An Analysis of Approaches to XML Schema Inference

Irena Mlynkova

irena.mlynkova@mff.cuni.cz

Charles University
Faculty of Mathematics and Physics
Department of Software Engineering
Prague, Czech Republic

Overview

1. Introduction
2. Existing approaches
3. Open issues
4. Conclusion

Introduction

- XML = a standard for data representation and manipulation
- XML documents + XML schema
  - Allowed data structure
  - W3C recommendations: DTD, XML Schema (XSD)
  - ISO standards: RELAX NG, Schematron, …
- Why schema?
  - Known structure, valid data, limited complexity of processing, …

    $\Rightarrow$ Optimization of XML processing
    - Storing, querying, updating, compressing, …

Real-World XML Schemas

- Statistical analyses of real-word XML data:
  - 52% of randomly crawled / 7.4% of semi-automatically collected documents: no schema
  - 0.09% of randomly crawled / 38% of semi-automatically collected documents with schema: use XSD
  - 85% of randomly crawled XSDs: equivalent to DTDs
- Problem:
  - Users do not use schemas at all
    - Extreme opinion: I do not want to follow the rules of an XML schema in my XML data.
  - Schema = a kind of documentation
    - Documents are not valid, schemas are not correct

Mlynkova, Toman, Pokorny: Statistical Analysis of Real XML Data Collections.
In COMAD '06, pages 20 – 31, New Delhi, India, 2006. Tata McGraw-Hill Publishing Co. Ltd.

Inference of XML Schemas

- **Solution:**
  - **Automatic inference of XML schema S_D for a given set of documents *D***

  ⇒ Multiple solutions
    - **Too general = accepts too many documents**
    - **Too restrictive = accepts only *D***
- **Advantages:**
  - **S_D = a good initial draft for user-specified schema**
  - **S_D = a reasonable representative when no schema is available**
  - **User-defined XML schemas are too general (*, +, recursion, …) ⇒ S_D can be more precise**

XML Schemas and Grammars

An <u>extended context-free grammar</u> is quadruple G = (N,T,P,S), where N and T are finite sets of nonterminals and terminals, P is a finite set of productions and S is a non terminal called a start symbol. Each production is of the form $A \rightarrow \alpha$, where $A \in N$ and α is a regular expression over alphabet $N \cup T$.

Given the alphabet Σ, a <u>regular expression</u> (RE) over Σ is inductively defined as follows:

- ∅ (empty set) and ε (empty string) are REs
- $\forall\, a \in \Sigma$: a is a RE
- If r and s are REs over Σ, then (rs) (concatenation), (r|s) (alternation) and (r*) (Kleene closure) are REs

- DTD adds: (s|ε) = (s?), (s s*) = (s+), concatenation = ','
- XML Schema adds: unordered sequence

Overview

1. Introduction
2. **Existing approaches**
3. Open issues
4. Conclusion

Classification of Approaches

- Type of the result (DTD vs. XSD)
  - DTDs are most common
    - Some works infer XSDs, but with expressive power of DTD
  - Key aim: Inference of REs (content models)
- The way we construct the result
  - Heuristic = no theoretic basis
    - Generalization of a trivial schema
    - Rules: "If there are > 3 occurrences of E, it can occur arbitrary times" ⇒ E* or E+
  - Inferring a grammar = inference of a set of regular expressions
    - Gold's theorem: Regular languages are not identifiable in the limit only from positive examples (valid XML documents)

  ⇒ Inference of subclasses of regular languages

Classical Steps

1. Derivation of initial grammar (IG)
   - For each element E and its subelements $E_1, E_2, \ldots, E_n$ we create production $E \rightarrow E_1 E_2 \ldots E_n$
2. Clustering of rules of IG
   - According to element names vs. broader context
3. Construction of prefix tree automaton (PTA) for each cluster
4. Generalization of PTAs
   - Merging state algorithms
5. Inference of simple data types and integrity constraints
   - Often ignored
6. Refactorization
   - Correction and simplification of the derived REs
7. Expressing the inferred REs in target XML schema language
   - Most common: Direct rewriting of REs to content models

Step 1: Initial Grammar

```
...
<person id="123">
  <name>
    <first>Irena</first>
    <surname>Mlynkova</surname>
  </name>
  <email>irena.mlynkova@gmail.com</email>
  <email>irena.mlynkova@mff.cuni.cz</email>
</person>
<person id="456" holiday="yes">
  <name>
    <surname>Necasky</surname>
    <first>Martin</first>
  </name>
  <phone>123-456-789</phone>
  <email>martin.necasky@mff.cuni.cz</email>
</person>
...
```

person → name email email
name → first surname
first → PCDATA
surname → PCDATA
email → PCDATA
email → PCDATA
person → name phone email
name → surname first
surname → PCDATA
first → PCDATA
phone → PCDATA
email → PCDATA

```
<book>
  <name>Sherlock Holmes</name>
</book>
```

```
<author>
  <name>
    <first>Arthur</first>
    <middle>Conan</middle>
    <last>Doyle</last>
  </name>
</author>
```

person → name email email
person → name phone email

name → first surname
name → surname first

first → PCDATA

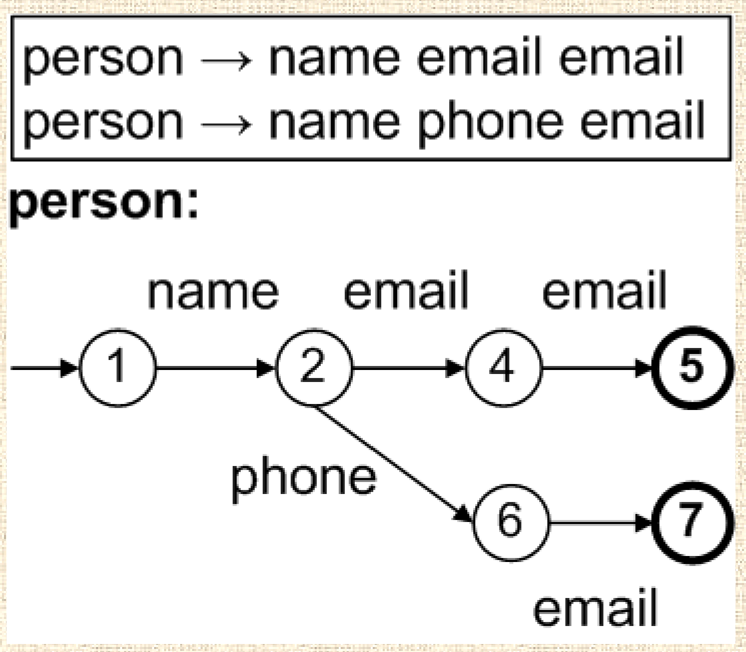
surname → PCDATA

email → PCDATA

phone → PCDATA

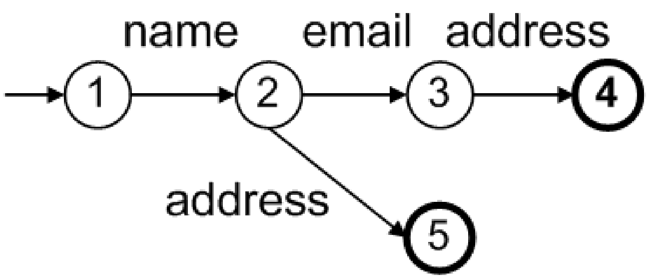
Step 2: Clustering

Step 3: Construction of PTA

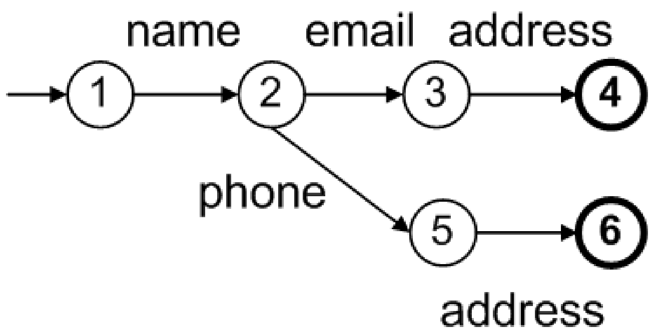


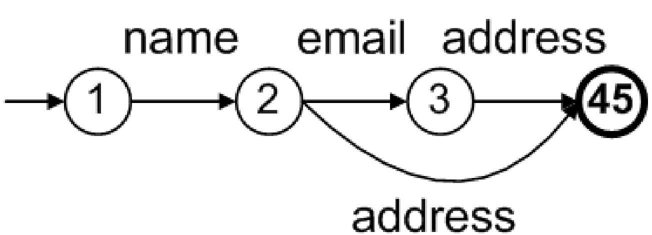
Step 4. PTA Generalization

person → name email address
person → name address

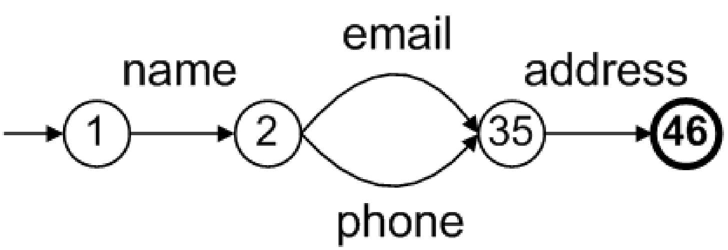


person → name email address
person → name phone address





person → name email? address



person → name (email | phone) address

Heuristic Approaches

- Various generalization rules
  - Observations of real-world data, common prefixes, suffixes, …
- Generalization process
  - Generalize IG until a satisfactory solution is reached
    - Problem: wrong step
  - Generate a set of candidates and choose the optimal one
    - Problem: space overhead
- How to generalize
  - Until any rule can be applied
  - Until a better schema can be found
    - Problems:
      - Evaluation of quality of schemas (MDL principle)
      - Efficient search strategy (greedy search vs. ACO heuristics)

Conciseness = bits required to describe schema

Preciseness = bits required for description of input data using schema

Approaches Inferring a Grammar

- Common idea: regular languages are not identifiable in the limit from positive examples

  $\Rightarrow$ inferring a subclass that can be
- Difference: The selected class of languages
  - k-contextual, (k,h)-contextual = having a limited context
  - f-distinguishable = having a distinguishing function
  - single-occurrence REs, chain REs, k-local single-occurrence = simple types of REs occurring in real-world XML schemas
- Approaches: Merging state algorithms
  - Merging criteria are given by the language class directly
- Note: Necessary requirement of W3C = 1-unambiguity
  - Deterministic content models
  - Example: (A,B) | (A,C) vs. A, (B | C)
  - Often ignored

Overview

1. Introduction
2. Existing approaches
3. **Open issues**
4. Conclusion

1. User Interaction

- Existing approaches: Automatic inference of an XML schema
- Problem: How to find the optimal generalization?
  - MDL principle: Good schema = tightly represents data, concise, compact
  - User's preferences can be different ⇒ resulting schema may be unnatural
- Bex et al. (VLDB'06, VLDB'07): Let us infer only schema constructs that occur in real-world XML data
- Natural improvement: user interaction
  - Refining the clustering, preferred merging, preferred schema constructs, refining the REs, …
- Problem:
  - A user may not be skilled in specifying complex REs
  - A user is not able to make too many decisions

2. Other Input Information

- Input in existing works: a set of positive examples
- Problem: Gold's theorem

⇒ Question: Are there any other ways?

Input 1: An obsolete XML schema

- Typical situation: a user creates an XML schema ⇒ updates only the data ⇒ schema is obsolete
- Idea: The schema contains partially correct information
- Note: XML schema evolution = opposite problem

Input 2: XML queries

- Idea: partial information on the structure

Input 3 - … : Negative examples, user requirements, statistical analysis of XML documents, …

Mlynkova: On Inference of XML Schema with the Knowledge of an Obsolete One.
In ADC'09 (to appear), volume 92, Wellington, New Zealand, 2009. ACS.

Necasky, Mlynkova: Enhancing XML Schema Inference with Keys and Foreign Keys.
In SAC'09 (to appear), Honolulu, Hawaii, USA, 2009. ACM.

3. XML Schema Simple Data Types

- Advantage of XML Schema: wide support of simple data types
  - 44 built-in data types
  - User-defined data types derived from existing simple types
- Natural improvement: precise inference of simple data types
- Current approaches:
  - Omit simple data types at all
  - Two exceptions: selected built-in data types
- Do we need simple data types?
  - Inferring within an XML editor: yes
  - Inferring for optimization purposes: not always necessary
    - Schema-driven XML-to-relational mapping methods
- Ideas: exploitation of additional information
  - Queries, semantics of element names, obsolete schema, …

4. XML Schema Advanced Constructs

- Advantage of XML Schema: object-oriented features
  - User-defined data types, inheritance, substitutability of both data types and elements, …
- Disadvantage: Do not extend the expressive power
  - "syntactic sugar"
- Advantages:
  - More user-friendly and realistic schemas
  - Can carry more precise information for optimization
    - Inheritance, shared globally defined items, …
- Problem: constructs are equivalent $\Rightarrow$ how to find the optimal expression?
  - User-interaction
  - Additional information

Vosta, Mlynkova, Pokorny. Even an Ant Can Create an XSD.
In DASFAA'08, LNCS 4947, pages 35–50. New Delhi, India, 2008. Springer-Verlag.

Mlynkova, Necasky: Towards Inference of More Realistic XSDs.
In SAC'09 (to appear), Honolulu, Hawaii, USA, 2009. ACM.

5. Integrity Constraints (ICs)

- DTD: ID, IDREF, IDREFS = keys and foreign keys
- XML Schema:
  - ID, IDREF, IDREFS
  - unique, key, keyref
    - More precise expression of keys and foreign keys + uniqueness
  - assert, report
    - Special constraints expressed using XPath
- More powerful ICs: Cannot be expressed in XML Schema but can be inferred
- Aim of ICs
  - Optimization of XML processing approaches
- Existing works:
  - Restricted cases of ICs in special situations (applications)
  - No general/universal approach

Necasky, Mlynkova: Enhancing XML Schema Inference with Keys and Foreign Keys.
In SAC'09 (to appear), Honolulu, Hawaii, USA, 2009. ACM.

6. Other Schema Definition Languages

- W3C: DTD, XML Schema
  - Most popular ones
- There are other languages
- RELAX NG
  - Similar strategy as XML Schema and DTD
  - Describes the structure of XML documents using content models
  - Simpler syntax than XSDs, richer set of simple data types than DTD
- Schematron
  - Different strategy
  - Specifies a set of conditions (ICs) the documents must follow
    - Expressed using XPath

⇒ A brand new method
    - A first step towards inference of general ICs

7. XML Data Streams

- Data streams
  - Special type of XML data
  - Recently became popular

⇒ Special processing
  - Parsing, validation, querying, transforming, …
  - Inference of XML schema?

- Features:
  - Cannot be kept in a memory
  - Cannot be read more than once
  - Processing cannot "wait" for the last portion
- The situation is complicated
- No inference method for XML data streams

Overview

1. Introduction
2. Existing approaches
3. Open issues
4. **Conclusion**

Conclusion

- Almost any approach can benefit from XML schemas = knowledge of data structure
- Currently
  - Data-exchange: inferred schema = candidate for further improving
  - Optimization: inferred schema = the only option
    - May be more precise
- Main observations:
  - Basic aspects (inference of REs) are solved
  - Advanced aspects are still waiting for solutions
- Aim of this study:
  - A good starting point for researchers searching a solution or a research topic

Thank you