Reasoning about XML Constraints based on XML-to-relational mappings

Matthias Niewerth Thomas Schwentick

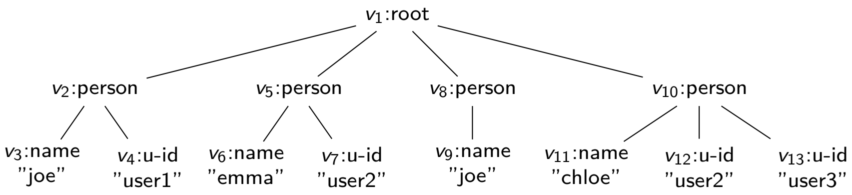
Problem Description

Foundations
of
XML

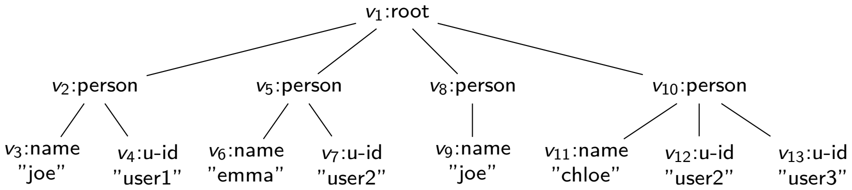
Project Tasks

- Develop a rich modelling language for XML constraints.
- Find an optimal trade-off between structure and data.

Example


v1:root
v2:person
v5:person
v8:person
v10:person
v3:name "joe"
v4:u-id "user1"
v6:name "emma"
v7:u-id "user2"
v9:name "joe"
v11:name "chloe"
v12:u-id "user2"
v13:u-id "user3"

Example

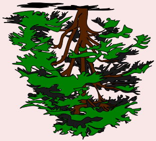


How to express that . . .

- . . . u-ids are unique?
- . . . every person has only one name?
- . . . every u-id has at most one associated name?

Contents

Definition



$\longrightarrow$

a	b	c

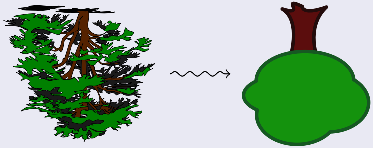
Existing Work

- XFDs
- Hierachical Constraints
- XML Schema Constraints

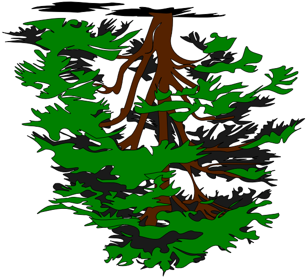
Reasoning

- Model-Checking
- Implication Problem

Tree Chase



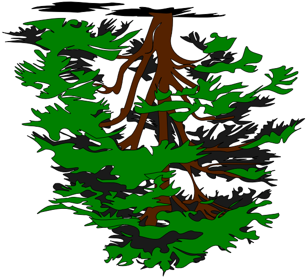
XML-to-relational Constraints



mapping →

a	b	c	d	e	f

XML-to-relational Constraints



mapping →

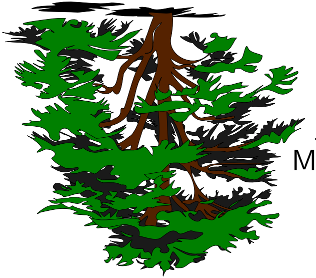
a	b	c	d	e	f

Definition: XML-to-rel. Constraint

- mapping $m : \mathcal{T} \to \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

XML-to-relational Constraints



mapping $\longrightarrow$ MSO,FO,XPath

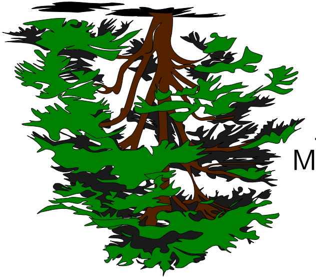
a	b	c	d	e	f

Definition: XML-to-rel. Constraint

- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

XML-to-relational Constraints



mapping

$\longrightarrow$

MSO,FO,XPath

a	b	c	d	e	f

Definition: XML-to-rel. Constraint

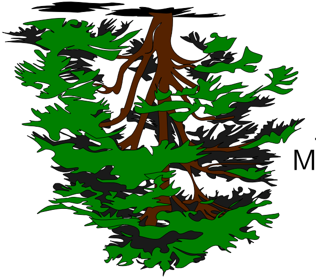
- mapping $m : \mathcal{T} \to \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



mapping
→
MSO,FO,XPath

a	b	c	d	e	f

Definition: XML-to-rel. Constraint

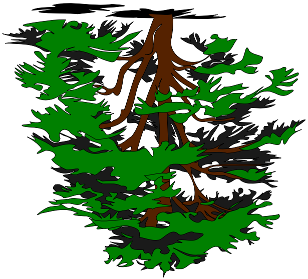
- mapping $m : \mathcal{T} \to \mathcal{R}$
- relational constraint σ

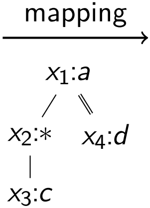
$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints





a	b	c	d	e	f

Definition: XML-to-rel. Constraint

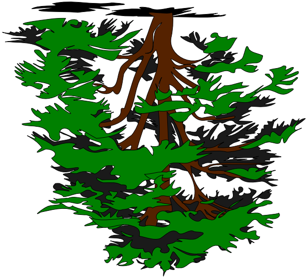
- mapping $m : \mathcal{T} \to \mathcal{R}$
- relational constraint σ

$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



mapping →

$x_1{:}a$

$x_2{:}*$ $x_4{:}d$

$x_3{:}c$

x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$

Definition: XML-to-rel. Constraint

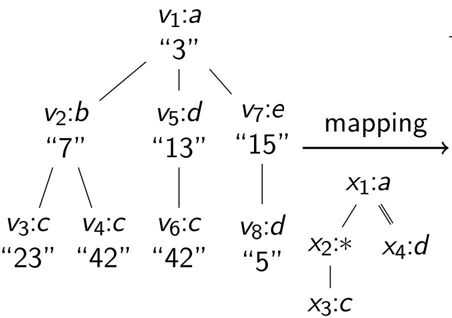
- mapping $m : \mathcal{T} \to \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$

Definition: XML-to-rel. Constraint

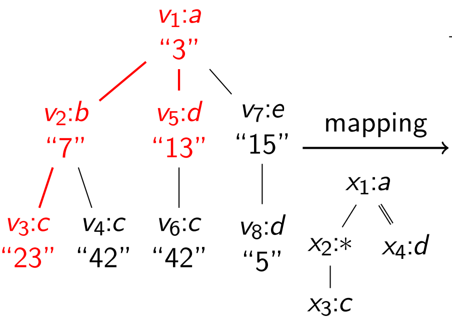
- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13

Definition: XML-to-rel. Constraint

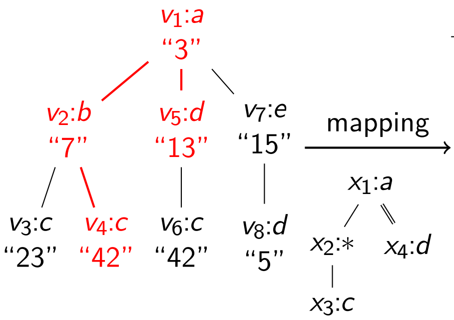
- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13

Definition: XML-to-rel. Constraint

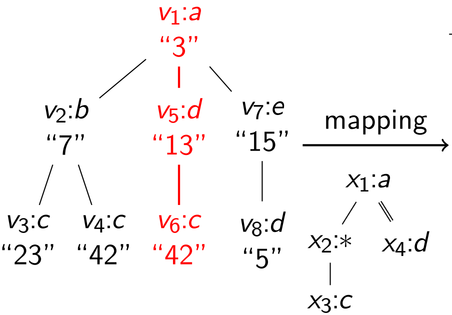
- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13

Definition: XML-to-rel. Constraint

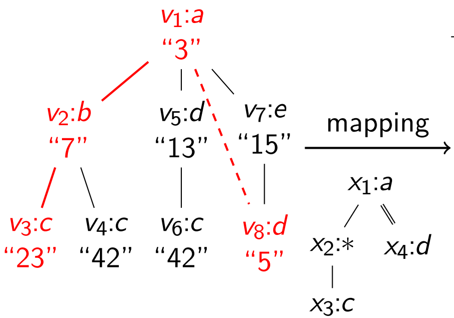
- mapping $m : \mathcal{T} \to \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5

Definition: XML-to-rel. Constraint

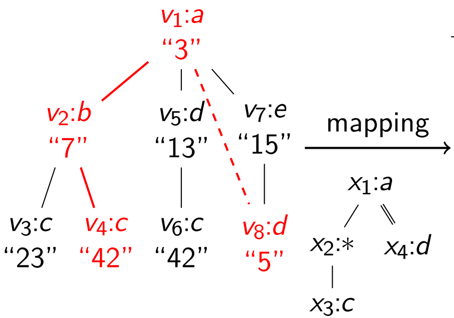
- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5
v_1	3	v_2	7	v_4	42	v_8	5

Definition: XML-to-rel. Constraint

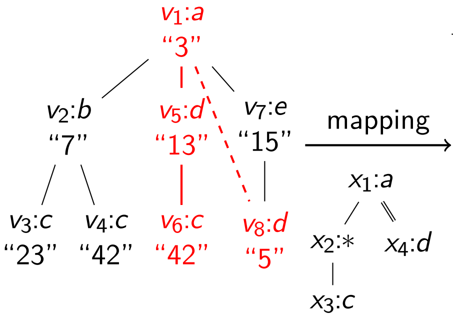
- mapping $m : \mathcal{T} \to \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \Leftrightarrow m(t) \models \sigma$$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5
v_1	3	v_2	7	v_4	42	v_8	5
v_1	3	v_5	13	v_6	42	v_8	5

Definition: XML-to-rel. Constraint

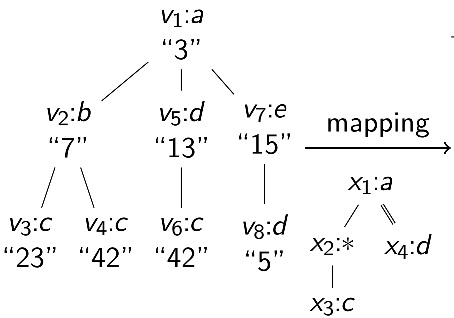
- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Semantical Restriction

Mappings should be invariant under renamings of datavalues.

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5
v_1	3	v_2	7	v_4	42	v_8	5
v_1	3	v_5	13	v_6	42	v_8	5

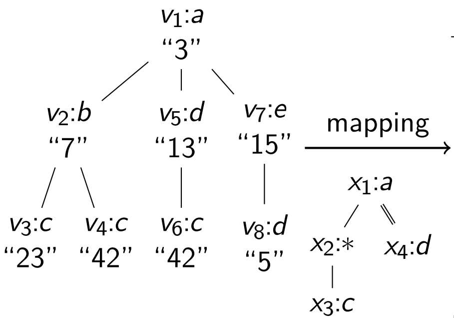
Definition: XML-to-rel. Constraint

- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \Leftrightarrow m(t) \models \sigma$$

Functional Dependencies:

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5
v_1	3	v_2	7	v_4	42	v_8	5
v_1	3	v_5	13	v_6	42	v_8	5

Definition: XML-to-rel. Constraint

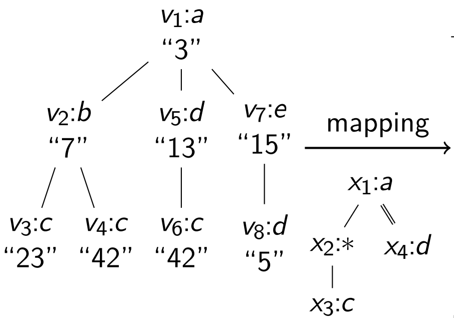
- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Functional Dependencies:

- $x_2.@ \rightarrow x_2$

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5
v_1	3	v_2	7	v_4	42	v_8	5
v_1	3	v_5	13	v_6	42	v_8	5

Definition: XML-to-rel. Constraint

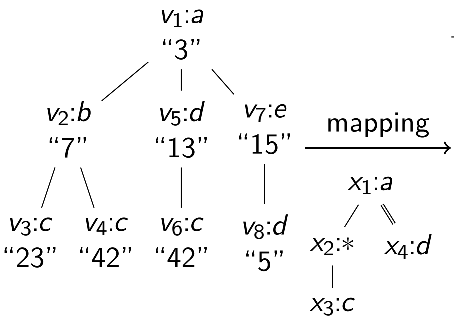
- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \Leftrightarrow m(t) \models \sigma$$

Functional Dependencies:

- $x_2.@ \rightarrow x_2$ ✓

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5
v_1	3	v_2	7	v_4	42	v_8	5
v_1	3	v_5	13	v_6	42	v_8	5

Definition: XML-to-rel. Constraint

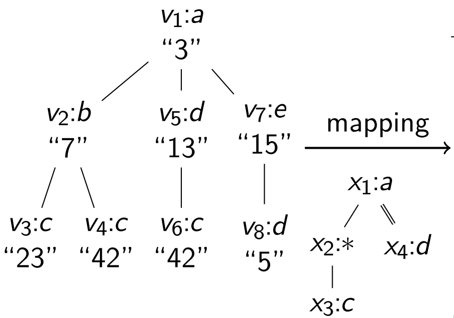
- mapping $m : \mathcal{T} \to \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$$

Functional Dependencies:

- $x_2.@ \rightarrow x_2$ ✓
- $x_3.@ \rightarrow x_3$

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5
v_1	3	v_2	7	v_4	42	v_8	5
v_1	3	v_5	13	v_6	42	v_8	5

Definition: XML-to-rel. Constraint

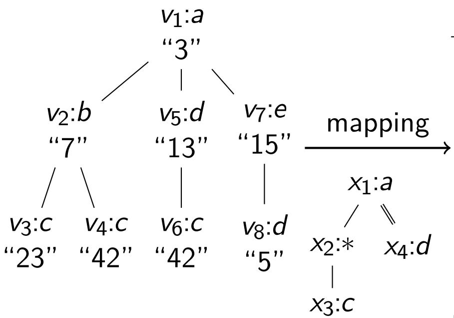
- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \Leftrightarrow m(t) \models \sigma$$

Functional Dependencies:

- $x_2.@ \rightarrow x_2$ ✓
- $x_3.@ \rightarrow x_3$ ✗

XML-to-relational Constraints



x_1	x_1.@	x_2	x_2.@	x_3	x_3.@	x_4	x_4.@
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5
v_1	3	v_2	7	v_4	42	v_8	5
v_1	3	v_5	13	v_6	42	v_8	5

Definition: XML-to-rel. Constraint

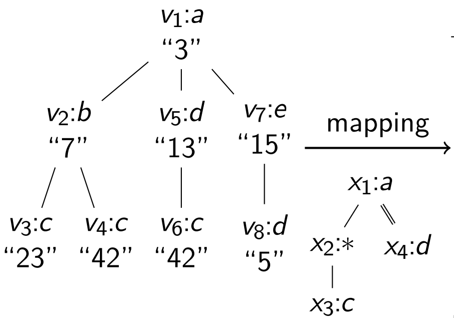
- mapping $m : \mathcal{T} \rightarrow \mathcal{R}$
- relational constraint σ

$$t \models (m, \sigma) \ \Leftrightarrow \ m(t) \models \sigma$$

Functional Dependencies:

- $x_2.@ \rightarrow x_2$ ✓
- $x_3.@ \rightarrow x_3$ ✗
- $x_2, x_3.@ \rightarrow x_3$

XML-to-relational Constraints



x_1	$x_1.@$	x_2	$x_2.@$	x_3	$x_3.@$	x_4	$x_4.@$
v_1	3	v_2	7	v_3	23	v_5	13
v_1	3	v_2	7	v_4	42	v_5	13
v_1	3	v_5	13	v_6	42	v_5	13
v_1	3	v_2	7	v_3	23	v_8	5
v_1	3	v_2	7	v_4	42	v_8	5
v_1	3	v_5	13	v_6	42	v_8	5

Definition: XML-to-rel. Constraint

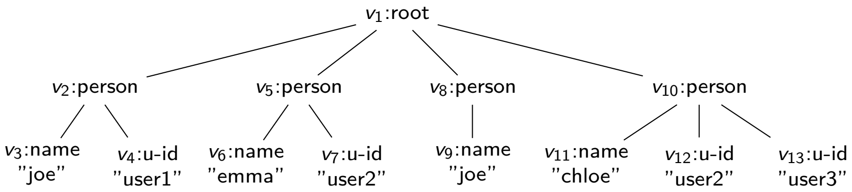
- mapping $m : \mathcal{T} \to \mathcal{R}$
- relational constraint σ

$t \models (m, \sigma) \;\Leftrightarrow\; m(t) \models \sigma$

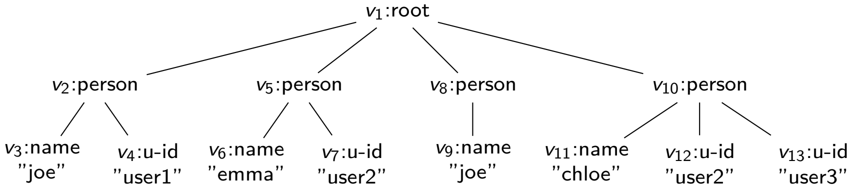
Functional Dependencies:

- $x_2.@ \to x_2$ ✓
- $x_3.@ \to x_3$ ✗
- $x_2, x_3.@ \to x_3$ ✓

Example


v1:root
v2:person
v5:person
v8:person
v10:person
v3:name "joe"
v4:u-id "user1"
v6:name "emma"
v7:u-id "user2"
v9:name "joe"
v11:name "chloe"
v12:u-id "user2"
v13:u-id "user3"

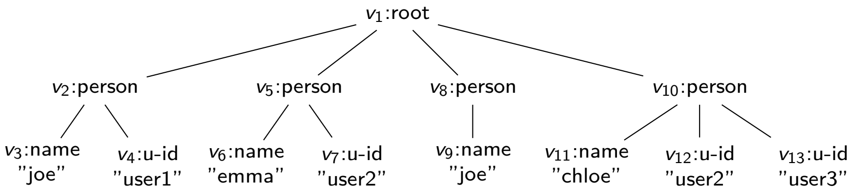
Example



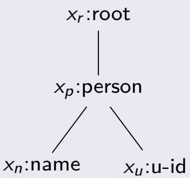
How to express that ...

- ... u-ids are unique?
- ... every person has only one name (node)?
- ... every u-id has at most one associated name (data value)?

Example



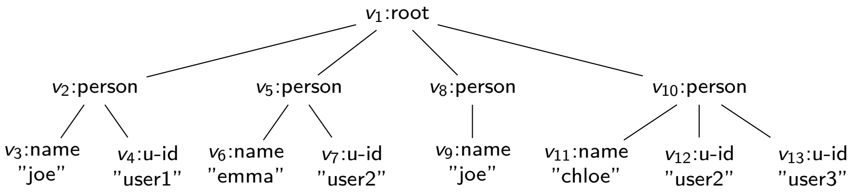
Pattern/Mapping



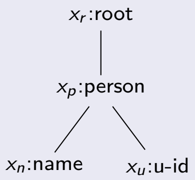
How to express that ...

- ... u-ids are unique?
- ... every person has only one name (node)?
- ... every u-id has at most one associated name (data value)?

Example



Pattern/Mapping



How to express that ...

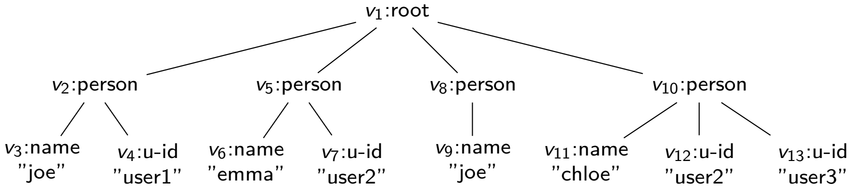
- ... u-ids are unique?

  $x_u.@ \longrightarrow x_u$

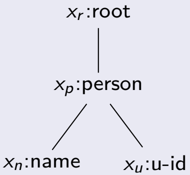
- ... every person has only one name (node)?

- ... every u-id has at most one associated name (data value)?

Example



Pattern/Mapping



How to express that ...

- ... u-ids are unique?

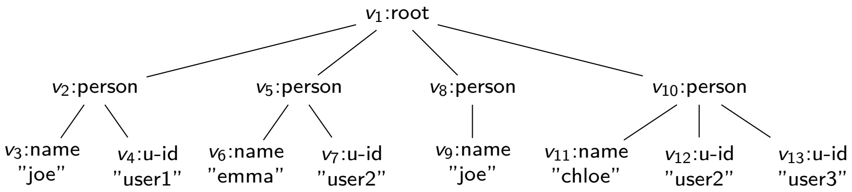
  $x_u.@ \rightarrow x_u$

- ... every person has only one name (node)?

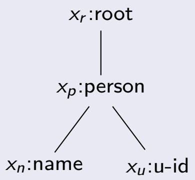
  $x_p \rightarrow x_n$

- ... every u-id has at most one associated name (data value)?

Example



Pattern/Mapping



How to express that ...

- ... u-ids are unique?

  $x_u.@ \rightarrow x_u$

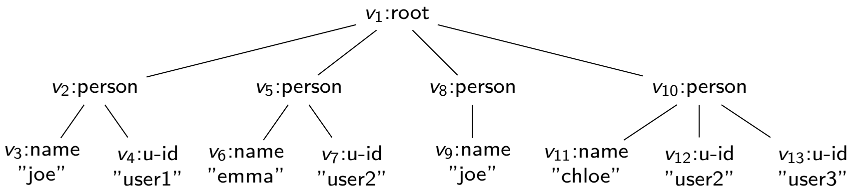
- ... every person has only one name (node)?

  $x_p \rightarrow x_n$

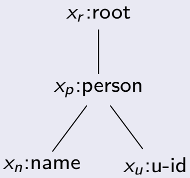
- ... every u-id has at most one associated name (data value)?

  $x_u.@ \rightarrow x_n.@$

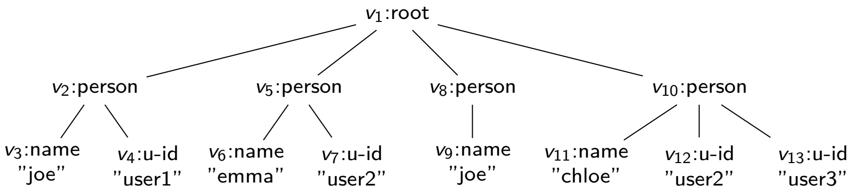
Incomplete/Missing Data



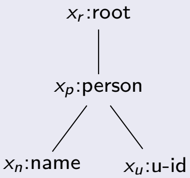
Pattern/Mapping



Incomplete/Missing Data



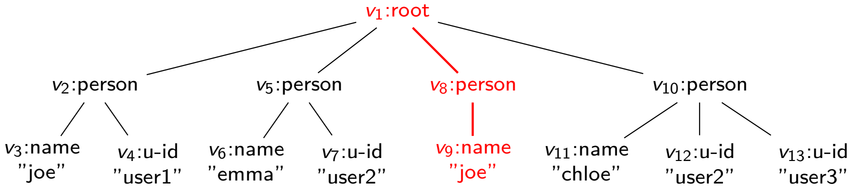
Pattern/Mapping



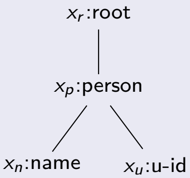
Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	joe	v_4	user1
v_1	v_5	v_6	emma	v_7	user2
v_1	v_{10}	v_{11}	chloe	v_{12}	user2
v_1	v_{10}	v_{11}	chloe	v_{13}	user3

Incomplete/Missing Data



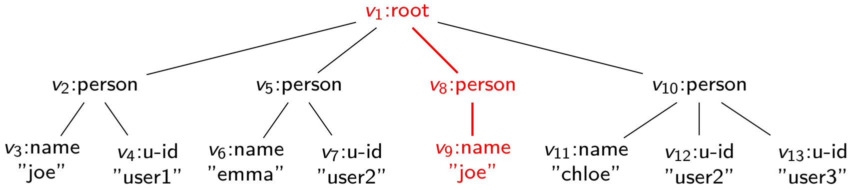
Pattern/Mapping



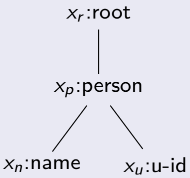
Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	joe	v_4	user1
v_1	v_5	v_6	emma	v_7	user2
v_1	v_{10}	v_{11}	chloe	v_{12}	user2
v_1	v_{10}	v_{11}	chloe	v_{13}	user3

Incomplete/Missing Data



Pattern/Mapping



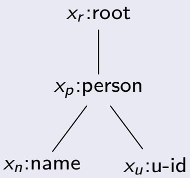
Relation

x_r	x_p	x_n	x_n.@	x_u	x_u.@
v_1	v_2	v_3	joe	v_4	user1
v_1	v_5	v_6	emma	v_7	user2
v_1	v_{10}	v_{11}	chloe	v_{12}	user2
v_1	v_{10}	v_{11}	chloe	v_{13}	user3

Incomplete/Missing Data

Semantics

Pattern/Mapping



Relation

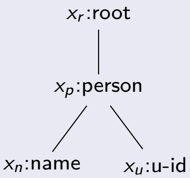
x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	joe	v_4	user1
v_1	v_5	v_6	emma	v_7	user2
v_1	v_8	v_9	joe	$\perp$	$\perp$
v_1	v_{10}	v_{11}	chloe	v_{12}	user2
v_1	v_{10}	v_{11}	chloe	v_{13}	user3

Incomplete/Missing Data

Semantics

- "Complete Data Semantics":
  One tuple for every complete embedding of the tree pattern

Pattern/Mapping



Relation

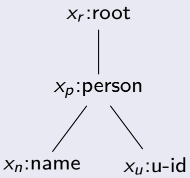
x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	joe	v_4	user1
v_1	v_5	v_6	emma	v_7	user2
v_1	v_8	v_9	joe	$\perp$	$\perp$
v_1	v_{10}	v_{11}	chloe	v_{12}	user2
v_1	v_{10}	v_{11}	chloe	v_{13}	user3

Incomplete/Missing Data

Semantics

- "Complete Data Semantics":
  One tuple for every complete embedding of the tree pattern
- "Incomplete Data Semantics":
  One tuple for every maximal embedding of the tree pattern

Pattern/Mapping



Relation

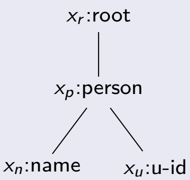
x_r	x_p	x_n	x_n.@	x_u	x_u.@
v_1	v_2	v_3	joe	v_4	user1
v_1	v_5	v_6	emma	v_7	user2
v_1	v_8	v_9	joe	$\perp$	$\perp$
v_1	v_{10}	v_{11}	chloe	v_{12}	user2
v_1	v_{10}	v_{11}	chloe	v_{13}	user3

Incomplete/Missing Data

Semantics

- "Complete Data Semantics":
  One tuple for every complete embedding of the tree pattern
- "Incomplete Data Semantics":
  One tuple for every maximal embedding of the tree pattern
- Ficticious Functional Dependecies to control Null Values

Pattern/Mapping



Relation

x_r	x_p	x_n	x_n.@	x_u	x_u.@
v_1	v_2	v_3	joe	v_4	user1
v_1	v_5	v_6	emma	v_7	user2
v_1	v_8	v_9	joe	$\perp$	$\perp$
v_1	v_{10}	v_{11}	chloe	v_{12}	user2
v_1	v_{10}	v_{11}	chloe	v_{13}	user3

Incomplete/Missing Data

Semantics

- "Complete Data Semantics":
  One tuple for every complete embedding of the tree pattern
- "Incomplete Data Semantics":
  One tuple for every maximal embedding of the tree pattern
- Ficticious Functional Dependecies to control Null Values

In this Paper

Incomplete/Missing Data

Semantics

- "Complete Data Semantics":
  One tuple for every complete embedding of the tree pattern
- "Incomplete Data Semantics":
  One tuple for every maximal embedding of the tree pattern
- Ficticious Functional Dependecies to control Null Values

In this Paper

- "Incomplete Data Semantics" to compare with other frameworks.

Incomplete/Missing Data

Semantics

- "Complete Data Semantics":
  One tuple for every complete embedding of the tree pattern
- "Incomplete Data Semantics":
  One tuple for every maximal embedding of the tree pattern
- Ficticious Functional Dependecies to control Null Values

In this Paper

- "Incomplete Data Semantics" to compare with other frameworks.
- "Complete Data Semantics" for complexity results.

Incomplete/Missing Data

Semantics

- "Complete Data Semantics":
  One tuple for every complete embedding of the tree pattern
- "Incomplete Data Semantics":
  One tuple for every maximal embedding of the tree pattern
- Ficticious Functional Dependecies to control Null Values

In this Paper

- "Incomplete Data Semantics" to compare with other frameworks.
- "Complete Data Semantics" for complexity results.
- Full version of paper will use "Incomplete Data Semantics"

Functional Dependencies vs. Key Constraints

Functional Dependencies vs. Key Constraints

Informal Definition

A key constraint uniquely identifies a real-world entity.

Functional Dependencies vs. Key Constraints

Informal Definition

A key constraint uniquely identifies a real-world entity.

Observation

In XML documents real-world entities correspond to nodes.

Functional Dependencies vs. Key Constraints

Informal Definition

A key constraint uniquely identifies a real-world entity.

Observation

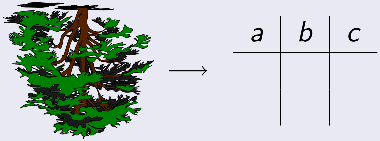
In XML documents real-world entities correspond to nodes.

Formal Definition: XKFD

An XML Key Functional Dependency (XKFD) is an FD $(m, A \rightarrow b)$, where b refers to a node.

Contents

Definition



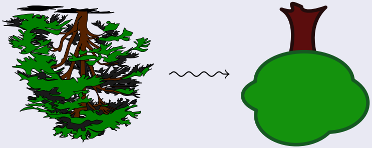
Existing Work

- XFDs
- Hierachical Constraints
- XML Schema Constraints

Reasoning

- Model-Checking
- Implication Problem

Tree Chase



XML Functional Dependencies (XFDs)

- Introduced by Arenas, Fan and Libkin 2002

XML Functional Dependencies (XFDs)

- Introduced by Arenas, Fan and Libkin 2002
- Further examined by Kot and White 2007

XML Functional Dependencies (XFDs)

- Introduced by Arenas, Fan and Libkin 2002
- Further examined by Kot and White 2007

Syntax/Example

Set of PATHs	$\rightarrow$	PATH
{/root/person/u-id.@}	$\rightarrow$	/root/person/u-id

XML Functional Dependencies (XFDs)

- Introduced by Arenas, Fan and Libkin 2002
- Further examined by Kot and White 2007

Syntax/Example

Set of PATHs	$\rightarrow$	PATH
{/root/person/u-id.@}	$\rightarrow$	/root/person/u-id

Semantics (in our framework)

- Combine all paths into a tree pattern.
- Use the canonical FD

XML Functional Dependencies (XFDs)

- Introduced by Arenas, Fan and Libkin 2002
- Further examined by Kot and White 2007

Syntax/Example

$$\text{Set of PATHs} \rightarrow \text{PATH}$$
$$\{/\text{root}/\text{person}/\text{u-id}.@\} \rightarrow /\text{root}/\text{person}/\text{u-id}$$

Semantics (in our framework)

- Combine all paths into a tree pattern.
- Use the canonical FD

Definition [Arenas, Fan, Libkin 2002]

An XML Database is in XML Normal Form, if all FDs are XKFDs.

Hierachical Key Constraints

Hierachical Key Constraints

- Introduced by Buneman et al. 2001

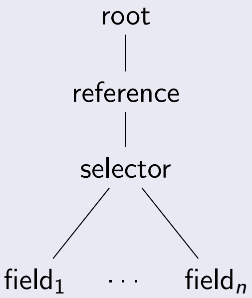
Hierachical Key Constraints

- Introduced by Buneman et al. 2001
- Used for XML Schema Integrity Constraints

Hierachical Key Constraints

- Introduced by Buneman et al. 2001
- Used for XML Schema Integrity Constraints

Syntax

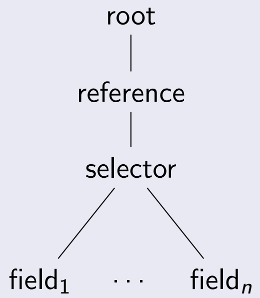


$$\text{reference}, \text{field}_1.@, \ldots, \text{field}_n.@ \rightarrow \text{selector}$$

Hierachical Key Constraints

- Introduced by Buneman et al. 2001
- Used for XML Schema Integrity Constraints

Syntax



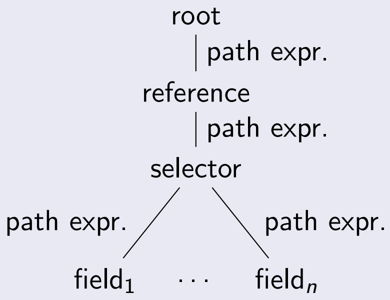
$\text{reference}, \text{field}_1.@, \ldots, \text{field}_n.@ \rightarrow \text{selector}$

- All hierachical key constraints are XKFDs.

Hierachical Key Constraints

- Introduced by Buneman et al. 2001
- Used for XML Schema Integrity Constraints

Syntax



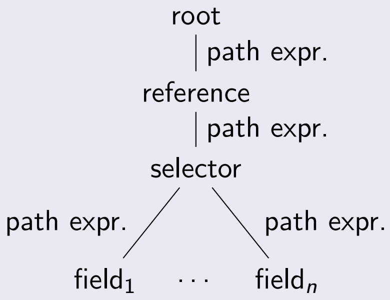
$$\text{reference}, \text{field}_1.@, \ldots, \text{field}_n.@ \rightarrow \text{selector}$$

- All hierachical key constraints are XKFDs.
- Path Expressions instead of simple paths

Hierachical Key Constraints

- Introduced by Buneman et al. 2001
- Used for XML Schema Integrity Constraints

Syntax



reference,field$_1$.@,...,field$_n$.@ $\rightarrow$ selector

- All hierachical key constraints are XKFDs.
- Path Expressions instead of simple paths
- Can be expressed with tree patterns if Linear XPath is used.

Contents

Definition

→

a	b	c

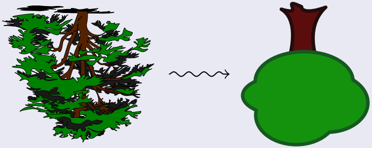
Existing Work

- XFDs
- Hierachical Constraints
- XML Schema Constraints

Reasoning

- Model-Checking
- Implication Problem

Tree Chase



Algorithmic Problems

Model Checking Problem: MC($\mathcal{M}, \mathcal{C}$)

Given: XML tree t,
XML-to-rel constraint $(m, \sigma) \in \mathcal{M} \times \mathcal{C}$

Question: Does $t \models (m, \sigma)$ hold?

Algorithmic Problems

Model Checking Problem: MC($\mathcal{M}, \mathcal{C}$)

Given: XML tree t,
XML-to-rel constraint $(m, \sigma) \in \mathcal{M} \times \mathcal{C}$

Question: Does $t \models (m, \sigma)$ hold?

Implication Problem: IMP($\mathcal{M}, \mathcal{C}, \mathcal{S}$)

Given: XML-to-rel. constaints
$(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$ and $(m, \sigma) \in \mathcal{M} \times \mathcal{C}$,
schema $S \in \mathcal{S}$ (optional)

Question: Does $(m_1, \sigma_1), \ldots, (m_n, \sigma_n) \models_S (m, \sigma)$ hold?

Algorithmic Problems

Model Checking Problem: $MC(\mathcal{M}, \mathcal{C})$

Given: XML tree t,
XML-to-rel constraint $(m, \sigma) \in \mathcal{M} \times \mathcal{C}$

Question: Does $t \models (m, \sigma)$ hold?

Implication Problem: $IMP(\mathcal{M}, \mathcal{C}, \mathcal{S})$

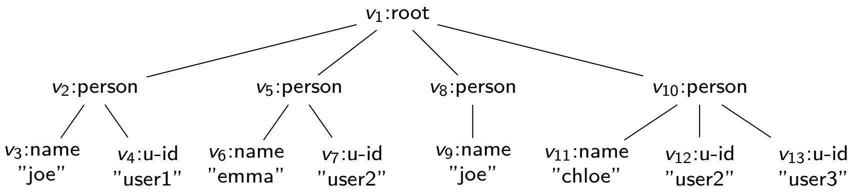
Given: XML-to-rel. constaints
$(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$ and $(m, \sigma) \in \mathcal{M} \times \mathcal{C}$,
schema $S \in \mathcal{S}$ (optional)

Question: Does $(m_1, \sigma_1), \ldots, (m_n, \sigma_n) \models_S (m, \sigma)$ hold?

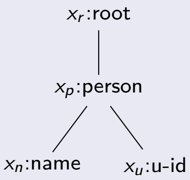
Theorem

MC(TP[/, //, $*$], FD) can be solved in PTIME

Model Checking



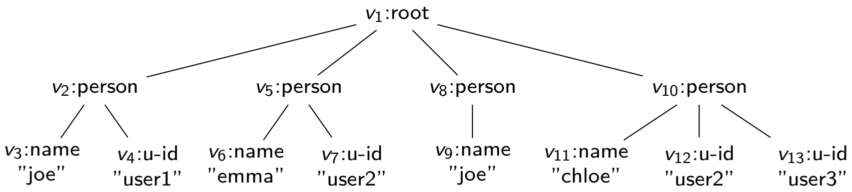
Pattern/Mapping



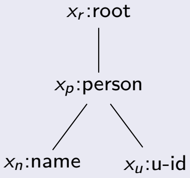
Functional Dependency

- U-IDs are unique: $x_u.@ \to x_u$

Model Checking



Pattern/Mapping



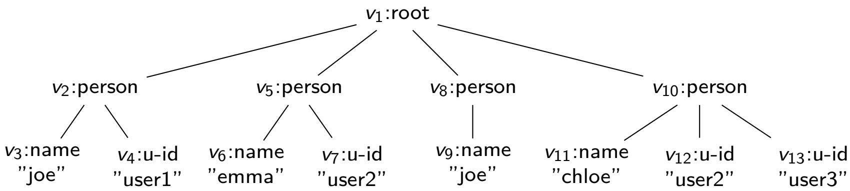
Functional Dependency

- U-IDs are unique: $x_u.@ \rightarrow x_u$

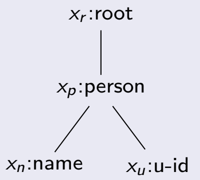
Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	joe	v_4	user1
v_1	v_5	v_6	emma	v_7	user2
v_1	v_{10}	v_{11}	chloe	v_{12}	user2
v_1	v_{10}	v_{11}	chloe	v_{13}	user3

Model Checking



Pattern/Mapping



Functional Dependency

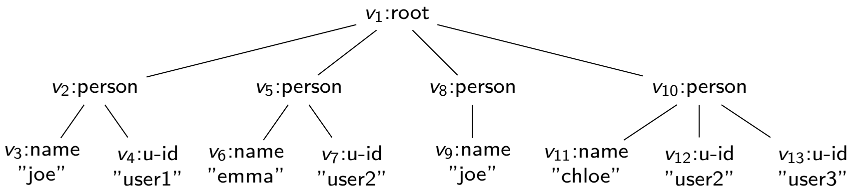
- U-IDs are unique: $x_u.@ \rightarrow x_u$

Relation

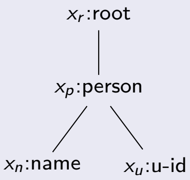
x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	joe	v_4	user1
v_1	v_5	v_6	emma	v_7	user2
v_1	v_{10}	v_{11}	chloe	v_{12}	user2
v_1	v_{10}	v_{11}	chloe	v_{13}	user3

Too Large

Model Checking



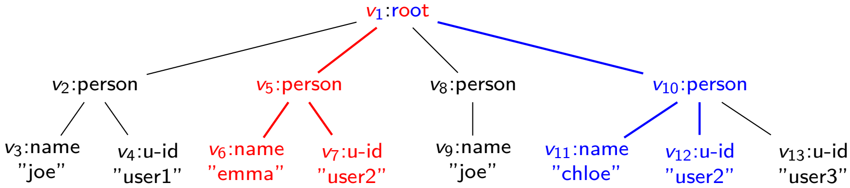
Pattern/Mapping



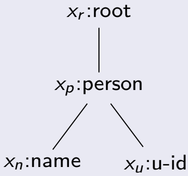
Functional Dependency

- U-IDs are unique: $x_u.@ \rightarrow x_u$

Model Checking



Pattern/Mapping

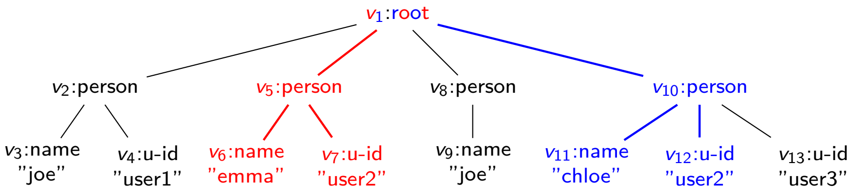


Functional Dependency

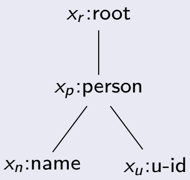
- U-IDs are unique: $x_u.@\rightarrow x_u$

- Compute Witness Pairs using dynamic programming

Model Checking



Pattern/Mapping



Functional Dependency

- U-IDs are unique: $x_u.@ \rightarrow x_u$

- Compute Witness Pairs using dynamic programming

Complexity

Combined Complexity: $\in$ DTIME($|t|^2|p|$)
Data Complexity: LOGSPACE

The Implication Problem

Problem: IMP($\mathcal{M}, \mathcal{C}, \mathcal{S}$)

Given: XML-to-rel. constaints
$(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$ and $(m, \sigma) \in \mathcal{M} \times \mathcal{C}$,
schema $S \in \mathcal{S}$ (optional)

Question: Does $(m_1, \sigma_1), \ldots, (m_n, \sigma_n) \models_S (m, \sigma)$ hold?

The Implication Problem

Problem: IMP($\mathcal{M}, \mathcal{C}, \mathcal{S}$)

Given: XML-to-rel. constaints
$(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$ and $(m, \sigma) \in \mathcal{M} \times \mathcal{C}$,
schema $S \in \mathcal{S}$ (optional)

Question: Does $(m_1, \sigma_1), \ldots, (m_n, \sigma_n) \models_S (m, \sigma)$ hold?

Theorem

IMP(TP$[/, //, *]$, FD, finite alphabet) is undecidable.

The Implication Problem

Problem: IMP($\mathcal{M}, \mathcal{C}, \mathcal{S}$)

Given: XML-to-rel. constaints
$(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$ and $(m, \sigma) \in \mathcal{M} \times \mathcal{C}$,
schema $S \in \mathcal{S}$ (optional)

Question: Does $(m_1, \sigma_1), \ldots, (m_n, \sigma_n) \models_S (m, \sigma)$ hold?

Theorem

IMP(TP$[/, //, *]$, FD, finite alphabet) is undecidable.

Theorem

IMP(MSO, XKFD, MSO) is decidable.

MSO formulas have no access to data values!

The Implication Problem

Theorem

IMP(MSO, XKFD, MSO) is decidable.

The Implication Problem

Theorem

IMP(MSO, XKFD, MSO) is decidable.

Proof

The Implication Problem

Theorem

IMP(MSO, XKFD, MSO) is decidable.

Proof

- A counterexample to an XKFD needs at most linearly many important data values, which occur more than once.

The Implication Problem

Theorem

IMP(MSO, XKFD, MSO) is decidable.

Proof

- A counterexample to an XKFD needs at most linearly many important data values, which occur more than once.
- Encode important data values in the labels.

The Implication Problem

Theorem

IMP(MSO, XKFD, MSO) is decidable.

Proof

- A counterexample to an XKFD needs at most linearly many important data values, which occur more than once.
- Encode important data values in the labels.
- MSO (without data values) is decidable on trees.

Results

Results

	TP[/]	TP[/, ∗]	TP[/, //]	TP[/, //, ∗]
FD	$\in$ P	$\in$ P	co-NP	co-NP-hard

Results

Results

	TP[/]	TP[/, ∗]	TP[/, //]	TP[/, //, ∗]
XKFD	$\in$ P	$\in$ P	co-NP	co-NP
FD	$\in$ P	$\in$ P	co-NP	co-NP-hard

Results

Results

	$TP[/]$	$TP[/,*]$	$TP[/,//]$	$TP[/,//,*]$
XKFD	$\in$ P	$\in$ P	co-NP	co-NP
FD	$\in$ P	$\in$ P	co-NP	co-NP-hard
XKFD + Schema	$\in$ P	co-NP	co-NP	PSPACE
FD + Schema	$\in$ P	co-NP-hard $\in$ EXPTIME	co-NP-hard	undecidable

Schema means:

- finite alphabet for lower bounds
- simple DTDs for upper bounds

Results

Results

	TP[/]	TP[/, *]	TP[/, //]	TP[/, //, *]
XKFD	$\in$ P	$\in$ P	co-NP	co-NP
FD	$\in$ P	$\in$ P	co-NP	co-NP-hard
XKFD + Schema	$\in$ P	co-NP	co-NP	PSPACE
FD + Schema	$\in$ P	co-NP-hard $\in$ EXPTIME	co-NP-hard	undecidable

Schema means:

- finite alphabet for lower bounds
- simple DTDs for upper bounds

Upper Bounds:

- chase algorithm
- small (encodings of) counter examples

Results

Results

	TP[/]	TP[/, *]	TP[/, //]	TP[/, //, *]
XKFD	$\in$ P	$\in$ P	co-NP	co-NP
FD	$\in$ P	$\in$ P	co-NP	co-NP-hard
XKFD + Schema	$\in$ P	co-NP	co-NP	PSPACE
FD + Schema	$\in$ P	co-NP-hard $\in$ EXPTIME	co-NP-hard	undecidable

Schema means:

- finite alphabet for lower bounds
- simple DTDs for upper bounds

Upper Bounds:

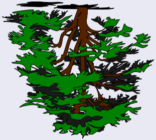
- chase algorithm
- small (encodings of) counter examples

Lower Bounds:

- reductions

Contents

Definition



$\longrightarrow$

a	b	c

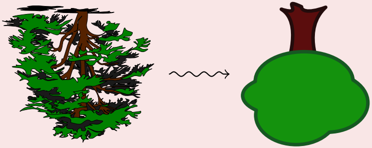
Existing Work

- XFDs
- Hierachical Constraints
- XML Schema Constraints

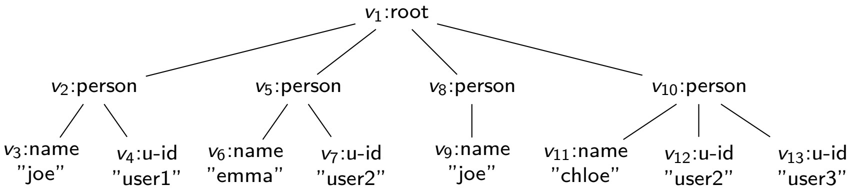
Reasoning

- Model-Checking
- Implication Problem

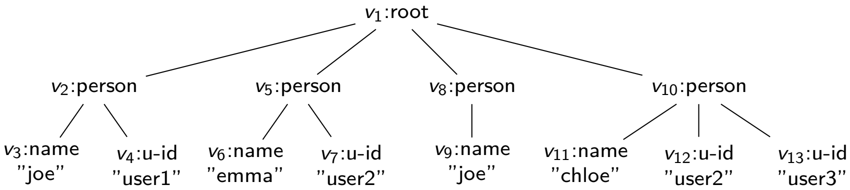
Tree Chase



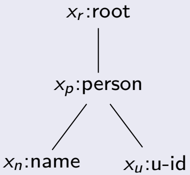
Our Tree Chase



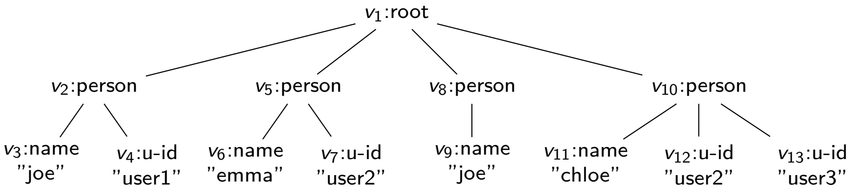
Our Tree Chase



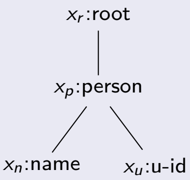
Pattern/Mapping



Our Tree Chase



Pattern/Mapping



Implication Instance

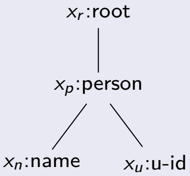
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma$?

Our Tree Chase

Pattern/Mapping



Implication Instance

$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

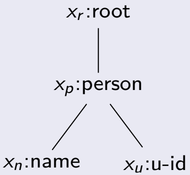
$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma?$

Our Tree Chase

Relation

Pattern/Mapping



Implication Instance

$\Sigma = \{x_u.@ \to x_u, x_p \to x_n\}$

$\sigma = x_u.@ \to x_n.@$

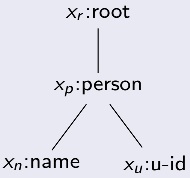
$\Sigma \models \sigma$?

Our Tree Chase

Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$

Pattern/Mapping



Implication Instance

$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

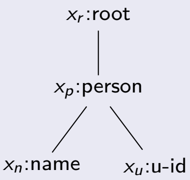
$\Sigma \models \sigma$?

Our Tree Chase

Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_5	v_6	v_7	3	v_8	2

Pattern/Mapping



Implication Instance

$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma$?

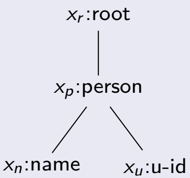
Our Tree Chase

Tree

Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_5	v_6	v_7	3	v_8	2

Pattern/Mapping



Implication Instance

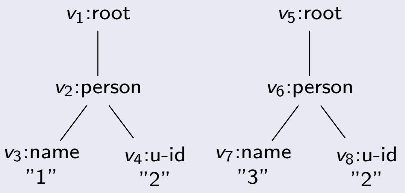
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma$?

Our Tree Chase

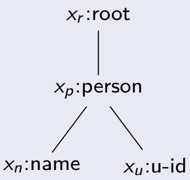
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_5	v_6	v_7	3	v_8	2

Pattern/Mapping



Implication Instance

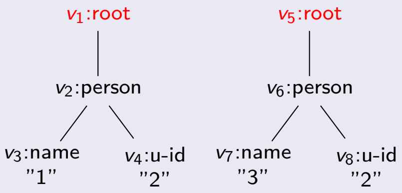
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma$?

Our Tree Chase

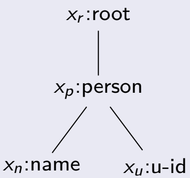
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_5	v_6	v_7	3	v_8	2

Pattern/Mapping



Implication Instance

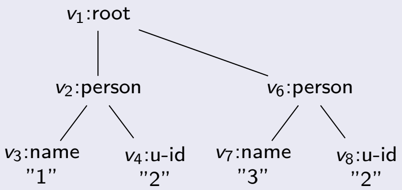
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma$?

Our Tree Chase

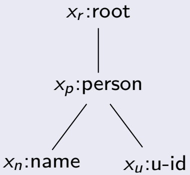
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_1	v_6	v_7	3	v_8	2

Pattern/Mapping



Implication Instance

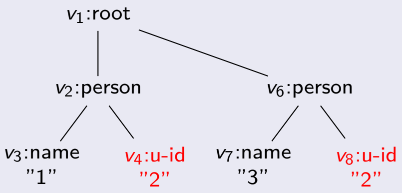
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma?$

Our Tree Chase

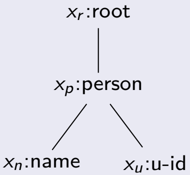
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_1	v_6	v_7	3	v_8	2

Pattern/Mapping



Implication Instance

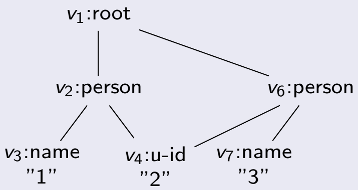
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma?$

Our Tree Chase

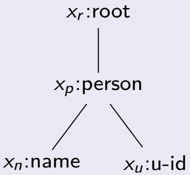
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_1	v_6	v_7	3	v_4	2

Pattern/Mapping



Implication Instance

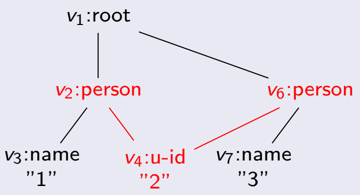
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma?$

Our Tree Chase

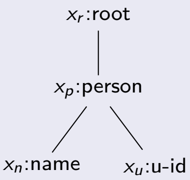
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_1	v_6	v_7	3	v_4	2

Pattern/Mapping



Implication Instance

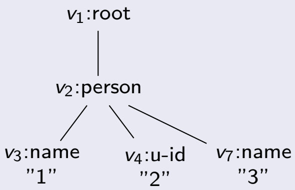
$$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$$

$$\sigma = x_u.@ \rightarrow x_n.@$$

$$\Sigma \models \sigma?$$

Our Tree Chase

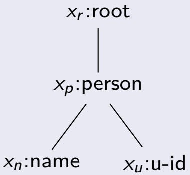
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_1	v_2	v_7	3	v_4	2

Pattern/Mapping



Implication Instance

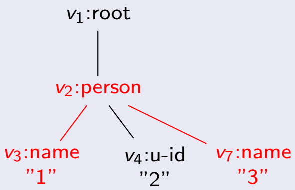
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma?$

Our Tree Chase

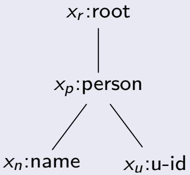
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_1	v_2	v_7	3	v_4	2

Pattern/Mapping



Implication Instance

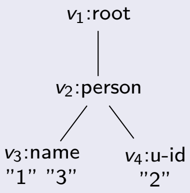
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma?$

Our Tree Chase

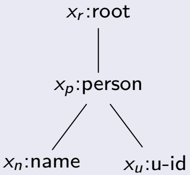
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_1	v_2	v_3	3	v_4	2

Pattern/Mapping



Implication Instance

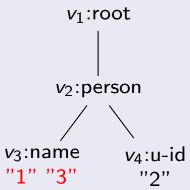
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma?$

Our Tree Chase

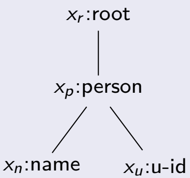
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_1	v_2	v_3	3	v_4	2

Pattern/Mapping



Implication Instance

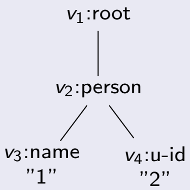
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma$?

Our Tree Chase

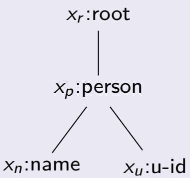
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2
v_1	v_2	v_3	1	v_4	2

Pattern/Mapping



Implication Instance

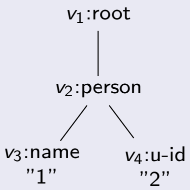
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma?$

Our Tree Chase

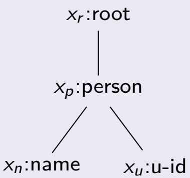
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2

Pattern/Mapping



Implication Instance

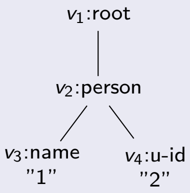
$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma?$

Our Tree Chase

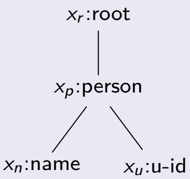
Tree



Relation

x_r	x_p	x_n	$x_n.@$	x_u	$x_u.@$
v_1	v_2	v_3	1	v_4	2

Pattern/Mapping



Implication Instance

$\Sigma = \{x_u.@ \rightarrow x_u, x_p \rightarrow x_n\}$

$\sigma = x_u.@ \rightarrow x_n.@$

$\Sigma \models \sigma$? ✓

Results & Further Work

Results & Further Work

- Defined a general framework for XML Constraints

Results & Further Work

- Defined a general framework for XML Constraints
- Showed how existing work integrates into the framework

Results & Further Work

- Defined a general framework for XML Constraints
- Showed how existing work integrates into the framework

Complexity Results for Implication

	TP[/]	TP[/, *]	TP[/, //]	TP[/, //, *]
XKFD	$\in$ P	$\in$ P	co-NP	co-NP
FD		$\in$ P	co-NP	co-NP-hard
XKFD + Schema		co-NP	co-NP	PSPACE
FD + Schema		co-NP-hard $\in$ EXPTIME	co-NP-hard	undecidable

Results & Further Work

- Defined a general framework for XML Constraints
- Showed how existing work integrates into the framework

Complexity Results for Implication

	TP[/]	TP[/, *]	TP[/, //]	TP[/, //, *]
XKFD	$\in$ P	$\in$ P	co-NP	co-NP
FD		$\in$ P	co-NP	co-NP-hard
XKFD + Schema		co-NP	co-NP	PSPACE
FD + Schema		co-NP-hard $\in$ EXPTIME	co-NP-hard	undecidable

Further Work

- Solve missing complexities
- Incomplete Data
- More general schemas
- Connections to XML Data Exchange