The Relative Complexity of Updates for a Class of Database Views

Stephen J. Hegner

Department of Computing Science
Umeå University, Sweden

Hegner Consulting, LLC
New Hampshire, USA

An Example from Relational Folklore

Terminology: (The k-submodel property of a relational schema)

- r satisfies the constraints of the schema iff every $s \subseteq r$ containing at most k tuples satisfies them.
- Thus, even without special data structures, satisfaction may be test in worst-case time $O(n^k)$, with n the number of tuples in the relation.

Main schema:

$$(R[ABCD],\ \ \mathcal{F} = \{A \to D,\ B \to D,\ CD \to A\})$$

Fact: The main schema has the 2-submodel property.

View:

$$\Pi_{ABC} = (R[ABC], \pi_{ABC})$$

Fact: The view does not have the k-submodel property for any $k \geqslant 0$.

Illustration of the Example

Claim: With distinct symbols identifying distinct values:

- The structure below is not a legal state of Π_{ABC}.
- Any proper subset is a legal state of Π_{ABC}.

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_3
a_2	b_3	c_4
a_3	b_3	c_5
a_3	b_4	c_6
a_4	b_4	c_7
a_4	b_5	c_8
$\vdots$	$\vdots$	$\vdots$
a_{k-1}	b_{k-1}	c_{2k-1}
a_{k-1}	b_k	c_{2k}
$\vdots$	$\vdots$	$\vdots$
a_{m-1}	b_{m-1}	c_{2n-1}
a_{m-1}	b_m	c_{2n}
a_m	b_m	c_1

Illustration of the Example

$$\{A \to D,\ B \to D,\ CD \to A\}$$

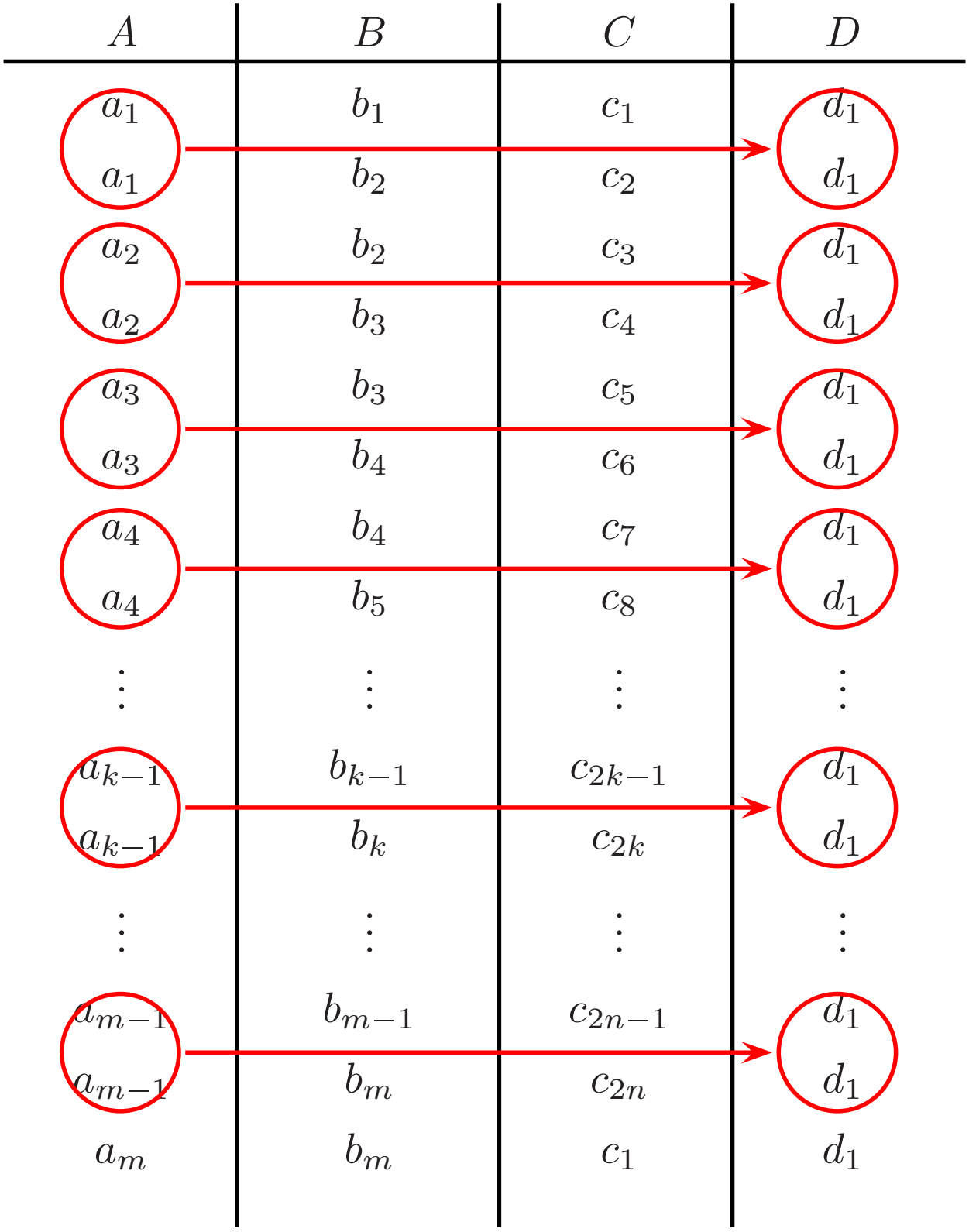


Illustration of the Example

$$\{A \to D,\ B \to D,\ CD \to A\}$$

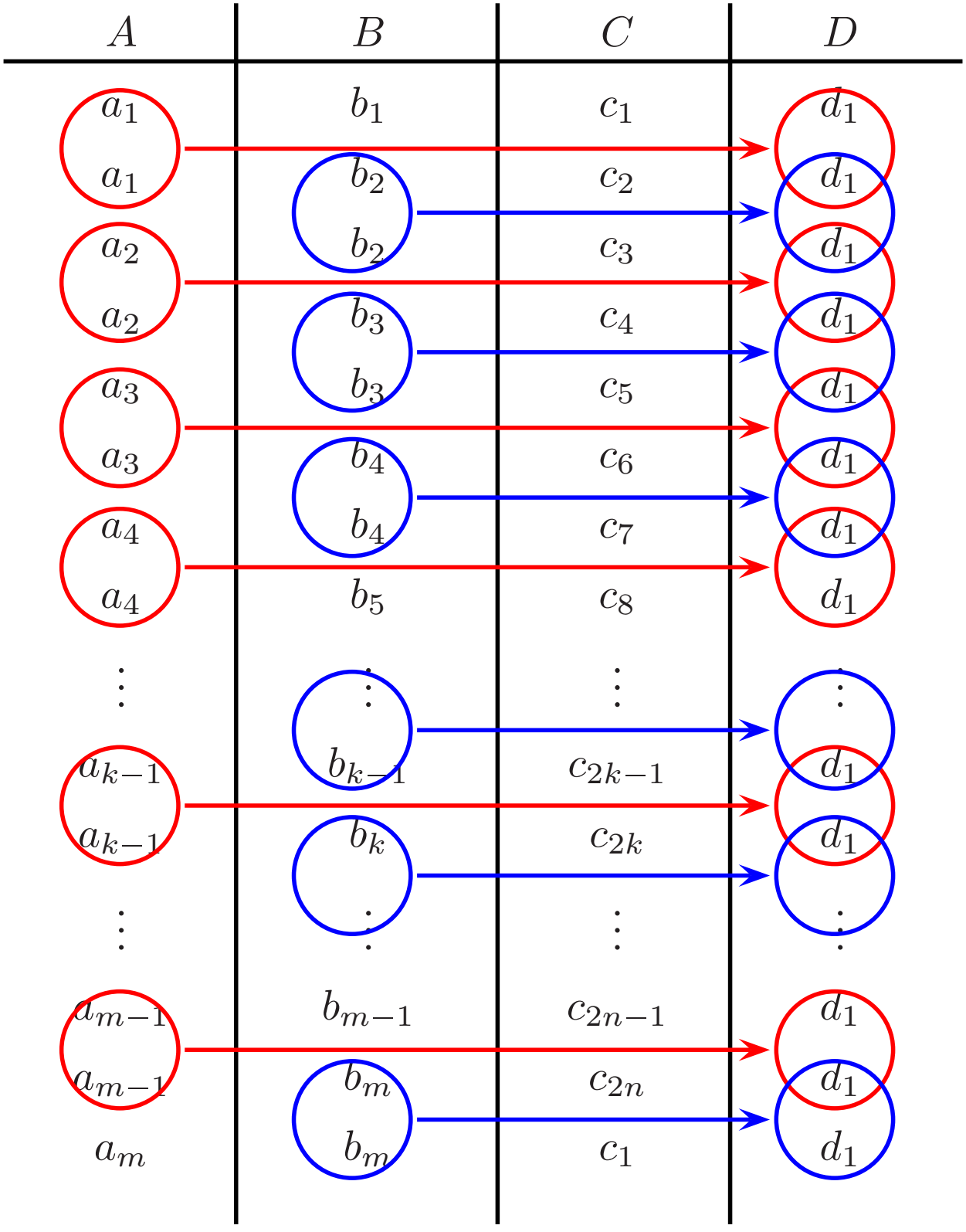


Illustration of the Example

$$\{A \to D,\ B \to D,\ CD \to A\}$$

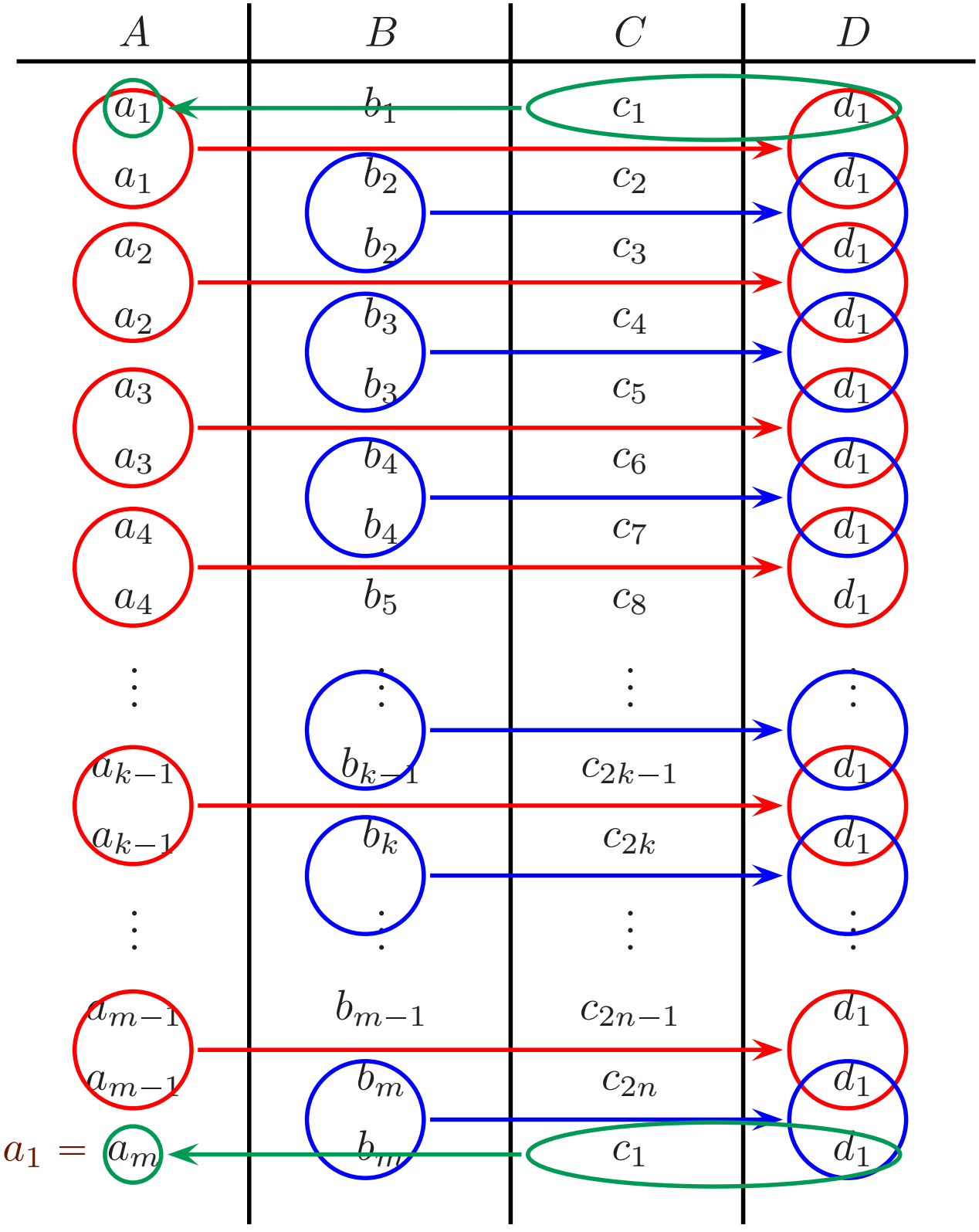
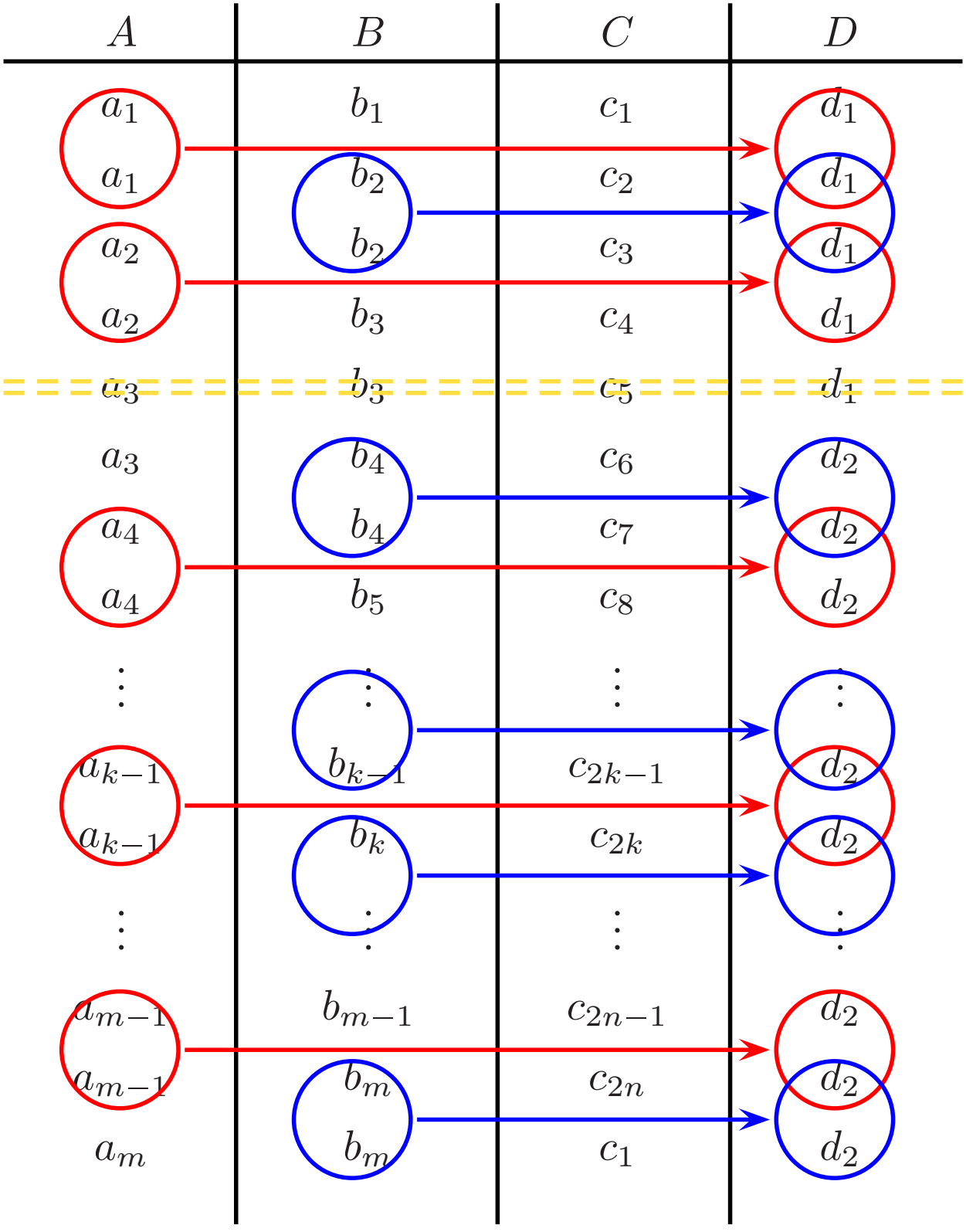


Illustration of the Example

$$\{A \to D,\ B \to D,\ CD \to A\}$$



Major Questions Addressed in this Work

Question: To what extent does this explosion of complexity impact the support for updates to views of database schemata?

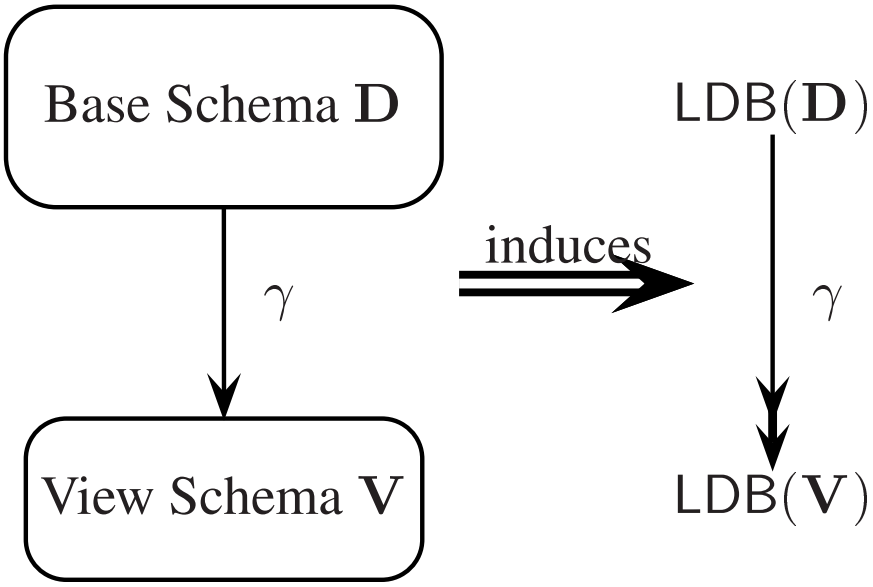
- Updates to views generally follow some prescribed *strategy*.
  - Arbitrary updates are not allowed.
  - Those updates which are allowed arise from specific changes to a database already known to satisfy the constraints

Question: Do these limitations result in a more manageable complexity for updates to views?

Answer: For the class of *closed update strategies*, the answer is **yes**.

- The complexity of testing a proposed update for correctness is no more complex than a corresponding test in the main schema.
- This is true even though the full axiomatization of the view schema may be much more complex than that of the main schema.

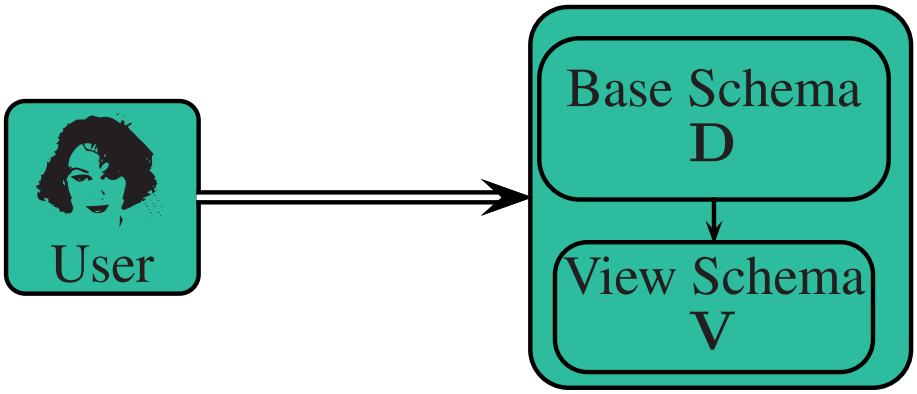
The Context of this Work



- To each schema $\mathbf{D}$ is associated a set $\mathsf{LDB}(\mathbf{D})$ of *legal states* or *legal databases*.
- The database mapping $\gamma : \mathsf{LDB}(\mathbf{D}) \rightarrow \mathsf{LDB}(\mathbf{V})$ which defines the view is surjective.
- This framework is general and not tied to any specific data model.
- Additional structure is added as necessary:
  - *Order structure* is employed to ensure uniqueness in update strategies.
  - *Powerset structure* is employed to address the complexity of updates.

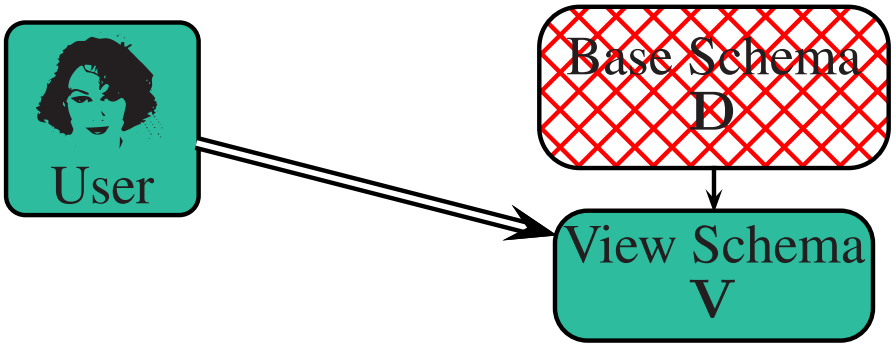
Open vs. Closed Views with Update

Open:



- The user has access to both the view and the base schema.
- The view is provided as a convenience.
- Uniqueness criteria suffice for update translation.

Closed: (context of this work)



- The user has access only to the view.
- The user has no direct knowledge of the base schema.
- The view must be self contained in terms of knowledge needed to effect updates.
- The view should preferably look "just" like a complete base schema.

Closed Update Families

- A *closed update family* for $\mathbf{V}$ is an equivalence relation U on $\mathsf{LDB}(\mathbf{V})$, the legal states of $\mathbf{V}$.

- $(M_1, M_2) \in U$ means that the update $M_1 \longrightarrow M_2$ is admissible on $\mathbf{V}$.

- Interpretation of equivalence relation properties:

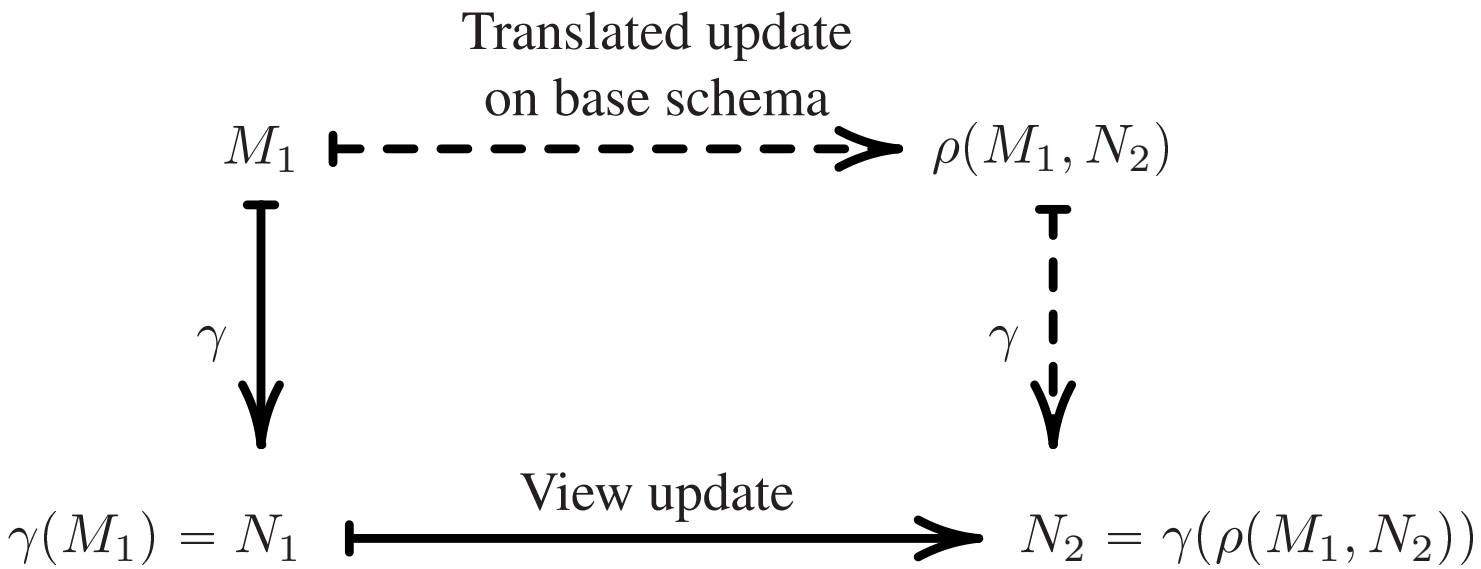
  - *Reflexivity* implies that the identity update is always allowed.

  - *Symmetry* implies that every update is reversible.

  - *Transitivity* implies that updates may be composed.

Fundamental modelling assumption: The set of admissible updates on a view forms a closed update family.

Translators for View Update



- An *update strategy* is a partial function:

  ρ : $\mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{V}) \rightarrow \mathsf{LDB}(\mathbf{D})$

  ρ : Base States $\times$ View States $\rightarrow$ Base States

  (Current Base State, New View State) $\mapsto$ New Base State.

- Not all view updates are allowed; thus ρ is a partial function, in general.

Closed Update Strategies

- Let T = a closed update family for the view $\mathbf{V}$.
  U = a closed update family for the base schema $\mathbf{D}$.
- In a closed setting, the following conditions are also imposed to yield a *closed update strategy* for T with respect to U.

(upt:1) [Only view updates from T are embodied in ρ.]

(upt:2) [Only base schema updates from U are embodied in ρ.]

(upt:3) [Identity updates are reflected as identities.]

(upt:4) [Every view update is globally reversible.]

(upt:5) [View update reflection is transitive.]

- In this work, only closed update strategies are considered.

Complement-Based Update Strategies

- Context: $\mathbf{D}$: base schema
  $\Gamma_1 = (\mathbf{V}_1,\ \gamma_1 : \mathbf{D} \to \mathbf{V}_1)$: a view of $\mathbf{D}$
  $\Gamma_2 = (\mathbf{V}_2,\ \gamma_2 : \mathbf{D} \to \mathbf{V}_2)$: a view of $\mathbf{D}$

- The view Γ_2 is a *(subdirect) complement* of Γ_1 if the associated decomposition is lossless. Formally,
  $$\gamma_1 \times \gamma_2 : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V}_1) \times \mathsf{LDB}(\mathbf{V}_2)$$
  $$M \mapsto (\gamma_1(M), \gamma_2(M))$$
  must be injective.

Observation [Bancilhon & Spyratos 81]: Every subdirect complement Γ_2 of Γ_1 defines a (not necessarily closed) update strategy on Γ_1 as follows:

$\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle(M_1, N_2) =$

$$\begin{cases} (\gamma_1 \times \gamma_2)^{-1}(N_2, \gamma_2(M_1)) & \text{if } (N_2, \gamma_2(M_1)) \in (\gamma_1 \times \gamma_2)(\mathsf{LDB}(\mathbf{D})) \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\begin{array}{ccc} \mathbf{D} & \longrightarrow & \mathbf{D} \\ \downarrow & & \downarrow \\ \mathbf{V}_1 \times \mathbf{V}_2 & \twoheadrightarrow & \mathbf{V}_1 \times \mathbf{V}_2 \end{array} \qquad \begin{array}{ccc} M_1 & \longrightarrow & \mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle(M_1, N_2) \\ \downarrow & & \downarrow \\ \gamma_1(M_1) \times \gamma_2(M_1) & \longrightarrow & N_2 \times \gamma_2(M_1) \end{array}$$

- This is called the update strategy defined by *constant complement* Γ_2.

Theorem [Bancilhon & Spyratos 81]: Every closed update strategy is defined by a constant complement update strategy. □

Closed Update in the Relational Context

Example base schema and instance:

$(E[\mathsf{ABC}], \{\mathsf{B} \rightarrow \mathsf{C}\})$:

A	B	C
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_1	c_1

View to be updated:

View $\Pi_{\mathsf{AB}} = (E[\mathsf{AB}], \varnothing)$

A	B
a_0	b_0
a_1	b_1
a_2	b_1

Complementary view:

View $\Pi_{\mathsf{BC}} = (E[\mathsf{BC}], \{\mathsf{B} \rightarrow \mathsf{C}\})$

B	C
b_0	c_0
b_1	c_1

- The updates which are allowed to $E[\mathsf{AB}]$ under constant complement Π_{BC} are precisely those which hold the view Π_{B} fixed.
- This is the part of Π_{BC} which is visible within Π_{AB}.
- The allowed updates to Π_{AB} do not depend upon other aspects of the view Π_{BC}.
- The view $\Pi_{\mathsf{B}} = (E[\mathsf{B}], \varnothing)$ is called the *meet* of Π_{AB} and Π_{BC}.

Non-Closed Update in the Relational Context

Example base schema and instance:

$(E[\mathsf{ABC}], \{\mathsf{A} \rightarrow \mathsf{C}, \mathsf{B} \rightarrow \mathsf{C}\})$:

A	B	C
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_1	c_1

View to be updated:

View $\Pi_{\mathsf{AB}} = (E[\mathsf{AB}], \varnothing)$

A	B
a_0	b_0
a_1	b_1
a_2	b_1

Complementary view:

View $\Pi_{\mathsf{BC}} = (E[\mathsf{BC}], \{\mathsf{B} \rightarrow \mathsf{C}\})$

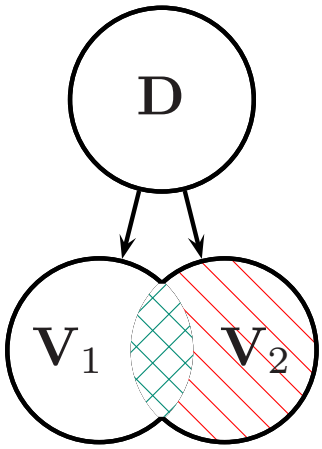
B	C
b_0	c_0
b_1	c_1

- This decomposition is not dependency preserving.
- The allowed updates to Π_{AB} now depend upon information contained in the view Π_{BC} other than that also contained in Π_{B}.
- The associated update strategy is not closed.

Complements and Closed Update Strategies

Fact: There exist subdirect complements which do not define closed update families. □

- To understand the problem, consider the following visualization, in which $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ is to be updated with constant complement $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$.



- When updating Γ_1, the part of the state of $\mathbf{V}_1$ which overlaps $\mathbf{V}_2$ shown in green must be held constant, while the rest may be modified at will.

- A closed update strategy on Γ_1 must depend only upon the part of $\mathbf{V}_2$ which overlaps $\mathbf{V}_1$ (shown in green), and not the rest of $\mathbf{V}_2$ (shown in red).

- The appropriate idea is a generalization of the notion of *dependency-preserving decomposition*.

Characterizing Closed Update Strategies via Complements

- The *congruence* $\mathsf{Congr}(\Gamma)$ of a view $\Gamma = (\mathbf{V}, \gamma)$ of $\mathbf{D}$ is the equivalence relation defined by
$$(M_1, M_2) \in \mathsf{Congr}(\Gamma) \text{ iff } \gamma(M_1) = \gamma(M_2).$$

- In a set-based context without additional structure, every view is defined, up to isomorphism, by its congruence.

- Thus, there is a natural correspondence:

Equivalence relations on states of the base schema	$\longleftrightarrow$	Isomorphism classes of views of the base schema

- Note that a closed update family U on $\mathbf{D}$ defines a view of $\mathbf{D}$, since it is an equivalence relation.

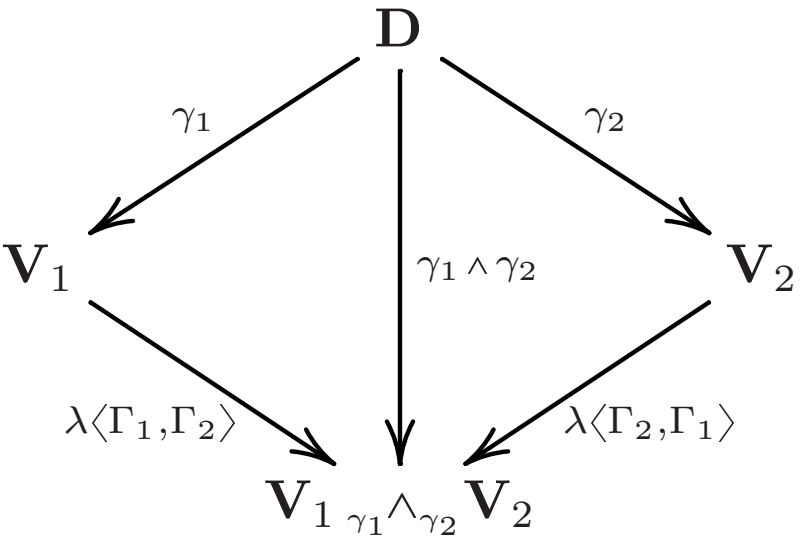
Commuting Congruences and Meet Complements

- A pair $\{\Gamma_1, \Gamma_2\}$ of views of $\mathbf{D}$ is called a *fully commuting pair* if $\mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2) = \mathsf{Congr}(\Gamma_2) \circ \mathsf{Congr}(\Gamma_1)$.

- In this case, $\mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2)$ is an equivalence relation.

- If $\{\Gamma_1, \Gamma_2\}$ is a fully commuting pair, the view whose congruence is $\Gamma_1 \circ \Gamma_2$ is called the *meet* of $\Gamma_1 \circ \Gamma_2$, and is denoted $\Gamma_1 \wedge \Gamma_2 = (\mathbf{V}_{1\ \gamma_1 \wedge_{\gamma_2}} \mathbf{V}_2, \gamma_1 \wedge \gamma_2)$.

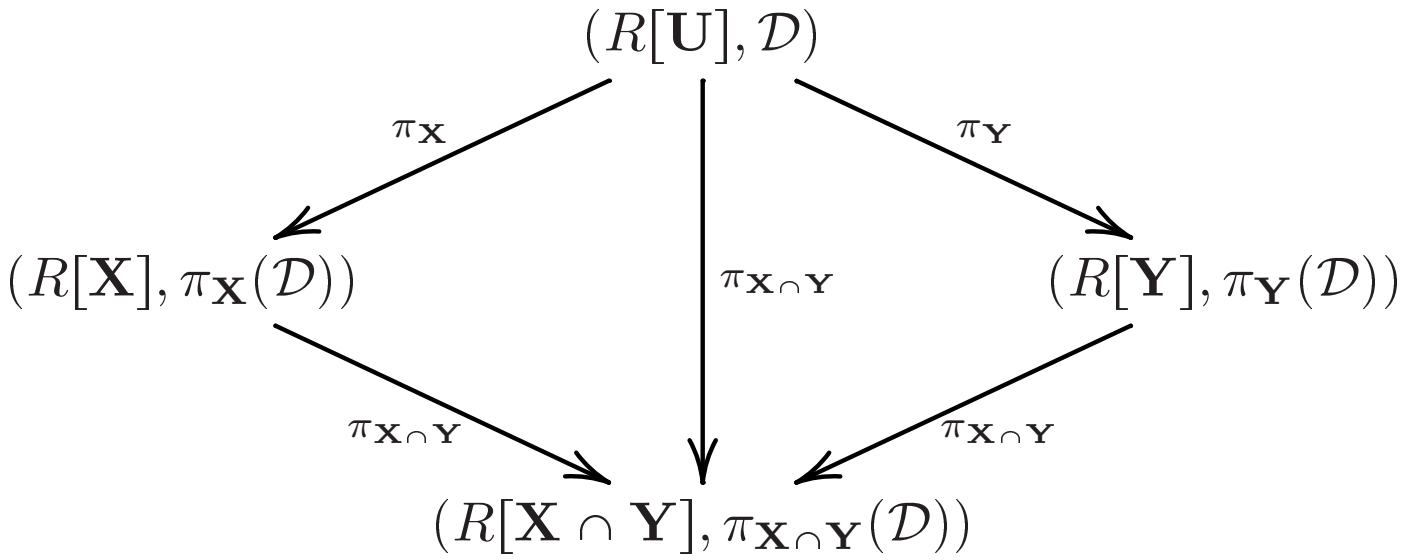
- In this case, $\Gamma_1 \wedge \Gamma_2$ may also be regarded as a view:
$\Lambda(\Gamma_1, \Gamma_2) = (\mathbf{V}_2, \lambda\langle\Gamma_1, \Gamma_2\rangle)$ of Γ_1.



- $\Gamma_1 \wedge \Gamma_2$ is effectively a glb of Γ_1 and Γ_2.

Meet in the Relational Context

- Under reasonable circumstances, the meet of two projections in the relational context is the projection on the intersection of the view attributes.
- Let:
  - $R[\mathbf{U}]$ = relation scheme on attribute set $\mathbf{U}$.
  - $\mathcal{D}$ = set of full dependencies.
  - $\mathbf{X}$, $\mathbf{Y}$ = subsets of $\mathbf{U}$.
- Suppose further that the decomposition is lossless and dependency preserving; *i.e.*,
  - $\bowtie [\mathbf{X}, \mathbf{Y}] \in \mathcal{D}^+$.
  - $(\pi_{\mathbf{X}}(\mathcal{D}) \cup \pi_{\mathbf{Y}}(\mathcal{D}))^+ = \mathcal{D}^+$.
- Then
  $(R[\mathbf{X}], \pi_{\mathbf{X}}(\mathcal{D})) \wedge (R[\mathbf{Y}], \pi_{\mathbf{Y}}(\mathcal{D})) = (R[\mathbf{X} \cap \mathbf{Y}], \pi_{\mathbf{X} \cap \mathbf{Y}}(\mathcal{D}))$.



Meet Complements and Closed Update Strategies

Theorem [Hegner ICDT90, FoIKS02]: The update strategy $\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle$ is closed iff $\{\Gamma_1, \Gamma_2\}$ forms a fully commuting pair. $\square$

- The view update family on Γ_1 defined by the meet complementary pair $\{\Gamma_1, \Gamma_2\}$ is just $\mathsf{Congr}(\Gamma_1 \wedge \Gamma_2)$.
  - In other words, the admissible updates on Γ_1 are under constant complement Γ_2 are precisely those which keep the meet $\Gamma_1 \wedge \Gamma_2$ constant.

Bottom line: For a given view Γ, there is a natural bijective correspondence:

Closed update strategies	$\longleftrightarrow$	Meet complements
ρ	$\longmapsto$	$\widetilde{\Gamma}^{\rho}$
$\mathsf{UpdStr}\langle\Gamma, \Gamma'\rangle$	$\longmapsfrom$	Γ'

Addressing the Uniqueness Question

- Generally speaking, distinct complements give rise to distinct update strategies under constant-complement translation.
- However, in the general sets-and-mappings framework, complements are never unique, except in degenerate cases.
- This is true even for complements with the same meet.

Observation: In many cases, there is an obvious "natural complement" which appears to be the only one which makes sense.

Question: How can this restriction to the "natural" complement be formalized?

Answer: [Hegner 02, 03] Use the natural order structure on database mappings and morphisms.

Classical examples:

- The natural order structure on relational schemata is defined via relation-by-relation inclusion.
- Select, project, and join mappings (SPJ) are all order preserving.

Key Features of the Order-Based Framework

- The legal databases of a schema $\mathbf{D}$ form a partially order set $(\mathsf{LDB}(\mathbf{D}), \leqslant_{\mathbf{D}})$.

- The mapping $\gamma : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V})$ of a view $\Gamma = (\mathbf{V}, \gamma)$ is an open poset morphism; *i.e.*,
$$\gamma(M_1) \leqslant_{\mathbf{V}} \gamma(M_2) \text{ iff } M_1 \leqslant_{\mathbf{D}} M_2.$$

- For $\{\Gamma_1 = (\mathbf{V}_1, \gamma_1), \Gamma_2 = (\mathbf{V}_2, \gamma_2)\}$ to be a pair of subdirect complements, the decomposition mapping
$$\gamma_1 \times \gamma_2 : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V})$$
must be a section (isomorphism into).

- A closed update family for $\mathbf{D}$ is an *order compatible* equivalence relation on $\mathsf{LDB}(\mathbf{D})$.

- A closed update strategy for the view $\Gamma = (\mathbf{V}, \gamma)$ of $\mathbf{D}$ is subject to three additional order-based conditions.

  (upt:6) [View update reflects order.]

  (upt:7) [Chain reflection.]

  (upt:8) [Order inheritance.]

Main uniqueness result: All closed update strategies agree on *order-based updates*; that is, updates which are realizable as sequences of insertions and deletions.

Extension to Recapture Update Complexity

Potential and legal database instances: It must be possible to recapture two distinct sets:

- $\mathsf{LDB}(\mathbf{D})$ – the databases which satisfy the extant constraints.
- $\mathsf{DB}(\mathbf{D})$ – all databases, regardless of whether or not they satisfy the constraints.

Database size: To measure complexity, there must be an underlying notion of the size of a database.

- The key idea is a *finite powerset construction*.
- $\mathsf{DB}(\mathbf{D})$ is the set of all finite subsets of a (not necessarily finite) set X, called the *foundation*.
- $\mathsf{LDB}(\mathbf{D})$ is a subset of $\mathsf{DB}(\mathbf{D})$, subject to the following constraints:
  - $\varnothing \in \mathsf{LDB}(\mathbf{D})$.
  - $x \in X$ implies $\{x\} \in \mathsf{LDB}(\mathbf{D})$.
- The ordering on both $\mathsf{LDB}(\mathbf{D})$ and $\mathsf{DB}(\mathbf{D})$ is induced by $\subseteq$.
- The size of a database is simply its cardinality, as a set.
- In the relational model:
  - $\mathsf{Foundation}(\mathbf{D})$ consists of all possible tuples.
  - Schemata constrained by full (universal) dependencies are recaptured.

Morphisms and Views of CFA-Schemata

Terminology: Entities in this framework are identified by the prefix *CFA* (***C**oncrete **F**initely **A**tomistic*).

Key principle: All concepts related to morphisms and equivalences are defined atomically; that is, based upon the atoms (singletons) of the schemata.

- A *CFA-morphism* $f : \mathbf{D}_1 \to \mathbf{D}_2$ of CFA-schemata has the property that it is defined by its action on atoms:

$$f(M) = \bigcup \{f(\{x\}) \mid x \in M\}$$

- In the relational context, projections and restrictions (but not joins) are CFA-morphisms.

- A CFA-view $\Gamma = (\mathbf{V}, \gamma)$ of the CFA-schema $\mathbf{D}$ is defined in the obvious way, with γ a *CFA-surjection*.

- CFA-views are characterized by a special class of equivalence relations, the *atomically generated equivalences*.

- A *closed update strategy* for the view $\Gamma = (\mathbf{V}, \gamma)$ of $\mathbf{D}$ is subject to an additional condition.

  (upt:9) [The update strategy is defined atomically.]

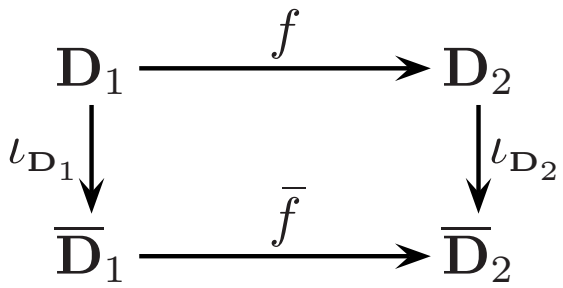
- Within this context, the natural correspondence between meet-complements and update strategies is maintained.

Lifting to the Unconstrained Context

- Given a CFA-schema $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$, let $(\overline{\mathbf{D}} = \mathsf{DB}(\mathbf{D}), \subseteq)$ denote the corresponding unconstrained schema consisting of all finite subsets of $\mathsf{Foundation}(\mathbf{D})$.

- The morphism $\iota_{\mathbf{D}_1} : \mathbf{D} \to \overline{\mathbf{D}}$ represents the natural embedding.

- Every morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ of CFA-schemata has a natural extension to unconstrained databases:

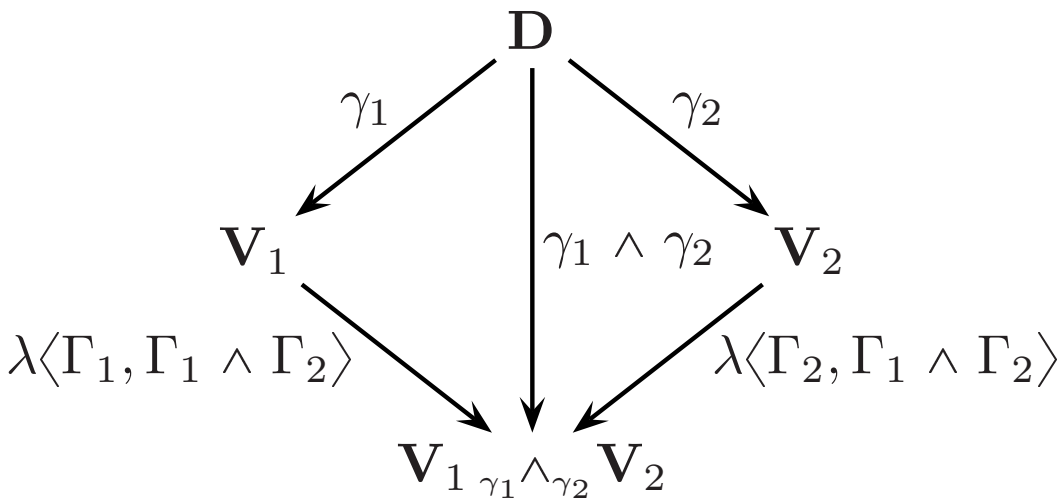


- **Observation**: Every CFA-view $\Gamma = (\mathbf{V}, \gamma)$ on $\mathbf{D}$ has a unique extension $\overline{\Gamma} = (\overline{\mathbf{V}}, \overline{\gamma})$ to $\overline{\mathbf{D}}$. $\square$

Lifting Equivalence

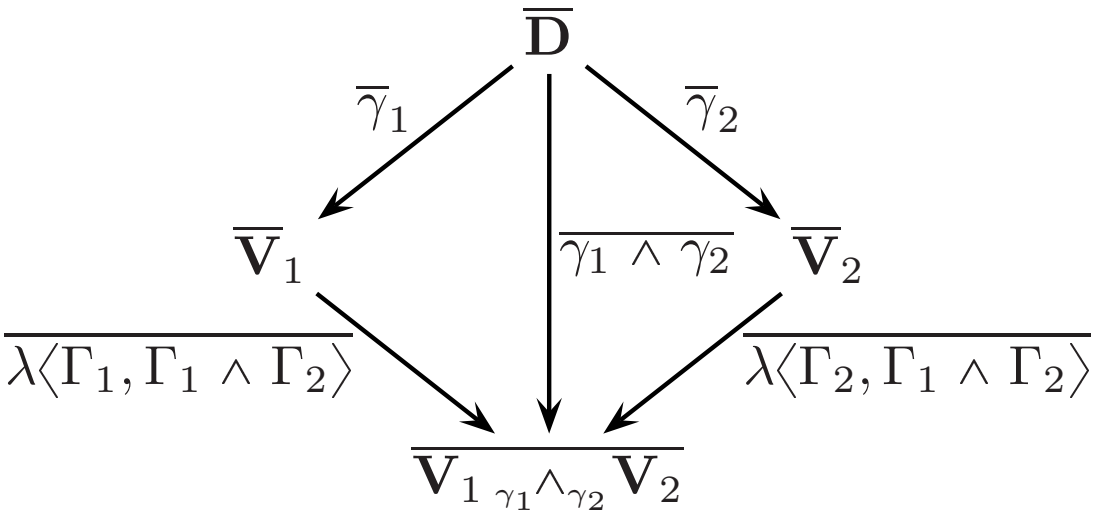
- A key construction is the extension of meets from the unconstrained to the constrained setting.

- The following diagram, which defines the meet of $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$, may be "lifted" to the unconstrained context in two ways.

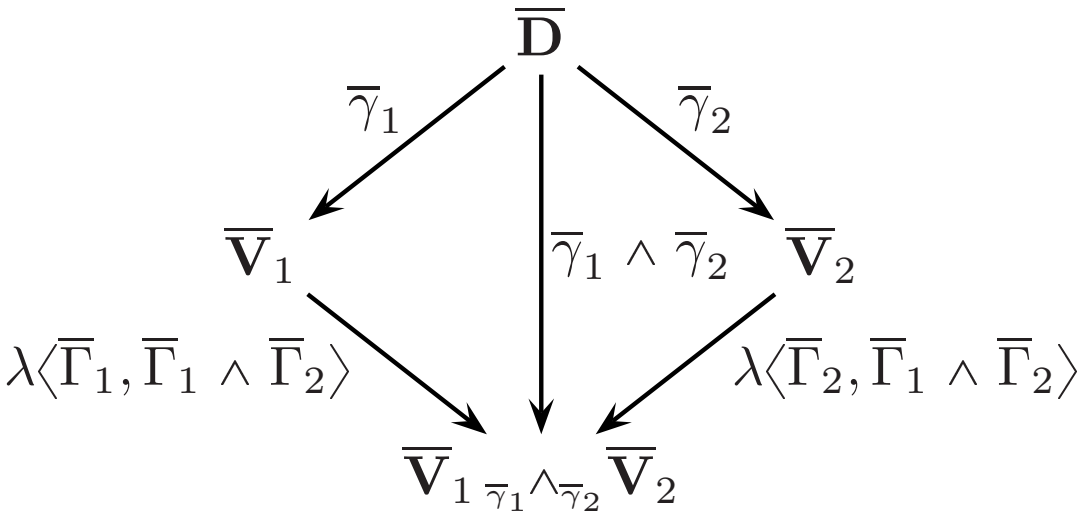


Lifting Equivalence (2)

- Lift the entire diagram at once:



- Lift each component separately:



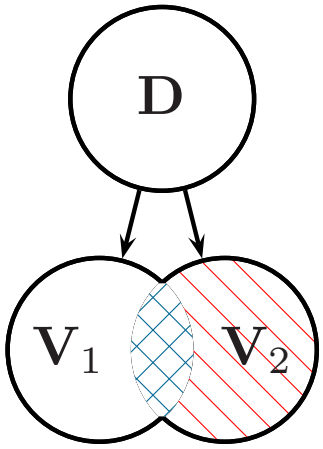
Theorem: These two liftings are the same.

The Main Result on Constraint Complexity in Views

If: $\mathbf{D}$ is a CFA-schema with the *k-submodel property*:

- $M \in \mathsf{LDB}(\mathbf{D})$ iff $N \in \mathsf{LDB}(\mathbf{D})$ for every $N \subseteq M$ with $\mathsf{Card}(N) \leqslant k$.

And: $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ is a CFA-view of $\mathbf{D}$, to be updated while holding CFA-meet complement $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ constant.



Then: $\mathbf{V}_1$ has the Γ_2-*relative k-submodel property*:

If: $\lambda\langle\Gamma_1, \Gamma_2\rangle(M) \in \mathsf{LDB}(\mathbf{V}_1 \,{}_{\gamma_1}\!\wedge_{\gamma_2} \mathbf{V}_2)$

Then: $M \in \mathsf{LDB}(\mathbf{V}_1)$ iff $N \in \mathsf{LDB}(\mathbf{V}_1)$ for every $N \subseteq M$ with $\mathsf{Card}(N) \leqslant k$. $\square$

Example: $k = 2$ in the classical relational setting recaptures functional dependencies.

Example of Constraint Complexity on a Closed View

Example base schema and instance:

$(E[\mathsf{ABCDE}], \mathcal{F} = \{\mathsf{A} \rightarrow \mathsf{D},\ \mathsf{B} \rightarrow \mathsf{D},\ \mathsf{CD} \rightarrow \mathsf{A},\ \mathsf{A} \rightarrow \mathsf{E}\})$:

A	B	C	D	E
a_0	b_0	c_0	d_0	e_0
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	d_2	e_2

View to be updated:

$\Pi_{\mathsf{ABCE}} = (E[\mathsf{ABCE}], \{\mathsf{A} \rightarrow \mathsf{E}\} \cup \pi_{\mathsf{ABC}}(\mathcal{F}))$.

A	B	C	E
a_0	b_0	c_0	e_0
a_1	b_1	c_1	e_1
a_2	b_1	c_2	e_2

Complementary view:

$\Pi_{\mathsf{ABCD}} = (E[\mathsf{ABCD}], \{\mathsf{A} \rightarrow \mathsf{D},\ \mathsf{B} \rightarrow \mathsf{D},\ \mathsf{CD} \rightarrow \mathsf{A}\})$.

A	B	C	D
a_0	b_0	c_0	d_0
a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2

- The view Π_{ABCE} is not finitely axiomatizable.
- Any update must hold $\Pi_{\mathsf{ABC}} = \Pi_{\mathsf{ABCE}} \wedge \Pi_{\mathsf{ABCD}}$ fixed.
- The allowed updates to Π_{ABCE} are limited to changes in the values of E, and these are governed solely by $\mathsf{A} \rightarrow \mathsf{E}$.

Syntactic Order-Based Updates

- The results developed in [Hegner 02 & 03] are *order based*.
  - The action of a closed update strategy is guaranteed to be unique only for updates which are realizable as sequences of insertions and deletions, *with all intermediate states legal*.
  - Updates which are not realizable as sequences of insertions and deletions, with all intermediate states legal, are not covered by the uniqueness results.

- The framework developed here extends these results within the context of CFA-views.

  If: the CFA-view $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ of $\mathbf{D}$ has meet complement $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$.

  Then: the view $\overline{\Gamma}_1 = (\overline{\mathbf{V}}_1, \overline{\gamma}_1)$ of $\overline{\mathbf{D}}$ has meet complement $\overline{\Gamma}_2 = (\overline{\mathbf{V}}_2, \overline{\gamma}_2)$.

  - In this extended context, *all* updates are *order based*.
  - Thus, *all* view updates under a closed strategy have a unique reflection into the main view.

Example of Syntactic Order-Based Update

Base schema:

$(E[\mathsf{ABC}], \{\mathsf{B} \rightarrow \mathsf{C}\})$

A	B	C
a_0	b_0	~~c_0~~ c_2
a_1	b_1	c_1
a_2	b_1	c_1

View to be updated:

$\Pi_{\mathsf{BC}} = (E[\mathsf{BC}], \{\mathsf{B} \rightarrow \mathsf{C}\})$

B	C
b_0	~~c_0~~ c_2
b_1	c_1

Constant complement view:

View $\Pi_{\mathsf{AB}} = (E[\mathsf{AB}], \varnothing)$

A	B
a_0	b_0
a_1	b_1
a_2	b_1

- Updates to Π_{BC} hold Π_{B} fixed.
- In view of $\mathsf{B} \rightarrow \mathsf{C}$, the only allowable updates are those which alter the value of C for a given B.
- None of these updates may be realized as a sequence of insertions and deletions with all intermediate states legal.
- Therefore, the results of [Hegner 02, 03] do not guarantee uniqueness of the update strategy.
- However, the theory given here does provide uniqueness of the update strategy, since all updates are syntactically order based.

Example of Syntactic Order-Based Update

Base schema:

$(E[\mathsf{ABC}], \{\mathsf{B} \to \mathsf{C}\})$

A	B	C
a_0	b_0	~~c_0~~ c_2
a_1	b_1	c_1
a_2	b_1	c_1

Constant complement view:

View $\Pi_{\mathsf{AB}} = (E[\mathsf{AB}], \varnothing)$

A	B
a_0	b_0
a_1	b_1
a_2	b_1

View to be updated:

$\Pi_{\mathsf{BC}} = (E[\mathsf{BC}], \{\mathsf{B} \to \mathsf{C}\})$

B	C
b_0	c_0
b_1	c_1

B	C
b_0	c_0
b_0	c_2
b_1	c_1

B	C
~~b_0~~	~~c_0~~
b_0	c_2
b_1	c_1

- Updates to Π_{BC} hold Π_{B} fixed.
- In view of $\mathsf{B} \to \mathsf{C}$, the only allowable updates are those which alter the value of C for a given B.
- None of these updates may be realized as a sequence of insertions and deletions with all intermediate states legal.
- Therefore, the results of [Hegner 02, 03] do not guarantee uniqueness of the update strategy.
- However, the theory given here does provide uniqueness of the update strategy, since all updates are syntactically order based.

Conclusions and Further Directions

Conclusion:

- Within the context of CFA-schemata and views, and constraints which are characterized by k-models, the complexity of update in closed views is no greater than that of update in the main schema.

Major question: Do similar results hold in other important frameworks?

⩾ Schemata involving referential constraints, such as foreign-key dependencies.

⩾ Database mappings which are not defined element-wise, such as those characterized by join dependencies.

Preliminary answer: Yes.

- Updates strategies for closed views appear to be very well behaved and similar complexity bounds apply in other contexts.
- The challenge is to identify a general framework in which general results may be established, rather than a long list of independently established special cases.