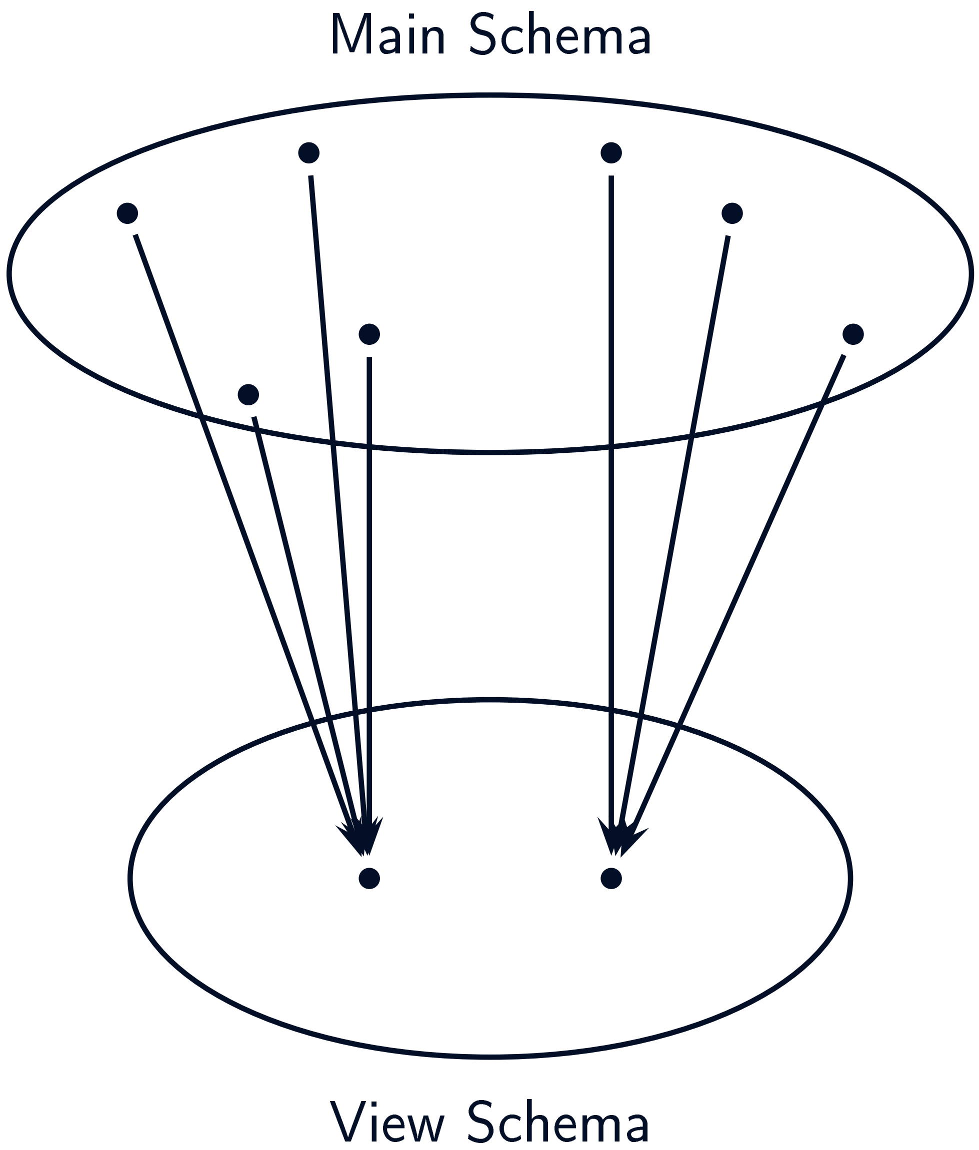
Semantic Bijectivity and the Uniqueness of Constant-Complement Updates in the Relational Context

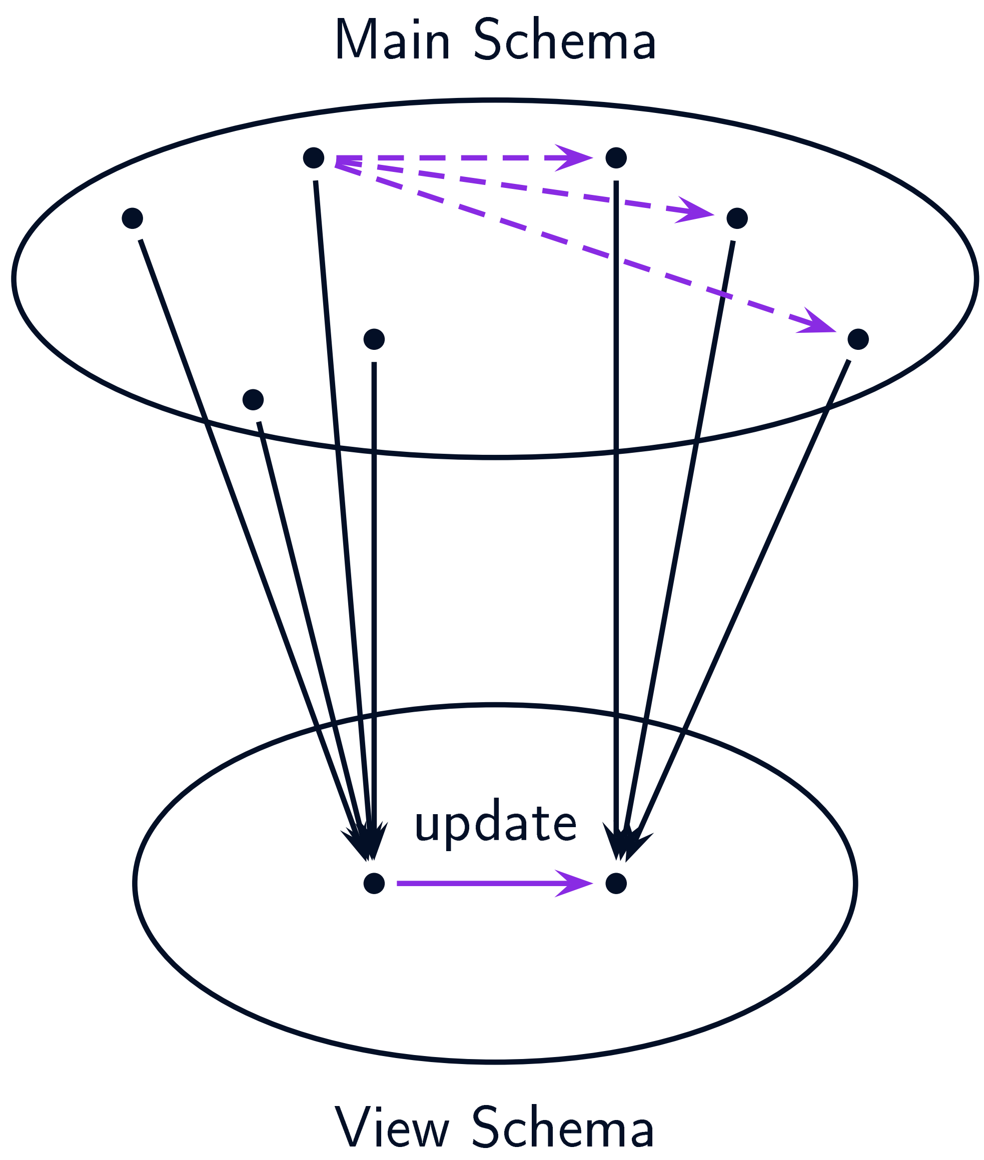
Stephen J. Hegner
Umeå University
Department of Computing Science
Sweden

The Update Problem for Database Views

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

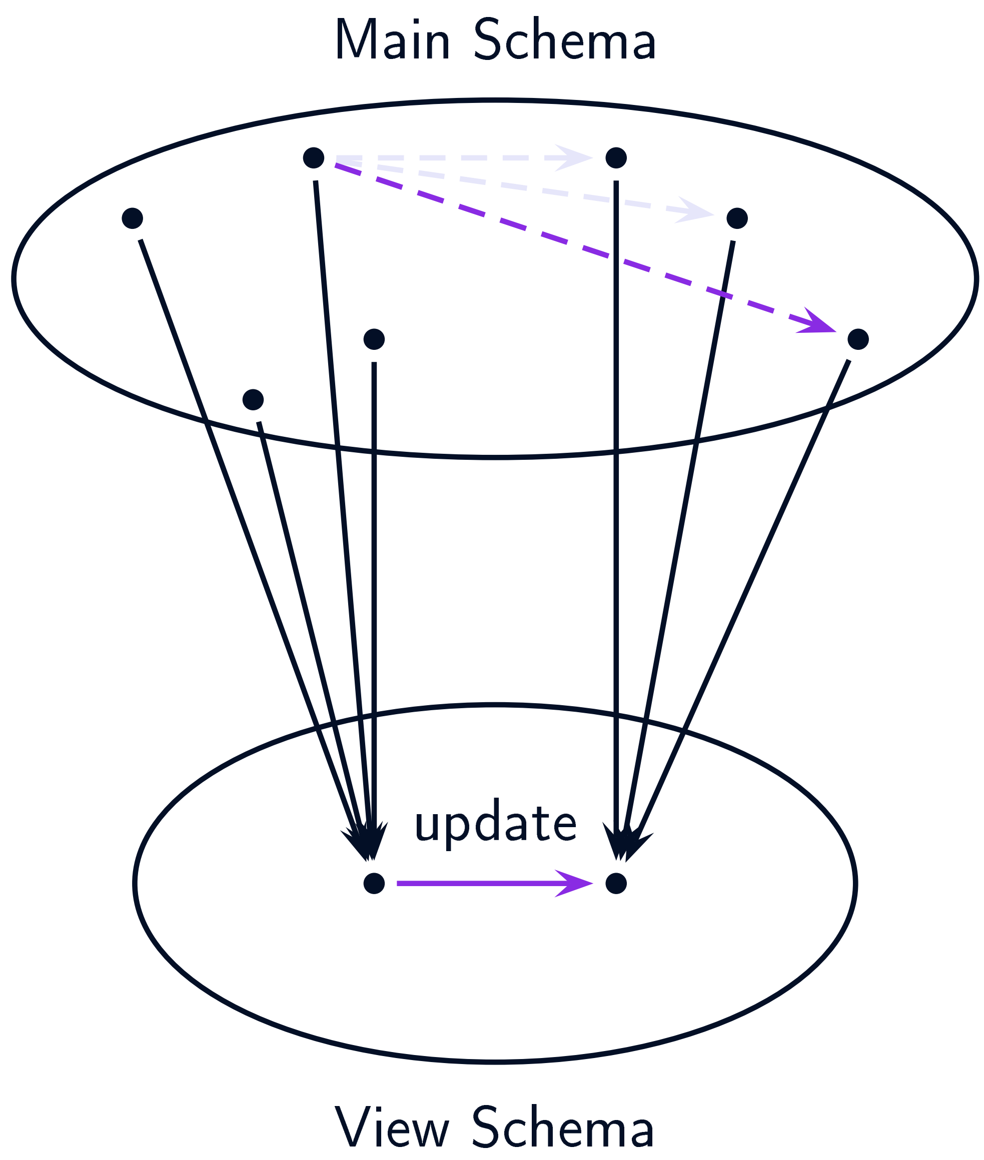


- On the underlying states, the view mapping is generally *surjective* (onto) but not *injective* (one-to-one).
- Thus, a view update has many possible *reflections* to the main schema.



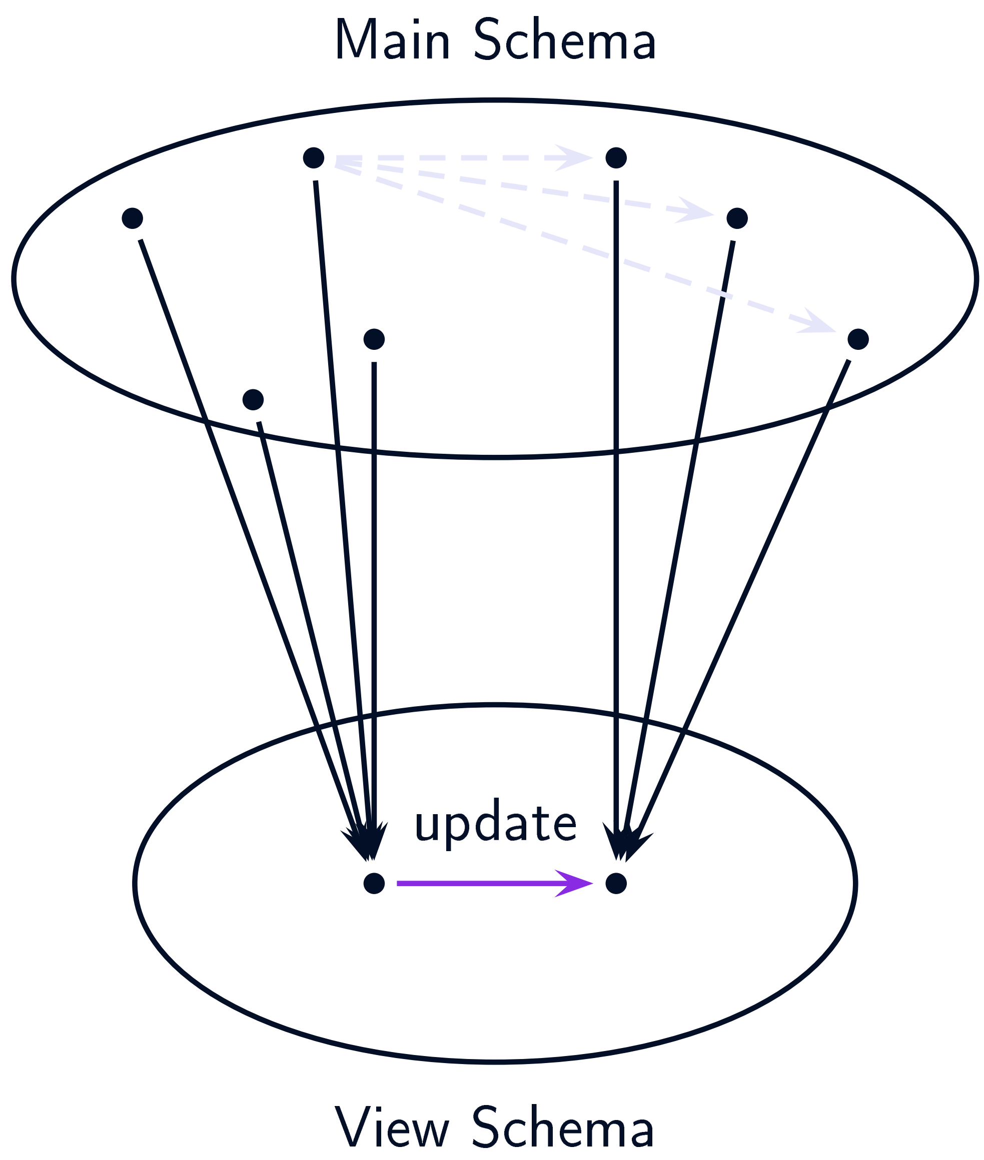
The Update Problem for Database Views

- 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*.

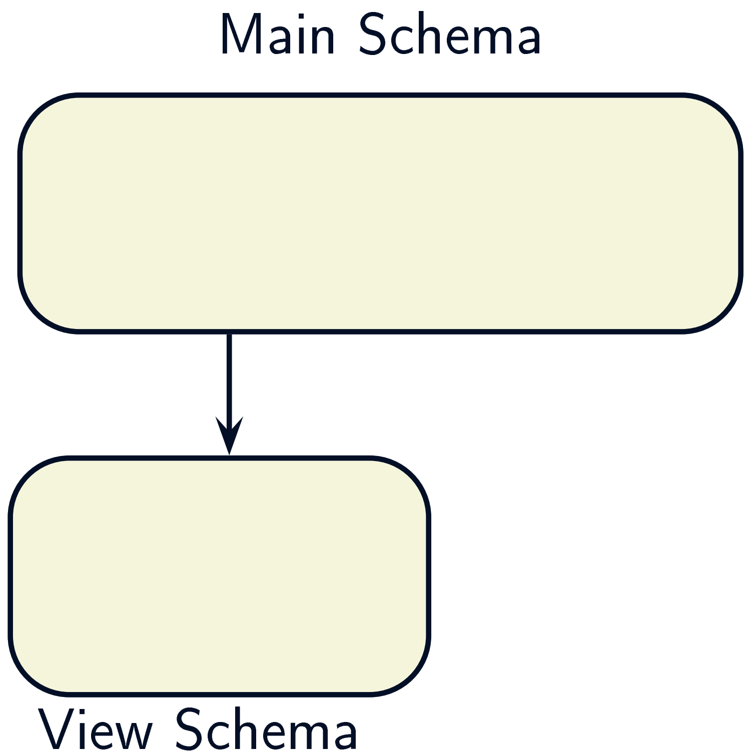


The Update Problem for Database Views

- 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.

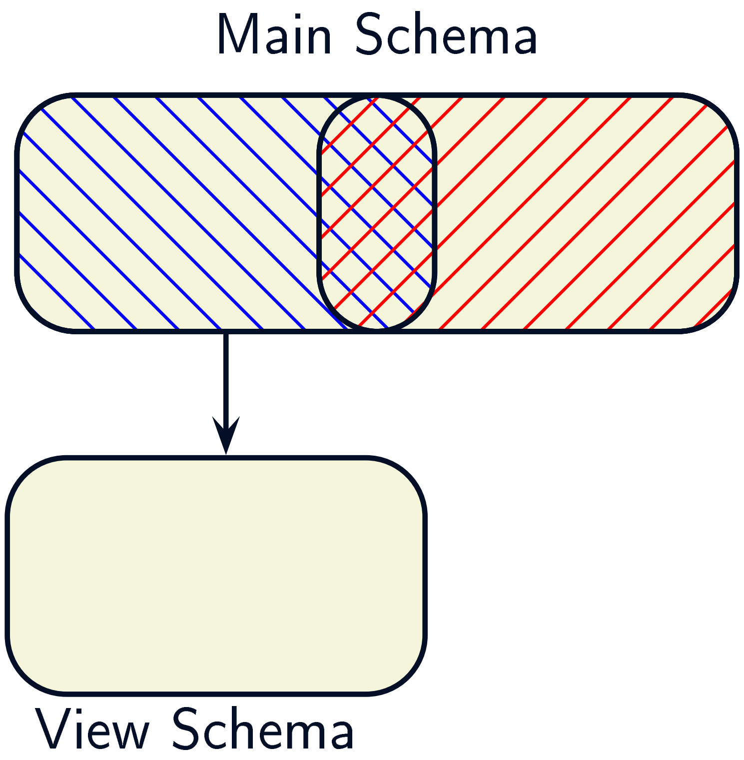


The Gold Standard — the Constant-Complement Strategy



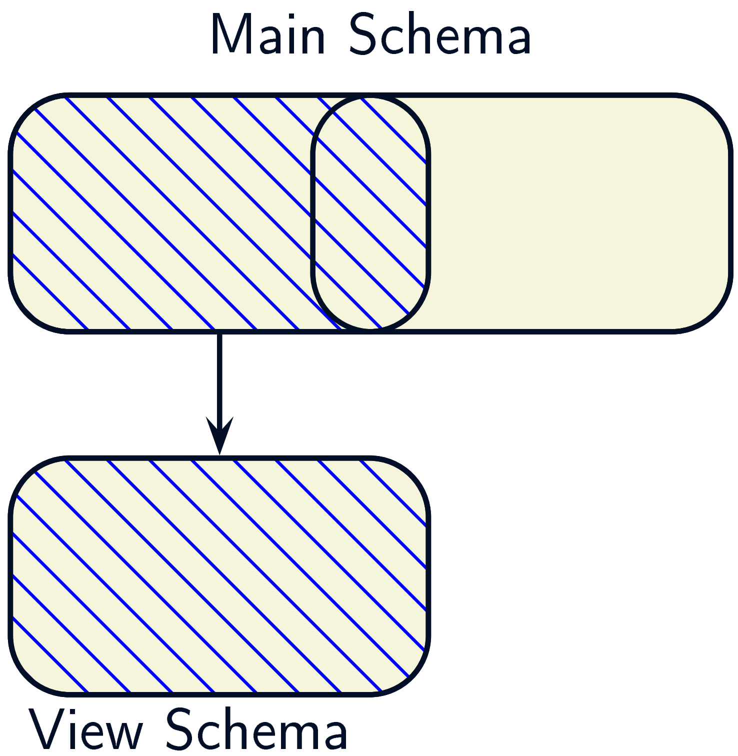
The Gold Standard — the Constant-Complement Strategy

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



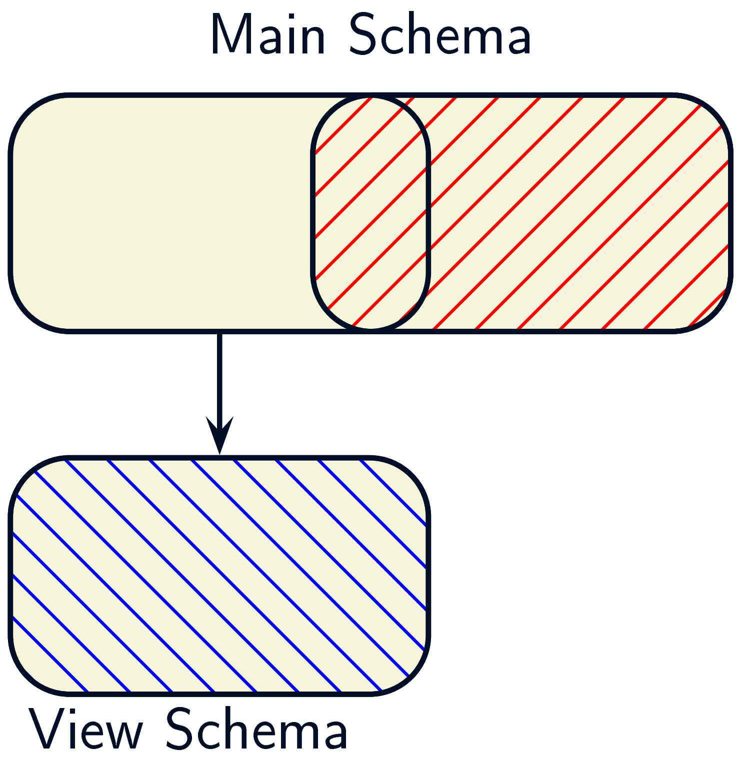
The Gold Standard — the Constant-Complement Strategy

- 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.



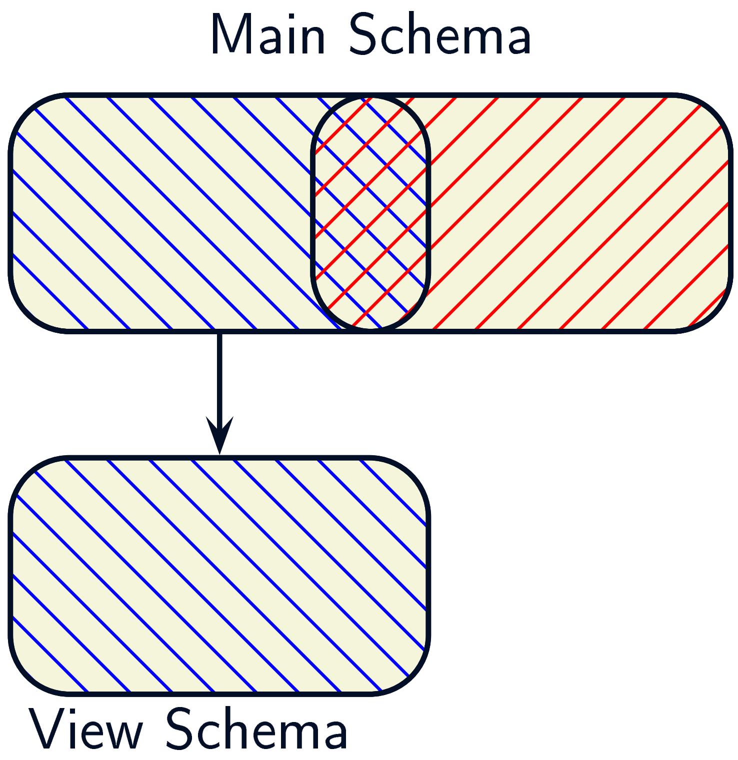
The Gold Standard — the Constant-Complement Strategy

- 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.



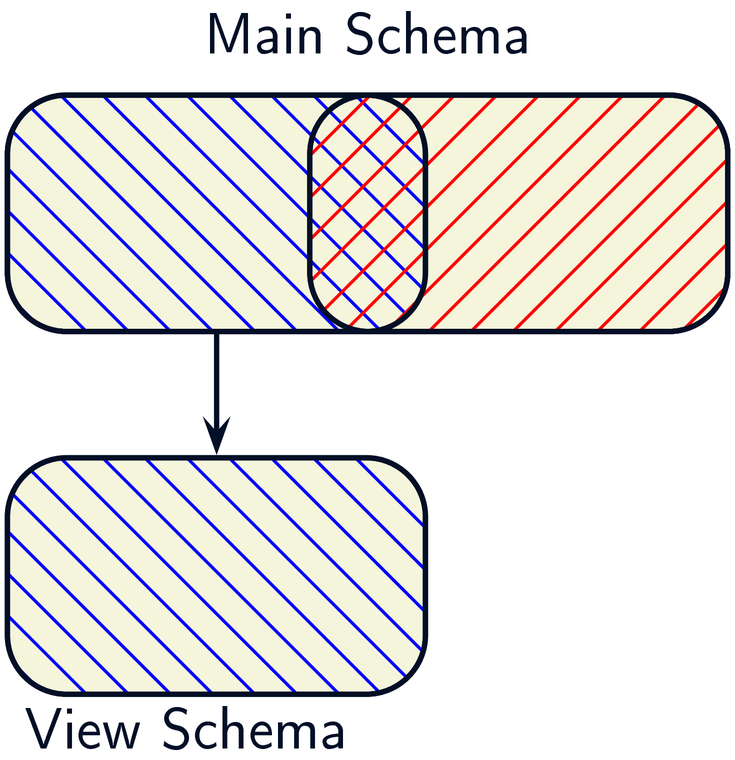
The Gold Standard — the Constant-Complement Strategy

- 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-Complement Strategy

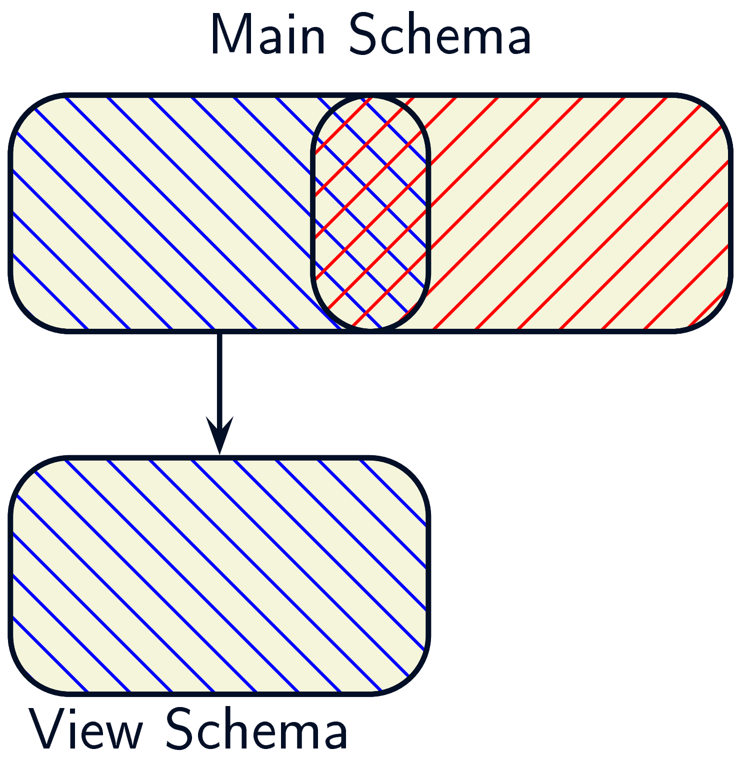
- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.
- One is isomorphic to the view schema and tracks its updates exactly.
- The other is held constant for all updates to the view.
- Although it is somewhat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



- It can be shown [Hegner 03] that this strategy is precisely that which avoids all *update anomalies*.

The Gold Standard — the Constant-Complement Strategy

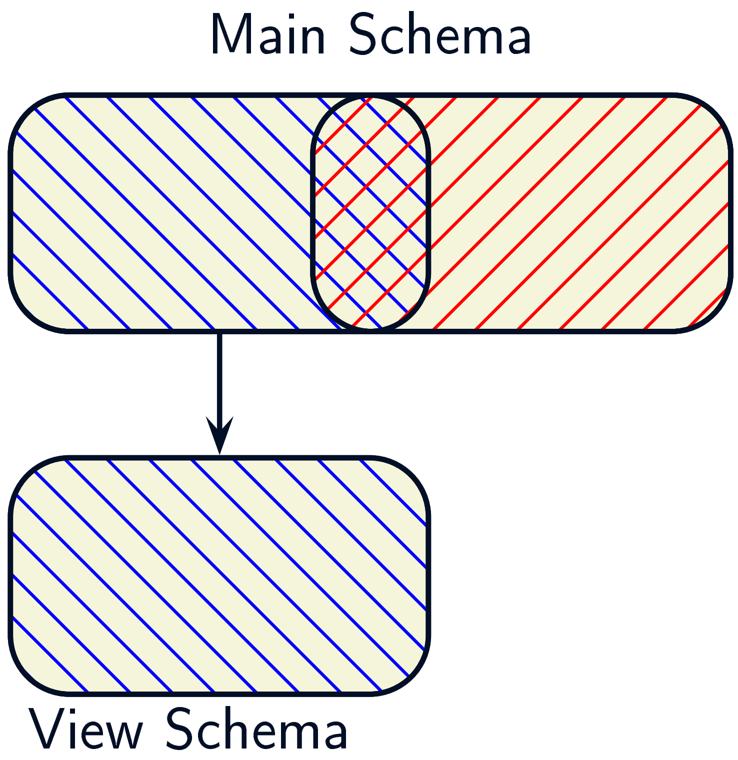
- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.
- One is isomorphic to the view schema and tracks its updates exactly.
- The other is held constant for all updates to the view.
- Although it is somewhat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



- It can be shown [Hegner 03] that this strategy is precisely that which avoids all *update anomalies*.
- However, this is complicated by the *complement uniqueness problem*.

The Gold Standard — the Constant-Complement Strategy

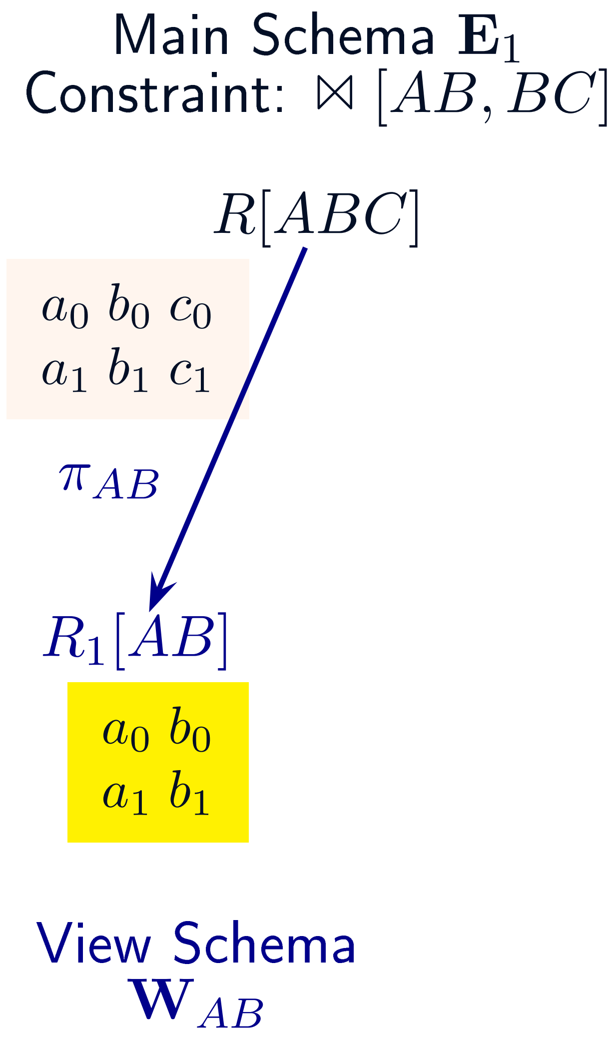
- In the constant-complement strategy [Bancilhon and Spyratos 81], [Hegner 04 AMAI], the main schema is decomposed into two *meet-complementary* views.
- One is isomorphic to the view schema and tracks its updates exactly.
- The other is held constant for all updates to the view.
- Although it is somewhat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



- It can be shown [Hegner 03] that this strategy is precisely that which avoids all *update anomalies*.
- However, this is complicated by the *complement uniqueness problem*.
- Some examples will help illustrate these ideas.

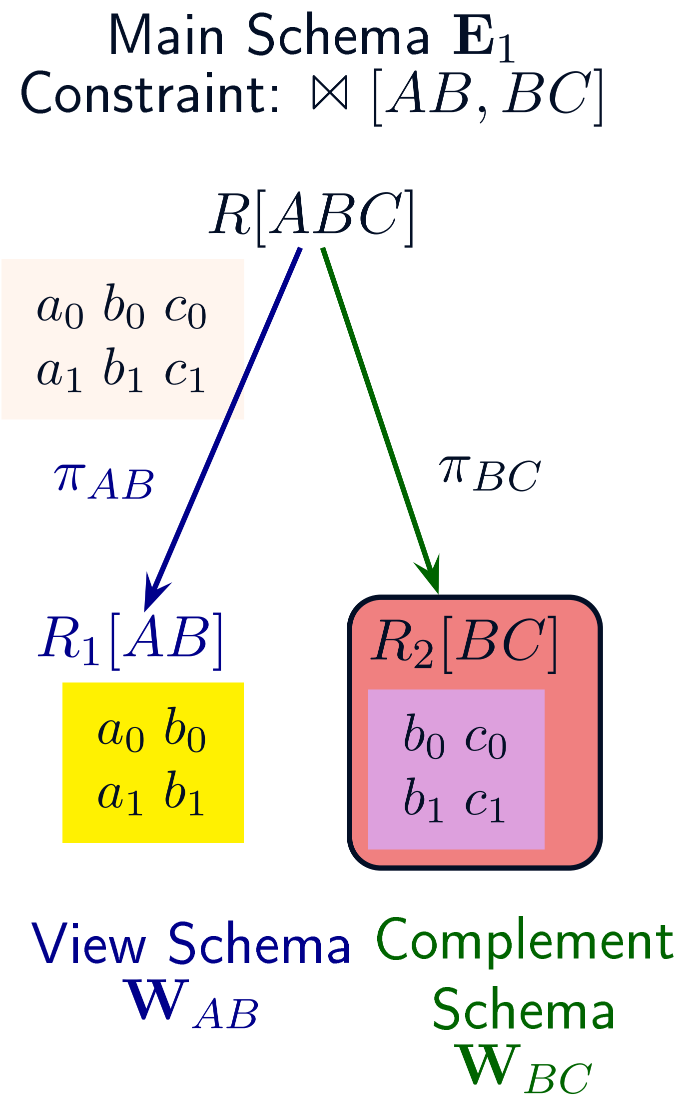
The Idea of Constant-Complement by Example

- Consider the classical example to the right.



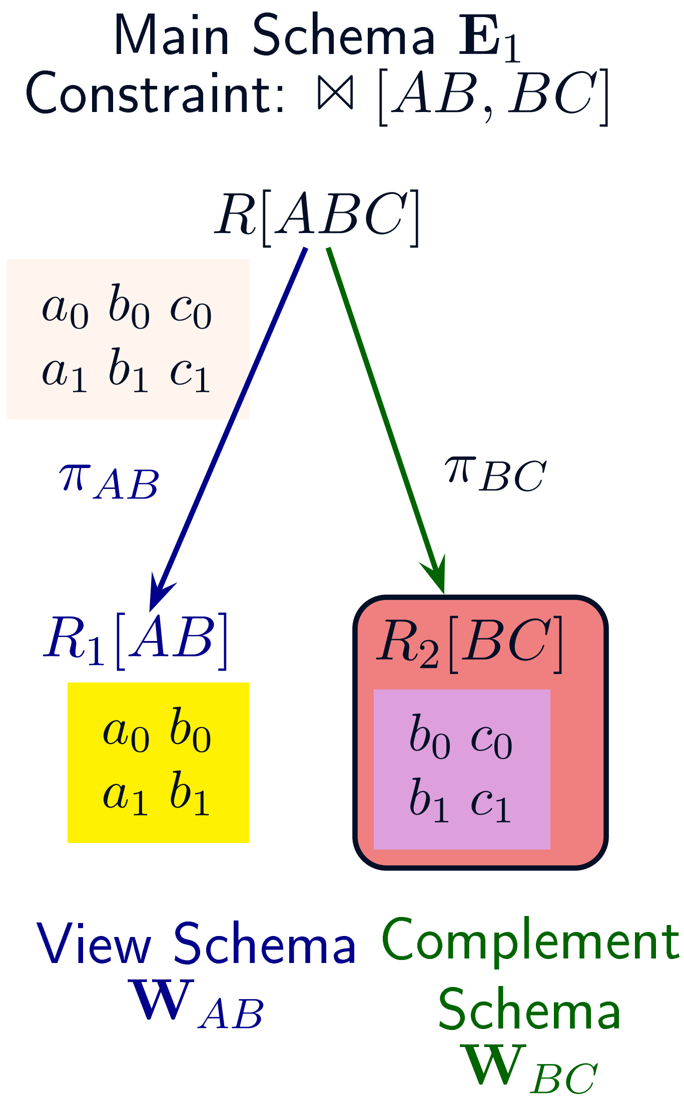
The Idea of Constant-Complement by Example

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



The Idea of Constant-Complement by Example

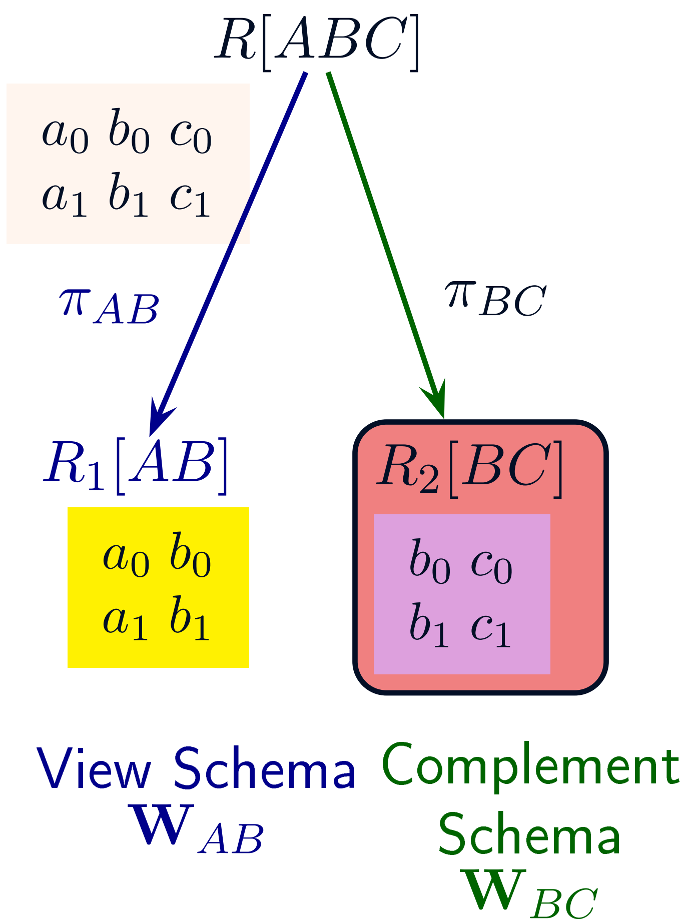
- Consider the classical example to the right.
- A natural complement to the AB-projection is the BC-projection.
- The *decomposed schema* $\mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$ has relation symbols $R_[AB]$ and $R_2[BC]$; the legal database are all states which are join compatible on B.



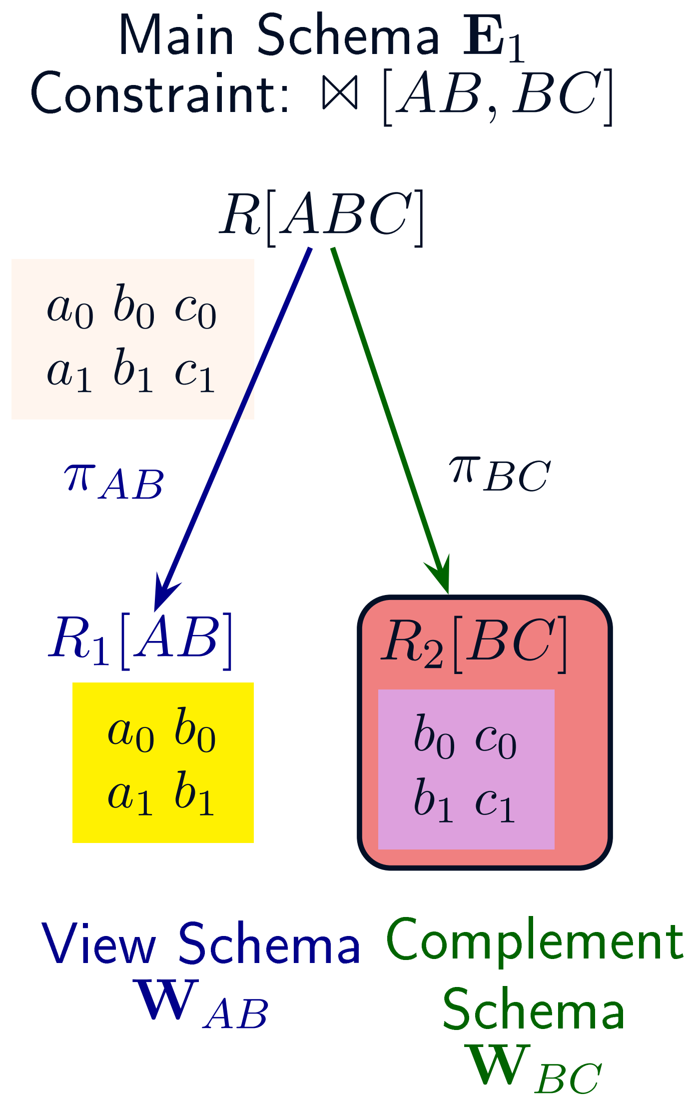
The Idea of Constant-Complement by Example

- 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* $\mathbf{W}_1 \rightarrow \mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$, and is always bijective for complements.

Main Schema $\mathbf{E}_1$
Constraint: $\bowtie [AB, BC]$

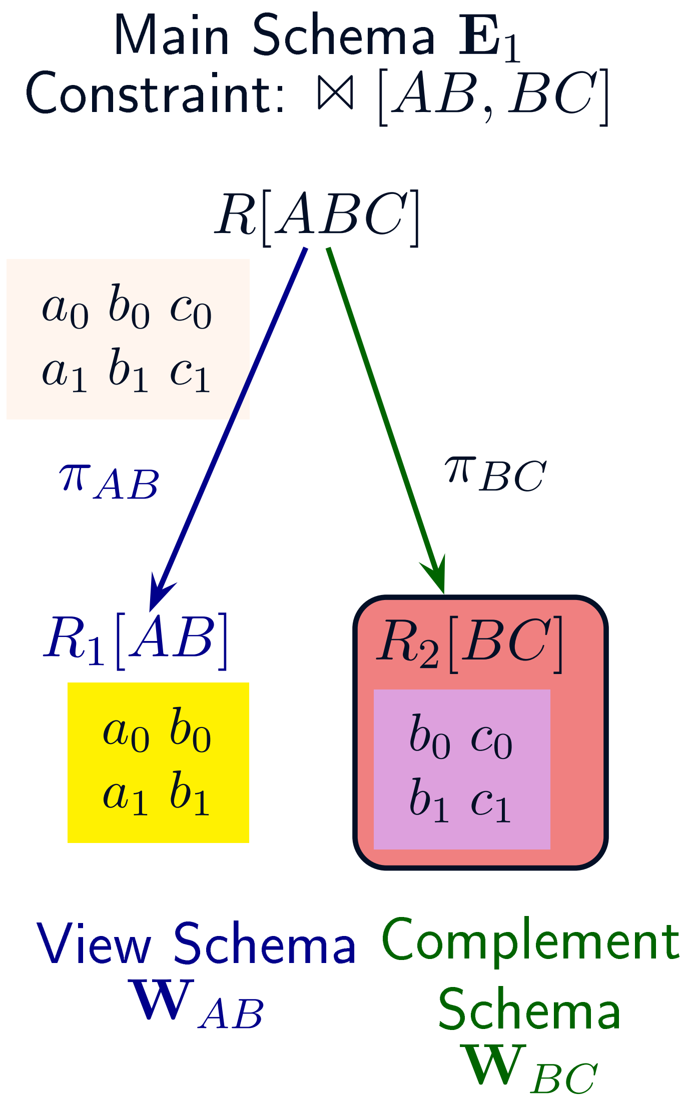


- Consider the classical example to the right.
- A natural complement to the AB-projection is the BC-projection.
- The *decomposed schema* $\mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$ has relation symbols $R_{[}AB]$ and $R_2[BC]$; the legal database are all states which are join compatible on B.
- The *decomposition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

- 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* $\mathbf{W}_1 \to \mathbf{W}_{AB} \otimes \mathbf{W}_{BC}$, 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 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 Uniqueness

- Given is the following two-relation main schema.

Main Schema $\mathbf{E}_0$
No dependencies

$R[A]$ $S[A]$

The Problem of Complement Uniqueness

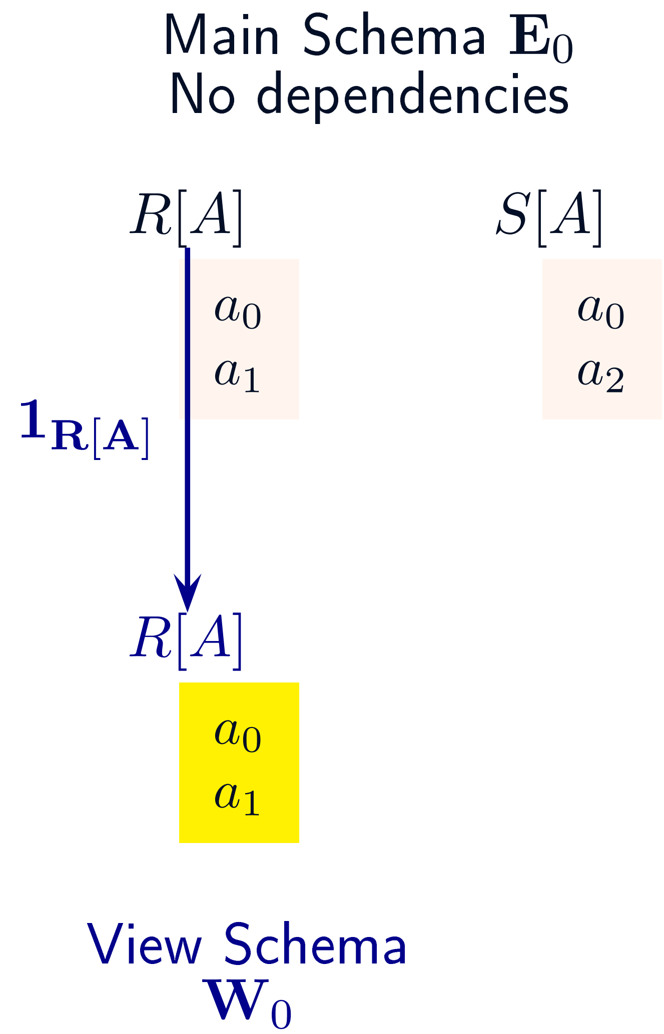
- Given is the following two-relation main schema.

Main Schema $\mathbf{E}_0$
No dependencies

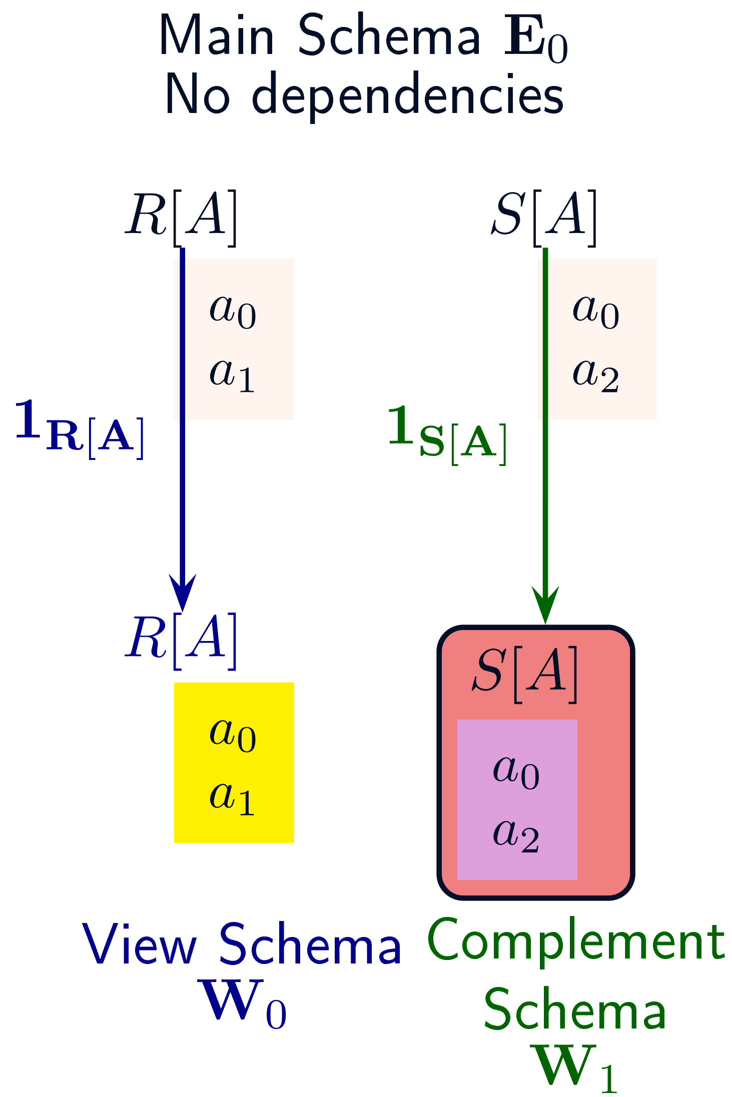
$R[A]$
a_0
a_1

$S[A]$
a_0
a_2

- 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 $S[A]$.

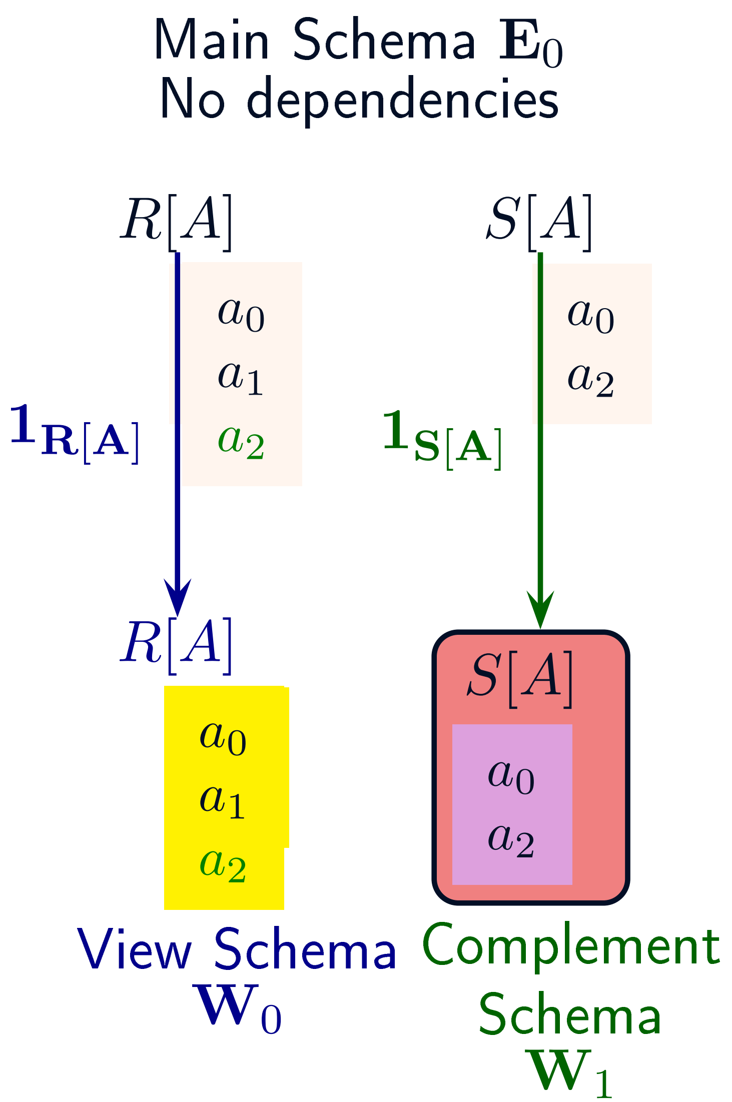


- Given is the following two-relation main schema.
- The view schema $\mathbf{W}_0$ to be updated is that which preserves $R[A]$ but discards $S[A]$.
- The *natural complement* $\mathbf{W}_1$ is the schema which preserves $S[A]$ but discards $R[A]$.



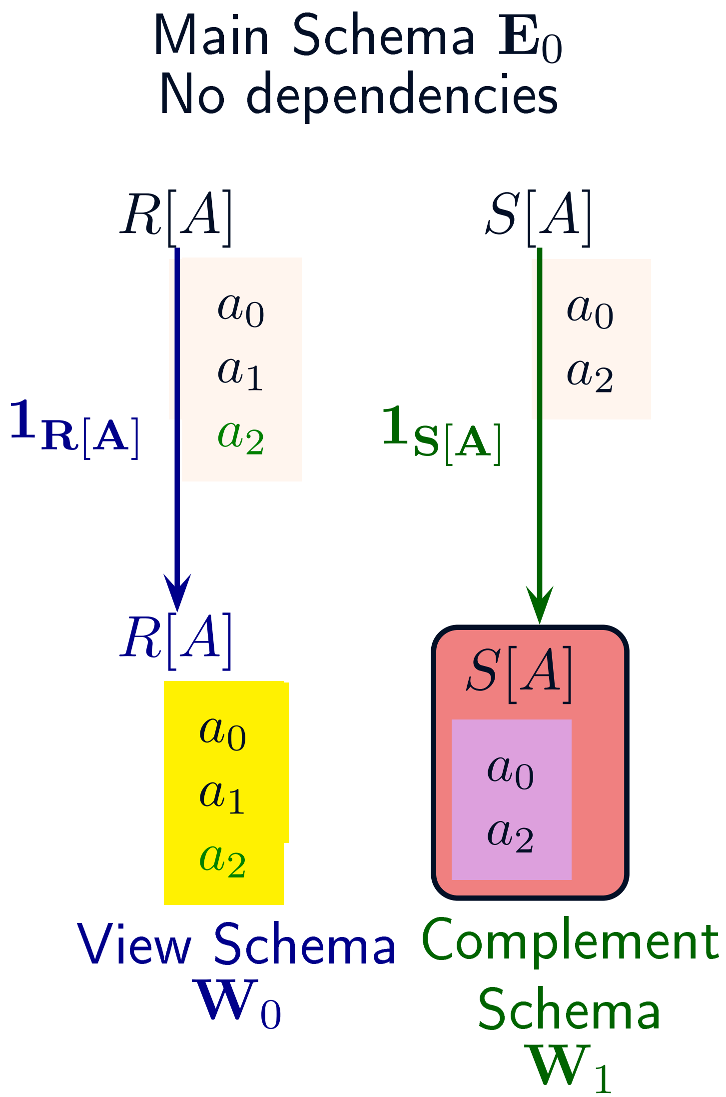
The Problem of Complement Uniqueness

- 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.



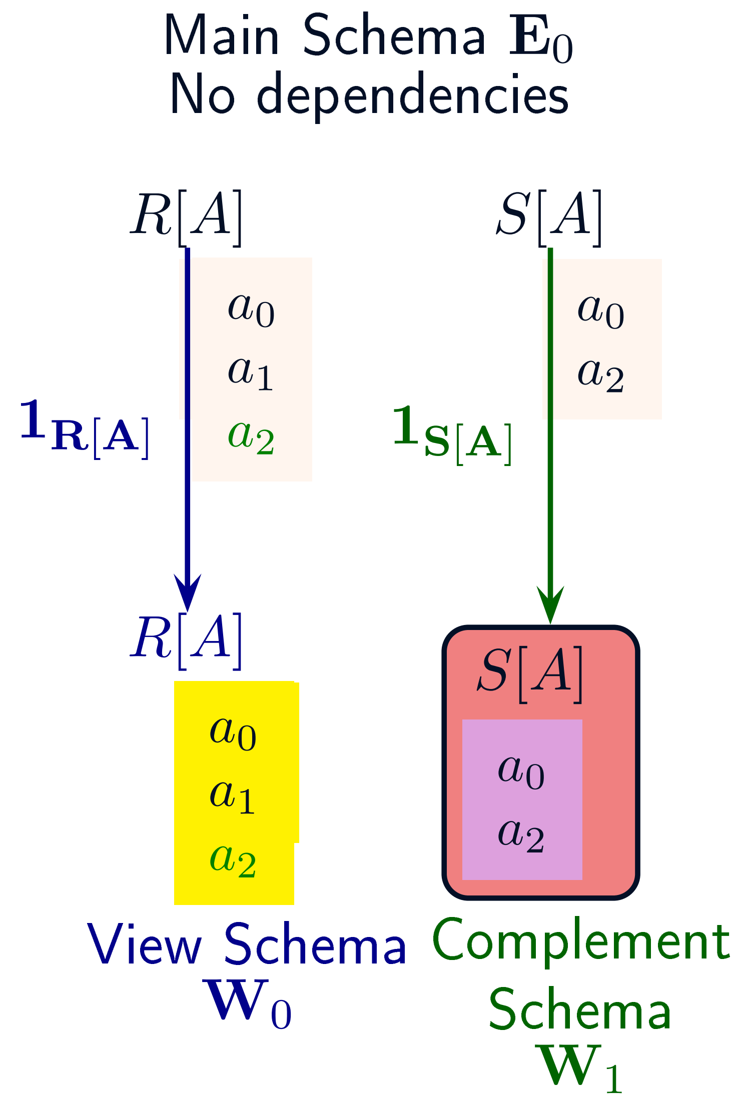
The Problem of Complement Uniqueness

- 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 $\mathbf{W}_0$.



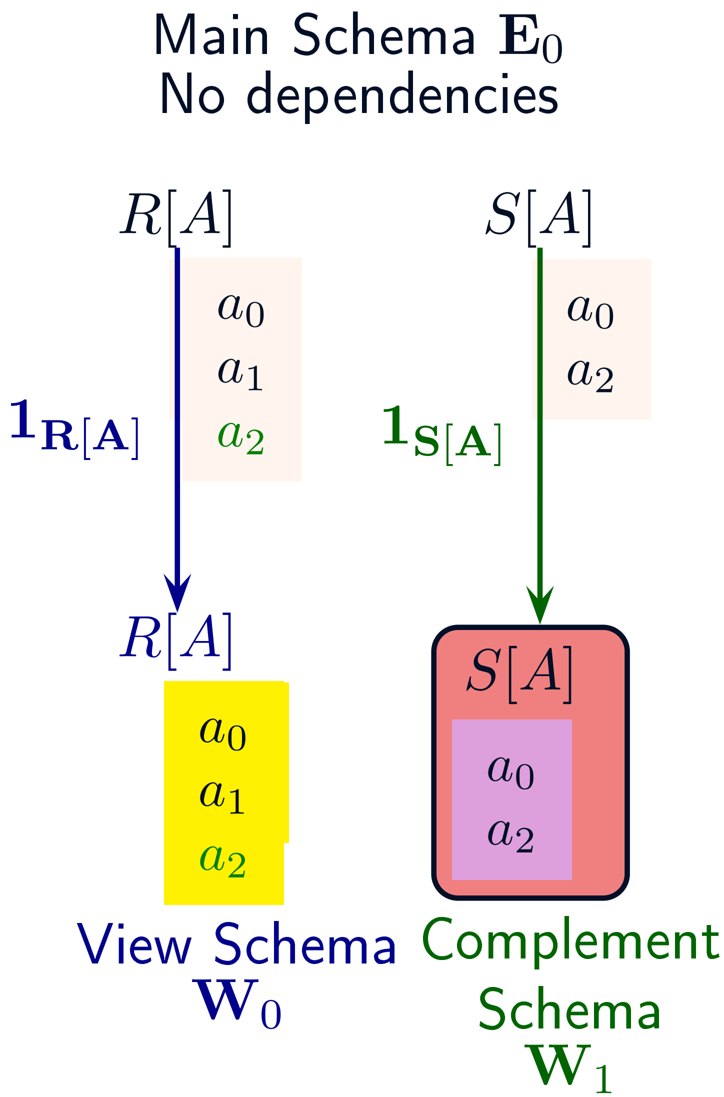
The Problem of Complement Uniqueness

- 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 Uniqueness

- 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

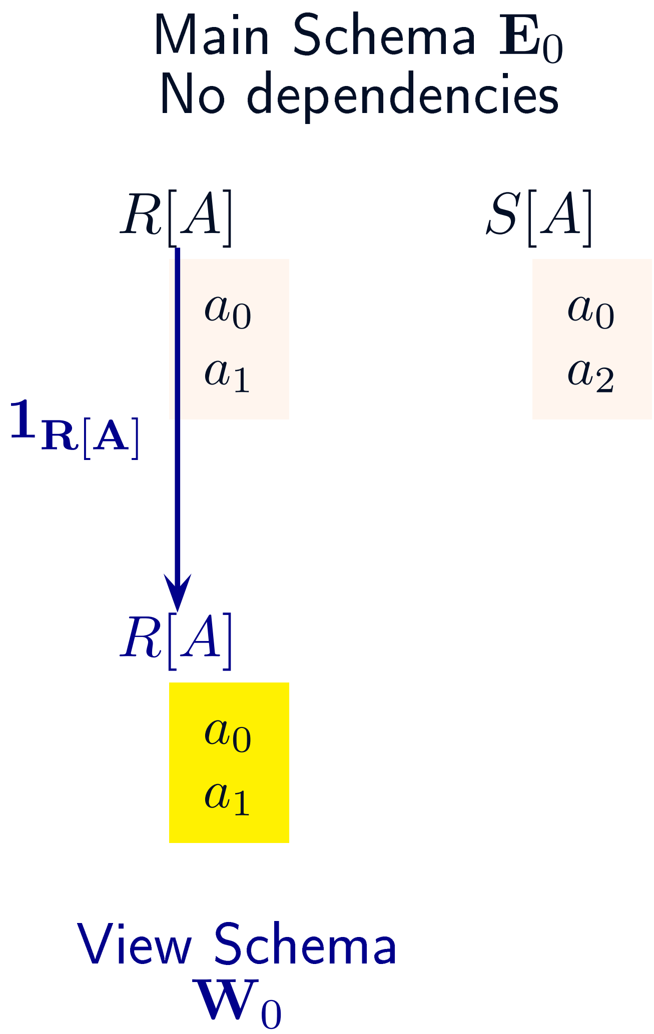
- The main schema is unchanged.

Main Schema $\mathbf{E}_0$
No dependencies

$R[A]$ $S[A]$

An Alternate Complement

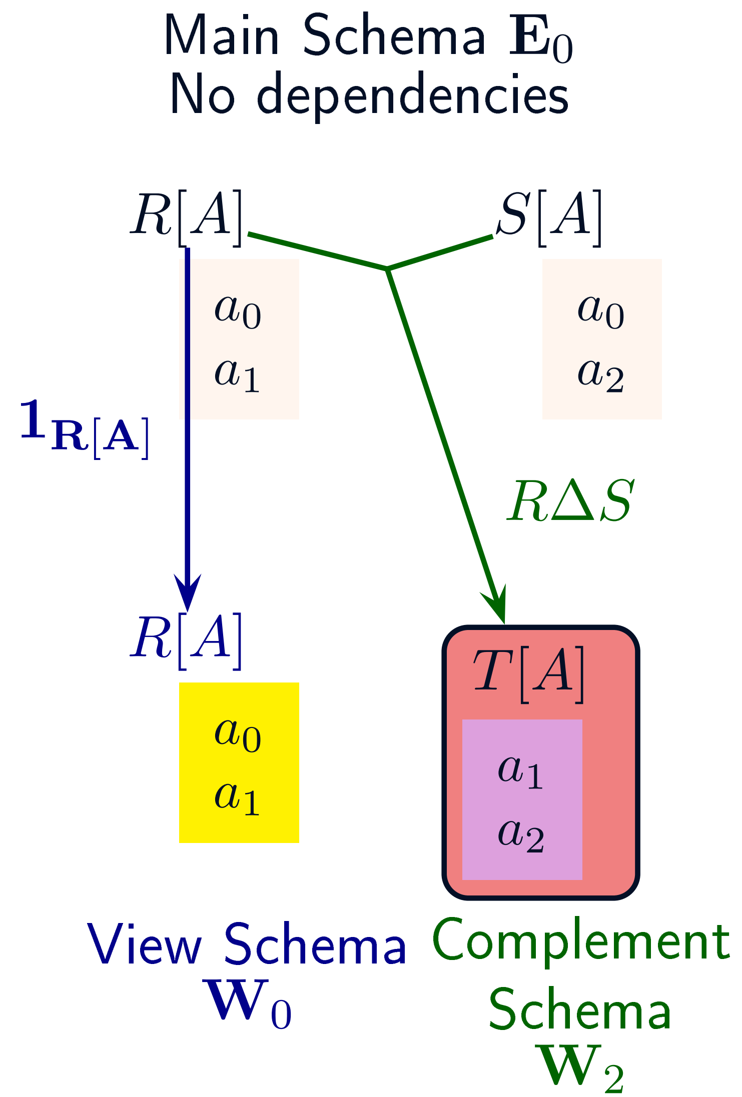
- The main schema is unchanged.
- The view schema $\mathbf{W}_0$ to be updated is also the same.



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:

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

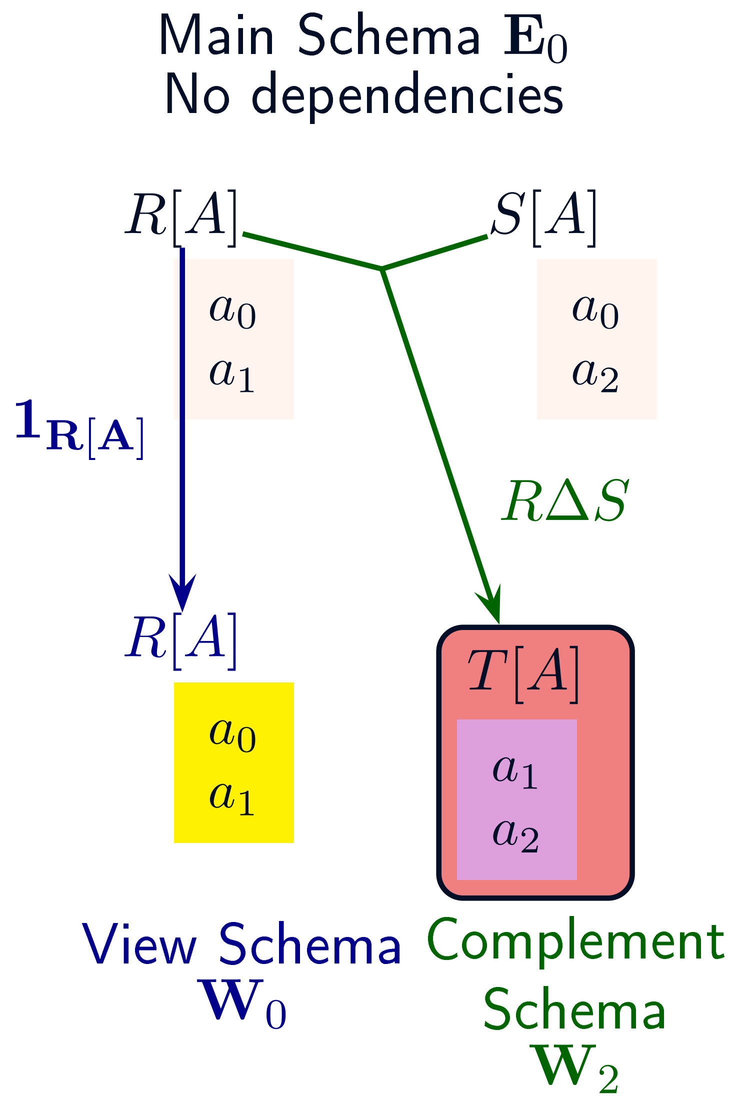


An Alternate Complement

- 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:

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

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

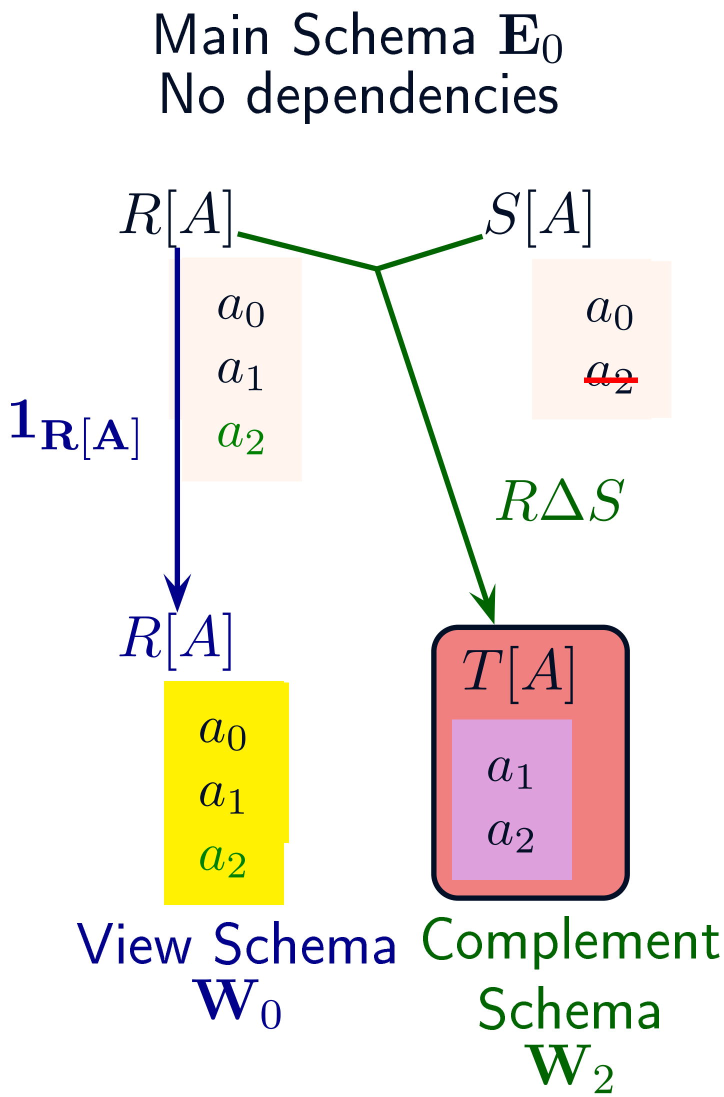


An Alternate Complement

- 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:

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

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



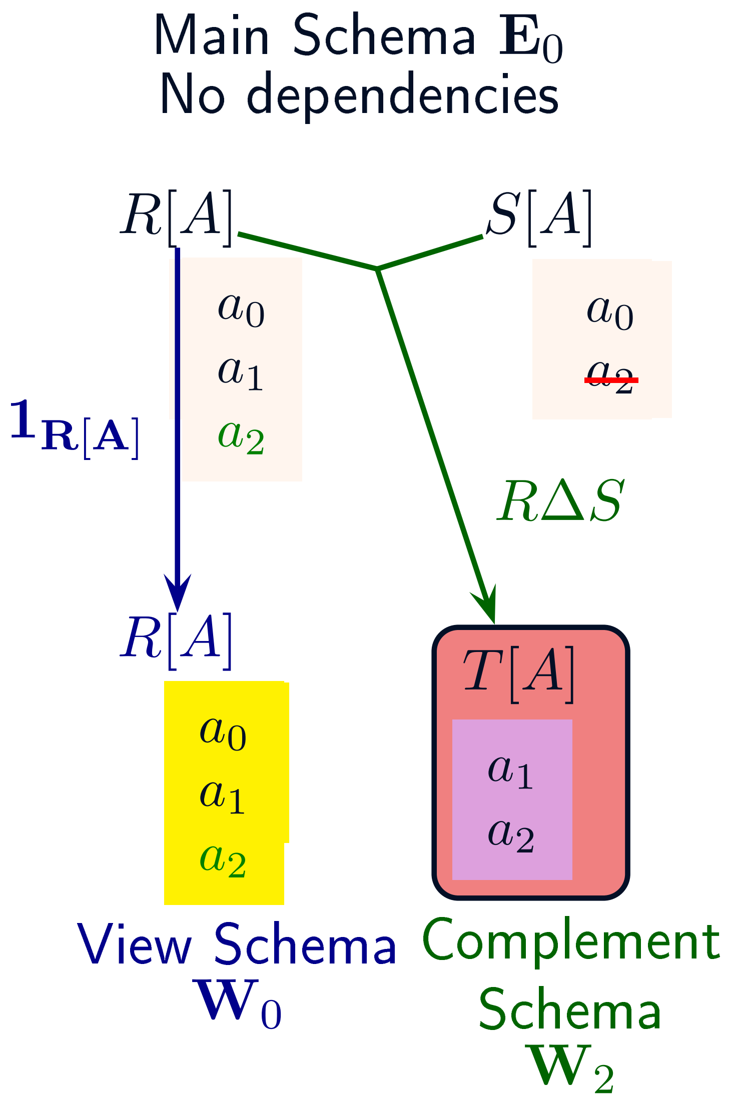
An Alternate Complement

- 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:

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

- 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?



A Partial Solution Based upon Monotonicity

- Note that the symmetric difference mapping is not monotonic with respect to the natural order of database states.

A Partial Solution Based upon Monotonicity

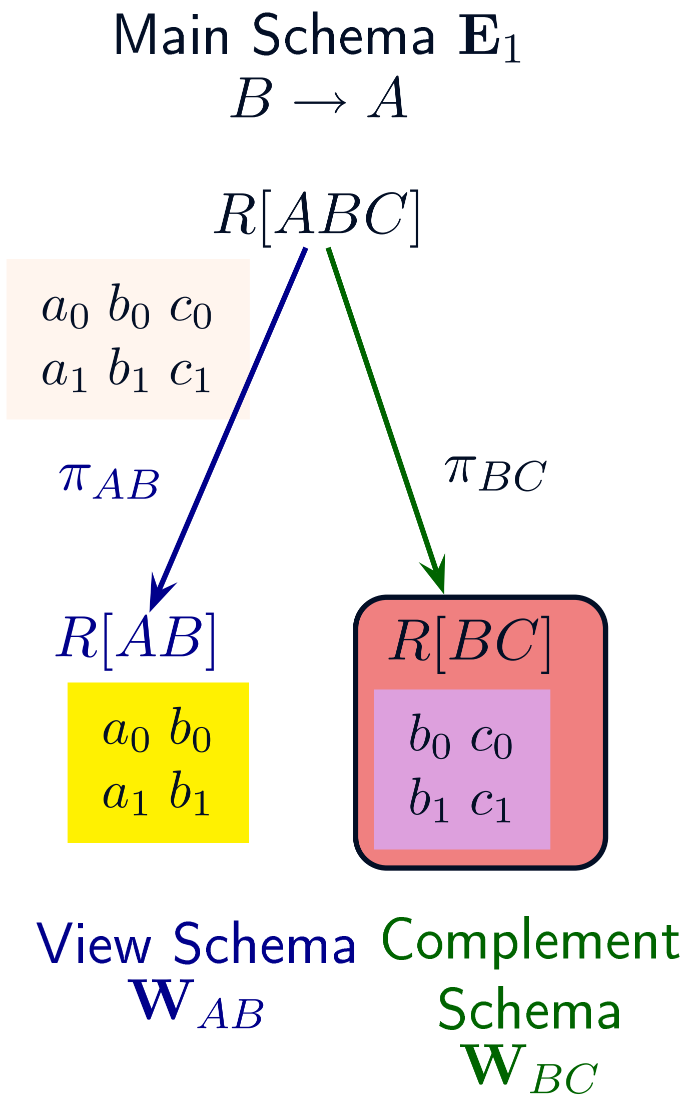
- Note that the symmetric difference mapping is not monotonic with respect to the natural order of database states.
- In earlier work [Hegner04 AMAI], it was shown that *order-realizable* update translations are unique when both view mappings are monotonic and the decomposition mapping is an order isomorphism.

A Partial Solution Based upon Monotonicity

- Note that the symmetric difference mapping is not monotonic with respect to the natural order of database states.
- In earlier work [Hegner04 AMAI], it was shown that *order-realizable* update translations are unique when both view mappings are monotonic and the decomposition mapping is an order isomorphism.
- An *order-realizable* update is one which is realizable as a sequence of **legal** insertions and deletions.

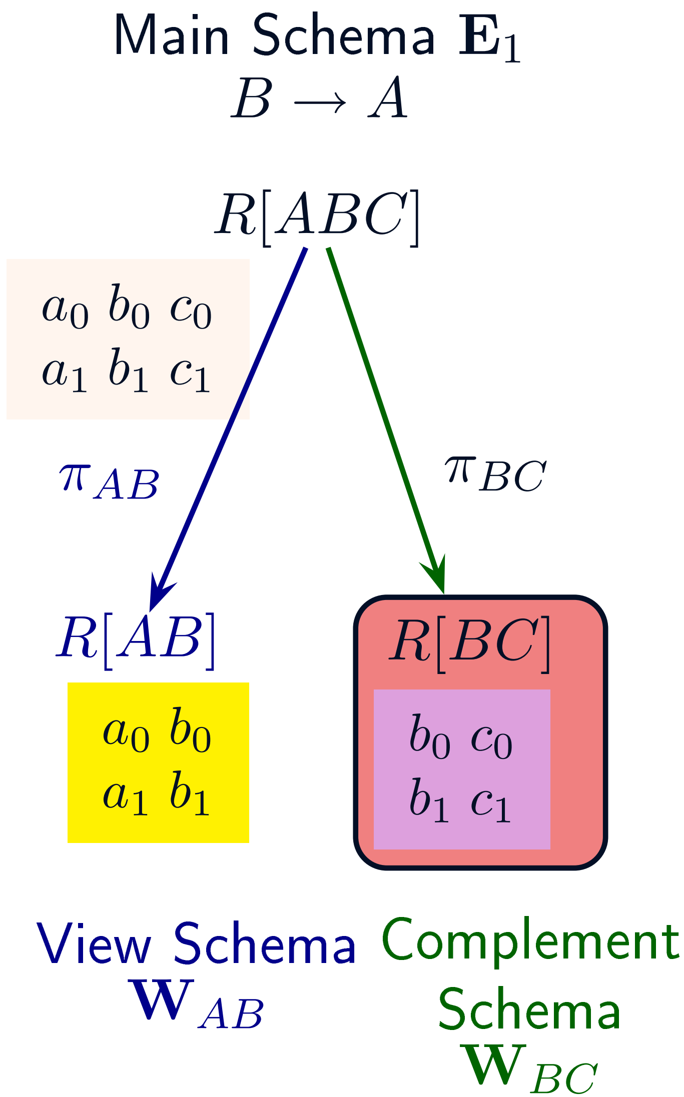
A Partial Solution Based upon Monotonicity

- Note that the symmetric difference mapping is not monotonic with respect to the natural order of database states.
- In earlier work [Hegner04 AMAI], it was shown that *order-realizable* update translations are unique when both view mappings are monotonic and the decomposition mapping is an order isomorphism.
- An *order-realizable* update is one which is realizable as a sequence of **legal** insertions and deletions.
- In the example to the right, no update to $\mathbf{W}_{AB}$ with $\mathbf{W}_{BC}$ constant is order realizable.

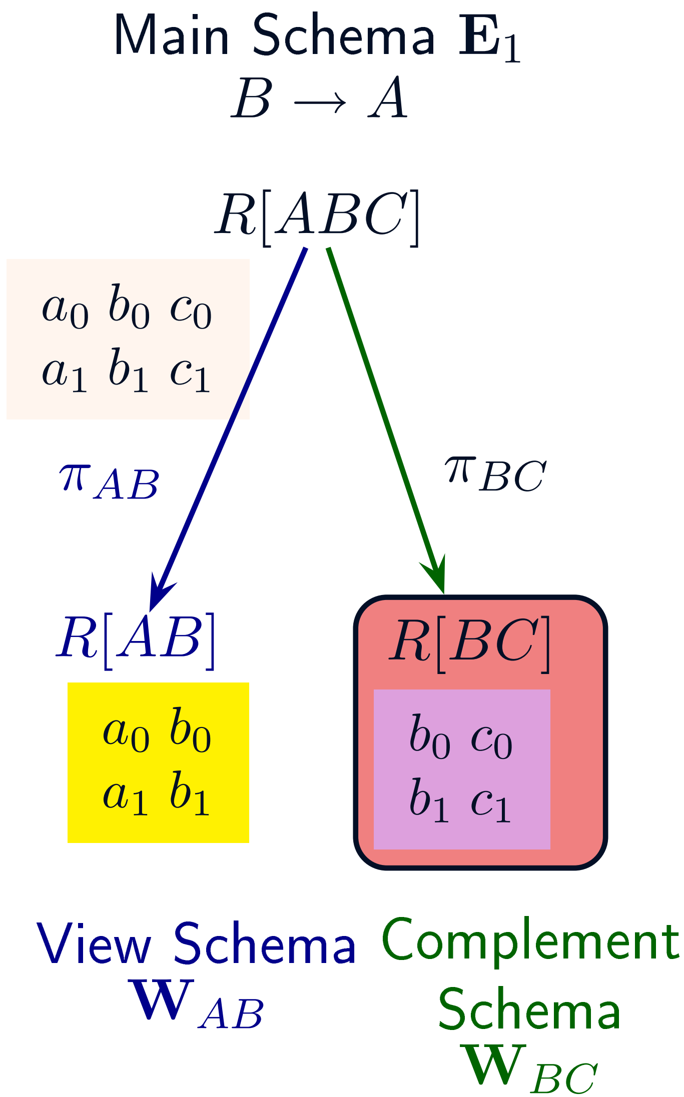


A Partial Solution Based upon Monotonicity

- Note that the symmetric difference mapping is not monotonic with respect to the natural order of database states.
- In earlier work [Hegner04 AMAI], it was shown that *order-realizable* update translations are unique when both view mappings are monotonic and the decomposition mapping is an order isomorphism.
- An *order-realizable* update is one which is realizable as a sequence of **legal** insertions and deletions.
- In the example to the right, no update to $\mathbf{W}_{AB}$ with $\mathbf{W}_{BC}$ constant is order realizable.
- The only allowable updates to $\mathbf{W}_{AB}$ keep the *meet* $R[B]$ constant.

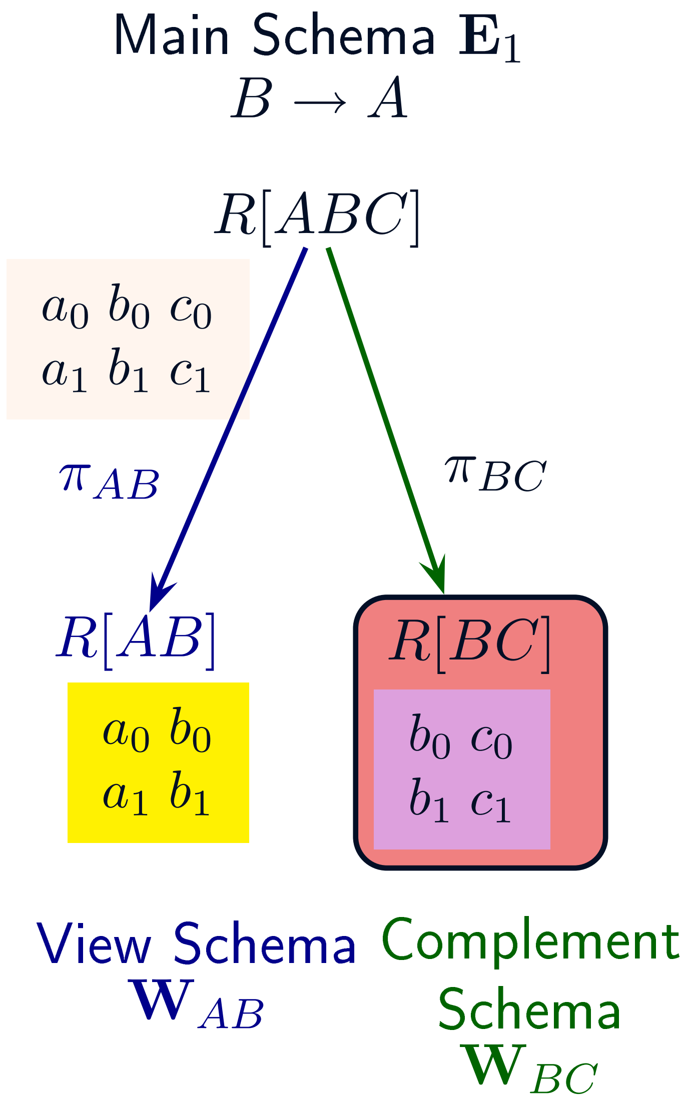


- Note that the symmetric difference mapping is not monotonic with respect to the natural order of database states.
- In earlier work [Hegner04 AMAI], it was shown that *order-realizable* update translations are unique when both view mappings are monotonic and the decomposition mapping is an order isomorphism.
- An *order-realizable* update is one which is realizable as a sequence of **legal** insertions and deletions.
- In the example to the right, no update to $\mathbf{W}_{AB}$ with $\mathbf{W}_{BC}$ constant is order realizable.
- The only allowable updates to $\mathbf{W}_{AB}$ keep the *meet* $R[B]$ constant.
- Thus, the only possible updates are those which change the A value of a tuple, and these are not order realizable.



A Partial Solution Based upon Monotonicity

- Note that the symmetric difference mapping is not monotonic with respect to the natural order of database states.
- In earlier work [Hegner04 AMAI], it was shown that *order-realizable* update translations are unique when both view mappings are monotonic and the decomposition mapping is an order isomorphism.
- An *order-realizable* update is one which is realizable as a sequence of **legal** insertions and deletions.
- In the example to the right, no update to $\mathbf{W}_{AB}$ with $\mathbf{W}_{BC}$ constant is order realizable.
- The only allowable updates to $\mathbf{W}_{AB}$ keep the *meet* $R[B]$ constant.
- Thus, the only possible updates are those which change the A value of a tuple, and these are not order realizable.



The Substitution Mapping of a Database Morphism

- Let WFS($\mathbf{D}$) denote the set of all well-formed sentences in the language of the relational database schema $\mathbf{D}$.

The Substitution Mapping of a Database Morphism

- Let $\mathsf{WFS}(\mathbf{D})$ denote the set of all well-formed sentences in the language of the relational database schema $\mathbf{D}$.
- A database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ between relational schemata is represented as a logical interpretation $\mathsf{Subst}\langle f, - \rangle : \mathsf{WFS}(\mathbf{D}_2) \to \mathsf{WFS}(\mathbf{D}_1)$.

The Substitution Mapping of a Database Morphism

- Let $\mathrm{WFS}(\mathbf{D})$ denote the set of all well-formed sentences in the language of the relational database schema $\mathbf{D}$.
- A database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ between relational schemata is represented as a logical interpretation $\mathrm{Subst}\langle f, - \rangle : \mathrm{WFS}(\mathbf{D}_2) \to \mathrm{WFS}(\mathbf{D}_1)$.

Example: The projection $\pi_{AB} : R[ABC] \to S[AB]$ is represented by the formula $(\exists x_3)(R(x_1, x_2, x_3))$ in the relational calculus.

The Substitution Mapping of a Database Morphism

- Let WFS($\mathbf{D}$) denote the set of all well-formed sentences in the language of the relational database schema $\mathbf{D}$.
- A database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ between relational schemata is represented as a logical interpretation Subst$\langle f, - \rangle$: WFS($\mathbf{D}_2$) $\to$ WFS($\mathbf{D}_1$).

Example: The projection $\pi_{AB} : R[ABC] \to S[AB]$ is represented by the formula $(\exists x_3)(R(x_1, x_2, x_3))$ in the relational calculus.

Subst$\langle f, - \rangle$: $(\exists y)(S(a_1, y) \wedge S(a_2, y)) \mapsto (\exists x_3)(\exists y)(R(a_1, y, x_3) \wedge R(a_2, y, x_3))$.

The Substitution Mapping of a Database Morphism

- Let WFS($\mathbf{D}$) denote the set of all well-formed sentences in the language of the relational database schema $\mathbf{D}$.
- A database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ between relational schemata is represented as a logical interpretation Subst$\langle f, - \rangle$: WFS($\mathbf{D}_2$) $\to$ WFS($\mathbf{D}_1$).

Example: The projection $\pi_{AB} : R[ABC] \to S[AB]$ is represented by the formula $(\exists x_3)(R(x_1, x_2, x_3))$ in the relational calculus.

Subst$\langle f, - \rangle$: $(\exists y)(S(a_1, y) \wedge S(a_2, y)) \mapsto (\exists x_3)(\exists y)(R(a_1, y, x_3) \wedge R(a_2, y, x_3))$.

- Let WFS($\mathbf{D}, \exists\wedge+$) denote the subset of WFS($\mathbf{D}$) consisting of all positive conjunctive sentences (no disjunction, no negation).

The Substitution Mapping of a Database Morphism

- Let WFS($\mathbf{D}$) denote the set of all well-formed sentences in the language of the relational database schema $\mathbf{D}$.
- A database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ between relational schemata is represented as a logical interpretation Subst$\langle f, - \rangle$: WFS($\mathbf{D}_2$) $\to$ WFS($\mathbf{D}_1$).

Example: The projection $\pi_{AB} : R[ABC] \to S[AB]$ is represented by the formula $(\exists x_3)(R(x_1, x_2, x_3))$ in the relational calculus.

Subst$\langle f, - \rangle$: $(\exists y)(S(a_1, y) \wedge S(a_2, y)) \mapsto (\exists x_3)(\exists y)(R(a_1, y, x_3) \wedge R(a_2, y, x_3))$.

- Let WFS($\mathbf{D}, \exists\wedge+$) denote the subset of WFS($\mathbf{D}$) consisting of all positive conjunctive sentences (no disjunction, no negation).
- The morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ is of *class* $\exists\wedge+$ if for every $\varphi \in$ WFS($\mathbf{D}_2$,), Subst$\langle f, - \rangle(\varphi)$ is equivalent to a sentence in WFS($\mathbf{D}_1, \exists\wedge+$).

The Substitution Mapping of a Database Morphism

- Let WFS($\mathbf{D}$) denote the set of all well-formed sentences in the language of the relational database schema $\mathbf{D}$.
- A database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ between relational schemata is represented as a logical interpretation Subst$\langle f, - \rangle$: WFS($\mathbf{D}_2$) $\to$ WFS($\mathbf{D}_1$).

Example: The projection $\pi_{AB} : R[ABC] \to S[AB]$ is represented by the formula $(\exists x_3)(R(x_1, x_2, x_3))$ in the relational calculus.

Subst$\langle f, - \rangle$: $(\exists y)(S(a_1, y) \wedge S(a_2, y)) \mapsto (\exists x_3)(\exists y)(R(a_1, y, x_3) \wedge R(a_2, y, x_3))$.

- Let WFS($\mathbf{D}, \exists\wedge+$) denote the subset of WFS($\mathbf{D}$) consisting of all positive conjunctive sentences (no disjunction, no negation).
- The morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ is of *class* $\exists\wedge+$ if for every $\varphi \in$ WFS($\mathbf{D}_2$,), Subst$\langle f, - \rangle(\varphi)$ is equivalent to a sentence in WFS($\mathbf{D}_1, \exists\wedge+$).
- Every SPJ-mapping (select-project-join) is of class $\exists\wedge+$.

The Substitution Mapping of a Database Morphism

- Let WFS($\mathbf{D}$) denote the set of all well-formed sentences in the language of the relational database schema $\mathbf{D}$.
- A database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ between relational schemata is represented as a logical interpretation Subst$\langle f, - \rangle$: WFS($\mathbf{D}_2$) $\to$ WFS($\mathbf{D}_1$).

Example: The projection $\pi_{AB} : R[ABC] \to S[AB]$ is represented by the formula $(\exists x_3)(R(x_1, x_2, x_3))$ in the relational calculus.

Subst$\langle f, - \rangle$: $(\exists y)(S(a_1, y) \wedge S(a_2, y)) \mapsto (\exists x_3)(\exists y)(R(a_1, y, x_3) \wedge R(a_2, y, x_3))$.

- Let WFS($\mathbf{D}, \exists\wedge+$) denote the subset of WFS($\mathbf{D}$) consisting of all positive conjunctive sentences (no disjunction, no negation).
- The morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ is of *class* $\exists\wedge+$ if for every $\varphi \in$ WFS($\mathbf{D}_2$,), Subst$\langle f, - \rangle(\varphi)$ is equivalent to a sentence in WFS($\mathbf{D}_1, \exists\wedge+$).
- Every SPJ-mapping (select-project-join) is of class $\exists\wedge+$.
- These are also called conjunctive queries.

Semantic Bijectivity

- $\equiv_{\mathbf{D}}$ denotes semantic equivalence of sentences in $\mathrm{WFS}(\mathbf{D})$ on the databases which satisfy all integrity constraints.

Semantic Bijectivity

- $\equiv_{\mathbf{D}}$ denotes semantic equivalence of sentences in $\mathsf{WFS}(\mathbf{D})$ on the databases which satisfy all integrity constraints.

Definition: The database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ of class $\exists\wedge+$ is *semantically bijective for* $\exists\wedge+$ if $\mathsf{Subst}\langle f, -\rangle$ induces a bijection

$$\mathsf{WFS}(\mathbf{D}_2, \exists\wedge+)/\equiv_{\mathbf{D}_2} \longleftrightarrow \mathsf{WFS}(\mathbf{D}_1, \exists\wedge+)/\equiv_{\mathbf{D}_1}.$$

Semantic Bijectivity

- $\equiv_{\mathbf{D}}$ denotes semantic equivalence of sentences in WFS$(\mathbf{D})$ on the databases which satisfy all integrity constraints.

Definition: The database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ of class $\exists\wedge+$ is *semantically bijective for* $\exists\wedge+$ if Subst$\langle f, -\rangle$ induces a bijection

$$\mathsf{WFS}(\mathbf{D}_2, \exists\wedge+)/\equiv_{\mathbf{D}_2} \longleftrightarrow \mathsf{WFS}(\mathbf{D}_1, \exists\wedge+)/\equiv_{\mathbf{D}_1}.$$

Fact (Semantic bijectivity is stronger than ordinary bijectivity): Every semantic bijection for $\exists\wedge+$ is also a bijection $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$ on the *legal database states* (those which satisfy the integrity constraints.)

- $\equiv_{\mathbf{D}}$ denotes semantic equivalence of sentences in $\mathsf{WFS}(\mathbf{D})$ on the databases which satisfy all integrity constraints.

Definition: The database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ of class $\exists\wedge+$ is *semantically bijective for* $\exists\wedge+$ if $\mathsf{Subst}\langle f, -\rangle$ induces a bijection

$$\mathsf{WFS}(\mathbf{D}_2, \exists\wedge+)/\equiv_{\mathbf{D}_2} \longleftrightarrow \mathsf{WFS}(\mathbf{D}_1, \exists\wedge+)/\equiv_{\mathbf{D}_1}.$$

Fact (Semantic bijectivity is stronger than ordinary bijectivity): Every semantic bijection for $\exists\wedge+$ is also a bijection $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$ on the *legal database states* (those which satisfy the integrity constraints.)

Proposition: Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be of class $\exists\wedge+$ and a bijection on database states. Then it is a semantic bijection iff its inverse is also of class $\exists\wedge+$. $\square$

- $\equiv_{\mathbf{D}}$ denotes semantic equivalence of sentences in $\mathsf{WFS}(\mathbf{D})$ on the databases which satisfy all integrity constraints.

Definition: The database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ of class $\exists\wedge+$ is *semantically bijective for* $\exists\wedge+$ if $\mathsf{Subst}\langle f, -\rangle$ induces a bijection

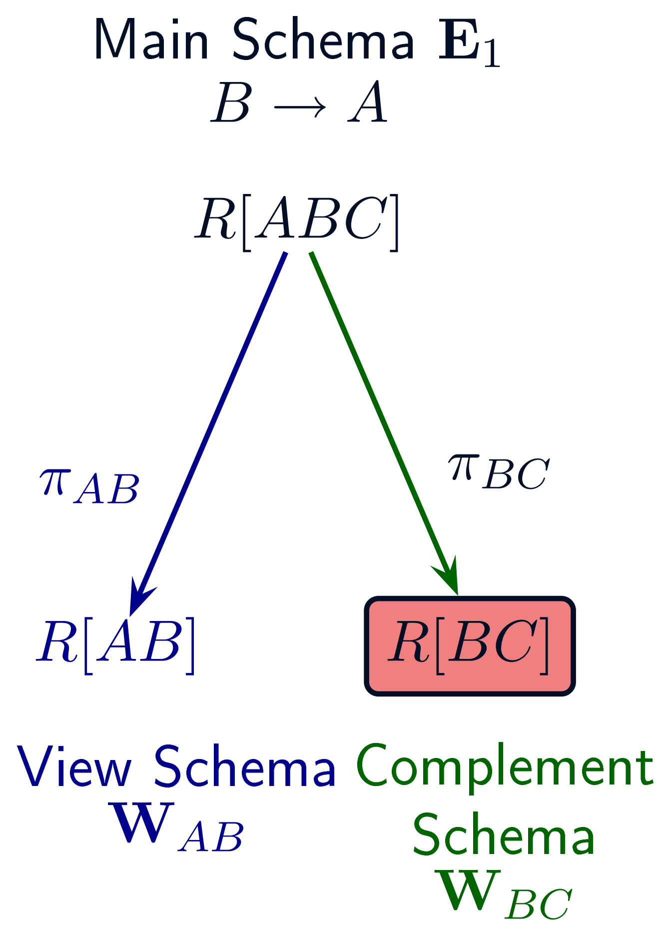
$$\mathsf{WFS}(\mathbf{D}_2, \exists\wedge+)/\equiv_{\mathbf{D}_2} \longleftrightarrow \mathsf{WFS}(\mathbf{D}_1, \exists\wedge+)/\equiv_{\mathbf{D}_1}.$$

Fact (Semantic bijectivity is stronger than ordinary bijectivity): Every semantic bijection for $\exists\wedge+$ is also a bijection $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$ on the *legal database states* (those which satisfy the integrity constraints.)

Proposition: Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be of class $\exists\wedge+$ and a bijection on database states. Then it is a semantic bijection iff its inverse is also of class $\exists\wedge+$. $\square$

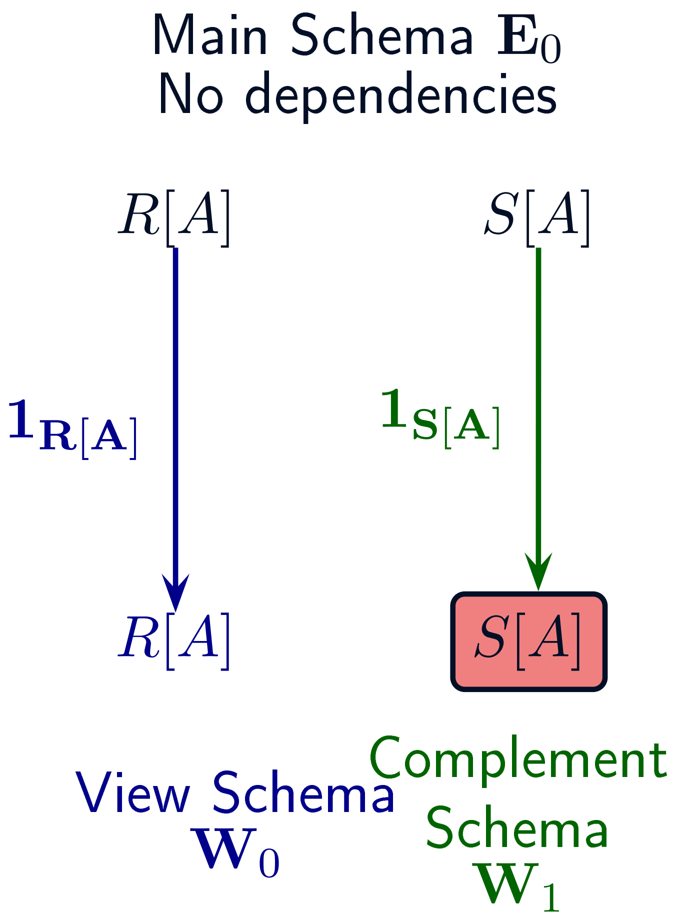
Theorem (Uniqueness of complements): A view whose morphism is of class $\exists\wedge+$ can have only one complement of class $\exists\wedge+$ for which the decomposition mapping is semantically bijective for $\exists\wedge+$. $\square$

- In the classical example to the right, all mappings are of class $\exists\wedge+$.
- Therefore, Π_{BC} is the only complement of Π_{AB} for which the reconstruction mapping is also of class $\exists\wedge+$.



Examples illustrating Uniqueness of Complements

- In the classical example to the right, all mappings are of class $\exists\wedge+$.
- Therefore, Π_{BC} is the only complement of Π_{AB} for which the reconstruction mapping is also of class $\exists\wedge+$.
- Likewise for the second example, now to the right.

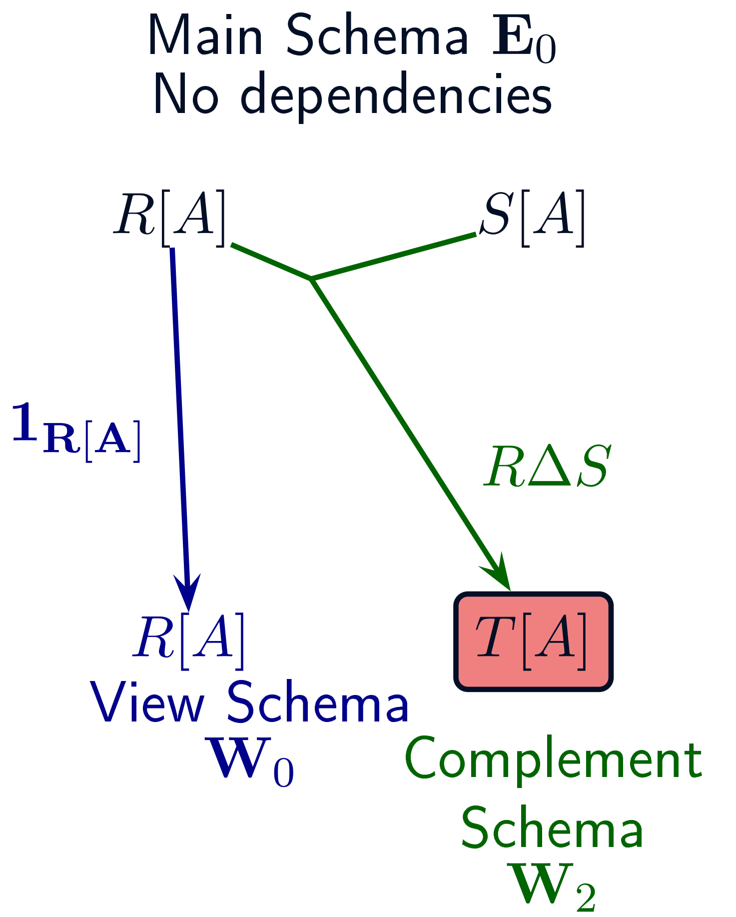


Examples illustrating Uniqueness of Complements

- In the classical example to the right, all mappings are of class $\exists\wedge+$.
- Therefore, Π_{BC} is the only complement of Π_{AB} for which the reconstruction mapping is also of class $\exists\wedge+$.
- Likewise for the second example, now to the right.
- In the third example, the view mapping for $\mathbf{W}_2$ is not of class $\exists\wedge+$:

$$T(x) \Leftrightarrow (R[x] \wedge \neg S(x)) \vee (S(x) \wedge \neg R(x))$$

- In particular, this complement does not define a reconstruction mapping of class $\exists\wedge+$.

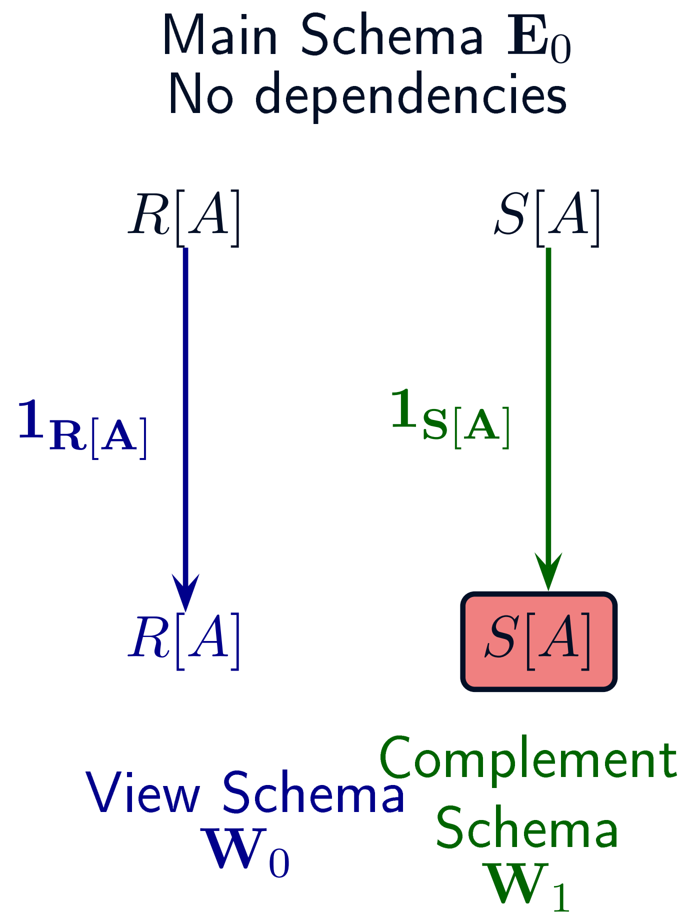


Examples illustrating Uniqueness of Complements

- In the classical example to the right, all mappings are of class $\exists\wedge+$.
- Therefore, Π_{BC} is the only complement of Π_{AB} for which the reconstruction mapping is also of class $\exists\wedge+$.
- Likewise for the second example, now to the right.
- In the third example, the view mapping for $\mathbf{W}_2$ is not of class $\exists\wedge+$:

$$T(x) \Leftrightarrow (R[x] \wedge \neg S(x)) \vee (S(x) \wedge \neg R(x))$$

- In particular, this complement does not define a reconstruction mapping of class $\exists\wedge+$.
- The complement defined by $\mathbf{W}_1$ is the only one for $\mathbf{W}_0$ which defines a reconstruction of class $\exists\wedge+$.



Guaranteeing Semantic Bijectivity

Question: Are there conditions which may be imposed on a schema $\mathbf{D}_1$ which guarantee that every bijective morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ of class $\exists\wedge+$ is semantically bijective?

Guaranteeing Semantic Bijectivity

Question: Are there conditions which may be imposed on a schema $\mathbf{D}_1$ which guarantee that every bijective morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ of class $\exists\wedge+$ is semantically bijective?

Theorem: If $\mathbf{D}_1$ admits *universal models*, then every such bijective morphism of class $\exists\wedge+$ is semantically bijective. $\square$

Guaranteeing Semantic Bijectivity

Question: Are there conditions which may be imposed on a schema $\mathbf{D}_1$ which guarantee that every bijective morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ of class $\exists\wedge+$ is semantically bijective?

Theorem: If $\mathbf{D}_1$ admits *universal models*, then every such bijective morphism of class $\exists\wedge+$ is semantically bijective. $\square$

Theorem (Chase generates universal models): Suppose that $\mathbf{D}_1$ is constrained by classical database dependencies: EGDs (equality-generating dependencies) and TGDs (tuple-generating dependencies, possibly embedded). If the classical *chase inference procedure* terminates when applied to every M which is a subset of a legal database, then $\mathbf{D}_1$ admits universal models. $\square$

Guaranteeing Semantic Bijectivity

Question: Are there conditions which may be imposed on a schema $\mathbf{D}_1$ which guarantee that every bijective morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ of class $\exists\wedge+$ is semantically bijective?

Theorem: If $\mathbf{D}_1$ admits *universal models*, then every such bijective morphism of class $\exists\wedge+$ is semantically bijective. □

Theorem (Chase generates universal models): Suppose that $\mathbf{D}_1$ is constrained by classical database dependencies: EGDs (equality-generating dependencies) and TGDs (tuple-generating dependencies, possibly embedded). If the classical *chase inference procedure* terminates when applied to every M which is a subset of a legal database, then $\mathbf{D}_1$ admits universal models. □

Fact: The chase procedure always terminates when restricted to EGDs and the *weakly acyclic TGDs* [Fagin et al TCS 2005]. □

Guaranteeing Semantic Bijectivity

Question: Are there conditions which may be imposed on a schema $\mathbf{D}_1$ which guarantee that every bijective morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ of class $\exists\wedge+$ is semantically bijective?

Theorem: If $\mathbf{D}_1$ admits *universal models*, then every such bijective morphism of class $\exists\wedge+$ is semantically bijective. $\square$

Theorem (Chase generates universal models): Suppose that $\mathbf{D}_1$ is constrained by classical database dependencies: EGDs (equality-generating dependencies) and TGDs (tuple-generating dependencies, possibly embedded). If the classical *chase inference procedure* terminates when applied to every M which is a subset of a legal database, then $\mathbf{D}_1$ admits universal models. $\square$

Fact: The chase procedure always terminates when restricted to EGDs and the *weakly acyclic TGDs* [Fagin et al TCS 2005]. $\square$

Bottom Line: If the main schema is constrained by EGDs and weakly acyclic TGDs, and all view mappings are of class $\exists\wedge+$, then view complements are unique. $\square$

- For M a database regarded as a set of ground atoms, the *information content* of M relative to $\exists\wedge+$ is:

$$\mathsf{Info}\langle M\rangle = \{\varphi \in \mathsf{WFS}(\mathbf{D}, \exists\wedge+) \mid M \models \varphi\}$$

Constant-Complement Update and Information Change

- For M a database regarded as a set of ground atoms, the *information content* of M relative to $\exists\wedge+$ is:

$$\mathsf{Info}\langle M\rangle = \{\varphi \in \mathsf{WFS}(\mathbf{D}, \exists\wedge+) \mid M \models \varphi\}$$

- For an update (M_1, M_2), the *information change* is:

$$\Delta\langle M_1, M_2\rangle = (\mathsf{Info}\langle M_2\rangle \setminus \mathsf{Info}\langle M_1\rangle) \cup (\mathsf{Info}\langle M_1\rangle \setminus \mathsf{Info}\langle M_2\rangle)$$

Constant-Complement Update and Information Change

- For M a database regarded as a set of ground atoms, the *information content* of M relative to $\exists\wedge+$ is:

$$\mathsf{Info}\langle M\rangle = \{\varphi \in \mathsf{WFS}(\mathbf{D}, \exists\wedge+) \mid M \models \varphi\}$$

- For an update (M_1, M_2), the *information change* is:

$$\Delta\langle M_1, M_2\rangle = (\mathsf{Info}\langle M_2\rangle \setminus \mathsf{Info}\langle M_1\rangle) \cup (\mathsf{Info}\langle M_1\rangle \setminus \mathsf{Info}\langle M_2\rangle)$$

Theorem (Constant-complement view update implies least information change):

- Γ_1 a view of class $\exists\wedge+$.
- (N_1, N_2) an update on view Γ_1.
- Γ_2 the unique complement of Γ_1 which is also of class $\exists\wedge+$.
- The decomposition morphism is semantically bijective.

Then the update (M_1, M_2) on the main schema which is defined by constant-complement Γ_2 has the least information change over all possible reflections. $\square$

- It has been shown that under suitable conditions which include common database dependencies and views defined by SPJ-mappings, complements of relational schemata are unique.

Conclusions

- It has been shown that under suitable conditions which include common database dependencies and views defined by SPJ-mappings, complements of relational schemata are unique.
- This in turn implies that there is a unique, natural realization for reflecting a view update to the main schema when using the the constant-complement strategy.

Conclusions

- It has been shown that under suitable conditions which include common database dependencies and views defined by SPJ-mappings, complements of relational schemata are unique.
- This in turn implies that there is a unique, natural realization for reflecting a view update to the main schema when using the the constant-complement strategy.
- It has also been shown that this natural realization is optimal in terms of information change to the main schema.

Extension to Other Logic-Based Data Models:

Further Directions

Extension to Other Logic-Based Data Models:

- The Nested Relational Model
- The Higher-Order Entity-Relationship Model

Further Directions

Extension to Other Logic-Based Data Models:

- The Nested Relational Model
- The Higher-Order Entity-Relationship Model

Question: To what extent is the $\exists\wedge+$ context applicable to such models?

Extension to Other Logic-Based Data Models:

- The Nested Relational Model
- The Higher-Order Entity-Relationship Model

Question: To what extent is the $\exists\wedge+$ context applicable to such models?

Rapprochement with the Order-Based Approach:

Extension to Other Logic-Based Data Models:

- The Nested Relational Model
- The Higher-Order Entity-Relationship Model

Question: To what extent is the $\exists\wedge+$ context applicable to such models?

Rapprochement with the Order-Based Approach:

- The work of [Hegner 04 AMAI] is not based upon logical models, but rather upon poset-based models.

Question: To what extent can these two approaches be merged?

Extension to Other Logic-Based Data Models:

- The Nested Relational Model
- The Higher-Order Entity-Relationship Model

Question: To what extent is the $\exists\wedge+$ context applicable to such models?

Rapprochement with the Order-Based Approach:

- The work of [Hegner 04 AMAI] is not based upon logical models, but rather upon poset-based models.

Question: To what extent can these two approaches be merged?

Relationship to the Inversion of Schema Mappings:

Further Directions

Extension to Other Logic-Based Data Models:

- The Nested Relational Model
- The Higher-Order Entity-Relationship Model

Question: To what extent is the $\exists\wedge+$ context applicable to such models?

Rapprochement with the Order-Based Approach:

- The work of [Hegner 04 AMAI] is not based upon logical models, but rather upon poset-based models.

Question: To what extent can these two approaches be merged?

Relationship to the Inversion of Schema Mappings:

- The work of Fagin and his colleagues on data translation makes use of ideas related to information content.

Question: To what extent are the techniques developed for this work applicable to problems in data translation?