An Incremental Correction Algorithm for XML Documents and Single Type Tree Grammars

Martin Svoboda, Irena Mlýnková

XML and Web Engineering Research Group

Charles University in Prague

The Czech Republic

24 April 2012

NDT 2012

Dubai, United Arab Emirates

Outline

- Introduction
  - Motivation
  - Objectives
- Approach
  - Corrections
  - Algorithms
  - Experiments
- Conclusion

Introduction

- Motivation
  - **Incorrect XML documents**
    - Well-formedness
    - Schema validity
    - Data consistency
    - …
  - Strategies
    - Adjusting algorithms
    - **Correcting data**

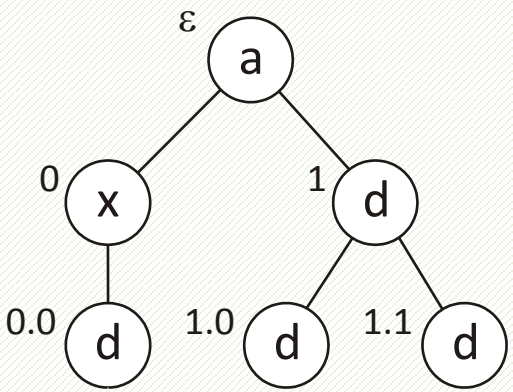
Introduction

- Problem
  - Input
    - **One XML document**
      - Well-formed but (potentially) invalid
    - **DTD or XML Schema**
  - Output
    - **All minimal repairs**
      - Structural corrections of elements

Definitions

- Document
  - **Trees**
    - Nodes for elements and texts
    - Prefix numbering of nodes
  - Example

```
<a>
   <x><d/></x>
   <d><d/><d/></d>
</a>
```



Definitions

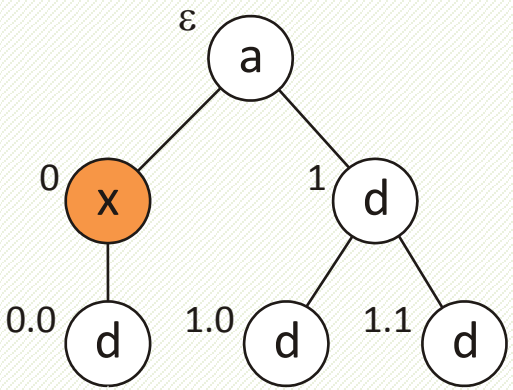
- Schema
  - Grammars
    - **Terminal symbols** for element names
    - **Nonterminal symbols** for types
    - **Production rules** based on regular expressions
  - Classes
    - Regular tree grammars
    - **Single type tree grammars** (XML Schema)
    - **Local tree grammars** (DTD)

Model

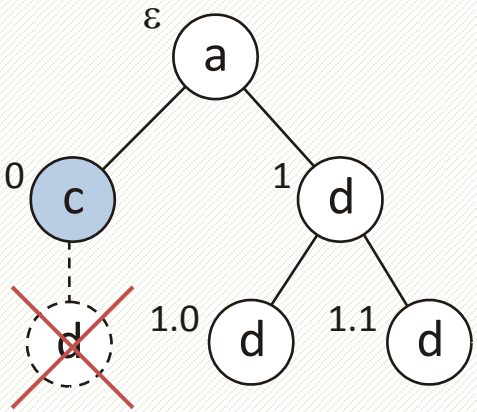
- Edit operations
  - ADD leaf, REMOVE leaf, RENAME label
- **Update operations**
  - Sequences of edit operations
  - **INSERT**, **DELETE**, **REPAIR**, **RENAME**
- Cost function
  - **Unit costs** of edit operations

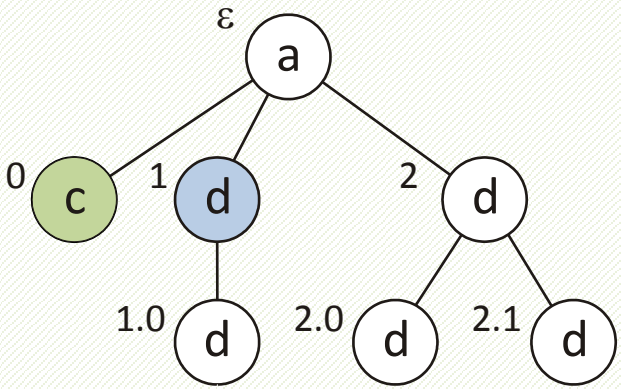
Model

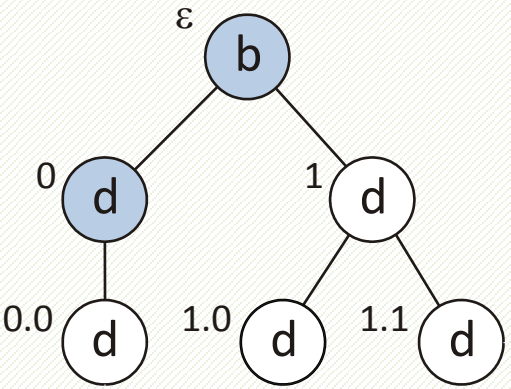
```
<a>
   <x><d/></x>
   <d>
      <d/><d/>
   </d>
</a>
```



Type	Name	Model
A	a	C.D*
B	b	D*
C	c	empty
D	d	D*





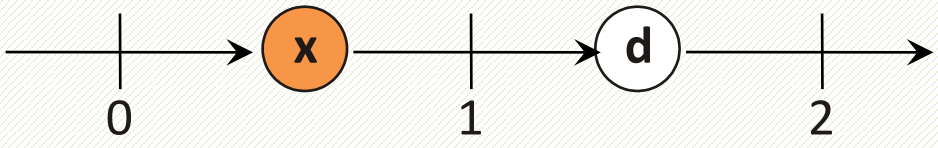


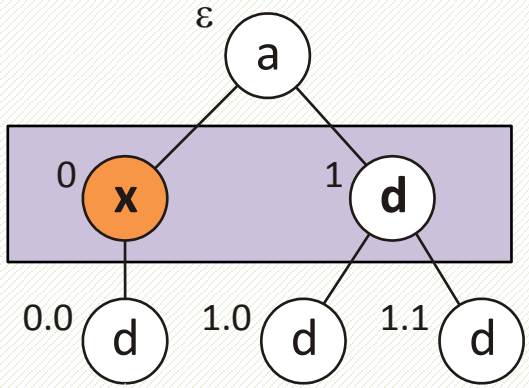
Algorithm

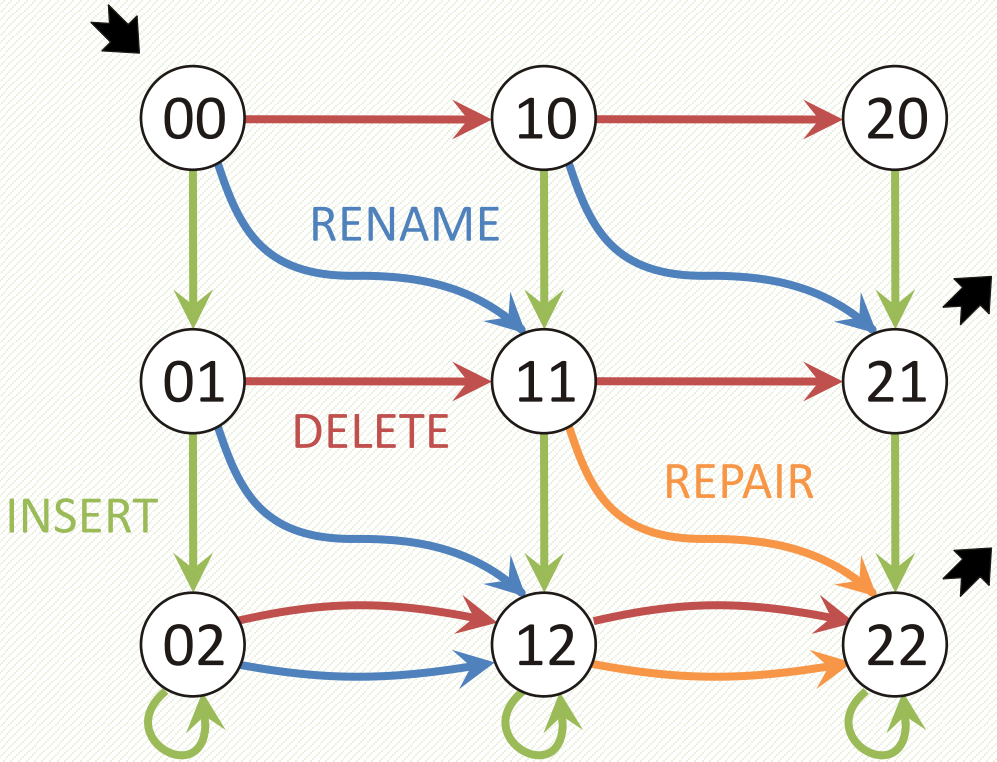
- Naive algorithm
  - Task
    - At each level of top-down tree processing…

      …find repairs for a sequence of sibling nodes
  - Steps
    - Construct a **repairing multigraph**
    - Recursively **repair subtrees**
    - Compose a **repairing structure**

Algorithm

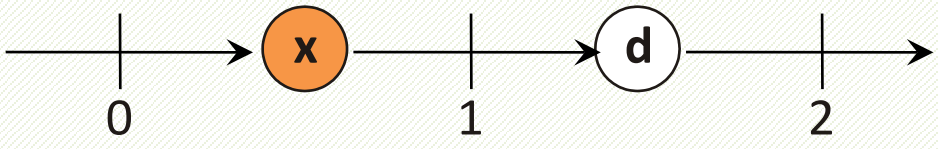


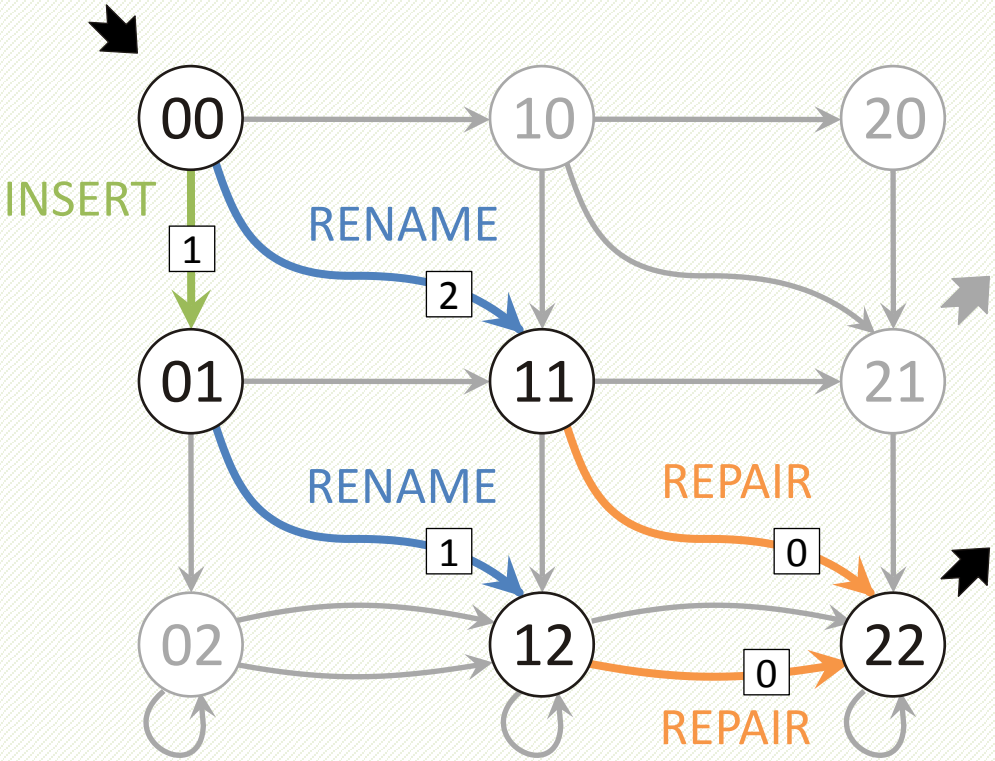


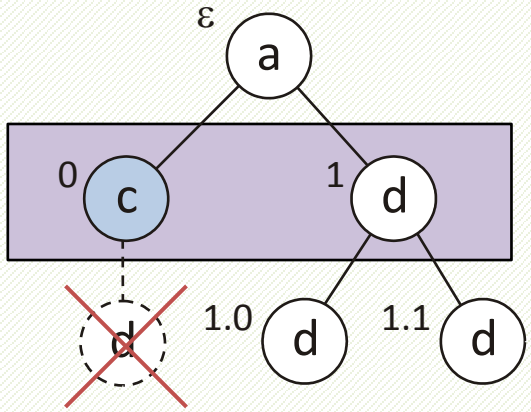


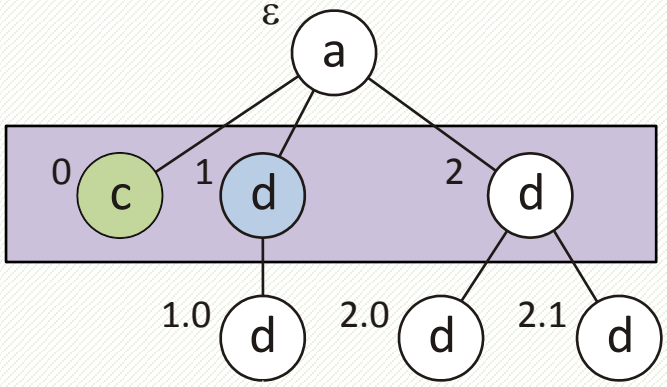
Type	Name	Model
A	**a**	**C.D***
B	b	D*
C	c	*empty*
D	d	D*

Algorithm


x
d
0
1
2


00
10
20
INSERT
1
RENAME
2
01
11
21
RENAME
1
REPAIR
0
02
12
22
0
REPAIR


ε
a
0
c
1
d
d
1.0
d
1.1
d


ε
a
0
c
1
d
2
d
1.0
d
2.0
d
2.1
d

Algorithms

- **Naive**
- **Dynamic**
  - Directly follows Dijkstra's algorithm and, thus, only required multigraph parts are explored
- **Caching**
  - Avoids repeated recursive computations by detecting and **caching identical repairs**
- **Incremental**
  - Evaluates repairing multigraphs step by step

Algorithms

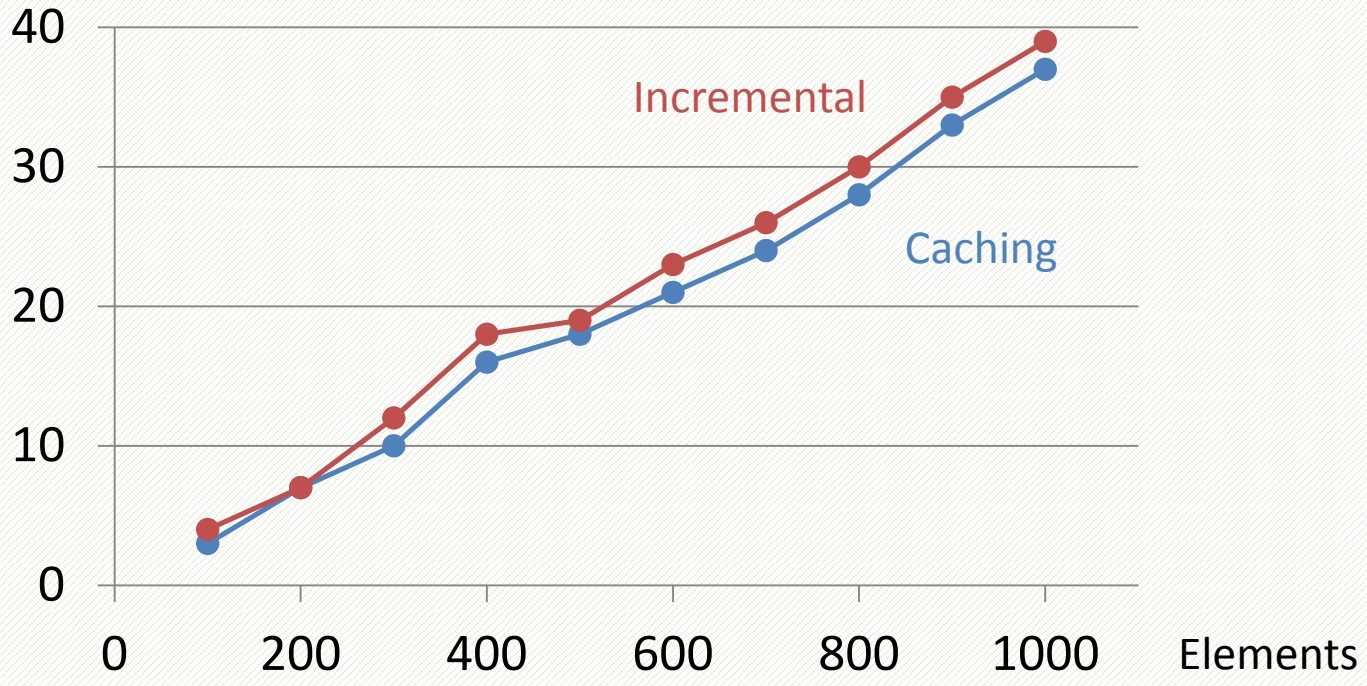
- **Incremental**
  - Task
    - Structure encapsulating multigraph evaluation
      - Multigraph structure
      - Dijkstra's variables
  - Scheduler
    - Processing of an activated task:
      - Request further refinement of perspective edges
      - Activate corresponding tasks for nested problems

Experiments

- Data
  - Single type tree grammar
    - 7 nonterminal symbols
    - 6 terminal symbols
    - Recursion, iteration
  - XML data trees
    - Maximal depth 5, fan-out 8
    - **Elements from 100 to 1,000**
    - 20 files for each particular size
    - Average values from 20 repeats

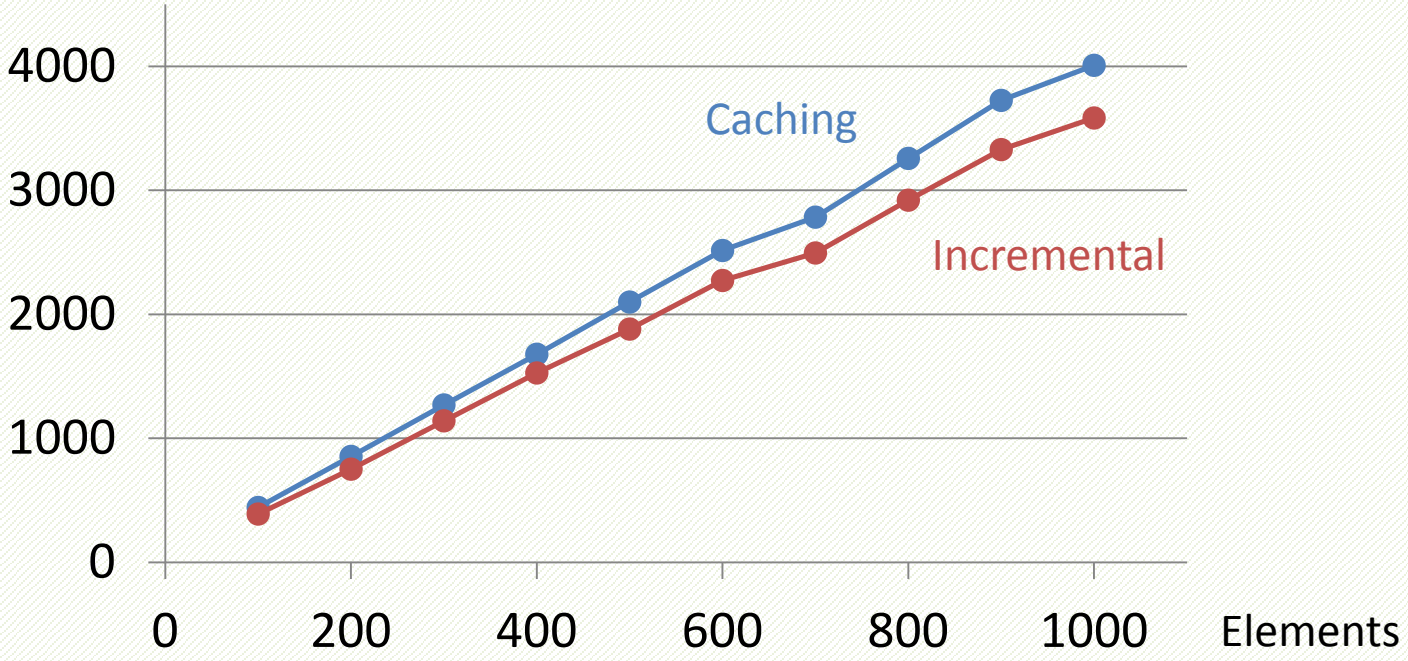
Experiments

- Execution time in miliseconds



Experiments

- Number of correction intents
  - Equals to a number of distinct multigraphs



Conclusion

- Contributions
  - **Single type tree grammars**
  - **Always all minimal repairs**
  - **New incremental algorithm**
- Advantages
  - **Compact repair structure**
  - **Prototype implementation**

Thank you for your attention…

XML and Web Engineering Research Group

Charles University in Prague