Nonadditive Lossless-Join Decomposition

Example 3

Schema < R=ABC , F = { A→B }> and partition D_2 = { AB, BC }

r

A	B	C
1	2	3
1	2	4
2	1	1
3	1	0
3	1	4

D_2

r1

A	B
1	2
2	1
3	1

r2

B	C
2	3
2	4
1	1
1	0
1	4

*

r'

A	B	C
1	2	3
1	2	4
2	1	1
3	1	0
3	1	4
2	**1**	**0**
2	**1**	**4**
3	**1**	**1**

Consider the decompositions
D_1 = { AB, AC } and
D_2 = { AB, BC }
Observe that D_1 is a 'good' decomposition (lossless) while D_2 is not.

Phantom /Spurious rows created when using decomposition D_2

Nonadditive Lossless-Join Decomposition

Example 3 *cont.*

Schema < R=ABC , F = { A→B }> and partition D_1 = { AB, AC }

r

A	B	C
1	2	3
1	2	4
2	1	1
3	1	0
3	1	4

D_1

r1

A	B
1	2
2	1
3	1

r2

A	C
1	3
1	4
2	1
3	0
3	4

*

r'

A	B	C
1	2	3
1	2	4
2	1	1
3	1	0
3	1	4

Consider the decompositions
D_1 = { AB, AC } and
D_2 = { AB, BC }
Observe that D_1 is a 'good' decomposition (lossless) while D_2 is not.

Notice that $r' = \Pi_{AB}(r) * \Pi_{AC}(r)$

Nonadditive Lossless-Join Decomposition

Example 3 *cont.*

Schemas < R_1= ABC, F_1 = { A→BC, C →B } > and
< R_2= BCD, F_2 = { C →B, B →C, D →B }

r_1

A	B	C
1	2	3
4	2	3
3	1	4
2	2	6

r_2

B	C	D
2	3	4
2	3	5
1	4	2

$r = r_1 \otimes r_2$

A	B	C	D
1	2	3	4
1	2	3	5
4	2	3	4
4	2	3	5
3	1	4	2

$r_1' = \Pi_{ABC}(r)$

A	B	C
1	2	3
4	2	3
3	1	4

$r_2' = \Pi_{BCD}(r)$

B	C	D
2	3	4
2	3	5
1	4	2

Observe that $r_1 \neq r_1'$

r_1 tuple <2,2,6> is lost in the JOIN and does not appear in r_1'

Testing Lossless-Join (or Non-Additive) Decomposition

Definition (good only on binary partition)

If D={ R_1, R_2 } is a decomposition of R and F is a set of FDs on R, then D has a lossless-join with respect to F if

$$\mathbf{F \Rightarrow (R_1 \cap R_2) \rightarrow (R_1 - R_2)} \quad \text{or} \quad \mathbf{F \Rightarrow (R_1 \cap R_2) \rightarrow (R_2 - R_1)}$$

Example 4

Consider the previous problem where **R=ABC** and **F = { A→B }.**
Let's assess the partition **D_1 = {AB, AC}.** Here R_1=AB and R_2= AC
therefore $R_1 \cap R_2 = A$
$R_1 - R_2 = B$
$R_2 - R_1 = C$

The question F⇒ ($R_1 \cap R_2$) → (R_1 - R_2) is equivalent to F⇒ A → B and we know this is true because F contains exactly this dependency.

We must conclude the decomposition D_1 is lossless with respect to F.