the area code and the population.

question: which FDs hold in this scenario?

- city names are unique within a state, i.e.
  $\{\text{name, state}\} \rightarrow \{\text{area code, population}\}$
- several cities can have the same area code, but only if they have different names, i.e.
  $\{\text{name, area code}\} \rightarrow \{\text{state, population}\}$

question: which key candidates do we obtain based on the FDs?

- $\{\text{name, state}\}$
- $\{\text{name, area code}\}$

Classification of Keys via attrclosure

problem: find key candidates of a relation R based on the given FDs

Definition (key (revisited))

$\gamma \subseteq \mathcal{R}$ is a key candidate or a key if the following conditions are satisfied:

1. $\gamma \rightarrow \mathcal{R}$
2. γ is minimal, i.e.. $\forall A \in \gamma : (\gamma - \{A\}) \not\rightarrow \mathcal{R}$

Classification of Keys via attrclosure

problem: find key candidates of a relation R based on the given FDs

Definition (key (revisited))

$\gamma \subseteq \mathcal{R}$ is a key candidate or a key if the following conditions are satisfied:

1. $\gamma \rightarrow \mathcal{R}$
2. γ is minimal, i.e.. $\forall A \in \gamma : (\gamma - \{A\}) \not\rightarrow \mathcal{R}$

1. $\gamma \rightarrow \mathcal{R}$, when attrclosure$(F, \gamma) = \mathcal{R}$

Classification of Keys via attrclosure

problem: find key candidates of a relation R based on the given FDs

Definition (key (revisited))

$\gamma \subseteq \mathcal{R}$ is a key candidate or a key if the following conditions are satisfied:

1. $\gamma \to \mathcal{R}$
2. γ is minimal, i.e.. $\forall A \in \gamma : (\gamma - \{A\}) \not\to \mathcal{R}$

1. $\gamma \to \mathcal{R}$, when attrclosure$(F, \gamma) = \mathcal{R}$
2. minimality is satisfied, if for every attribute A in γ:
attrclosure$(F, \gamma - \{A\}) \neq \mathcal{R}$

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \to BDAE\}$

- time consuming procedure: trying out all the one-element, two-element, three-element key candidates using attrclosure

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \rightarrow BDAE\}$

- time consuming procedure: trying out all the one-element, two-element, three-element key candidates using attrclosure
- alternative: using the following heuristics: all attributes, that do not occur on the right side cannot be derived via attrclosure and have therefore to occur in the key

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \rightarrow BDAE\}$

- time consuming procedure: trying out all the one-element, two-element, three-element key candidates using attrclosure
- alternative: using the following heuristics: all attributes, that do not occur on the right side cannot be derived via attrclosure and have therefore to occur in the key
- here: C and F do not occur on the right side, therefore the following attempt:

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \rightarrow BDAE\}$

- time consuming procedure: trying out all the one-element, two-element, three-element key candidates using attrclosure
- alternative: using the following heuristics: all attributes, that do not occur on the right side cannot be derived via attrclosure and have therefore to occur in the key
- here: C and F do not occur on the right side, therefore the following attempt: attrclosure($\{C \rightarrow BDAE\}, CF) = \{C, F, B, D, A, E\}$

$$\Rightarrow$$

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \rightarrow BDAE\}$

- time consuming procedure: trying out all the one-element, two-element, three-element key candidates using attrclosure
- alternative: using the following heuristics: all attributes, that do not occur on the right side cannot be derived via attrclosure and have therefore to occur in the key
- here: C and F do not occur on the right side, therefore the following attempt: attrclosure$(\{C \rightarrow BDAE\}, CF) = \{C, F, B, D, A, E\}$

$$\Rightarrow$$

CF is the key of $\mathcal{R}$

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \rightarrow BD, D \rightarrow AE, E \rightarrow CF, F \rightarrow E\}$
heuristics cannot be used here, as all attributes are derivable

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \rightarrow BD, D \rightarrow AE, E \rightarrow CF, F \rightarrow E\}$
heuristics cannot be used here, as all attributes are derivable

C: attrclosure$(F_d, C) = \{C, B, D, A, E, F\} \Rightarrow C$ is key of $\mathcal{R}$

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \to BD, D \to AE, E \to CF, F \to E\}$
heuristics cannot be used here, as all attributes are derivable

- C: attrclosure$(F_d, C) = \{C, B, D, A, E, F\} \Rightarrow C$ is key of $\mathcal{R}$
  attention: C derivable from E, therefore:
- E: attrclosure$(F_d, E) = \{E, C, B, D, A, F\} \Rightarrow E$ is also key of $\mathcal{R}$

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \rightarrow BD, D \rightarrow AE, E \rightarrow CF, F \rightarrow E\}$
heuristics cannot be used here, as all attributes are derivable

C: attrclosure$(F_d, C) = \{C, B, D, A, E, F\} \Rightarrow C$ is key of $\mathcal{R}$
attention: C derivable from E, therefore:

E: attrclosure$(F_d, E) = \{E, C, B, D, A, F\} \Rightarrow E$ is also key of $\mathcal{R}$
attention: E derivable from D or F, therefore:

D: attrclosure$(F_d, D) = \{D, A, E, C, F, B\} \Rightarrow D$ is key of $\mathcal{R}$,

F: attrclosure$(F_d, F) = \{F, E, C, B, D, A\} \Rightarrow F$ is key of $\mathcal{R}$

Classification of Keys via attrclosure

Example

$\mathcal{R} = \{ABCDEF\}, F_d = \{C \rightarrow BD, D \rightarrow AE, E \rightarrow CF, F \rightarrow E\}$
heuristics cannot be used here, as all attributes are derivable

C: attrclosure$(F_d, C) = \{C, B, D, A, E, F\} \Rightarrow C$ is key of $\mathcal{R}$
attention: C derivable from E, therefore:

E: attrclosure$(F_d, E) = \{E, C, B, D, A, F\} \Rightarrow E$ is also key of $\mathcal{R}$
attention: E derivable from D or F, therefore:

D: attrclosure$(F_d, D) = \{D, A, E, C, F, B\} \Rightarrow D$ is key of $\mathcal{R}$,

F: attrclosure$(F_d, F) = \{F, E, C, B, D, A\} \Rightarrow F$ is key of $\mathcal{R}$

$$\Rightarrow$$

C, D, E, F are the keys of $\mathcal{R}$

Algorithm for the Calculation of all Keys

input: $(F, \mathcal{R})$: set F of FDs and schema $\mathcal{R}$

output: set of all keys of $\mathcal{R}$

allkeys $(F, \mathcal{R})$

$\quad$ $keys = \{minimize(F, \mathcal{R}, \mathcal{R})\}$ // find 1. key

$\quad$ **for each** $key \in keys$:

$\quad\quad$ **for each** $att \in key$:

$\quad\quad\quad$ **for each** $\alpha \rightarrow \beta \in F$:

$\quad\quad\quad\quad$ **if** $att \in \beta$:

$\quad\quad\quad\quad\quad$ $nkey = (key \setminus \{att\}) \cup \alpha$

$\quad\quad\quad\quad\quad$ **if** $\nexists k \in keys$ with $k \subseteq nkey$:

$\quad\quad\quad\quad\quad\quad$ $keys = keys \cup \{minimize(F, nkey, \mathcal{R})\}$

$\quad$ **return** keys

$minimize(F, \gamma, \delta)$: returns a minimal subset $\gamma' \subseteq \gamma$ such that $\delta \subseteq$ attrclosure(F, γ').

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \rightarrow BEF, DE \rightarrow AC, A \rightarrow BDG\}$

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \rightarrow BEF, DE \rightarrow AC, A \rightarrow BDG\}$

1. find 1. key:

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \rightarrow BEF, DE \rightarrow AC, A \rightarrow BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \rightarrow BEF, DE \rightarrow AC, A \rightarrow BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from *keys*:
   - $key = A$, $att = A$:
     - $B \rightarrow BEF$, $A \rightarrow BDG$: $A \notin \beta$
     - $DE \rightarrow AC$:

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from *keys*:
   - $key = A$, $att = A$:
     - $B \to BEF$, $A \to BDG$: $A \notin \beta$
     - $DE \to AC$: $nkey = DE$

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from $keys$:
   - $key = A$, $att = A$:
     - $B \to BEF$, $A \to BDG$: $A \notin \beta$
     - $DE \to AC$: $nkey = DE$
       $minimize$: attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \rightarrow BEF, DE \rightarrow AC, A \rightarrow BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow$ *keys* $= \{A\}$
2. construct new key from *keys*:
   - *key* $= A$, *att* $= A$:
     - $B \rightarrow BEF$, $A \rightarrow BDG$: $A \notin \beta$
     - $DE \rightarrow AC$: *nkey* $= DE$
       minimize : attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$
       $\Rightarrow$ *keys* $= \{A, DE\}$

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \rightarrow BEF, DE \rightarrow AC, A \rightarrow BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from $keys$:
   - $key = A$, $att = A$:
     - $B \rightarrow BEF$, $A \rightarrow BDG$: $A \notin \beta$
     - $DE \rightarrow AC$: $nkey = DE$
       $minimize$: attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = D$:
     - $B \rightarrow BEF$, $DE \rightarrow AC$: $D \notin \beta$
     - $A \rightarrow BDG$:

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from $keys$:
   - $key = A$, $att = A$:
     - $B \to BEF$, $A \to BDG$: $A \notin \beta$
     - $DE \to AC$: $nkey = DE$
       $minimize$: attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = D$:
     - $B \to BEF$, $DE \to AC$: $D \notin \beta$
     - $A \to BDG$: $nkey = AE$;

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from $keys$:
   - $key = A$, $att = A$:
     - $B \to BEF$, $A \to BDG$: $A \notin \beta$
     - $DE \to AC$: $nkey = DE$
       $minimize$: attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = D$:
     - $B \to BEF$, $DE \to AC$: $D \notin \beta$
     - $A \to BDG$: $nkey = AE$; $A \subseteq AE$

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from $keys$:
   - $key = A$, $att = A$:
     - $B \to BEF$, $A \to BDG$: $A \notin \beta$
     - $DE \to AC$: $nkey = DE$
       $minimize$: attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = D$:
     - $B \to BEF$, $DE \to AC$: $D \notin \beta$
     - $A \to BDG$: $nkey = AE$; $A \subseteq AE \Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = E$:
     - $A \to BDG$, $DE \to AC$: $E \notin \beta$
     - $B \to BEF$:

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from $keys$:
   - $key = A$, $att = A$:
     - $B \to BEF$, $A \to BDG$: $A \notin \beta$
     - $DE \to AC$: $nkey = DE$
       $minimize$: attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = D$:
     - $B \to BEF$, $DE \to AC$: $D \notin \beta$
     - $A \to BDG$: $nkey = AE$; $A \subseteq AE \Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = E$:
     - $A \to BDG$, $DE \to AC$: $E \notin \beta$
     - $B \to BEF$: $nkey = BD$;

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from $keys$:
   - $key = A$, $att = A$:
     - $B \to BEF$, $A \to BDG$: $A \notin \beta$
     - $DE \to AC$: $nkey = DE$
       $minimize$: attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = D$:
     - $B \to BEF$, $DE \to AC$: $D \notin \beta$
     - $A \to BDG$: $nkey = AE$; $A \subseteq AE \Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = E$:
     - $A \to BDG$, $DE \to AC$: $E \notin \beta$
     - $B \to BEF$: $nkey = BD$;
       $minimize$: attrclosure$(F_d, \{B\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{D\}) \neq \mathcal{R}$

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \rightarrow BEF, DE \rightarrow AC, A \rightarrow BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from $keys$:
   - $key = A$, $att = A$:
     - $B \rightarrow BEF$, $A \rightarrow BDG$: $A \notin \beta$
     - $DE \rightarrow AC$: $nkey = DE$
       $minimize$: attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = D$:
     - $B \rightarrow BEF$, $DE \rightarrow AC$: $D \notin \beta$
     - $A \rightarrow BDG$: $nkey = AE$; $A \subseteq AE \Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = E$:
     - $A \rightarrow BDG$, $DE \rightarrow AC$: $E \notin \beta$
     - $B \rightarrow BEF$: $nkey = BD$;
       $minimize$: attrclosure$(F_d, \{B\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{D\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE, BD\}$

Computation of all Keys of a Schema

Example

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

1. find 1. key: attrclosure$(F_d, \{A\}) = \mathcal{R} \Rightarrow keys = \{A\}$
2. construct new key from *keys*:
   - $key = A$, $att = A$:
     - $B \to BEF$, $A \to BDG$: $A \notin \beta$
     - $DE \to AC$: $nkey = DE$
       minimize : attrclosure$(F_d, \{D\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{E\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = D$:
     - $B \to BEF$, $DE \to AC$: $D \notin \beta$
     - $A \to BDG$: $nkey = AE$; $A \subseteq AE \Rightarrow keys = \{A, DE\}$
   - $key = DE$, $att = E$:
     - $A \to BDG$, $DE \to AC$: $E \notin \beta$
     - $B \to BEF$: $nkey = BD$;
       minimize : attrclosure$(F_d, \{B\}) \neq \mathcal{R}$ and attrclosure$(F_d, \{D\}) \neq \mathcal{R}$
       $\Rightarrow keys = \{A, DE, BD\}$

Computation of all Keys of a Schema

Example (continuation)

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

2. construct new key from *keys*:
   - $key = BD$, $att = B$:
     - $DE \to AC$: $B \notin \beta$
     - $A \to BDG$:

Computation of all Keys of a Schema

Example (continuation)

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

2. construct new key from *keys*:
   - $key = BD$, $att = B$:
     - $DE \to AC$: $B \notin \beta$
     - $A \to BDG$: $nkey = AD$; $A \subseteq AD \Rightarrow keys = \{A, DE, BD\}$

Computation of all Keys of a Schema

Example (continuation)

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

2. construct new key from *keys*:
   - $key = BD$, $att = B$:
     - $DE \to AC$: $B \notin \beta$
     - $A \to BDG$: $nkey = AD$; $A \subseteq AD \Rightarrow keys = \{A, DE, BD\}$
     - $B \to BEF$:

Computation of all Keys of a Schema

Example (continuation)

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

2. construct new key from *keys*:
   - $key = BD$, $att = B$:
     - $DE \to AC$: $B \notin \beta$
     - $A \to BDG$: $nkey = AD$; $A \subseteq AD \Rightarrow keys = \{A, DE, BD\}$
     - $B \to BEF$: $nkey = BD$; $BD \subseteq BD \Rightarrow keys = \{A, DE, BD\}$

Computation of all Keys of a Schema

Example (continuation)

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \to BEF, DE \to AC, A \to BDG\}$

2. construct new key from *keys*:
   - $key = BD$, $att = B$:
     - $DE \to AC$: $B \notin \beta$
     - $A \to BDG$: $nkey = AD$; $A \subseteq AD \Rightarrow keys = \{A, DE, BD\}$
     - $B \to BEF$: $nkey = BD$; $BD \subseteq BD \Rightarrow keys = \{A, DE, BD\}$
   - $key = BD$, $att = D$:
     - $DE \to AC$, $B \to BEF$: $D \notin \beta$
     - $A \to BDG$:

Computation of all Keys of a Schema

Example (continuation)

$\mathcal{R} = \{ABCDEFG\}$, $F_d = \{B \rightarrow BEF, DE \rightarrow AC, A \rightarrow BDG\}$

2. construct new key from *keys*:
   - $key = BD$, $att = B$:
     - $DE \rightarrow AC$: $B \notin \beta$
     - $A \rightarrow BDG$: $nkey = AD$; $A \subseteq AD \Rightarrow keys = \{A, DE, BD\}$
     - $B \rightarrow BEF$: $nkey = BD$; $BD \subseteq BD \Rightarrow keys = \{A, DE, BD\}$
   - $key = BD$, $att = D$:
     - $DE \rightarrow AC$, $B \rightarrow BEF$: $D \notin \beta$
     - $A \rightarrow BDG$: $nkey = AB$; $A \subseteq AB \Rightarrow keys = \{A, DE, BD\}$
3. $keys = \{A, DE, BD\}$

Learning Objectives

- What are FDs?
  - When is a FD satisfied, how can we check this condition?
- What is the attribute closure and the closure of FDs?
  - How do we compute them?
- When are two sets of FDs equivalent?
- What are the Armstrong axioms?
  - What are they needed for, how do they look like?
- What is the canonical cover?
  - How can we compute it?
- What are (super) keys?
  - How can we recognize/check them?