Schema Refinement

R&G Ch 19

Functional Dependencies

- A functional dependency X → Y holds over relation R if for every pair of tuples t_1, t_2 in R, it holds that if $\pi_X t_1 = \pi_X t_2$, then $\pi_Y t_1 = \pi_Y t_2$.
  - X and Y are sets of columns
- A FD isn't just a statement about a particular instance of R, but about application semantics.
  - We can check to see of an FD holds over R, but can't check to see if R has an FD.

Example

Contracts(***c****id*, ***s****id*, ***j****id*, ***d****id*, ***p****id*, ***q****ty*, ***v****alue*)

1. C is a key: C → CSJDPQV
2. ProJects purchase **P**arts using a single **C**ontract: JP → C
3. **D**epts. purchase at most one **P**art from any **S**upplier: SD → P
4. (1), (2) imply that JP → CSJDPQV
5. (3) implies that SDJ → JP
6. (4), (5) imply that SDJ → CSJDPQV

Decomposition

- Replace R(A,B,C,D) with, for example,
    - R1(A,B), R2(B,C,D) or
    - R1(A,C,D), R2(A,B,D)
- When is it useful to decompose?
    - What are the costs of decomposition?

Normal Forms

Normal Forms

- If a relation is in one of the normal forms (BCNF, 3NF) certain problems are avoided/minimized.
- Decomposition can produce relations in/closer to a normal form.
- FDs help us detect redundancy
    - For R(A,B,C), if A → B, and several tuples have the same A value, they'll all have the same Bs.

Boyce-Codd Normal Form (BCNF)

- R (with FDs F) is in BCNF if for all $X \rightarrow A \in F^+$:
    - $A \subseteq X$ (the trivial FD), or
    - X contains a key for R
- In other words, R is in BCNF if the only non-trivial FDs that hold over R are key constraints.

BCNF Isn’t Always Viable

Ship	**Crew Role**	**Officer**
Enterprise	Captain	Kirk
Enterprise	Science	Spock
Enterprise	Medical	McCoy
Excelsior	Captain	Sulu

Ship, Crew Role → Officer

Officer → Crew Role

Keys: {Ship, Crew Role}, {Crew Role, Officer}

BCNF Isn't Always Viable

Ship	Crew Role	Officer
Enterprise	Captain	Kirk
Enterprise	Science	Spock
Enterprise	Medical	McCoy
Excelsior	Captain	Sulu

Ship, Crew Role → Officer

Officer → Crew Role

Keys: {Ship, Crew Role}, {Crew Role, Officer}

3rd Normal Form

- R (with FDs F) is in BCNF if for all X → A $\in F^+$:
  - A $\subseteq$ X (the trivial FD), or
  - X contains a key for R, or
  - A is a subset of any key for R
    - Recall that keys are minimal sets of attributes.
- Weaker form of BCNF
  - …used when BCNF impractical, impossible.

3rd Normal Form

- If 3NF is violated by X → A then:
    - X is a subset of some key K
        - Some (X,A)s are being stored redundantly.
    - X is not a proper subset of any key
        - So there exists redundancy: K → X → A
            - But this can still happen in 3NF.

3NF Isn’t Always Perfect

Ship	Crew Role	Officer
Enterprise	Captain	Kirk
Enterprise	Science	Spock
Enterprise	Medical	McCoy
Excelsior	Captain	Sulu

Ship, Crew Role → Officer

Officer → Crew Role

Keys: {Ship, Crew Role}, {Crew Role, Officer}

3NF Isn’t Always Perfect

Ship	Crew Role	Officer
Enterprise	Captain	Kirk
Enterprise	Science	Spock
Enterprise	Medical	McCoy
Excelsior	Captain	Sulu

Ship, Crew Role → Officer

Officer → Crew Role

Keys: {Ship, Crew Role}, {Crew Role, Officer}

3NF Isn't Always Perfect

- BCNF can't always be decomposed (as in example)
- 3NF is a compromise:
  - Guaranteed to be possible to decompose to 3NF.
  - Not guaranteed to lack redundancy.

Decomposition

- Starting with $R(A_1, \ldots ,A_n)$, a decomposition creates relations $R_1, R_2, \ldots$ such that
  - $R_i \subset R$ (R_i contains only attributes in R)
  - $R \equiv R_1 \cup R_2 \cup \ldots$ (each attribute appears at least once in a decomposed rel)
- We store instances of the R_is instead of R.

Example

- Officers(**O**id, **N**ame, **P**ost, **R**ank, **S**alary)
  - F = {O → N,P,R,S; R → S}
  - R → S violates 3NF
- **Store**: Officers'(ONPR), Salaries(RS)
  - Can we just project Officers down to O',S?
  - What problems could occur?

Decomposition Costs

- Queries become more expensive:
  - How much does Sheridan earn? (2 way join)
- May not be possible to reconstruct original relation from instances.
  - $R_1(A,B), R_2(B,C), R_3(A,C)$
- Checking dependencies may require reconstituting the decomposed relation.

Lossy Decompositions

R

A	B	C
1	1	1
1	2	2
2	1	2
2	2	1

Lossy Decompositions

R

A	B	C
1	1	1
1	2	2
2	1	2
2	2	1

R_1

A	B
1	1
1	2
2	1
2	2

R_2

B	C
1	1
2	2
1	2
2	1

R_3

A	B
1	1
1	2
2	2
2	1

Lossy Decompositions

R

A	B	C
1	1	1
1	2	2
2	1	2
2	2	1
2	2	2
2	1	1
1	2	1
1	1	2

R_1

A	B
1	1
1	2
2	1
2	2

R_2

B	C
1	1
2	2
1	2
2	1

R_3

A	B
1	1
1	2
2	2
2	1

Lossless Join Decompositions

- For a relation **R** with FDs **F**:
  - A decomposition of R into R_1, R_2 is <u>lossless</u> iff F^+ contains $R_1 \cap R_2 \rightarrow R_1$, or $R_1 \cap R_2 \rightarrow R_2$.
- In other words, $R_1 \cap R_2$ must contain a key for R.
- Don't let data loss happen to you.
  - Practice lossless decomposition.

Decomposition into BCNF

- Start with relation R with FDs F
- If X→Y Violates BCNF
  - Decompose R into R_1=(R-Y), R_2=XY.
- Recur on R_1,R_2 until all satisfy BCNF.
  - Guaranteed to terminate.
- There might be multiple violations, the order in which they are resolved drastically changes the output.

Example

Contracts(***c****id*, ***s****id*, ***j****id*, ***d****id*, ***p****id*, ***q****ty*, ***v****alue*)

1. C is a key: C → CSJDPQV
2. Pro**J**ects purchase **P**arts using a single **C**ontract: JP → C
3. **D**epts. purchase at most one **P**art from any **S**upplier: SD → P
4. Each Pro**J**ect uses one **S**upplier: J → S

Example

Contracts(***c****id*, ***s****id*, ***j****id*, ***d****id*, ***p****id*, ***q****ty*, ***v****alue*)

1. C is a key: C → CSJDPQV
2. Pro**J**ects purchase **P**arts using a single **C**ontract: JP → C
3. **D**epts. purchase at most one **P**art from any **S**upplier: SD → P
4. Each Pro**J**ect uses one **S**upplier: J → S

(1): OK, C is a key

Example

Contracts(***c****id*, ***s****id*, ***j****id*, ***d****id*, ***p****id*, ***q****ty*, ***v****alue*)

1. C is a key: C → CSJDPQV
2. Pro**J**ects purchase **P**arts using a single **C**ontract: JP → C
3. **D**epts. purchase at most one **P**art from any **S**upplier: SD → P
4. Each Pro**J**ect uses one **S**upplier: J → S

(2): OK, JP is a key

Example

Contracts(***c***_id_, ***s***_id_, ***j***_id_, ***d***_id_, ***p***_id_, ***q***_ty_, ***v***_alue_)

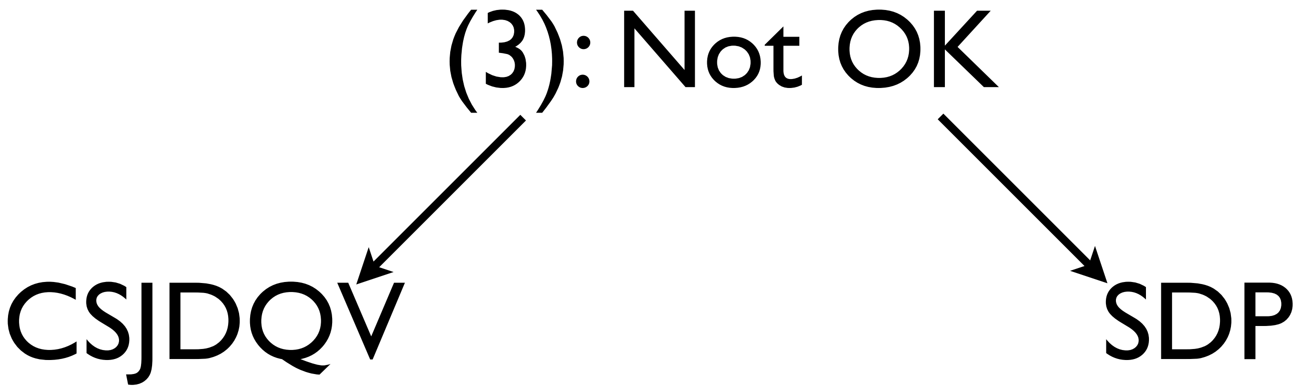
1. C is a key: C → CSJDPQV
2. ProJects purchase **P**arts using a single **C**ontract: JP → C
3. **D**epts. purchase at most one **P**art from any **S**upplier: SD → P
4. Each Pro**J**ect uses one **S**upplier: J → S

(3): Not OK

Example

Contracts(***c****id*, ***s****id*, ***j****id*, ***d****id*, ***p****id*, ***q****ty*, ***v****alue*)

1. C is a key: C → CSJDPQV
2. Pro**J**ects purchase **P**arts using a single **C**ontract: JP → C
3. **D**epts. purchase at most one **P**art from any **S**upplier: SD → P
4. Each Pro**J**ect uses one **S**upplier: J → S



Example

Contracts(***c****id*, ***s****id*, ***j****id*, ***d****id*, ***q****ty*, ***v****alue*)

PartSupp(***s****id*, ***d****id*, ***p****id*)

1. C is a key: C → CSJDPQV
2. Pro**J**ects purchase **P**arts using a single **C**ontract: JP → C
3. **D**epts. purchase at most one **P**art from any **S**upplier: SD → P
4. Each Pro**J**ect uses one **S**upplier: J → S

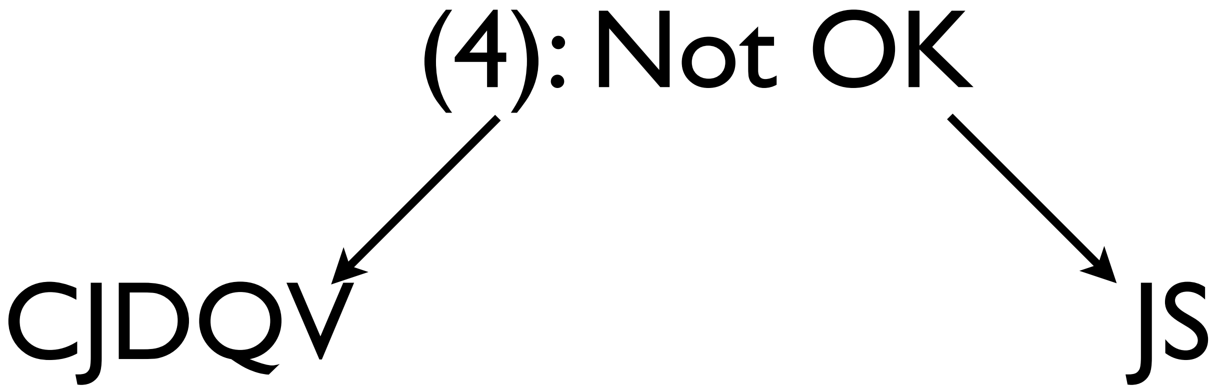
(4): Not OK

Example

Contracts(***c**id*, ***s**id*, ***j**id*, ***d**id*, ***q**ty*, ***v**alue*)

PartSupp(***s**id*, ***d**id*, ***p**id*)

1. C is a key: C → CSJDPQV
2. Pro**J**ects purchase **P**arts using a single **C**ontract: JP → C
3. **D**epts. purchase at most one **P**art from any **S**upplier: SD → P
4. Each Pro**J**ect uses one **S**upplier: J → S



(4): Not OK

CJDQV

JS

Example

Contracts(***c****id*, ***j****id*, ***d****id*, ***q****ty*, ***v****alue*)

PartSupp(***s****id*, ***d****id*, ***p****id*)

ProjectSupp(***j****id*, ***s****id*)

1. C is a key: C → CSJDPQV
2. Pro**J**ects purchase **P**arts using a single **C**ontract: JP → C
3. **D**epts. purchase at most one **P**art from any **S**upplier: SD → P
4. Each Pro**J**ect uses one **S**upplier: J → S

BCNF Isn't Always Viable

Ship	Crew Role	Officer
Enterprise	Captain	Kirk
Enterprise	Science	Spock
Enterprise	Medical	McCoy
Excelsior	Captain	Sulu

Ship, Crew Role → Officer

Officer → Crew Role

BCNF Isn’t Always Viable

Ship	Crew Role	Officer
Enterprise	Captain	Kirk
Enterprise	Science	Spock
Enterprise	Medical	McCoy
Excelsior	Captain	Sulu

Ship, Crew Role → Officer

Officer → Crew Role

No decomposition preserves all FDs

No Dependency Preserving Decomposition

Dependency Preserving Decomposition

- Simple modification to the algorithm for 3NF:
  - If decomposition can't enforce X→Y, add relation XY.
- But XY may still violate 3NF
  - E.g., Relation ABC with FDs AB→C, B→C
- **Refinement**: Only enforce FDs in the Minimal Cover.

Minimal Cover

- For a set of FDs F, the minimal cover G satisfies:
  - Closure of F = Closure of G
  - RHS of each FD in G is a single attribute
  - Any deletion of an FD in G or attributes in an FD in G changes its closure.

Example

For the following set of FDs

$A \rightarrow B$ $ABCD \rightarrow E$ $EF \rightarrow GH$ $ACDF \rightarrow EG$

$A \rightarrow B$ $ACD \rightarrow E$ $EF \rightarrow G$ $EF \rightarrow H$

… is the minimal cover

Example

For the following set of FDs

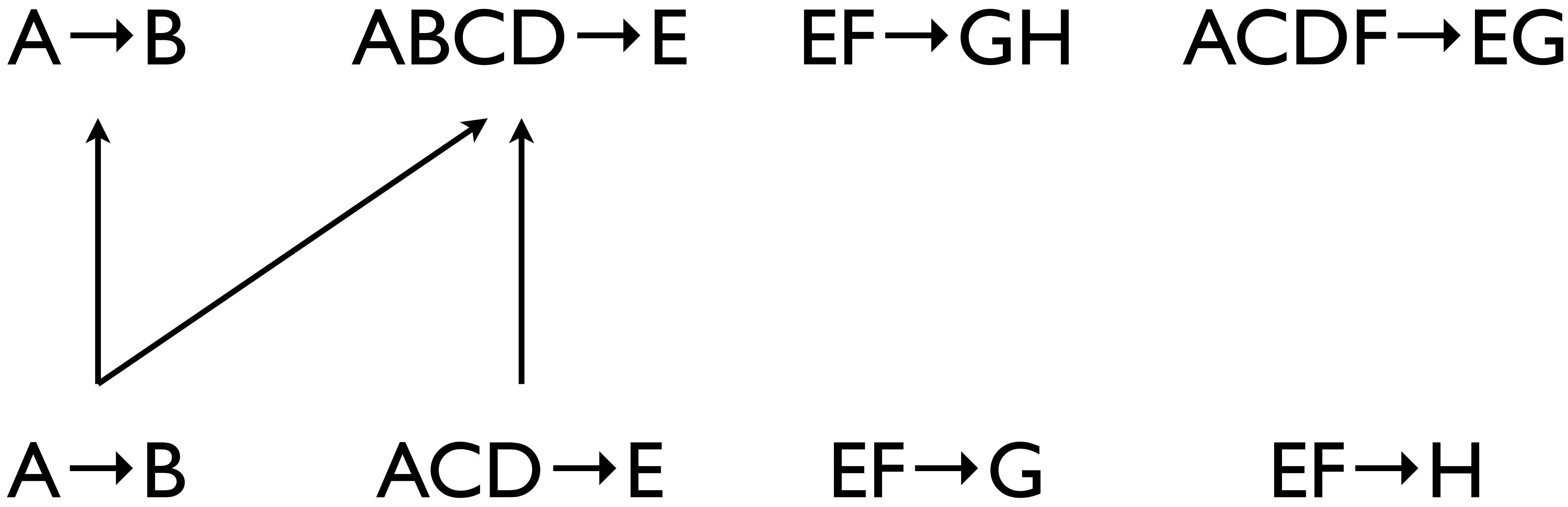
$A \rightarrow B$ $ABCD \rightarrow E$ $EF \rightarrow GH$ $ACDF \rightarrow EG$

$A \rightarrow B$ $ACD \rightarrow E$ $EF \rightarrow G$ $EF \rightarrow H$

… is the minimal cover

Example

For the following set of FDs



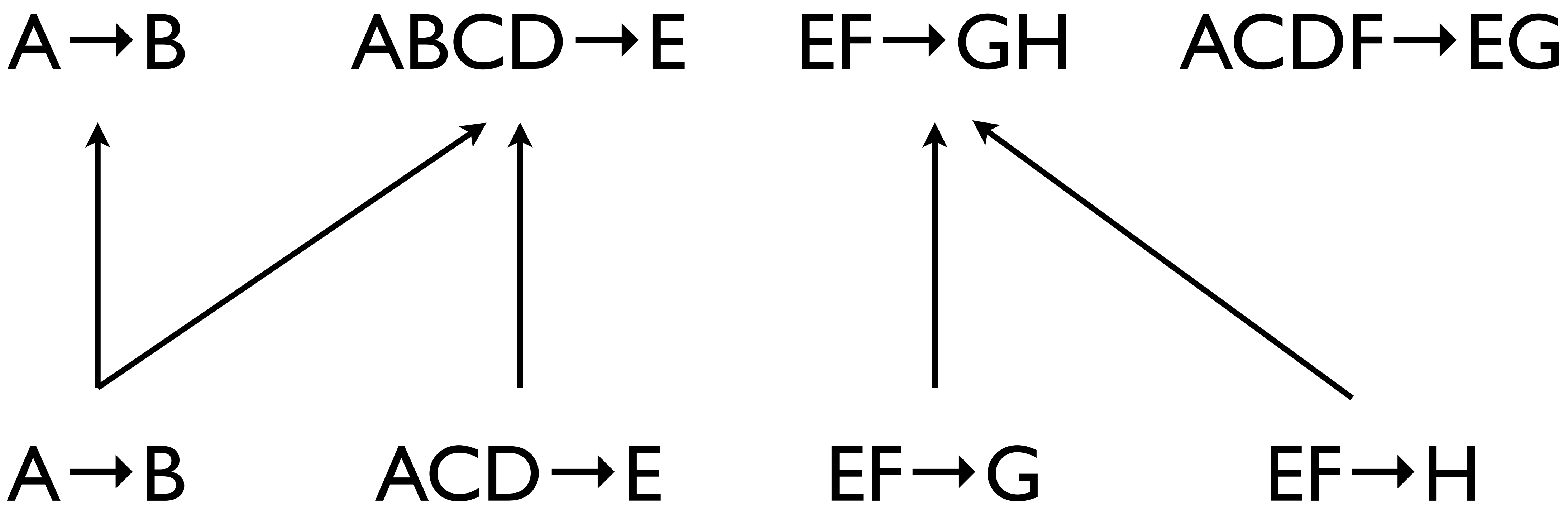
A→B ABCD→E EF→GH ACDF→EG

A→B ACD→E EF→G EF→H

… is the minimal cover

Example

For the following set of FDs



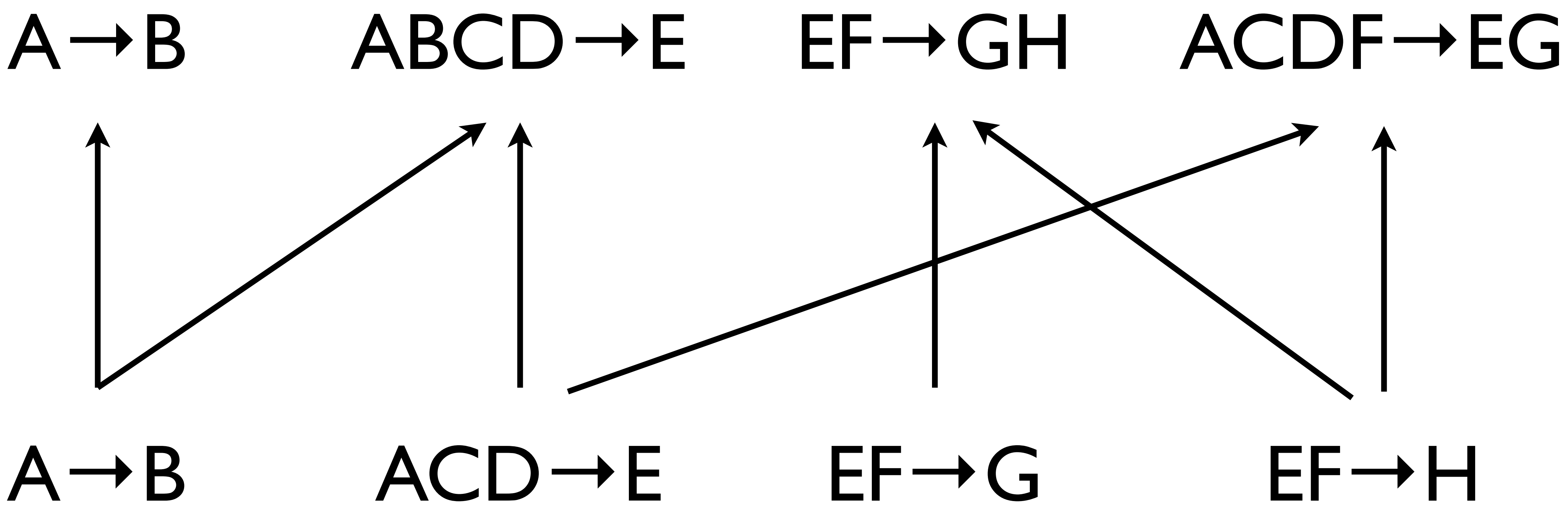
$A \rightarrow B$ $ABCD \rightarrow E$ $EF \rightarrow GH$ $ACDF \rightarrow EG$

$A \rightarrow B$ $ACD \rightarrow E$ $EF \rightarrow G$ $EF \rightarrow H$

… is the minimal cover

Example

For the following set of FDs



A→B ABCD→E EF→GH ACDF→EG

A→B ACD→E EF→G EF→H

… is the minimal cover