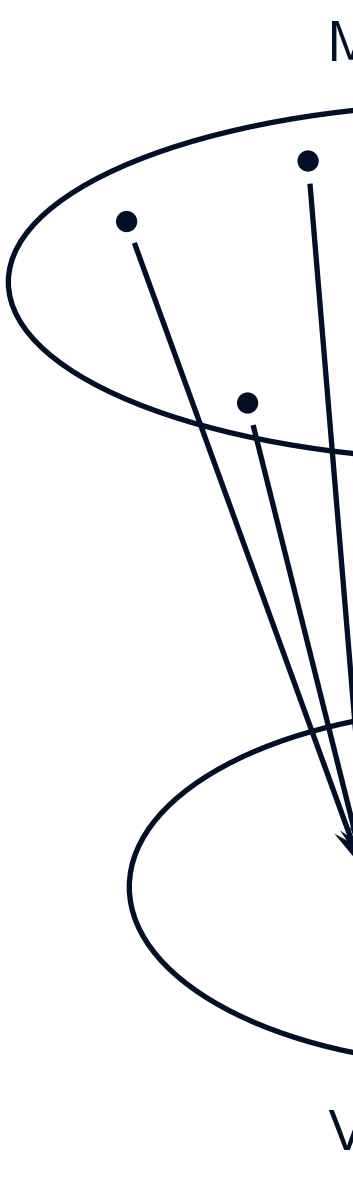
Optimal Complements for a Class of Relational V

Stephen J. Hegner
Umeå University
Department of Computing Science
Sweden

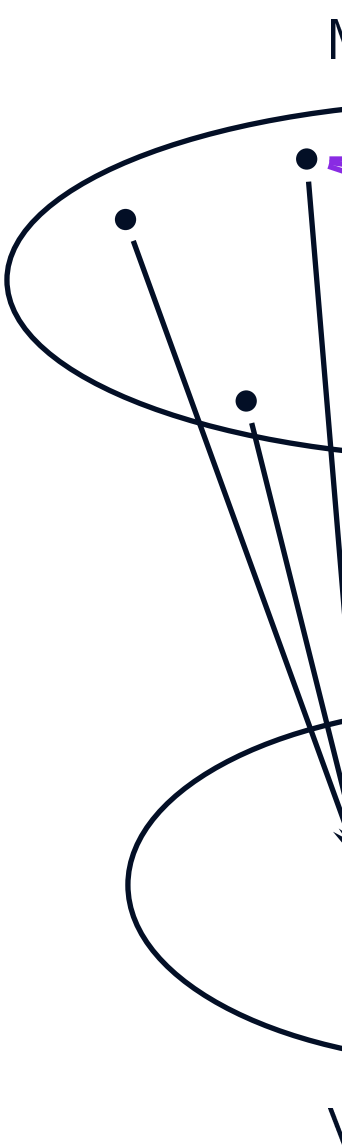
The Update Problem for Database V

- On the underlying states, the view mapping is generally *surjective* (onto) but not *injective* (one-to-one).



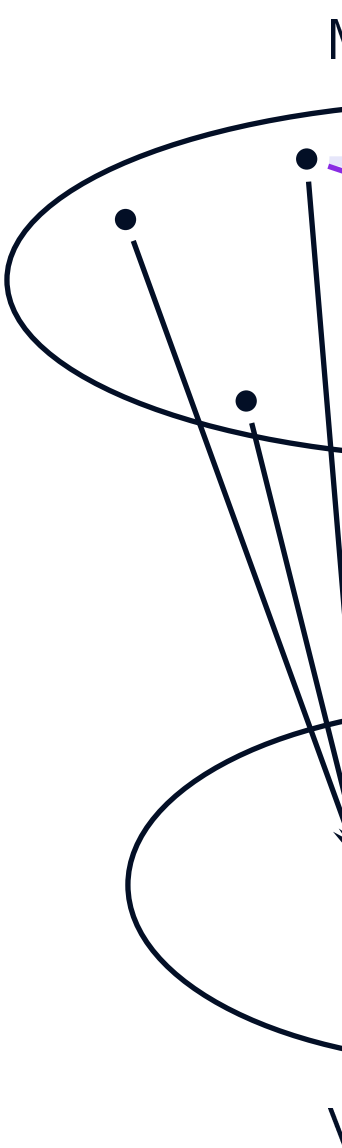
The Update Problem for Database V

- On the underlying states, the view mapping is generally *surjective* (onto) but not *injective* (one-to-one).
- Thus, a view update has many possible *reflections* to the main schema.



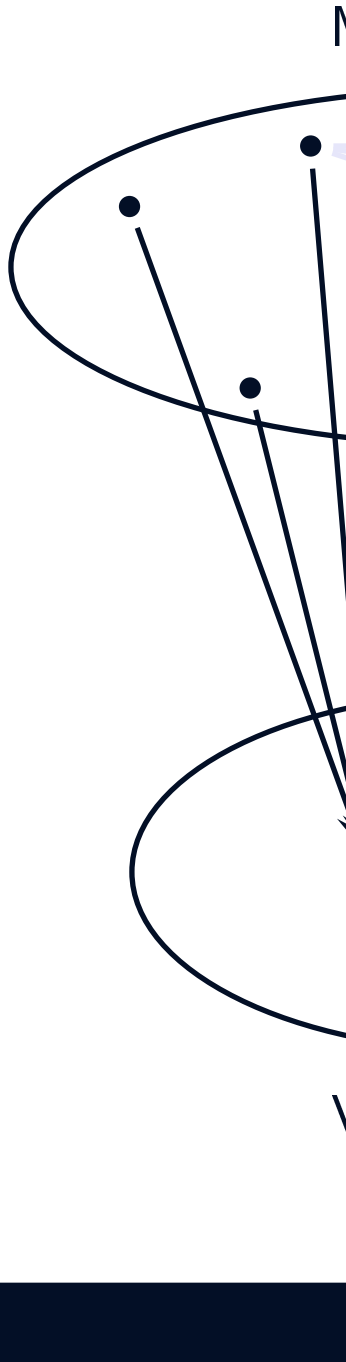
The Update Problem for Database V

- On the underlying states, the view mapping is generally *surjective* (onto) but not *injective* (one-to-one).
- Thus, a view update has many possible *reflections* to the main schema.
- The problem of identifying a suitable reflection is known as the *update translation problem* or *update reflection problem*.



The Update Problem for Database V

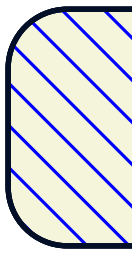
- On the underlying states, the view mapping is generally *surjective* (onto) but not *injective* (one-to-one).
- Thus, a view update has many possible *reflections* to the main schema.
- The problem of identifying a suitable reflection is known as the *update translation problem* or *update reflection problem*.
- With a reasonable definition of suitability, it may not be the case that every view update has a suitable translation.



The Gold Standard — the Constant-Complen

The Gold Standard — the Constant-Complen

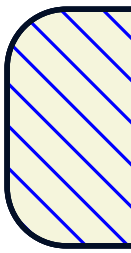
- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.

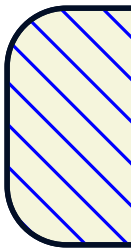


View

The Gold Standard — the Constant-Complen

- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.
- One is isomorphic to the view schema and tracks its updates exactly.





View

The Gold Standard — the Constant-Complen

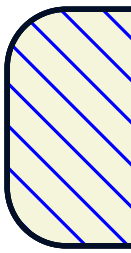
- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.
- One is isomorphic to the view schema and tracks its updates exactly.
- The other is held constant for all updates to the view.

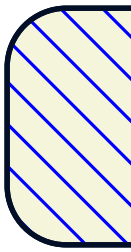
The Gold Standard — the Constant-Complen

- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.
- One is isomorphic to the view schema and tracks its updates exactly.
- The other is held constant for all updates to the view.
- Although it is somewhat limited in the view updates which it allows, they are supported in an optimal manner.

The Gold Standard — the Constant-Complen

- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.
- One is isomorphic to the view schema and tracks its updates exactly.
- The other is held constant for all updates to the view.
- Although it is somewhat limited in the view updates which it allows, they are supported in an optimal manner.



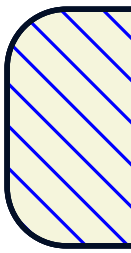


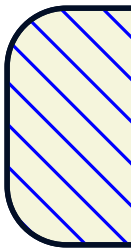
View

- It can be shown [Hegner 03 AMAI] that this strategy is pr all *update anomalies*.

The Gold Standard — the Constant-Complen

- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.
- One is isomorphic to the view schema and tracks its updates exactly.
- The other is held constant for all updates to the view.
- Although it is somewhat limited in the view updates which it allows, they are supported in an optimal manner.



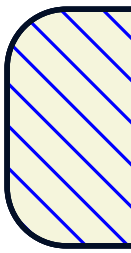


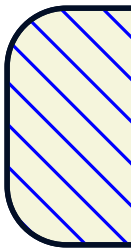
View

- It can be shown [Hegner 03 AMAI] that this strategy is pr all *update anomalies*.
- However, this is complicated by the *complement uniquene*

The Gold Standard — the Constant-Complen

- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.
- One is isomorphic to the view schema and tracks its updates exactly.
- The other is held constant for all updates to the view.
- Although it is somewhat limited in the view updates which it allows, they are supported in an optimal manner.





View

- It can be shown [Hegner 03 AMAI] that this strategy is pr all *update anomalies*.
- However, this is complicated by the *complement uniquene*
- Some examples will help illustrate these ideas.

The Idea of Constant-Complement by E

- Consider the classical example to the right.

C

The Idea of Constant-Complement by E

- Consider the classical example to the right.
- A natural complement to the AB-projection is the BC-projection.

The Idea of Constant-Complement by E

- Consider the classical example to the right.
- A natural complement to the AB-projection is the BC-projection.
- The *decomposed schema* $\mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$ has relation symbols $R_[AB]$ and $R_2[BC]$; the legal database are all states which are join compatible on B.

The Idea of Constant-Complement by E

- Consider the classical example to the right.
- A natural complement to the AB-projection is the BC-projection.
- The *decomposed schema* $\mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$ has relation symbols $R_[AB]$ and $R_2[BC]$; the legal database are all states which are join compatible on B.
- The *decomposition mapping* $\mathbf{E}_1 \to \mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$, and is always bijective for complements.

The Idea of Constant-Complement by E

- Consider the classical example to the right.
- A natural complement to the AB-projection is the BC-projection.
- The *decomposed schema* $\mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$ has relation symbols $R_[AB]$ and $R_2[BC]$; the legal database are all states which are join compatible on B.
- The *decomposition mapping* $\mathbf{E}_1 \to \mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$, and is always bijective for complements.
- The *reconstruction mapping* $\mathbf{W}_{AB} \otimes \mathbf{W}_{BC} \to \mathbf{W}_1$ is the inverse of the decomposition mapping. It is the natural join in this case.

The Idea of Constant-Complement by E

- Consider the classical example to the right.
- A natural complement to the AB-projection is the BC-projection.
- The *decomposed schema* $\mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$ has relation symbols $R_[AB]$ and $R_2[BC]$; the legal database are all states which are join compatible on B.
- The *decomposition mapping* $\mathbf{E}_1 \rightarrow \mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$, and is always bijective for complements.
- The *reconstruction mapping* $\mathbf{W}_{AB} \otimes \mathbf{W}_{BC} \rightarrow \mathbf{W}_1$ is the inverse of the decomposition mapping. It is the natural join in this case.
- The view which is the projection on B is the *meet* of $\mathbf{W}_{AB}$ and $\mathbf{W}_{BC}$, and is precisely that which must be held constant under a constant-complement update.

The Problem of Complement Unique

- Given is the following two-relation main schema.

M
N

$R[$

The Problem of Complement Unique

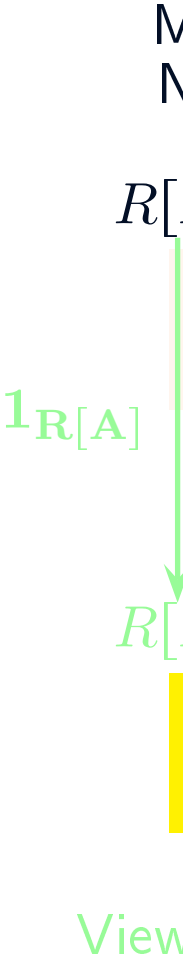
- Given is the following two-relation main schema.

M
N

$R[$

The Problem of Complement Unique

- Given is the following two-relation main schema.
- The view schema $\mathbf{W}_0$ to be updated is that which preserves $R[A]$ but discards $S[A]$.

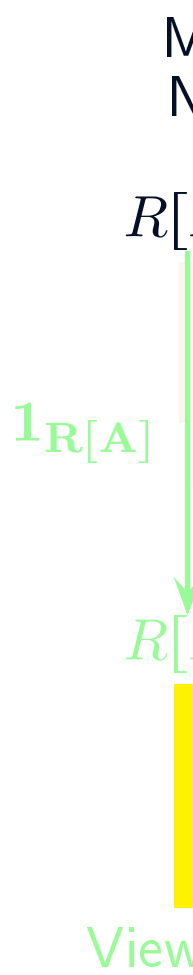


The Problem of Complement Unique

- Given is the following two-relation main schema.
- The view schema $\mathbf{W}_0$ to be updated is that which preserves $R[A]$ but discards $S[A]$.
- The *natural complement* $\mathbf{W}_1$ is the schema which preserves $S[A]$ but discards $R[A]$.

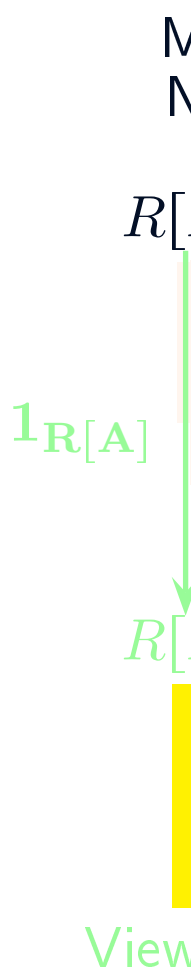
The Problem of Complement Unique

- Given is the following two-relation main schema.
- The view schema $\mathbf{W}_0$ to be updated is that which preserves $R[A]$ but discards $S[A]$.
- The *natural complement* $\mathbf{W}_1$ is the schema which preserves $S[A]$ but discards $R[A]$.
- With $\mathbf{W}_1$ constant, all updates to $R[A]$ are allowed.



The Problem of Complement Unique

- Given is the following two-relation main schema.
- The view schema $\mathbf{W}_0$ to be updated is that which preserves $R[A]$ but discards $S[A]$.
- The *natural complement* $\mathbf{W}_1$ is the schema which preserves $S[A]$ but discards $R[A]$.
- With $\mathbf{W}_1$ constant, all updates to $R[A]$ are allowed.
- Clearly, this is the only reasonable update strategy for $\mathbf{W}_0$.



The Problem of Complement Unique

- Given is the following two-relation main schema.
- The view schema $\mathbf{W}_0$ to be updated is that which preserves $R[A]$ but discards $S[A]$.
- The *natural complement* $\mathbf{W}_1$ is the schema which preserves $S[A]$ but discards $R[A]$.
- With $\mathbf{W}_1$ constant, all updates to $R[A]$ are allowed.
- Clearly, this is the only reasonable update strategy for $\mathbf{W}_0$.
- However, $\mathbf{W}_1$ does not define the only complement.

The Problem of Complement Unique

- Given is the following two-relation main schema.
- The view schema $\mathbf{W}_0$ to be updated is that which preserves $R[A]$ but discards $S[A]$.
- The *natural complement* $\mathbf{W}_1$ is the schema which preserves $S[A]$ but discards $R[A]$.
- With $\mathbf{W}_1$ constant, all updates to $R[A]$ are allowed.
- Clearly, this is the only reasonable update strategy for $\mathbf{W}_0$.
- However, $\mathbf{W}_1$ does not define the only complement.
- Without further restrictions, complements are almost never unique.

An Alternate Complement

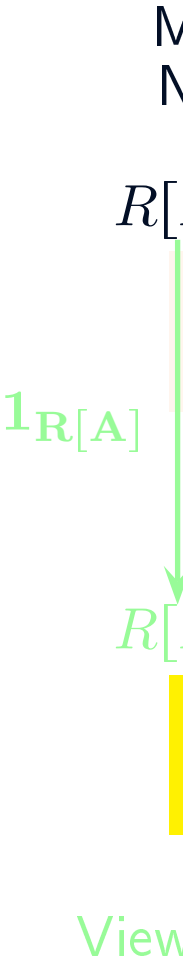
- The main schema is unchanged.

M
N

$R[$

An Alternate Complement

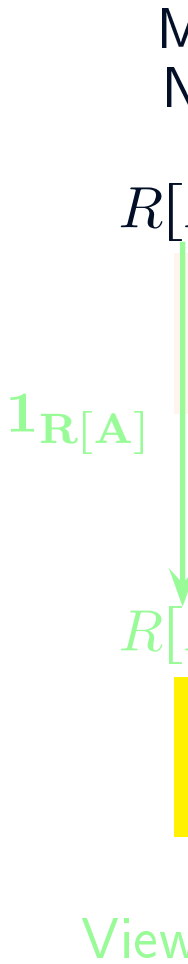
- The main schema is unchanged.
- The view schema $\mathbf{W}_0$ to be updated is also the same.



An Alternate Complement

- The main schema is unchanged.
- The view schema $\mathbf{W}_0$ to be updated is also the same.
- An alternative complement $\mathbf{W}_2$ is defined by the symmetric difference:

$$T[A] = (R[A] \backslash S[A]) \cup (S[A] \backslash R[A])$$



An Alternate Complement

- The main schema is unchanged.
- The view schema $\mathbf{W}_0$ to be updated is also the same.
- An alternative complement $\mathbf{W}_2$ is defined by the symmetric difference:

$$T[A] = (R[A] \backslash S[A]) \cup (S[A] \backslash R[A])$$

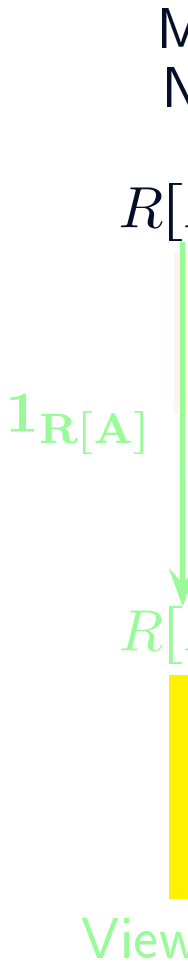
- With this alternative complement, the update strategy is different — $S[A]$ is altered.

An Alternate Complement

- The main schema is unchanged.
- The view schema $\mathbf{W}_0$ to be updated is also the same.
- An alternative complement $\mathbf{W}_2$ is defined by the symmetric difference:

$$T[A] = (R[A] \backslash S[A]) \cup (S[A] \backslash R[A])$$

- With this alternative complement, the update strategy is different — $S[A]$ is altered.
- Clearly, this is not a desirable complement.



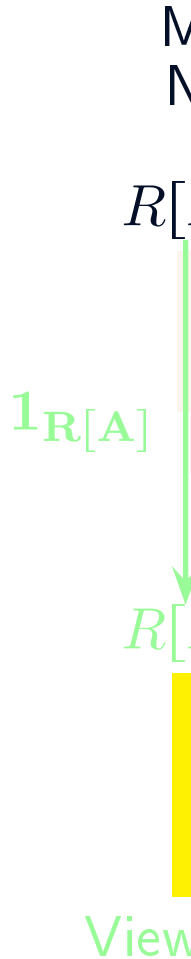
An Alternate Complement

- The main schema is unchanged.
- The view schema $\mathbf{W}_0$ to be updated is also the same.
- An alternative complement $\mathbf{W}_2$ is defined by the symmetric difference:

$$T[A] = (R[A] \backslash S[A]) \cup (S[A] \backslash R[A])$$

- With this alternative complement, the update strategy is different — $S[A]$ is altered.
- Clearly, this is not a desirable complement.

Question: How can these two complements be distinguished formally?



The Sufficiency of Monotonicity

- Note that the symmetric difference mapping is not monot
  natural order of database states.

The Sufficiency of Monotonicity

- Note that the symmetric difference mapping is not monot
  natural order of database states.

Theorem: If the view to be updated and is defined by a mono
the reflection of a given view update to the main schema i
choice of complement, provided that the complement is al
monotonic morphism.

The Sufficiency of Monotonicity

- Note that the symmetric difference mapping is not monot
  natural order of database states.

Theorem: If the view to be updated and is defined by a mono
the reflection of a given view update to the main schema i
choice of complement, provided that the complement is al
monotonic morphism.

Proof: [Hegner 04 AMAI], [Hegner 08 SDKB], [Hegner 09 LID

The Sufficiency of Monotonicity

- Note that the symmetric difference mapping is not monot
  natural order of database states.

Theorem: If the view to be updated and is defined by a mono
the reflection of a given view update to the main schema i
choice of complement, provided that the complement is al
monotonic morphism.

Proof: [Hegner 04 AMAI], [Hegner 08 SDKB], [Hegner 09 LID

However: It is not necessarily the case that all such view upda
using the same complement.

The Sufficiency of Monotonicity

- Note that the symmetric difference mapping is not monot
  natural order of database states.

Theorem: If the view to be updated and is defined by a mono
the reflection of a given view update to the main schema i
choice of complement, provided that the complement is al
monotonic morphism.

Proof: [Hegner 04 AMAI], [Hegner 08 SDKB], [Hegner 09 LID

However: It is not necessarily the case that all such view upda
using the same complement.

- It is useful to illustrate with a simple example.

Incompatible View Updates

- The view Π_{ABC} of the schema to the right has Π_{BD} as a natural monotonic meet complement.

Incompatible View Updates

- The view Π_{ABC} of the schema to the right has Π_{BD} as a natural monotonic meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_B constant.

Incompatible View Updates

- The view Π_{ABC} of the schema to the right has Π_{BD} as a natural monotonic meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_B constant.
- However, Π_{ABC} also has Π_{CD} as a natural meet complement.

Incompatible View Updates

- The view Π_{ABC} of the schema to the right has Π_{BD} as a natural monotonic meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_B constant.
- However, Π_{ABC} also has Π_{CD} as a natural meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_C constant.

Incompatible View Updates

- The view Π_{ABC} of the schema to the right has Π_{BD} as a natural monotonic meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_B constant.
- However, Π_{ABC} also has Π_{CD} as a natural meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_C constant.
- The only updates allowable with both complements are those which hold Π_{BC} constant.

Incompatible View Updates

- The view Π_{ABC} of the schema to the right has Π_{BD} as a natural monotonic meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_B constant.
- However, Π_{ABC} also has Π_{CD} as a natural meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_C constant.
- The only updates allowable with both complements are those which hold Π_{BC} constant.
- The combined complement is effectively Π_{BCD}.

Incompatible View Updates

- The view Π_{ABC} of the schema to the right has Π_{BD} as a natural monotonic meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_B constant.
- However, Π_{ABC} also has Π_{CD} as a natural meet complement.
- With this complement, the allowable updates on Π_{ABC} are precisely those which keep Π_C constant.
- The only updates allowable with both complements are those which hold Π_{BC} constant.
- The combined complement is effectively Π_{BCD}.
- There is no Π-complement which is more general than Π_{BD} or Π_{CD}.

Comparison of Projections and Optimal Co

Context: A universal relational schema $R[\mathbf{U}]$ constrained by s

Comparison of Projections and Optimal Co

Context: A universal relational schema $R[\mathbf{U}]$ constrained by s

- A Π-*view* is defined by a single projection.

Comparison of Projections and Optimal Co

Context: A universal relational schema $R[\mathbf{U}]$ constrained by s

- A Π-*view* is defined by a single projection.

Notation: Π_W is the projection onto attribute set W.

Comparison of Projections and Optimal Co

Context: A universal relational schema $R[\mathbf{U}]$ constrained by s

- A Π-*view* is defined by a single projection.

Notation: Π_W is the projection onto attribute set W.

- Projective views may be compared via their attributes.

$$\Pi_{\mathbf{W}_1} \preceq \Pi_{\mathbf{W}_2} \text{ iff } \mathbf{W}_1 \subseteq \mathbf{W}_2 \qquad (\mathbf{W}_1, \mathbf{W}$$

Comparison of Projections and Optimal Co

Context: A universal relational schema $R[\mathbf{U}]$ constrained by s

- A Π-*view* is defined by a single projection.

Notation: Π_W is the projection onto attribute set W.

- Projective views may be compared via their attributes.

$$\Pi_{\mathbf{W}_1} \preceq \Pi_{\mathbf{W}_2} \text{ iff } \mathbf{W}_1 \subseteq \mathbf{W}_2 \qquad (\mathbf{W}_1, \mathbf{W}$$

- Given a projective view $\Pi_{\mathbf{W}}$, a complement $\Pi_{\mathbf{W}'}$ is

Comparison of Projections and Optimal Co

Context: A universal relational schema $R[\mathbf{U}]$ constrained by s

- A Π-*view* is defined by a single projection.

Notation: Π_W is the projection onto attribute set W.

- Projective views may be compared via their attributes.

$$\Pi_{\mathbf{W}_1} \preceq \Pi_{\mathbf{W}_2} \text{ iff } \mathbf{W}_1 \subseteq \mathbf{W}_2 \qquad (\mathbf{W}_1, \mathbf{W}$$

- Given a projective view $\Pi_{\mathbf{W}}$, a complement $\Pi_{\mathbf{W}'}$ is
  - *minimal* if for no other complement $\Pi_{\mathbf{W}''}$ is it the cas

Comparison of Projections and Optimal Co

Context: A universal relational schema $R[\mathbf{U}]$ constrained by s

- A Π-*view* is defined by a single projection.

Notation: Π_W is the projection onto attribute set W.

- Projective views may be compared via their attributes.

$$\Pi_{\mathbf{W}_1} \preceq \Pi_{\mathbf{W}_2} \text{ iff } \mathbf{W}_1 \subseteq \mathbf{W}_2 \qquad (\mathbf{W}_1, \mathbf{W}$$

- Given a projective view $\Pi_{\mathbf{W}}$, a complement $\Pi_{\mathbf{W}'}$ is
  - *minimal* if for no other complement $\Pi_{\mathbf{W}''}$ is it the cas
  - *optimal* if for every other complement $\Pi_{\mathbf{W}''}$ it is the

A Context for Optimal Complements of P

- Let $R[\mathbf{U}]$ be universal relational schema constrained by sc

A Context for Optimal Complements of P

- Let $R[\mathbf{U}]$ be universal relational schema constrained by sc
- A *governing* JD is a representation of all JDs which hold c

A Context for Optimal Complements of P

- Let $R[\mathbf{U}]$ be universal relational schema constrained by sc
- A *governing* JD is a representation of all JDs which hold c
- More precisely, call a join dependency (JD) $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_$ (w.r.t. $\mathcal{F}$) if it defines a lossless decomposition of $R[\mathbf{U}]$ s properties:

A Context for Optimal Complements of P

- Let $R[\mathbf{U}]$ be universal relational schema constrained by sc
- A *governing* JD is a representation of all JDs which hold c
- More precisely, call a join dependency (JD) $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k$ (w.r.t. $\mathcal{F}$) if it defines a lossless decomposition of $R[\mathbf{U}]$ s properties:
  - *full*: $\mathbf{U}_1 \cup \ldots \cup \mathbf{U}_k = \mathbf{U}$;

A Context for Optimal Complements of P

- Let $R[\mathbf{U}]$ be universal relational schema constrained by sc
- A *governing* JD is a representation of all JDs which hold c
- More precisely, call a join dependency (JD) $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$ (w.r.t. $\mathcal{F}$) if it defines a lossless decomposition of $R[\mathbf{U}]$ s properties:
  - *full*: $\mathbf{U}_1 \cup \ldots \cup \mathbf{U}_k = \mathbf{U}$;
  - *entailed*: $\mathcal{F} \models \bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$;

A Context for Optimal Complements of P

- Let $R[\mathbf{U}]$ be universal relational schema constrained by sc
- A *governing* JD is a representation of all JDs which hold o
- More precisely, call a join dependency (JD) $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$ (w.r.t. $\mathcal{F}$) if it defines a lossless decomposition of $R[\mathbf{U}]$ s properties:
  - *full*: $\mathbf{U}_1 \cup \ldots \cup \mathbf{U}_k = \mathbf{U}$;
  - *entailed*: $\mathcal{F} \models \bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$;
  - *covering*: $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k] \models \varphi$ for every entailed JD

A Context for Optimal Complements of P

- Let $R[\mathbf{U}]$ be universal relational schema constrained by sc
- A *governing* JD is a representation of all JDs which hold o
- More precisely, call a join dependency (JD) $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$ (w.r.t. $\mathcal{F}$) if it defines a lossless decomposition of $R[\mathbf{U}]$ s properties:
  - *full*: $\mathbf{U}_1 \cup \ldots \cup \mathbf{U}_k = \mathbf{U}$;
  - *entailed*: $\mathcal{F} \models \bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$;
  - *covering*: $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k] \models \varphi$ for every entailed JD

Example: For $(R[ABCD]$ with $\mathcal{F} = \{B \to CD, C \to B\}$, the JD $\bowtie[ABC$

A Context for Optimal Complements of P

- Let $R[\mathbf{U}]$ be universal relational schema constrained by sc
- A *governing* JD is a representation of all JDs which hold c
- More precisely, call a join dependency (JD) $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$ (w.r.t. $\mathcal{F}$) if it defines a lossless decomposition of $R[\mathbf{U}]$ s properties:
  - *full*: $\mathbf{U}_1 \cup \ldots \cup \mathbf{U}_k = \mathbf{U}$;
  - *entailed*: $\mathcal{F} \models \bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$;
  - *covering*: $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k] \models \varphi$ for every entailed JD

Example: For $(R[ABCD]$ with $\mathcal{F} = \{B \to CD, C \to B\}$, the JD $\bowtie[ABC$

Example: For $(R[ABCD]$ with $\mathcal{F} = \{\bowtie[AB, BC]\}$, there is governing JD.

Normalization and Nonredundanc

- To address the non-uniqueness of complements illustrated
  following condition is introduced for the JD $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}$

Normalization and Nonredundanc

- To address the non-uniqueness of complements illustrated
following condition is introduced for the JD $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}$
- *nonredundant*: for no proper $J \subsetneq \{\mathbf{U}_1, \ldots, \mathbf{U}_k\}$ is $\bowtie [J]$ b

Normalization and Nonredundanc

- To address the non-uniqueness of complements illustrated
following condition is introduced for the JD $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}$
- *nonredundant*: for no proper $J \subsetneq \{\mathbf{U}_1, \ldots, \mathbf{U}_k\}$ is $\bowtie[J]$ b
- There are two flavors of redundancy:

Normalization and Nonredundanc

- To address the non-uniqueness of complements illustrated
  following condition is introduced for the JD $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}$
- *nonredundant*: for no proper $J \subsetneq \{\mathbf{U}_1, \ldots, \mathbf{U}_k\}$ is $\bowtie [J]$ b
- There are two flavors of redundancy:
  - $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}_k]$ is *trivially redundant* (not *normalized*
    distinct i, j.

Normalization and Nonredundanc

- To address the non-uniqueness of complements illustrated
  following condition is introduced for the JD $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}$
- *nonredundant*: for no proper $J \subsetneq \{\mathbf{U}_1, \ldots, \mathbf{U}_k\}$ is $\bowtie [J]$ b
- There are two flavors of redundancy:
  - $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}_k]$ is *trivially redundant* (not *normalized*
    distinct i, j.
  - This flavor of redundancy is "trivial" in the sense that
    without any further knowledge of the underlying depe

Normalization and Nonredundanc

- To address the non-uniqueness of complements illustrated
  following condition is introduced for the JD $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}$
- *nonredundant*: for no proper $J \subsetneq \{\mathbf{U}_1, \ldots, \mathbf{U}_k\}$ is $\bowtie [J]$ b
- There are two flavors of redundancy:
  - $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}_k]$ is *trivially redundant* (not *normalized*
    distinct i, j.
  - This flavor of redundancy is "trivial" in the sense that
    without any further knowledge of the underlying depe
  - It may always be removed without changing the sema

Normalization and Nonredundanc

- To address the non-uniqueness of complements illustrated
  following condition is introduced for the JD $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}$
- *nonredundant*: for no proper $J \subsetneq \{\mathbf{U}_1, \ldots, \mathbf{U}_k\}$ is $\bowtie[J]$ b
- There are two flavors of redundancy:
  - $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$ is *trivially redundant* (not *normalized*
    distinct i, j.
  - This flavor of redundancy is "trivial" in the sense that
    without any further knowledge of the underlying depe
  - It may always be removed without changing the sema

  Example: $\bowtie[AC, ABC, CD]$ is trivially redundant.

Normalization and Nonredundanc

- To address the non-uniqueness of complements illustrated
following condition is introduced for the JD $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}$
- *nonredundant*: for no proper $J \subsetneq \{\mathbf{U}_1, \ldots, \mathbf{U}_k\}$ is $\bowtie [J]$ b
- There are two flavors of redundancy:
  - $\bowtie [\mathbf{U}_1, \ldots, \mathbf{U}_k]$ is *trivially redundant* (not *normalized*
  distinct i, j.
  - This flavor of redundancy is "trivial" in the sense that
  without any further knowledge of the underlying depe
  - It may always be removed without changing the sema

  Example: $\bowtie [AC, ABC, CD]$ is trivially redundant.
  - Otherwise, redundancy is *essential* and must be deter
  underlying dependencies.

Normalization and Nonredundanc

- To address the non-uniqueness of complements illustrated
following condition is introduced for the JD $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}$
- *nonredundant*: for no proper $J \subsetneq \{\mathbf{U}_1, \ldots, \mathbf{U}_k\}$ is $\bowtie[J]$ b
- There are two flavors of redundancy:
  - $\bowtie[\mathbf{U}_1, \ldots, \mathbf{U}_k]$ is *trivially redundant* (not *normalized*
  distinct i, j.
  - This flavor of redundancy is "trivial" in the sense that
  without any further knowledge of the underlying depe
  - It may always be removed without changing the sema

  Example: $\bowtie[AC, ABC, CD]$ is trivially redundant.
  - Otherwise, redundancy is *essential* and must be deter
  underlying dependencies.

  Example: For $R[ABCD]$ with $\mathcal{F} = \{B \to CD, C \to B\}$
  the JD $\bowtie[ABC, CD, BD]$ is governing but essential
  since $\bowtie[ABC, CD]$ (as well as $\bowtie[ABC, BD]$) is bo

Characterization of Optimal Π-Comple

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s

Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Characterization of Optimal Π-Comple

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Theorem: Let $\mathbf{W} \subseteq \mathbf{U}$, and define $\mathbf{W}' = \bigcup\{\mathbf{U}_i \in \mathcal{A} \mid \mathbf{U}_i \nsubseteq \mathbf{V}$
Then $\Pi_{\mathbf{W}'}$ is an optimal Π-complement of $\Pi_{\mathbf{W}}$.
preserving, then it is furthermore a meet complement. $\square$

Characterization of Optimal Π-Comple

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Theorem: Let $\mathbf{W} \subseteq \mathbf{U}$, and define $\mathbf{W}' = \bigcup\{\mathbf{U}_i \in \mathcal{A} \mid \mathbf{U}_i \nsubseteq \mathbf{V}$
Then $\Pi_{\mathbf{W}'}$ is an optimal Π-complement of $\Pi_{\mathbf{W}}$.
preserving, then it is furthermore a meet complement. $\square$

Example context: $R[ABCDE]$ $C \to D$ $D \to E$ $\models$
$\bowtie[ABC, CD, DE]$ gove

Characterization of Optimal Π-Comple

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Theorem: Let $\mathbf{W} \subseteq \mathbf{U}$, and define $\mathbf{W}' = \bigcup\{\mathbf{U}_i \in \mathcal{A} \mid \mathbf{U}_i \nsubseteq \mathbf{V}$
Then $\Pi_{\mathbf{W}'}$ is an optimal Π-complement of $\Pi_{\mathbf{W}}$.
preserving, then it is furthermore a meet complement. $\square$

Example context: $R[ABCDE]$ $C \to D$ $D \to E$ $\models$
$\bowtie[ABC, CD, DE]$ gove

- The optimal Π-complement of Π_{ABC} is $\Pi_{CD \cup DE} =$

Characterization of Optimal Π-Comple

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Theorem: Let $\mathbf{W} \subseteq \mathbf{U}$, and define $\mathbf{W}' = \bigcup\{\mathbf{U}_i \in \mathcal{A} \mid \mathbf{U}_i \nsubseteq \mathbf{V}$
Then $\Pi_{\mathbf{W}'}$ is an optimal Π-complement of $\Pi_{\mathbf{W}}$.
preserving, then it is furthermore a meet complement. $\square$

Example context: $R[ABCDE]$ $C \to D$ $D \to E$ $\models$
$\bowtie[ABC, CD, DE]$ gove

- The optimal Π-complement of Π_{ABC} is $\Pi_{CD \cup DE} =$
- The optimal Π-complement of Π_{ABCE} is $\Pi_{CD \cup DE} =$

Characterization of Optimal Π-Comple

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Theorem: Let $\mathbf{W} \subseteq \mathbf{U}$, and define $\mathbf{W}' = \bigcup\{\mathbf{U}_i \in \mathcal{A} \mid \mathbf{U}_i \nsubseteq \mathbf{V}$
Then $\Pi_{\mathbf{W}'}$ is an optimal Π-complement of $\Pi_{\mathbf{W}}$.
preserving, then it is furthermore a meet complement. $\square$

Example context: $R[ABCDE]$ $C \to D$ $D \to E$ $\models$
$\bowtie[ABC, CD, DE]$ gove

- The optimal Π-complement of Π_{ABC} is $\Pi_{CD \cup DE} =$
- The optimal Π-complement of Π_{ABCE} is $\Pi_{CD \cup DE} =$
- The optimal Π-complement of Π_{ABCD} is Π_{DE}.

Characterization of Optimal Π-Comple

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Theorem: Let $\mathbf{W} \subseteq \mathbf{U}$, and define $\mathbf{W}' = \bigcup\{\mathbf{U}_i \in \mathcal{A} \mid \mathbf{U}_i \nsubseteq \mathbf{V}$
Then $\Pi_{\mathbf{W}'}$ is an optimal Π-complement of $\Pi_{\mathbf{W}}$.
preserving, then it is furthermore a meet complement. $\square$

Example context: $R[ABCDE]$ $C \to D$ $D \to E$ $\models$
$\bowtie[ABC, CD, DE]$ gove

- The optimal Π-complement of Π_{ABC} is $\Pi_{CD \cup DE} =$
- The optimal Π-complement of Π_{ABCE} is $\Pi_{CD \cup DE} =$
- The optimal Π-complement of Π_{ABCD} is Π_{DE}.
- The optimal Π-complement of Π_{AB} is $\Pi_{ABC \cup CD \cup DE}$

Characterization of Optimal Π-Comple

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Theorem: Let $\mathbf{W} \subseteq \mathbf{U}$, and define $\mathbf{W}' = \bigcup\{\mathbf{U}_i \in \mathcal{A} \mid \mathbf{U}_i \nsubseteq \mathbf{V}$
Then $\Pi_{\mathbf{W}'}$ is an optimal Π-complement of $\Pi_{\mathbf{W}}$.
preserving, then it is furthermore a meet complement. $\square$

Example context: $R[ABCDE]$ $C \to D$ $D \to E$ $\models$
$\bowtie[ABC, CD, DE]$ gove

- The optimal Π-complement of Π_{ABC} is $\Pi_{CD \cup DE} =$
- The optimal Π-complement of Π_{ABCE} is $\Pi_{CD \cup DE} =$
- The optimal Π-complement of Π_{ABCD} is Π_{DE}.
- The optimal Π-complement of Π_{AB} is $\Pi_{ABC \cup CD \cup DE}$
- The optimal Π-complement of Π_{CD} is $\Pi_{ABC \cup DE} =$

An Issue of Suboptimality within a Wider

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ $\bowtie [ABC, CD, DE]$ gove

An Issue of Suboptimality within a Wide

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ $\bowtie [ABC, CD, DE]$ gove

- For Π_{CD}, the optimal Π-complement Π_{ABCDE} allows no constant-complement strategy.

An Issue of Suboptimality within a Wide

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ $\bowtie [ABC, CD, DE]$ gove

- For Π_{CD}, the optimal Π-complement Π_{ABCDE} allows no constant-complement strategy.
- However, some updates are clearly possible.

An Issue of Suboptimality within a Wide

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ $\bowtie [ABC, CD, DE]$ gove

- For Π_{CD}, the optimal Π-complement Π_{ABCDE} allows no constant-complement strategy.
- However, some updates are clearly possible.
- Let $M = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1, \mathrm{e}_1), R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2, \mathrm{d}_2, \mathrm{e}_2)\}$ be the main schema.

An Issue of Suboptimality within a Wide

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ $\bowtie [ABC, CD, DE]$ gove

- For Π_{CD}, the optimal Π-complement Π_{ABCDE} allows no constant-complement strategy.
- However, some updates are clearly possible.
- Let $M = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1, \mathrm{e}_1), R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2, \mathrm{d}_2, \mathrm{e}_2)\}$ be the main schema.
- Consider the update (N, N') to Π_{CD} with $N = \{R(\mathrm{c}_1, \mathrm{d}_1$ $N' = \{R(\mathrm{c}_1, \mathrm{d}_2), R(\mathrm{c}_2, \mathrm{d}_1)\}$.

An Issue of Suboptimality within a Wide

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ $\bowtie [ABC, CD, DE]$ gove

- For Π_{CD}, the optimal Π-complement Π_{ABCDE} allows no constant-complement strategy.
- However, some updates are clearly possible.
- Let $M = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1, \mathrm{e}_1), R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2, \mathrm{d}_2, \mathrm{e}_2)\}$ be the main schema.
- Consider the update (N, N') to Π_{CD} with $N = \{R(\mathrm{c}_1, \mathrm{d}_1$ $N' = \{R(\mathrm{c}_1, \mathrm{d}_2), R(\mathrm{c}_2, \mathrm{d}_1)\}$.
- The reflection $M' = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_2, \mathrm{e}_2) R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2, \mathrm{d}_1, \mathrm{e}$ and Π_{CD} constant.

An Issue of Suboptimality within a Wide

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ $\bowtie [ABC, CD, DE]$ gove

- For Π_{CD}, the optimal Π-complement Π_{ABCDE} allows no constant-complement strategy.
- However, some updates are clearly possible.
- Let $M = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1, \mathrm{e}_1), R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2, \mathrm{d}_2, \mathrm{e}_2)\}$ be the main schema.
- Consider the update (N, N') to Π_{CD} with $N = \{R(\mathrm{c}_1, \mathrm{d}_1$ $N' = \{R(\mathrm{c}_1, \mathrm{d}_2), R(\mathrm{c}_2, \mathrm{d}_1)\}$.
- The reflection $M' = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_2, \mathrm{e}_2) R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2, \mathrm{d}_1, \mathrm{e}$ and Π_{CD} constant.
- The view $\Pi_{ABC} \vee \Pi_{DE}$ which contains two projections, F a complement of Π_{CD}.

An Issue of Suboptimality within a Wide

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ $\bowtie [ABC, CD, DE]$ gove

- For Π_{CD}, the optimal Π-complement Π_{ABCDE} allows no constant-complement strategy.
- However, some updates are clearly possible.
- Let $M = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1, \mathrm{e}_1), R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2, \mathrm{d}_2, \mathrm{e}_2)\}$ be the main schema.
- Consider the update (N, N') to Π_{CD} with $N = \{R(\mathrm{c}_1, \mathrm{d}_1$ $N' = \{R(\mathrm{c}_1, \mathrm{d}_2), R(\mathrm{c}_2, \mathrm{d}_1)\}$.
- The reflection $M' = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_2, \mathrm{e}_2) R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2, \mathrm{d}_1, \mathrm{e}$ and Π_{CD} constant.
- The view $\Pi_{ABC} \vee \Pi_{DE}$ which contains two projections, F a complement of Π_{CD}.
- Thus, (M, M') is a constant-complement reflection of $(N$ $\Pi_{ABC} \vee \Pi_{DE}$.

Complements of $\vee\Pi$-views

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s

Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Complements of $\bigvee\Pi$-views

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

- A $\bigvee\Pi$-view is defined by a set of projections on a (univers

Complements of $\bigvee\Pi$-views

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

- A $\bigvee\Pi$-view is defined by a set of projections on a (univers

Example and notation: $\Pi_{ABC} \vee \Pi_{DE} = \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \{ABC$

Complements of $\bigvee\Pi$-views

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

- A $\bigvee\Pi$-view is defined by a set of projections on a (univers

Example and notation: $\Pi_{ABC} \vee \Pi_{DE} = \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \{ABC$

Theorem: For any partition $\{\mathcal{A}_1, \mathcal{A}_2\}$ of $\mathcal{A}$, $\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{A}_1\}$
are complements. They are furthermore meet complement
dependency preserving. $\square$

Complements of $\bigvee\Pi$-views

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

- A $\bigvee\Pi$-view is defined by a set of projections on a (univers

Example and notation: $\Pi_{ABC} \vee \Pi_{DE} = \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \{ABC$

Theorem: For any partition $\{\mathcal{A}_1, \mathcal{A}_2\}$ of $\mathcal{A}$, $\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{A}_1\}$
are complements. They are furthermore meet complement
dependency preserving. □

Example context: $R[ABCDE]$ $C \rightarrow D$ $D \rightarrow E$ $E \rightarrow F$ $\models$
$\bowtie[ABC, CD, DE, EF]$ is gove

Complements of $\bigvee\Pi$-views

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

- A $\bigvee\Pi$-view is defined by a set of projections on a (univers

Example and notation: $\Pi_{ABC} \vee \Pi_{DE} = \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \{ABC$

Theorem: For any partition $\{\mathcal{A}_1, \mathcal{A}_2\}$ of $\mathcal{A}$, $\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{A}_1\}$
are complements. They are furthermore meet complement
dependency preserving. □

Example context: $R[ABCDE]$ $C \to D$ $D \to E$ $E \to F$ $\models$
$\bowtie[ABC, CD, DE, EF]$ is gove

- $\Pi_{ABC} \vee \Pi_{DE}$ and $\Pi_{CD} \vee \Pi_{EF}$ are meet complements.

Complements of $\bigvee\Pi$-views

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

- A $\bigvee\Pi$-view is defined by a set of projections on a (univers

Example and notation: $\Pi_{ABC} \vee \Pi_{DE} = \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \{ABC$

Theorem: For any partition $\{\mathcal{A}_1, \mathcal{A}_2\}$ of $\mathcal{A}$, $\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{A}_1\}$
are complements. They are furthermore meet complement
dependency preserving. □

Example context: $R[ABCDE]$ $C \to D$ $D \to E$ $E \to F$ $\models$
$\bowtie[ABC, CD, DE, EF]$ is gove

- $\Pi_{ABC} \vee \Pi_{DE}$ and $\Pi_{CD} \vee \Pi_{EF}$ are meet complements.
- Note that $\Pi_{ABC} \vee \Pi_{DE} \neq \Pi_{ABCDE}$ $\Pi_{CD} \vee \Pi_{EF} \neq \Pi_{CDEF}$; they are not even

Comparison of $\bigvee\Pi$-complements

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

Comparison of $\bigvee\Pi$-complements

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

- A first attempt at a definition of comparison:

$$\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_1\} \preceq \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\} \quad \text{iff} \quad (\forall \mathbf{Y} \in \mathcal{B}_1$$

Comparison of $\bigvee\Pi$-complements

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

- A first attempt at a definition of comparison:

$$\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_1\} \leq \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\} \quad \text{iff} \quad (\forall \mathbf{Y} \in \mathcal{B}_1$$

Counterexample: $R[ABCDE]$ $C \to D$ $D \to E \models \bowtie[ABC$
Since the embedded JD $\bowtie[ABC, CD]$ is implied, Π_{ABCD}
as $\Pi_{ABC} \vee \Pi_{CD}$.

Comparison of $\bigvee\Pi$-complements

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

- A first attempt at a definition of comparison:

$$\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_1\} \leq \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\} \quad \text{iff} \quad (\forall \mathbf{Y} \in \mathcal{B}_1$$

Counterexample: $R[ABCDE]$ $C \rightarrow D$ $D \rightarrow E \models \bowtie[ABC$
Since the embedded JD $\bowtie[ABC, CD]$ is implied, Π_{ABCD}
as $\Pi_{ABC} \vee \Pi_{CD}$.

- A better definition of comparison: every LHS attribute set
join of a RHS set.

$$\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_1\} \leq \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\} \quad \text{iff}$$

$$(\forall \mathbf{Y} \in \mathcal{B}_1)(\exists \mathcal{B}_3 \subseteq \mathcal{B}_2)((\bowtie[\mathcal{B}_3] \text{ v}$$

Optimal $\bigvee\Pi$-complements

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s

Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

$$\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_1\} \leq \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\} \quad \text{iff}$$

$$(\forall \mathbf{Y} \in \mathcal{B}_1)(\exists \mathcal{B}_3 \subseteq \mathcal{B}_2)((\bowtie[\mathcal{B}_3] \text{ v}$$

Optimal $\bigvee\Pi$-complements

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s

Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

$$\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_1\} \leq \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\} \quad \text{iff}$$

$$(\forall \mathbf{Y} \in \mathcal{B}_1)(\exists \mathcal{B}_3 \subseteq \mathcal{B}_2)((\bowtie[\mathcal{B}_3] \text{ v}$$

- $\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\}$, is an *optimal* complement of $\bigvee\{\Pi_{\mathbf{Y}} \mid$

  $\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\} \leq \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in$

  for every other $\bigvee\Pi$-comple

Optimal $\bigvee\Pi$-complements

Context: Universal relational schema $R[\mathbf{U}]$ constrained by s
Nonredundant governing JD $\bowtie[\mathcal{A}]$ with $\mathcal{A} \stackrel{\text{def}}{=} \{\mathbf{U}$

$$\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_1\} \leq \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\} \quad \text{iff}$$

$$(\forall \mathbf{Y} \in \mathcal{B}_1)(\exists \mathcal{B}_3 \subseteq \mathcal{B}_2)((\bowtie[\mathcal{B}_3] \text{ v}$$

- $\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\}$, is an *optimal* complement of $\bigvee\{\Pi_{\mathbf{Y}} \mid$
$\bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in \mathcal{B}_2\} \leq \bigvee\{\Pi_{\mathbf{Y}} \mid \mathbf{Y} \in$
for every other $\bigvee\Pi$-comple
- For $\mathbf{W} \subseteq \mathbf{U}$, define JCompl$\langle \mathbf{W}, \mathcal{A} \rangle = \{\mathbf{U}_i \in \mathcal{A} \mid \mathbf{U}_i \nsubseteq \mathbf{W}$

Theorem: $\bigvee\{\Pi_{\mathbf{U}_i} \mid \mathbf{U}_i \in \text{JCompl}\langle \mathbf{W}, \mathcal{A} \rangle\}$ is an optimal $\bigvee\Pi$-

Issues with $\vee\Pi$-complements

- For a wide variety of constraints on the main schema, the are well behaved first-order database dependencies [Fagin [Hull 84 JACM].

Issues with $\bigvee\Pi$-complements

- For a wide variety of constraints on the main schema, the are well behaved first-order database dependencies [Fagin [Hull 84 JACM].
- For $\bigvee\Pi$-views, the situation is very different.

Issues with $\vee\Pi$-complements

- For a wide variety of constraints on the main schema, the
  are well behaved first-order database dependencies [Fagin
  [Hull 84 JACM].

- For $\vee\Pi$-views, the situation is very different.

Example context continued: $R[ABCDE]$ $C \rightarrow D$ $D \rightarrow E$ |

Issues with $\vee\Pi$-complements

- For a wide variety of constraints on the main schema, the
  are well behaved first-order database dependencies [Fagin
  [Hull 84 JACM].

- For $\vee\Pi$-views, the situation is very different.

Example context continued: $R[ABCDE]$ $C \rightarrow D$ $D \rightarrow E$ |

- On $\Pi_{ABC} \vee \Pi_{DE}$, the constraint Cardinality$(\Pi_D) \leqslant$ Card

Issues with $\bigvee\Pi$-complements

- For a wide variety of constraints on the main schema, the
  are well behaved first-order database dependencies [Fagin
  [Hull 84 JACM].

- For $\bigvee\Pi$-views, the situation is very different.

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ |

- On $\Pi_{ABC} \vee \Pi_{DE}$, the constraint Cardinality$(\Pi_D) \leqslant$ Card

- It is not even first order for infinite databases.

Issues with $\bigvee\Pi$-complements

- For a wide variety of constraints on the main schema, the
  are well behaved first-order database dependencies [Fagin
  [Hull 84 JACM].
- For $\bigvee\Pi$-views, the situation is very different.

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ |

- On $\Pi_{ABC} \vee \Pi_{DE}$, the constraint Cardinality$(\Pi_D) \leqslant$ Card
- It is not even first order for infinite databases.
- Fortunately, it does not matter.

Issues with $\bigvee\Pi$-complements

- For a wide variety of constraints on the main schema, the
  are well behaved first-order database dependencies [Fagin
  [Hull 84 JACM].
- For $\bigvee\Pi$-views, the situation is very different.

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ |

- On $\Pi_{ABC} \vee \Pi_{DE}$, the constraint Cardinality$(\Pi_D) \leqslant$ Card
- It is not even first order for infinite databases.
- Fortunately, it does not matter.
- The truth value of such constraints is never altered by a c
  update [Hegner 06 AMAI].

Issues with $\bigvee\Pi$-complements

- For a wide variety of constraints on the main schema, the
  are well behaved first-order database dependencies [Fagin
  [Hull 84 JACM].
- For $\bigvee\Pi$-views, the situation is very different.

Example context continued: $R[ABCDE]$ $C \to D$ $D \to E$ |

- On $\Pi_{ABC} \vee \Pi_{DE}$, the constraint Cardinality$(\Pi_D) \leqslant$ Card
- It is not even first order for infinite databases.
- Fortunately, it does not matter.
- The truth value of such constraints is never altered by a c
  update [Hegner 06 AMAI].
- Only "simple" constraints must be checked for an update.

Moving Beyond the Framework of Proj

Goal: Carry the theory of optimal complements beyond projec

Moving Beyond the Framework of Proj

Goal: Carry the theory of optimal complements beyond projec

- At least include selections, and preferably joins.

Moving Beyond the Framework of Proj

Goal: Carry the theory of optimal complements beyond projec

- At least include selections, and preferably joins.

Principles: Look for a more general theory, as opposed to an a individual cases.

Moving Beyond the Framework of Proj

Goal: Carry the theory of optimal complements beyond projec

- At least include selections, and preferably joins.

Principles: Look for a more general theory, as opposed to an a individual cases.

General contexts:

Moving Beyond the Framework of Proj

Goal: Carry the theory of optimal complements beyond projec

- At least include selections, and preferably joins.

Principles: Look for a more general theory, as opposed to an a individual cases.

General contexts:

- For general principles of schemata and views, for the a simple set-based context.

Moving Beyond the Framework of Proj

Goal: Carry the theory of optimal complements beyond projec

- At least include selections, and preferably joins.

Principles: Look for a more general theory, as opposed to an a individual cases.

General contexts:

- For general principles of schemata and views, for the a simple set-based context.
- For the characterization of views, *information* based u

Moving Beyond the Framework of Proj

Goal: Carry the theory of optimal complements beyond projec

- At least include selections, and preferably joins.

Principles: Look for a more general theory, as opposed to an a individual cases.

General contexts:

- For general principles of schemata and views, for the a simple set-based context.
- For the characterization of views, *information* based u
- For decomposition, the *information semilattice* of equ Boolean queries on the main schema.

Optimality in a General Context

- Comparison of views in a general setting is easy.

Optimality in a General Context

- Comparison of views in a general setting is easy.
- A database schema $\mathbf{D}$ has a set LDB($\mathbf{D}$) of legal states.

Optimality in a General Context

- Comparison of views in a general setting is easy.
- A database schema $\mathbf{D}$ has a set LDB($\mathbf{D}$) of legal states.
- A view $\Gamma = (\mathbf{V}, \gamma)$ of $\mathbf{D}$ consists of a schema $\mathbf{V}$
  together with a surjective morphism γ

Optimality in a General Context

- Comparison of views in a general setting is easy.
- A database schema $\mathbf{D}$ has a set LDB($\mathbf{D}$) of legal states.
- A view $\Gamma = (\mathbf{V}, \gamma)$ of $\mathbf{D}$ consists of a schema $\mathbf{V}$
  together with a surjective morphism γ
- The *congruence* of $\Gamma = (\mathbf{V}, \gamma)$ is
  $\mathrm{Congr}(\Gamma) = \{(M_1, M_2) \in \mathrm{LDB}(\mathbf{D}) \times \mathrm{LDB}$

Optimality in a General Context

- Comparison of views in a general setting is easy.
- A database schema $\mathbf{D}$ has a set LDB($\mathbf{D}$) of legal states.
- A view $\Gamma = (\mathbf{V}, \gamma)$ of $\mathbf{D}$ consists of a schema $\mathbf{V}$
  together with a surjective morphism γ
- The *congruence* of $\Gamma = (\mathbf{V}, \gamma)$ is
  $\mathsf{Congr}(\Gamma) = \{(M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}$
- Define $\Gamma_1 \preceq \Gamma_2$ iff $\mathsf{Congr}(\Gamma_2) \subseteq \mathsf{Congr}(\Gamma_1)$.

Optimality in a General Context

- Comparison of views in a general setting is easy.
- A database schema $\mathbf{D}$ has a set LDB($\mathbf{D}$) of legal states.
- A view $\Gamma = (\mathbf{V}, \gamma)$ of $\mathbf{D}$ consists of a schema $\mathbf{V}$ together with a surjective morphism γ
- The *congruence* of $\Gamma = (\mathbf{V}, \gamma)$ is Congr(Γ) = $\{(M_1, M_2) \in \text{LDB}(\mathbf{D}) \times \text{LDB}$
- Define $\Gamma_1 \preceq \Gamma_2$ iff Congr(Γ_2) $\subseteq$ Congr(Γ_1).
- This definition agrees with those given for Π-views and $\bigvee$

Optimality in a General Context

- Comparison of views in a general setting is easy.
- A database schema $\mathbf{D}$ has a set LDB($\mathbf{D}$) of legal states.
- A view $\Gamma = (\mathbf{V}, \gamma)$ of $\mathbf{D}$ consists of a schema $\mathbf{V}$ together with a surjective morphism γ
- The *congruence* of $\Gamma = (\mathbf{V}, \gamma)$ is Congr(Γ) = $\{(M_1, M_2) \in \text{LDB}(\mathbf{D}) \times \text{LDB}$
- Define $\Gamma_1 \preceq \Gamma_2$ iff Congr(Γ_2) $\subseteq$ Congr(Γ_1).
- This definition agrees with those given for Π-views and $\bigvee$
- Thus, view Γ is optimal in a class $\mathcal{V}$ if its congruence is le $\mathcal{V}$.

Optimality in a General Context

- Comparison of views in a general setting is easy.
- A database schema $\mathbf{D}$ has a set LDB($\mathbf{D}$) of legal states.
- A view $\Gamma = (\mathbf{V}, \gamma)$ of $\mathbf{D}$ consists of a schema $\mathbf{V}$ together with a surjective morphism γ
- The *congruence* of $\Gamma = (\mathbf{V}, \gamma)$ is Congr(Γ) = $\{(M_1, M_2) \in \text{LDB}(\mathbf{D}) \times \text{LDB}$
- Define $\Gamma_1 \preceq \Gamma_2$ iff Congr(Γ_2) $\subseteq$ Congr(Γ_1).
- This definition agrees with those given for Π-views and $\bigvee$
- Thus, view Γ is optimal in a class $\mathcal{V}$ if its congruence is le $\mathcal{V}$.
- Such a view is unique up to the isomorphism class defined

Views Based upon Conjunctive Que

- A *conjunctive query* on a relational schema is a formula de

Views Based upon Conjunctive Que

- A *conjunctive query* on a relational schema is a formula de
- (Single-value) selection, projection, and join are defined by relational algebra:

Views Based upon Conjunctive Que

- A *conjunctive query* on a relational schema is a formula de
- (Single-value) selection, projection, and join are defined by relational algebra:

  Example: $\pi_{AB}(R[ABC])$ is defined by $(\exists z)(R(x_A, x_B, z$

Views Based upon Conjunctive Que

- A *conjunctive query* on a relational schema is a formula de
- (Single-value) selection, projection, and join are defined by relational algebra:

  Example: $\pi_{AB}(R[ABC])$ is defined by $(\exists z)(R(x_A, x_B, z$

  Example: $\sigma_{A=\mathrm{a}}(R[ABC])$ is defined by $R(\mathrm{a}, x_B, x_C)$.

Views Based upon Conjunctive Que

- A *conjunctive query* on a relational schema is a formula de
- (Single-value) selection, projection, and join are defined by relational algebra:

  Example: $\pi_{AB}(R[ABC])$ is defined by $(\exists z)(R(x_A, x_B, z$

  Example: $\sigma_{A=\mathrm{a}}(R[ABC])$ is defined by $R(\mathrm{a}, x_B, x_C)$.

- These define the $\exists\wedge+$-views.

Views Based upon Conjunctive Que

- A *conjunctive query* on a relational schema is a formula de
- (Single-value) selection, projection, and join are defined by relational algebra:

  Example: $\pi_{AB}(R[ABC])$ is defined by $(\exists z)(R(x_A, x_B, z$

  Example: $\sigma_{A=\mathrm{a}}(R[ABC])$ is defined by $R(\mathrm{a}, x_B, x_C)$.

- These define the $\exists\wedge+$-views.
- A *Boolean conjunctive query* or $\exists\wedge+$*-query* contains no fr

Views Based upon Conjunctive Que

- A *conjunctive query* on a relational schema is a formula de
- (Single-value) selection, projection, and join are defined by relational algebra:

  Example: $\pi_{AB}(R[ABC])$ is defined by $(\exists z)(R(x_A, x_B, z$

  Example: $\sigma_{A=\mathrm{a}}(R[ABC])$ is defined by $R(\mathrm{a}, x_B, x_C)$.

- These define the $\exists\wedge+$-views.
- A *Boolean conjunctive query* or *$\exists\wedge+$-query* contains no fr
- The tuples in $\exists\wedge+$-views correspond to Boolean conjuncti schema

Views Based upon Conjunctive Que

- A *conjunctive query* on a relational schema is a formula de
- (Single-value) selection, projection, and join are defined by relational algebra:

  Example: $\pi_{AB}(R[ABC])$ is defined by $(\exists z)(R(x_A, x_B, z$

  Example: $\sigma_{A=\mathrm{a}}(R[ABC])$ is defined by $R(\mathrm{a}, x_B, x_C)$.

- These define the $\exists\wedge+$-views.
- A *Boolean conjunctive query* or *$\exists\wedge+$-query* contains no fr
- The tuples in $\exists\wedge+$-views correspond to Boolean conjuncti schema

  Example: The tuple (b, c) for the view defined by $\pi_{AB}(R$ the Boolean query $(\exists z)(R(\mathrm{a}, \mathrm{b}, z))$.

Views Based upon Conjunctive Que

- A *conjunctive query* on a relational schema is a formula de
- (Single-value) selection, projection, and join are defined by relational algebra:

  Example: $\pi_{AB}(R[ABC])$ is defined by $(\exists z)(R(x_A, x_B, z$

  Example: $\sigma_{A=\mathrm{a}}(R[ABC])$ is defined by $R(\mathrm{a}, x_B, x_C)$.

- These define the $\exists\wedge+$-views.
- A *Boolean conjunctive query* or *$\exists\wedge+$-query* contains no fr
- The tuples in $\exists\wedge+$-views correspond to Boolean conjuncti schema

  Example: The tuple (b, c) for the view defined by $\pi_{AB}(R$ the Boolean query $(\exists z)(R(\mathrm{a}, \mathrm{b}, z))$.

  Example: The tuple $(\mathrm{a}, \mathrm{b}, \mathrm{c})$ for the view defined by $\sigma_{A=}$ to the Boolean query $R(\mathrm{a}, \mathrm{b}, \mathrm{c})$.

Extending and Limiting Views Based upon Conj

Extensions:

- A limitation of $\exists\wedge+$-views is that they recapture only sing

Extending and Limiting Views Based upon Conj

Extensions:

- A limitation of $\exists\wedge+$-views is that they recapture only sing

Example: A selection such as $\sigma_{(A\leqslant 30)}(R[ABC])$ is not recapt

Extending and Limiting Views Based upon Conj

Extensions:

- A limitation of $\exists\wedge+$-views is that they recapture only sing

Example: A selection such as $\sigma_{(A\leqslant 30)}(R[ABC])$ is not recapt

- The developed framework supports such $\sigma\exists\wedge+$-queries for

Extending and Limiting Views Based upon Conj

Extensions:

- A limitation of $\exists\wedge+$-views is that they recapture only sing

Example: A selection such as $\sigma_{(A\leqslant 30)}(R[ABC])$ is not recapt

- The developed framework supports such $\sigma\exists\wedge+$-queries for
- Any subset selection is allowed.

Extending and Limiting Views Based upon Conj

Extensions:

- A limitation of $\exists\wedge+$-views is that they recapture only sing

Example: A selection such as $\sigma_{(A\leqslant 30)}(R[ABC])$ is not recapt

- The developed framework supports such $\sigma\exists\wedge+$-queries for
- Any subset selection is allowed.

Limitations:

- For technical reasons, view definitions which "hide" consta

Extending and Limiting Views Based upon Conj

Extensions:

- A limitation of $\exists\wedge+$-views is that they recapture only sing

Example: A selection such as $\sigma_{(A\leqslant 30)}(R[ABC])$ is not recapt

- The developed framework supports such $\sigma\exists\wedge+$-queries for
- Any subset selection is allowed.

- For technical reasons, view definitions which "hide" consta

  Example: The two definitions $\pi_{BC}(\sigma_{A=\mathrm{a}_1}(R[ABC])$ and
  hide their selection constant in the sense that it is no

The Representation of $\sigma\exists\wedge+$-Views using sets of

- A relation R in a $\sigma\exists\wedge+$-view Γ may be represented by the
  $\exists\wedge+$-queries which are obtained by grounding its defining

The Representation of $\sigma\exists\wedge+$-Views using sets of

- A relation R in a $\sigma\exists\wedge+$-view Γ may be represented by the
  $\exists\wedge+$-queries which are obtained by grounding its defining
- In this approach, each Boolean query corresponds to a pos

The Representation of $\sigma\exists\wedge+$-Views using sets of

- A relation R in a $\sigma\exists\wedge+$-view Γ may be represented by the
  $\exists\wedge+$-queries which are obtained by grounding its defining
- In this approach, each Boolean query corresponds to a pos

  Example: The view defined by $\pi_{AB}(R[ABC])$ correspon
  $\{(\exists z)(R($

The Representation of $\sigma\exists\wedge+$-Views using sets of

- A relation R in a $\sigma\exists\wedge+$-view Γ may be represented by the $\exists\wedge+$-queries which are obtained by grounding its defining
- In this approach, each Boolean query corresponds to a pos

  Example: The view defined by $\pi_{AB}(R[ABC])$ correspon $\{(\exists z)(R($

  Example: The view defined by $\sigma_{A=\mathrm{a}}(R[ABC])$ correspon $\{(R$

The Representation of $\sigma\exists\wedge+$-Views using sets of

- A relation R in a $\sigma\exists\wedge+$-view Γ may be represented by the
  $\exists\wedge+$-queries which are obtained by grounding its defining
- In this approach, each Boolean query corresponds to a pos

  Example: The view defined by $\pi_{AB}(R[ABC])$ correspon
  $\{(\exists z)(R($

  Example: The view defined by $\sigma_{A=\mathrm{a}}(R[ABC])$ correspon
  $\{(R$

  Example: The view defined by $\sigma_{A\in S}(R[ABC])$ correspon
  $\{(R(a,b,c)) \mid$

The Representation of $\sigma\exists\wedge+$-Views using sets of

- A relation R in a $\sigma\exists\wedge+$-view Γ may be represented by the
$\exists\wedge+$-queries which are obtained by grounding its defining
- In this approach, each Boolean query corresponds to a pos

  Example: The view defined by $\pi_{AB}(R[ABC])$ correspon
  $\{(\exists z)(R($

  Example: The view defined by $\sigma_{A=\mathrm{a}}(R[ABC])$ correspo
  $\{(R$

  Example: The view defined by $\sigma_{A\in S}(R[ABC])$ correspo
  $\{(R(a,b,c)) \mid$
- The goal is to be able to represent all relations in the view
Boolean queries.

The Representation of $\sigma\exists\wedge+$-Views using sets of

- A relation R in a $\sigma\exists\wedge+$-view Γ may be represented by the
$\exists\wedge+$-queries which are obtained by grounding its defining
- In this approach, each Boolean query corresponds to a pos

  Example: The view defined by $\pi_{AB}(R[ABC])$ correspon
  $\{(\exists z)(R($

  Example: The view defined by $\sigma_{A=\mathrm{a}}(R[ABC])$ correspo
  $\{(R$

  Example: The view defined by $\sigma_{A\in S}(R[ABC])$ correspo
  $\{(R(a,b,c)) \mid$

- The goal is to be able to represent all relations in the view
Boolean queries.
- This means that the view relation must be recoverable fro
Boolean query.

The Representation of $\sigma\exists\wedge+$-Views using sets of

- A relation R in a $\sigma\exists\wedge+$-view Γ may be represented by the
$\exists\wedge+$-queries which are obtained by grounding its defining
- In this approach, each Boolean query corresponds to a pos

  Example: The view defined by $\pi_{AB}(R[ABC])$ correspon
  $\{(\exists z)(R($

  Example: The view defined by $\sigma_{A=\mathrm{a}}(R[ABC])$ correspo
  $\{(R$

  Example: The view defined by $\sigma_{A\in S}(R[ABC])$ correspo
  $\{(R(a,b,c)) \mid$

- The goal is to be able to represent all relations in the view
Boolean queries.
- This means that the view relation must be recoverable fro
Boolean query.
- The $\exists\wedge+$-formula defining the view is called its *pattern*.

The Representation of $\sigma\exists\wedge+$-Views using sets of

- A relation R in a $\sigma\exists\wedge+$-view Γ may be represented by the
$\exists\wedge+$-queries which are obtained by grounding its defining
- In this approach, each Boolean query corresponds to a pos

  Example: The view defined by $\pi_{AB}(R[ABC])$ correspon
  $\{(\exists z)(R($

  Example: The view defined by $\sigma_{A=\mathrm{a}}(R[ABC])$ correspo
  $\{(R$

  Example: The view defined by $\sigma_{A\in S}(R[ABC])$ correspo
  $\{(R(a,b,c)) \mid$

- The goal is to be able to represent all relations in the view
Boolean queries.
- This means that the view relation must be recoverable fro
Boolean query.
- The $\exists\wedge+$-formula defining the view is called its *pattern*.
- Each Boolean query must correspond to a single pattern.

Concrete and Abstract Views

- A *concrete* view is defined in the usual way, using the relat
  to the $\sigma\exists\wedge+$-context.

Concrete and Abstract Views

- A *concrete* view is defined in the usual way, using the relat
  to the $\sigma\exists\wedge+$-context.
- An *abstract view* consists of a set of Boolean queries, subj
  that it is of *finite pattern index*.

Concrete and Abstract Views

- A *concrete* view is defined in the usual way, using the relat to the $\sigma\exists\wedge+$-context.
- An *abstract view* consists of a set of Boolean queries, subj that it is of *finite pattern index*.
    - This means that there is a finite set of patterns, and matches one of those patterns.

Concrete and Abstract Views

- A *concrete* view is defined in the usual way, using the relat to the $\sigma\exists\wedge+$-context.
- An *abstract view* consists of a set of Boolean queries, subj that it is of *finite pattern index*.
  - This means that there is a finite set of patterns, and matches one of those patterns.
  - This property is essential for recovering a concrete vie

Concrete and Abstract Views

- A *concrete* view is defined in the usual way, using the relat to the $\sigma\exists\wedge+$-context.
- An *abstract view* consists of a set of Boolean queries, subj that it is of *finite pattern index*.
  - This means that there is a finite set of patterns, and matches one of those patterns.
  - This property is essential for recovering a concrete vie

Theorem; There is a natural correspondence between concrete

The Information Semilattice and the Decomp

- Write $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ if the two Boolean queries have the same
  $M \in \mathsf{LDB}(\mathbf{D})$.

The Information Semilattice and the Decomp

- Write $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ if the two Boolean queries have the same $M \in \mathsf{LDB}(\mathbf{D})$.

  Example: $R(\mathrm{a}, \mathrm{b}, \mathrm{c}) \equiv_{\mathbf{D}} (\exists z)(R(\mathrm{a}, \mathrm{b}, z) \wedge (\exists x)(R(x, \mathrm{b}, \mathrm{c}))$ holds.

- Write $[\varphi]$ for the induced equivalence class on φ.

The Information Semilattice and the Decomp

- Write $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ if the two Boolean queries have the same $M \in \mathsf{LDB}(\mathbf{D})$.

  Example: $R(\mathrm{a}, \mathrm{b}, \mathrm{c}) \equiv_{\mathbf{D}} (\exists z)(R(\mathrm{a}, \mathrm{b}, z) \wedge (\exists x)(R(x, \mathrm{b}, \mathrm{c}))$ holds.

- Write $[\varphi]$ for the induced equivalence class on φ.
- Write $[\varphi_1] \sqsubseteq_{\equiv_{\mathbf{D}}} [\varphi_2]$ if $[\varphi_2]$ is true on $\mathbf{D}$ whenever $[\varphi_1]$

The Information Semilattice and the Decomp

- Write $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ if the two Boolean queries have the same $M \in \mathsf{LDB}(\mathbf{D})$.

  Example: $R(\mathrm{a}, \mathrm{b}, \mathrm{c}) \equiv_{\mathbf{D}} (\exists z)(R(\mathrm{a}, \mathrm{b}, z) \wedge (\exists x)(R(x, \mathrm{b}, \mathrm{c}))$ holds.

- Write $[\varphi]$ for the induced equivalence class on φ.
- Write $[\varphi_1] \sqsubseteq_{\equiv_{\mathbf{D}}} [\varphi_2]$ if $[\varphi_2]$ is true on $\mathbf{D}$ whenever $[\varphi_1]$
- This set forms a meet semilattice with top element $[\mathbf{false}]$ $[\mathbf{true}]$.

The Information Semilattice and the Decomp

- Write $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ if the two Boolean queries have the same $M \in \mathsf{LDB}(\mathbf{D})$.

  Example: $R(\mathrm{a}, \mathrm{b}, \mathrm{c}) \equiv_{\mathbf{D}} (\exists z)(R(\mathrm{a}, \mathrm{b}, z) \wedge (\exists x)(R(x, \mathrm{b}, \mathrm{c}))$ holds.

- Write $[\varphi]$ for the induced equivalence class on φ.
- Write $[\varphi_1] \sqsubseteq_{\equiv_{\mathbf{D}}} [\varphi_2]$ if $[\varphi_2]$ is true on $\mathbf{D}$ whenever $[\varphi_1]$
- This set forms a meet semilattice with top element $[\mathbf{false}]$ $[\mathbf{true}]$.
- The key idea is to look for a *decomposition* basis in this se sentence is in the decomposition basis if

The Information Semilattice and the Decomp

- Write $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ if the two Boolean queries have the same $M \in \mathsf{LDB}(\mathbf{D})$.

  Example: $R(\mathrm{a}, \mathrm{b}, \mathrm{c}) \equiv_{\mathbf{D}} (\exists z)(R(\mathrm{a}, \mathrm{b}, z) \wedge (\exists x)(R(x, \mathrm{b}, \mathrm{c}))$ holds.

- Write $[\varphi]$ for the induced equivalence class on φ.
- Write $[\varphi_1] \sqsubseteq_{\equiv_{\mathbf{D}}} [\varphi_2]$ if $[\varphi_2]$ is true on $\mathbf{D}$ whenever $[\varphi_1]$
- This set forms a meet semilattice with top element $[\mathbf{false}]$ $[\mathbf{true}]$.
- The key idea is to look for a *decomposition* basis in this se sentence is in the decomposition basis if
  - it is a useful in a nontrivial way in the representation

The Information Semilattice and the Decomp

- Write $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ if the two Boolean queries have the same $M \in \mathsf{LDB}(\mathbf{D})$.

  Example: $R(\mathrm{a}, \mathrm{b}, \mathrm{c}) \equiv_{\mathbf{D}} (\exists z)(R(\mathrm{a}, \mathrm{b}, z) \wedge (\exists x)(R(x, \mathrm{b}, \mathrm{c}))$ holds.

- Write $[\varphi]$ for the induced equivalence class on φ.
- Write $[\varphi_1] \sqsubseteq_{\equiv_{\mathbf{D}}} [\varphi_2]$ if $[\varphi_2]$ is true on $\mathbf{D}$ whenever $[\varphi_1]$
- This set forms a meet semilattice with top element $[\mathbf{false}]$ $[\mathbf{true}]$.
- The key idea is to look for a *decomposition* basis in this se sentence is in the decomposition basis if
  - it is a useful in a nontrivial way in the representation
  - it cannot be further decomposed in a nontrivial way.

The Information Semilattice and the Decomp

- Write $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ if the two Boolean queries have the same $M \in \mathsf{LDB}(\mathbf{D})$.

  Example: $R(\mathrm{a}, \mathrm{b}, \mathrm{c}) \equiv_{\mathbf{D}} (\exists z)(R(\mathrm{a}, \mathrm{b}, z) \wedge (\exists x)(R(x, \mathrm{b}, \mathrm{c}))$ holds.

- Write $[\varphi]$ for the induced equivalence class on φ.
- Write $[\varphi_1] \sqsubseteq_{\equiv_{\mathbf{D}}} [\varphi_2]$ if $[\varphi_2]$ is true on $\mathbf{D}$ whenever $[\varphi_1]$
- This set forms a meet semilattice with top element $[\mathbf{false}]$ $[\mathbf{true}]$.
- The key idea is to look for a *decomposition* basis in this se sentence is in the decomposition basis if
  - it is a useful in a nontrivial way in the representation
  - it cannot be further decomposed in a nontrivial way.

Example: For the schema $R[ABC]$ constrained by $\bowtie [AB, B$ basis consists of elements of the form $(\exists z)(R(\mathrm{a}, \mathrm{b}, z))$ and

Optimal Complements in a General S

Let $\Gamma = (\mathbf{V}, \gamma)$ be the view whose optimal complement is to b

- All elements of the decomposition basis which "fit" into Γ

Optimal Complements in a General S

Let $\Gamma = (\mathbf{V}, \gamma)$ be the view whose optimal complement is to b

- All elements of the decomposition basis which "fit" into Γ
  - In the case of a JD, this corresponds to identifying th
    are subsumed by some projection of the JD.

Optimal Complements in a General S

Let $\Gamma = (\mathbf{V}, \gamma)$ be the view whose optimal complement is to b

- All elements of the decomposition basis which "fit" into Γ
  - In the case of a JD, this corresponds to identifying th are subsumed by some projection of the JD.
- All other elements of the decomposition basis are used to
  - In the case of a JD, this corresponds to generating a projections of the JD which are not subsumed by the complemented.

Optimal Complements in a General S

Let $\Gamma = (\mathbf{V}, \gamma)$ be the view whose optimal complement is to b

- All elements of the decomposition basis which "fit" into Γ
  - In the case of a JD, this corresponds to identifying th are subsumed by some projection of the JD.
- All other elements of the decomposition basis are used to
  - In the case of a JD, this corresponds to generating a projections of the JD which are not subsumed by the complemented.
- The condition for optimality of a complement is that upon decompositions of tuples using the decomposition basis are

Optimal Complements in a General S

Let $\Gamma = (\mathbf{V}, \gamma)$ be the view whose optimal complement is to b

- All elements of the decomposition basis which "fit" into Γ
  - In the case of a JD, this corresponds to identifying th are subsumed by some projection of the JD.
- All other elements of the decomposition basis are used to
  - In the case of a JD, this corresponds to generating a projections of the JD which are not subsumed by the complemented.
- The condition for optimality of a complement is that upon decompositions of tuples using the decomposition basis are
  - In the context of a single JD, this reduces exactly to nonredundant.

Optimal Complements in a General S

Let $\Gamma = (\mathbf{V}, \gamma)$ be the view whose optimal complement is to b

- All elements of the decomposition basis which "fit" into Γ
  - In the case of a JD, this corresponds to identifying th are subsumed by some projection of the JD.
- All other elements of the decomposition basis are used to
  - In the case of a JD, this corresponds to generating a projections of the JD which are not subsumed by the complemented.
- The condition for optimality of a complement is that upon decompositions of tuples using the decomposition basis are
  - In the context of a single JD, this reduces exactly to nonredundant.
- There are of course many details which have been omitted

Conclusions

- A study of the notion of optimal complements for views of
  been initiated.

Conclusions

- A study of the notion of optimal complements for views of
been initiated.
- A simple and concrete representation for complements of v
projections has been developed fully.

Conclusions

- A study of the notion of optimal complements for views of been initiated.
- A simple and concrete representation for complements of v projections has been developed fully.
- If the governing JD is dependency preserving, then this rep produces meet complements and so is appropriate for the update strategy.

Conclusions

- A study of the notion of optimal complements for views of been initiated.
- A simple and concrete representation for complements of v projections has been developed fully.
- If the governing JD is dependency preserving, then this rep produces meet complements and so is appropriate for the update strategy.
- It identifies in particular the situations in which all of the u are supportable via constant-complement are supportable v and hence via a single update strategy.

Conclusions

- A study of the notion of optimal complements for views of been initiated.
- A simple and concrete representation for complements of v projections has been developed fully.
- If the governing JD is dependency preserving, then this rep produces meet complements and so is appropriate for the update strategy.
- It identifies in particular the situations in which all of the u are supportable via constant-complement are supportable v and hence via a single update strategy.
- A more general theory, not restricted to projections but ra information and Boolean queries has also been developed.

Conclusions

- A study of the notion of optimal complements for views of been initiated.
- A simple and concrete representation for complements of v projections has been developed fully.
- If the governing JD is dependency preserving, then this rep produces meet complements and so is appropriate for the update strategy.
- It identifies in particular the situations in which all of the u are supportable via constant-complement are supportable v and hence via a single update strategy.
- A more general theory, not restricted to projections but ra information and Boolean queries has also been developed.
- That theory provides a beginning to a more general theory further directions.

Further Directions

Effective identification of meet complements:

Further Directions

Effective identification of meet complements:

- The constant-complement update strategy requires m

Further Directions

Effective identification of meet complements:

- The constant-complement update strategy requires m
- This is equivalent to the dependency preservation of t

Further Directions

Effective identification of meet complements:

- The constant-complement update strategy requires m
- This is equivalent to the dependency preservation of t
- An effective means to check this property for more ge constraints and views is needed.

Further Directions

Effective identification of meet complements:

- The constant-complement update strategy requires m
- This is equivalent to the dependency preservation of t
- An effective means to check this property for more ge constraints and views is needed.

Explicit development of a theory of decomposition whic

Further Directions

Effective identification of meet complements:

- The constant-complement update strategy requires m
- This is equivalent to the dependency preservation of t
- An effective means to check this property for more ge constraints and views is needed.

Explicit development of a theory of decomposition whic

- A simple theory for $\bigvee\Pi$-views has been developed.

Further Directions

Effective identification of meet complements:

- The constant-complement update strategy requires m
- This is equivalent to the dependency preservation of t
- An effective means to check this property for more ge constraints and views is needed.

Explicit development of a theory of decomposition whic

- A simple theory for $\bigvee\Pi$-views has been developed.
- It rests upon well-known results for Π-views and basic

Further Directions

Effective identification of meet complements:

- The constant-complement update strategy requires m
- This is equivalent to the dependency preservation of t
- An effective means to check this property for more ge constraints and views is needed.

Explicit development of a theory of decomposition whic

- A simple theory for $\bigvee\Pi$-views has been developed.
- It rests upon well-known results for Π-views and basic
- An expanded framework which includes views defined suggested.

Further Directions

Effective identification of meet complements:

- The constant-complement update strategy requires m
- This is equivalent to the dependency preservation of t
- An effective means to check this property for more ge constraints and views is needed.

Explicit development of a theory of decomposition whic

- A simple theory for $\bigvee\Pi$-views has been developed.
- It rests upon well-known results for Π-views and basic
- An expanded framework which includes views defined suggested.