Composing Schema Mapping
An Overview

Phokion G. Kolaitis

UC Santa Cruz & IBM Research – Almaden

Joint work with
R. Fagin, L. Popa, and W.C. Tan

Data Interoperability

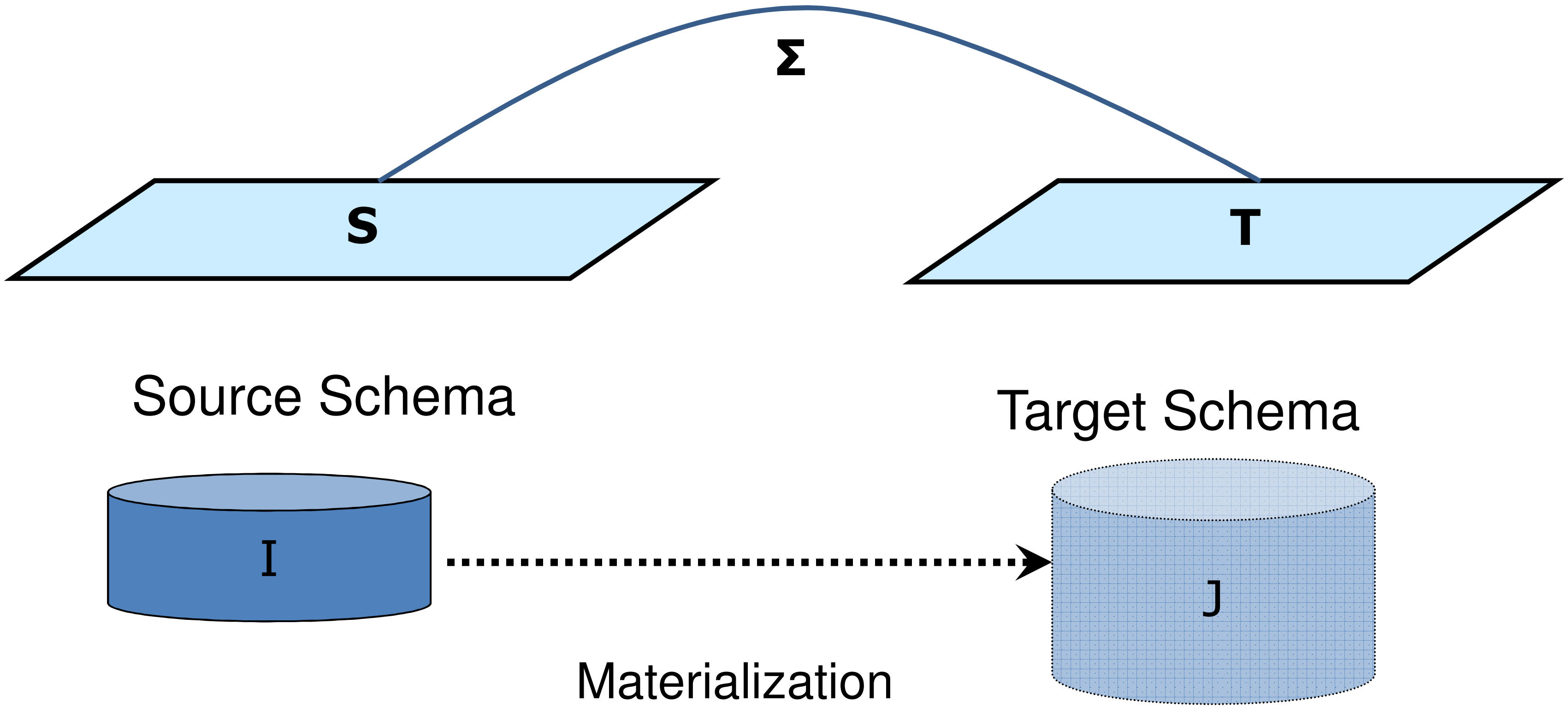
- Data may reside
  - at several different sites
  - in several different formats (relational, XML, …).

- Applications need to access, process, and query these data.

- Data Exchange:
  - A fundamental problem in data interoperability
  - Described as the "oldest problem in databases"
  - Formalized and studied in depth in the past 15 years.

Data Exchange

Transform data structured under a source schema into data structured under a different target schema.



Schema Mappings

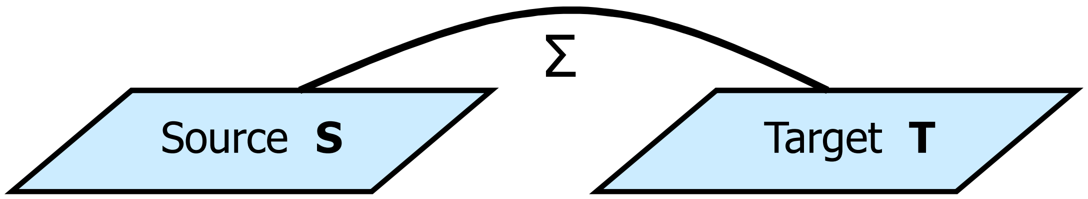
- Schema mappings:

  High-level, declarative assertions that specify the relationship between two database schemas.

- Schema mappings constitute the essential building blocks in formalizing and studying data interoperability tasks, including data exchange.

- Schema mappings make it possible to separate the design of the relationship between schemas from its implementation.

Schema Mappings



Schema Mapping **M** = (**S**, **T**, Σ)

- Source schema **S**, Target schema **T**
- Σ: High-level, declarative assertions that specify the relationship between **S** and **T**.

Question: What is a "good" schema-mapping specification language?

Schema-Mapping Specification Languages

- **Obvious Idea**:

  Use a logic-based language to specify schema mappings.

  In particular, use first-order logic.

- **Warning:**

  Unrestricted use of first-order logic as a schema-mapping specification language gives rise to **undecidability** of basic algorithmic problems about schema mappings.

Schema Mapping Specification Languages

Let us consider some simple tasks that every schema-mapping specification language should support:

- Copy (Nicknaming):
  - Copy each source table to a target table and rename it.
- Projection:
  - Form a target table by projecting on one or more columns of a source table.
- Column Augmentation:
  - Form a target table by adding one or more columns to a source table.
- Decomposition:
  - Decompose a source table into two or more target tables.
- Join:
  - Form a target table by joining two or more source tables.
- Combinations of the above (e.g., “join + column augmentation + …”)

Schema Mapping Specification Languages

- Copy (Nicknaming):
  - $\forall x_1, \ldots, x_n (P(x_1,\ldots,x_n) \rightarrow R(x_1,\ldots,x_n))$
- Projection:
  - $\forall x,y,z (P(x,y,z) \rightarrow R(x,y))$
- Column Augmentation:
  - $\forall x,y\ (P(x,y) \rightarrow \exists\, z\ R(x,y,z))$
- Decomposition:
  - $\forall x,y,z\ (P(x,y,z) \rightarrow R(x,y) \wedge T(y,z))$
- Join:
  - $\forall x,y,z (E(x,z) \wedge F(z,y) \rightarrow R(x,y,z))$
- Combinations of the above (e.g., "join + column augmentation + …")
  - $\forall x,y,z (E(x,z) \wedge F(z,y) \rightarrow \exists\, w\ (R(x,y) \wedge T(x,y,z,w)))$

Schema Mapping Specification Languages

- Question: What do all these tasks (copy, projection, column augmentation, decomposition, join) have in common?
- Answer:

  They can be specified using

  GLAV (global-and-local-as-view) constraints,

  also known as

  source-to-target tuple generating dependencies (s-t tgds).

  .

Schema Mapping Specification Language

The relationship between source and target is given by
GLAV constraints (s-t tgds)

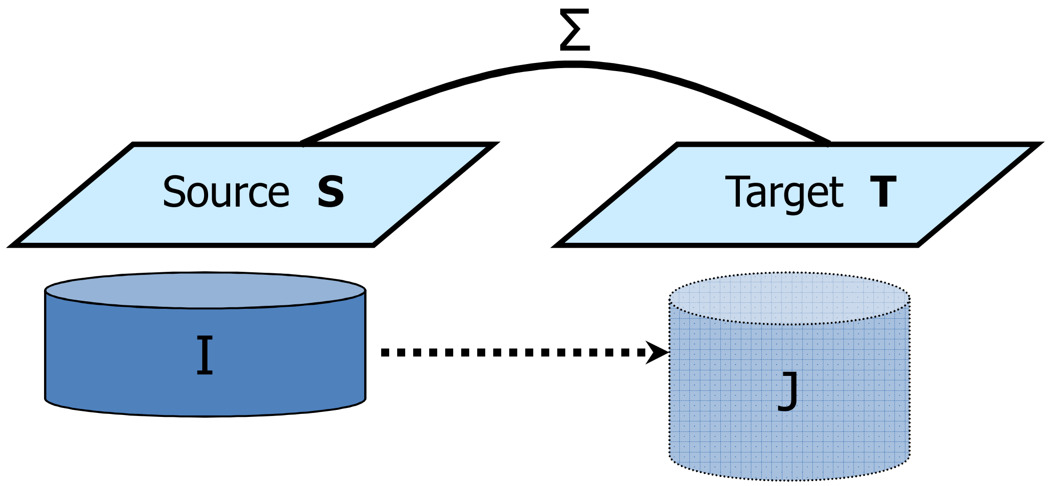
$$\forall \mathbf{x}\ (\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})),\ \text{where}$$

- $\varphi(\mathbf{x})$ is a conjunction of atoms over the source;
- $\psi(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms over the target.

- GLAV constraints assert that:
  some conjunctive query over the source is contained in some other conjunctive query over the target.

Example:
$\forall s\ \forall c\ (\text{Student}(s) \wedge \text{Enrolls}(s,c)) \rightarrow \exists t\ \exists g\ (\text{Teaches}(t,c) \wedge \text{Grade}(s,c,g))$

Schema Mappings & Data Exchange



- Data Exchange via the schema mapping **M** = (**S**, **T**, Σ)

Given a source instance I, construct a target instance J, so that (I, J) satisfy the specifications Σ of **M**.

Such a J is called a solution for I.

Difficulty:

- Usually, there are multiple solutions
- Which one is the "best" to materialize?

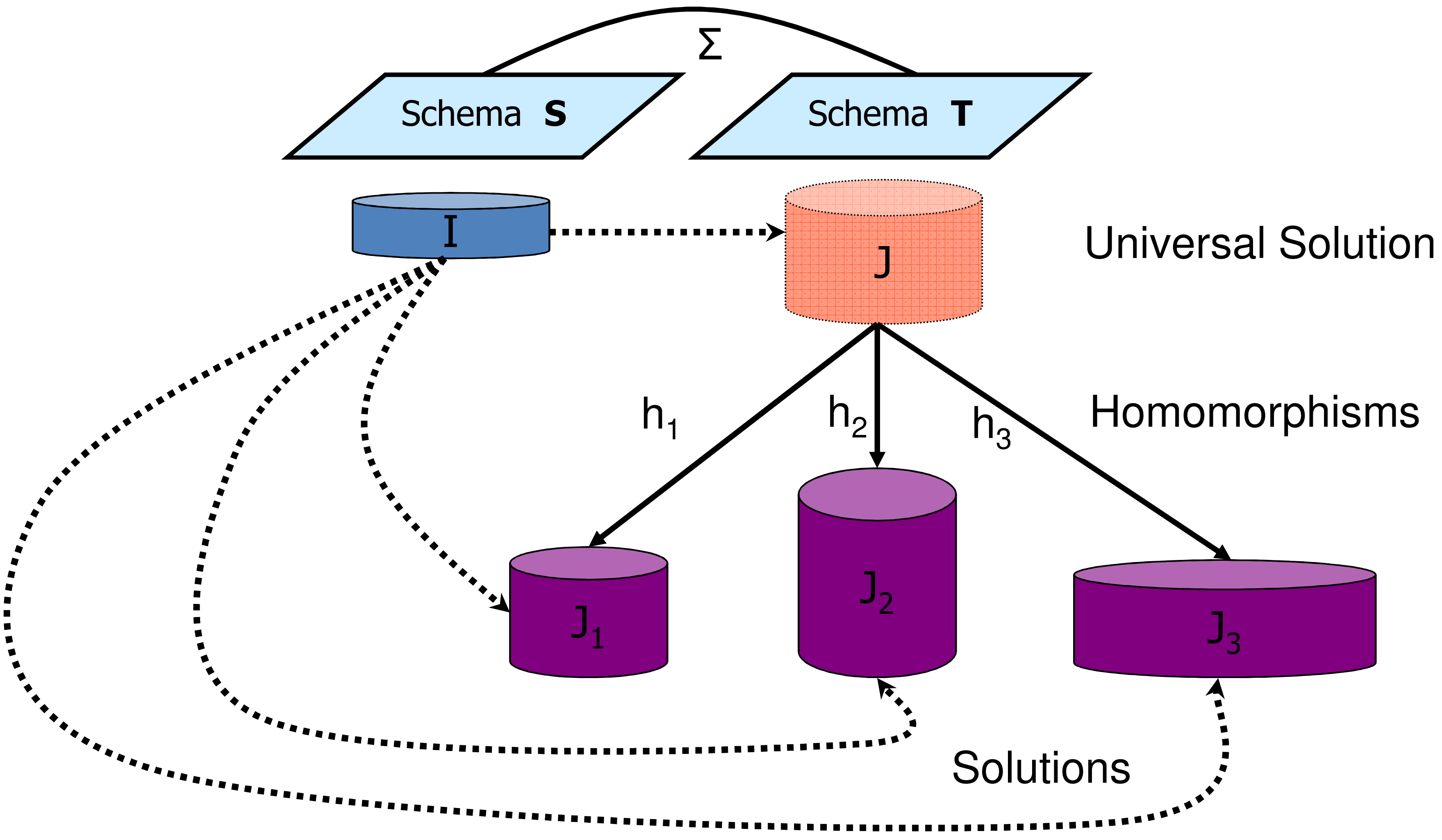
Data Exchange & Universal solutions

Fagin, K …, Miller, Popa:

Identified and studied the concept of a universal solution for GLAV mappings, i.e., schema mappings specified by a f finite set of GLAV constraints.

- A universal solutions is a most general solution.
- A universal solution “represents” the entire space of solutions.
- A “canonical” universal solution can be generated efficiently using the chase procedure.

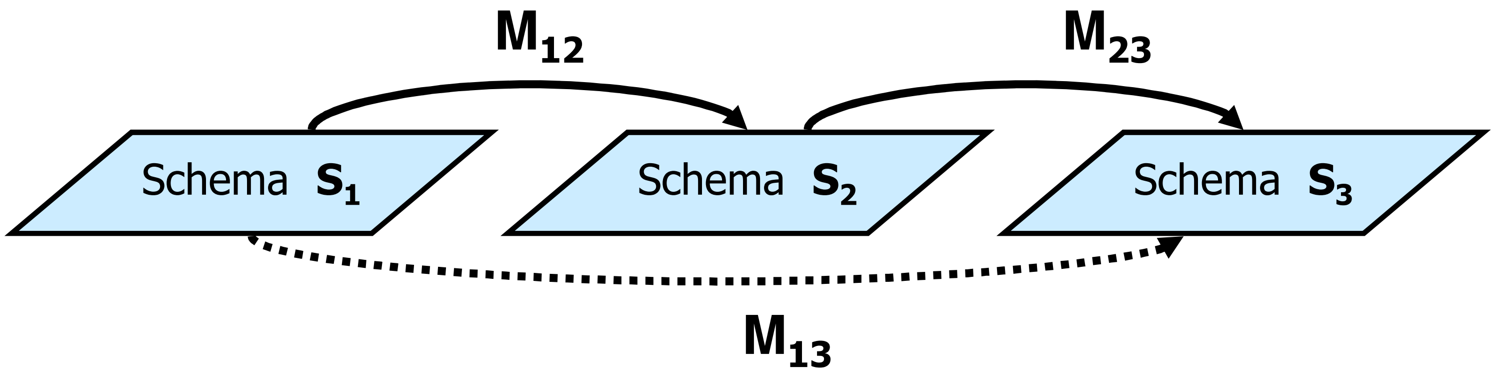
Universal Solutions in Data Exchange



Managing Schema Mappings

- Schema mappings can be quite complex.
- Methods and tools are needed to automate or semi-automate schema-mapping management.
- Metadata Management Framework – Bernstein 2003
  based on generic schema-mapping operators:
  - Match operator
  - Merge operator
  - Composition operator
  - Inverse operator

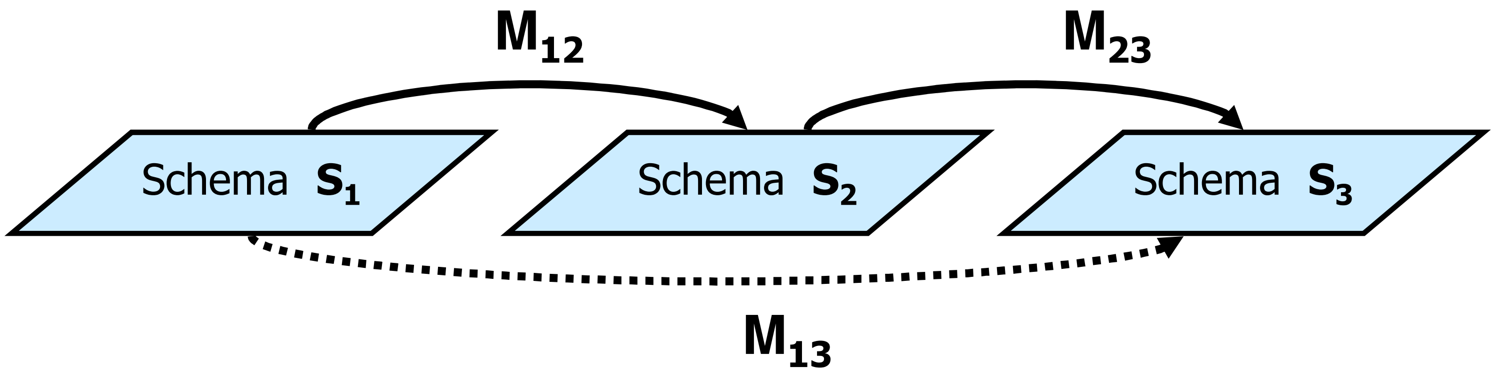
Composing Schema Mappings



- Given $\mathbf{M_{12}} = (\mathbf{S_1}, \mathbf{S_2}, \Sigma_{12})$ and $\mathbf{M_{23}} = (\mathbf{S_2}, \mathbf{S_3}, \Sigma_{23})$, derive a schema mapping $\mathbf{M_{13}} = (\mathbf{S_1}, \mathbf{S_3}, \Sigma_{13})$ that is "equivalent" to the sequential application of $\mathbf{M_{12}}$ and $\mathbf{M_{23}}$.
- $\mathbf{M_{13}}$ is a composition of $\mathbf{M_{12}}$ and $\mathbf{M_{23}}$, denoted

$$\mathbf{M_{13}} = \mathbf{M_{12}} \circ \mathbf{M_{23}}$$

Composing Schema Mappings



- Given $\mathbf{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathbf{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, derive a schema mapping $\mathbf{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ that is "equivalent" to the sequence $\mathbf{M}_{12}$ and $\mathbf{M}_{23}$.

What does it mean for $\mathbf{M}_{13}$ to be "equivalent" to the composition of $\mathbf{M}_{12}$ and $\mathbf{M}_{23}$?

Earlier Work

- Metadata Model Management (Bernstein in CIDR 2003)
  – Composition is one of the fundamental operators
  – However, no precise semantics is given

- Composing Mappings among Data Sources (Madhavan & Halevy in VLDB 2003)
  – First to propose a semantics for composition
  – Their notion of composition depends on the class of queries; it may not be unique up to logical equivalence.

Semantics of Composition

- Every schema mapping **M** = (**S**, **T**, Σ) defines a binary relationship Inst(**M**) between instances*:*

  $$\text{Inst}(\mathbf{M}) = \{ (I,J) \mid (I,J) \models \Sigma \}.$$

  From a semantic point of view, a schema mapping **M** can be identified with the set Inst(**M**).

- **Definition:** (FKPT)

  A schema mapping $\mathbf{M}_{13}$ is a composition of $\mathbf{M}_{12}$ and $\mathbf{M}_{23}$ if

  $$\text{Inst}(\mathbf{M}_{13}) = \text{Inst}(\mathbf{M}_{12}) \circ \text{Inst}(\mathbf{M}_{23}), \text{ that is,}$$

  $$(I_1,I_3) \models \Sigma_{13}$$

  if and only if

  there exists I_2 such that $(I_1,I_2) \models \Sigma_{12}$ and $(I_2,I_3) \models \Sigma_{23}$.

The Composition of Schema Mappings

Fact: If both $\mathbf{M} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma)$ and $\mathbf{M'} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma')$ are compositions of $\mathbf{M}_{12}$ and $\mathbf{M}_{23}$, then Σ are Σ' are logically equivalent.

For this reason:

- We say that **M** (or **M')** is *the* composition of $\mathbf{M}_{12}$ and $\mathbf{M}_{23}$.
- We write $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ to denote it

Issues in Composition of Schema Mappings

- The semantics of composition was the first main issue.
- The second main issue is the language of the composition.
  - Is the language of GLAV constraints closed under composition?
    If $\mathbf{M}_{12}$ and $\mathbf{M}_{23}$ are GLAV mappings,
    is $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ a GLAV mapping as well?
  - If not, what is the "right" language for composing schema mappings?

Inexpressibility of Composition

Theorem:

- GLAV mappings are **not** closed under composition.

- In fact, there are GLAV mappings $\mathbf{M}_{12}$ and $\mathbf{M}_{23}$ such that their composition $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is **not** expressible in least fixed-point logic LFP; in particular, $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is **not** expressible in first-order logic FO.

Lower Bounds for Composition

- $\mathbf{M}_{12}$:
  $\forall x \forall y\, (E(x,y) \rightarrow \exists u \exists v\, (C(x,u) \wedge C(y,v)))$
  $\forall x \forall y\, (E(x,y) \rightarrow F(x,y))$
- $\mathbf{M}_{23}$:
  $\forall x \forall y \forall u \forall v\, (C(x,u) \wedge C(y,v) \wedge F(x,y) \rightarrow D(u,v))$

- Given graph **G**=(V, E):
  - Let $I_1 = E$
  - Let $I_3 = \{ (r,g), (g,r), (b,r), (r,b), (g,b), (b,g) \}$

Fact:
G is 3-colorable iff $\langle I_1, I_3 \rangle \in \text{Inst}(\mathbf{M}_{12}) \circ \text{Inst}(\mathbf{M}_{23})$

- **Theorem (Dawar – 1998):**
  3-Colorability is **not** expressible in LFP.

Complexity of Composition

Definition: The model checking problem for a schema mapping **M** = (**S**, **T**, Σ) asks: given a source instanced I and a target instance J, does $\langle I,J \rangle \models \Sigma$?

Fact: If **M** is a GLAV mapping, then the model checking problem for **M** is in LOGSPACE.

Fact: There are GLAV mappings $\mathbf{M}_{12}$ and $\mathbf{M}_{23}$ such that the model checking problem for their composition $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is NP-complete.

Employee Example

$\mathbf{M}_{12}$: $\forall e\ (\mathrm{Emp}(e) \rightarrow \exists m\ \mathrm{Rep}(e,m))$

$\mathbf{M}_{23}$: $\forall e\ \forall m\ (\mathrm{Rep}(e,m) \rightarrow \mathrm{Mgr}(e,m))$

$\forall e\ \ (\mathrm{Rep}(e,e) \rightarrow \mathrm{SelfMgr}(e))$

Theorem:

- The composition $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is not definable by any set (finite or infinite) of GLAV constraints.
- The composition $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is definable in a fragment of Second-Order Logic that extends GLAV constraints with Skolem functions.

Employee Example - revisited

M_{12} :

- $\forall e\ (\mathrm{Emp}(e) \rightarrow \exists m\ \mathrm{Rep}(e,m))$

M_{23} :

- $\forall e\ \forall m(\mathrm{Rep}(e,m) \rightarrow \mathrm{Mgr}(e,m))$
- $\forall e\ (\mathrm{Rep}(e,e) \rightarrow \mathrm{SelfMgr}(e))$

Fact: **$M_{12} \circ M_{23}$** is definable by the following SO-tgd

$$\exists \mathbf{f}\ (\forall e(\mathrm{Emp}(e) \rightarrow \mathrm{Mgr}(e,\mathbf{f(e)})) \wedge \\ \forall e(\mathrm{Emp}(e) \wedge (\mathbf{e=f(e)}) \rightarrow \mathrm{SelfMgr}(e)))$$

Second-Order Tgds

Definition: Let **S** be a source schema and **T** a target schema.

A second-order tuple-generating dependency (SO tgd) is a formula of the form:

$\exists f_1 \ldots \exists f_m((\forall \mathbf{x_1}(\phi_1 \rightarrow \psi_1)) \wedge \ldots \wedge (\forall \mathbf{x}_n(\phi_n \rightarrow \psi_n)))$, where

- Each f_i is a function symbol.
- Each ϕ_i is a conjunction of atoms from **S** and equalities of terms.
- Each ψ_i is a conjunction of atoms from **T.**

Example: $\exists \mathbf{f}\, (\forall e(\mathrm{Emp}(e) \rightarrow \mathrm{Mgr}(e,\mathbf{f(e)})) \wedge$
$\forall e(\mathrm{Emp}(e) \wedge (\mathbf{e=f(e)}) \rightarrow \mathrm{SelfMgr}(e)))$

Composing SO-Tgds and Data Exchange

Theorem (FKPT):

- The composition of two SO-tgds is definable by a SO-tgd.
- There is an algorithm for composing SO-tgds.
- The chase procedure can be extended to SO-tgds; it produces universal solutions in polynomial time.
- Every SO tgd is the composition of finitely many finite sets of s-t tgds. Hence, SO tgds are the "right" language for the composition of s-t tgds

When is the composition FO-definable?

Fact:

- It is an undecidable problem to tell whether the composition of two GLAV mappings is FO-definable.
- However, there are certain sufficient conditions that guarantee that the composition of two GLAV mappings is FO-definable.

LAV, Extended LAV, and GAV Mappings

GLAV constraints: $\forall \mathbf{x} \, (\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y} \, \psi(\mathbf{x}, \mathbf{y}))$,

- **LAV (local-as-view)** constraints:

  $\forall \mathbf{x} \; (P(\mathbf{x}) \rightarrow \psi(\mathbf{x}))$, where each variable occurs ***only once*** in P(x).

  – Copy, Projection, Column Augmentation, Decomposition, …

- **Extended LAV** constraints:

  $\forall \mathbf{x} \; (P(\mathbf{x}) \rightarrow \psi(\mathbf{x}))$, where P is a source relation

  (a variable may occur more than once in P(x))

  – $\forall e \, (Rep(e,e) \rightarrow SelfMgr(e))$

- **GAV (global-as-view)** constraints:

  $\forall \mathbf{x} \; (\varphi(\mathbf{x}) \rightarrow R(\mathbf{x}))$, where R is a target relation

  – Copy, Projection, Join

Composing GLAV Schema Mappings

Composition	**Logically Equivalent**
GAV ∘ GAV	GAV
GAV ∘ GLAV	GLAV
GLAV ∘ LAV	GLAV
LAV ∘ Extended LAV	not GLAV
LAV ∘ GAV	not GLAV

Note:

- LAV ∘ LAV equivalent to GLAV (special case of GLAV ∘ LAV) was established by Arocena, Fuxman, Miller (2010).
- LAV ∘ Extended LAV - Employee schema mapping
- LAV ∘ GAV - 3-Colorability schema mapping

Synopsis of Schema Mapping Composition

- GLAV mappings are not closed under composition.
- SO-tgds form a well-behaved fragment of second-order logic.
  - SO-tgds are closed under composition; they are the "right" language for composing GLAV mappings.
  - SO-tgds are "chasable": Polynomial-time data exchange with universal solutions.
- SO-tgds and the composition algorithm have been incorporated in the IBM InfoSphere Information Server.

Related Work

- Composition with respect to conjunctive-query equivalence
  Madhavan-Halevy – 2003, K … and Fagin – 2012,
  Arenas, Pérez, Reutter - 2013

- Composing more expressive schema mappings
  Nash, Bernstein, Melnik – 2007

- Composing XML Schema Mappings
  Amano, Libkin, Murlak – 2009

- Categorical treatment of data migration
  Spivak – 2012, Spivak and Wisnesky - 2015

“The notion of composition of maps leads to the most natural account of fundamental notions of mathematics, from multiplication, addition, and exponentiation, through the basic notions of logic."

"Conceptual Mathematics"
by
W. Lawvere and S. Schanuel