Correction of Invalid XML Documents

with Respect to Single Type Tree Grammars

Martin Svoboda

svoboda@ksi.mff.cuni.cz

Charles University in Prague
The Czech Republic

11th July 2011

Outline

- Introduction
- Solution
- Example
- Algorithms
- Experiments
- Conclusion

Introduction

- Problem description
  - Incorrect XML documents
    - Well-formedness, validity, data consistency
  - DTD or XML Schema
    - Different allowed constructs
- Approach objectives
  - One document with its schema
  - Finding repairs of elements

Introduction

- XML documents
  - Trees
    - Element and data nodes
- XML schemata
  - Regular tree grammars
  - Competing nonterminals
    - DTD: local tree grammar class
    - XML Schema: single type tree grammar class

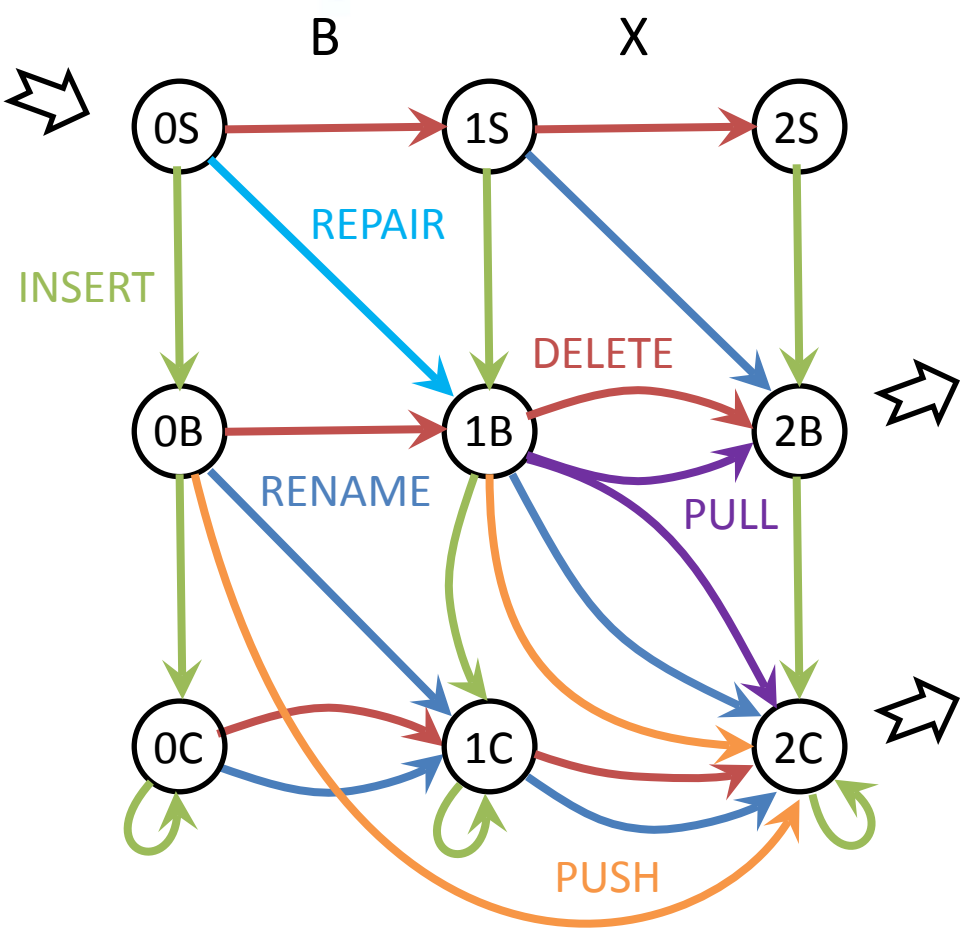
Solution

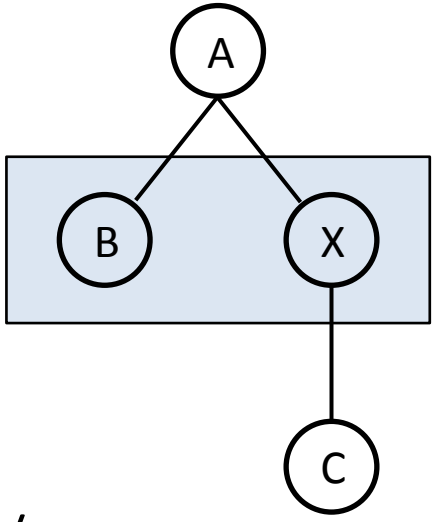
- Edit operations
  - ADD leaf, REMOVE leaf, RENAME label
- Update operations
  - Sequences of edit operations
  - INSERT, DELETE, REPAIR, RENAME, PUSH, PULL
- Cost function
  - Unit costs of edit operations

Solution

- Naive algorithm idea
  - At each level of top-down processing:
    - Correction multigraph
      - Represents all possible corrections
      - Edges correspond to nested problems
    - Recursive nesting
      - Evaluating multigraph edges
    - Minimal repairs
      - Finding all multigraph shortest paths
      - Creating a compact repair structure

Example

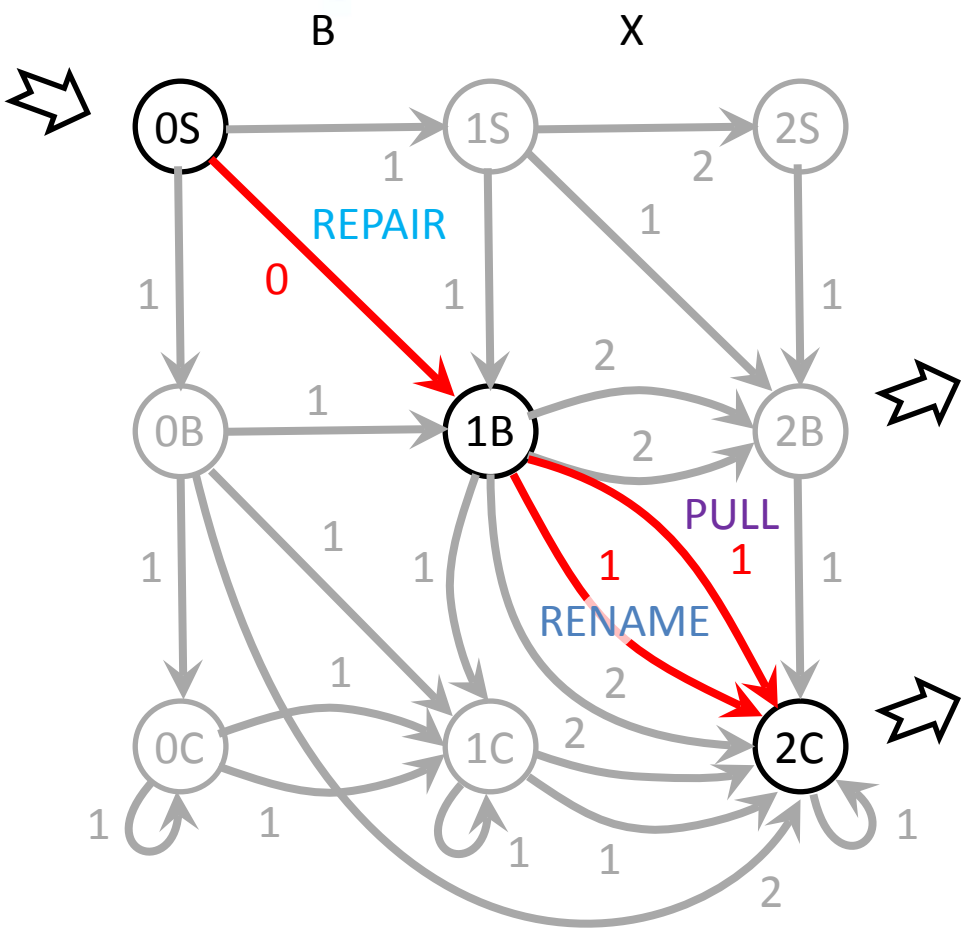


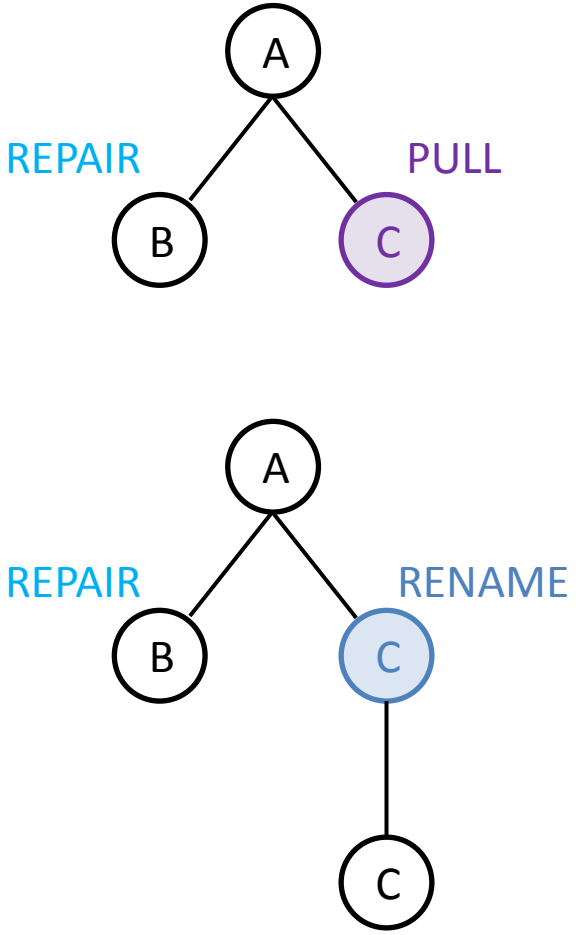



```
<A>
   <B/>
   <X><C/></X>
</A>
```

A: B.C*
B: EMPTY
C: C*

Example


B
X
0S
1S
2S
0B
1B
2B
0C
1C
2C
REPAIR
0
PULL
1
RENAME
1


A
REPAIR
PULL
B
C
A
REPAIR
RENAME
B
C
C

Algorithms

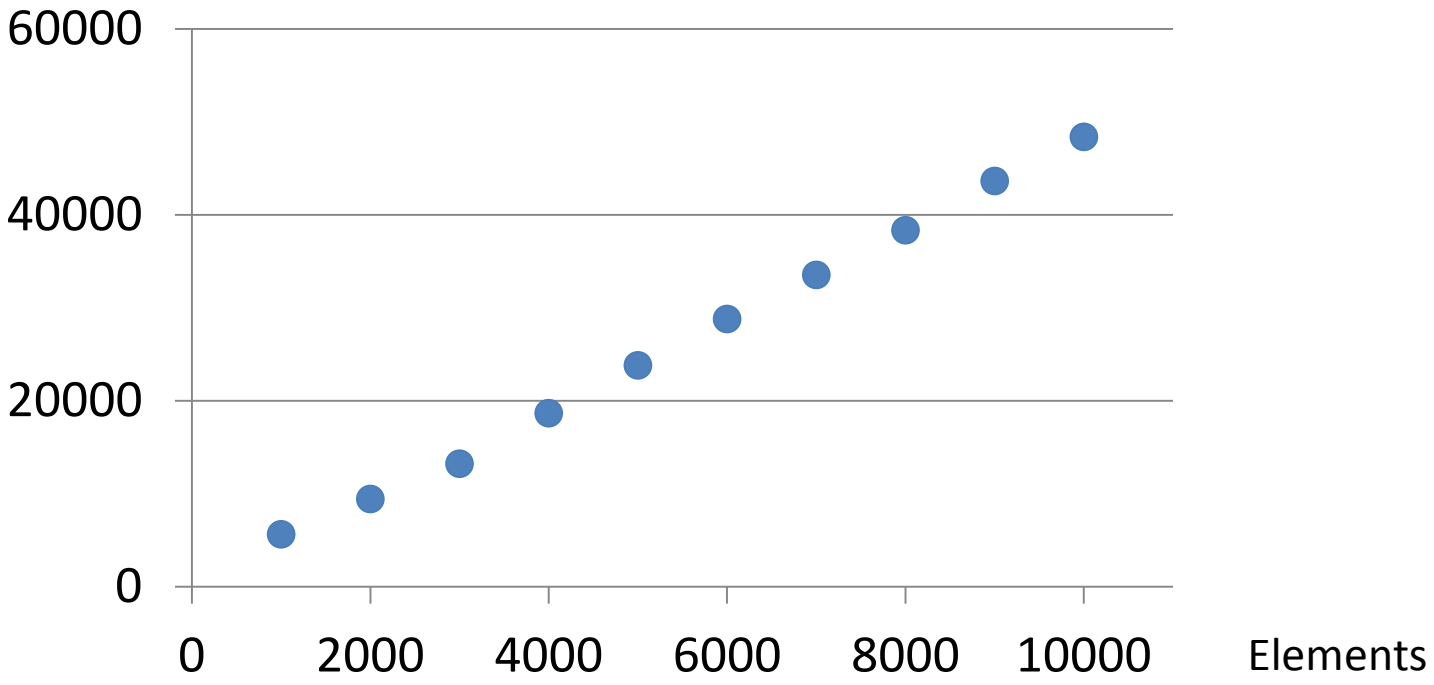
- Naive algorithm
- Dynamic algorithm
  - Incremental multigraph construction
    - Based on Dijkstra algorithm
    - Following only perspective paths
- Caching algorithm
  - Avoiding repeated repair computations
    - Repairs with identical signatures

Experiments

- Data description
  - Single type tree grammar
    - 7 nonterminal and 6 terminal symbols
    - Recursion, iteration
  - XML data trees
    - Depth 4, fan-out 10 to 25
    - Elements 1,000 to 10,000
  - Allowed operations
    - INSERT, DELETE, REPAIR and RENAME

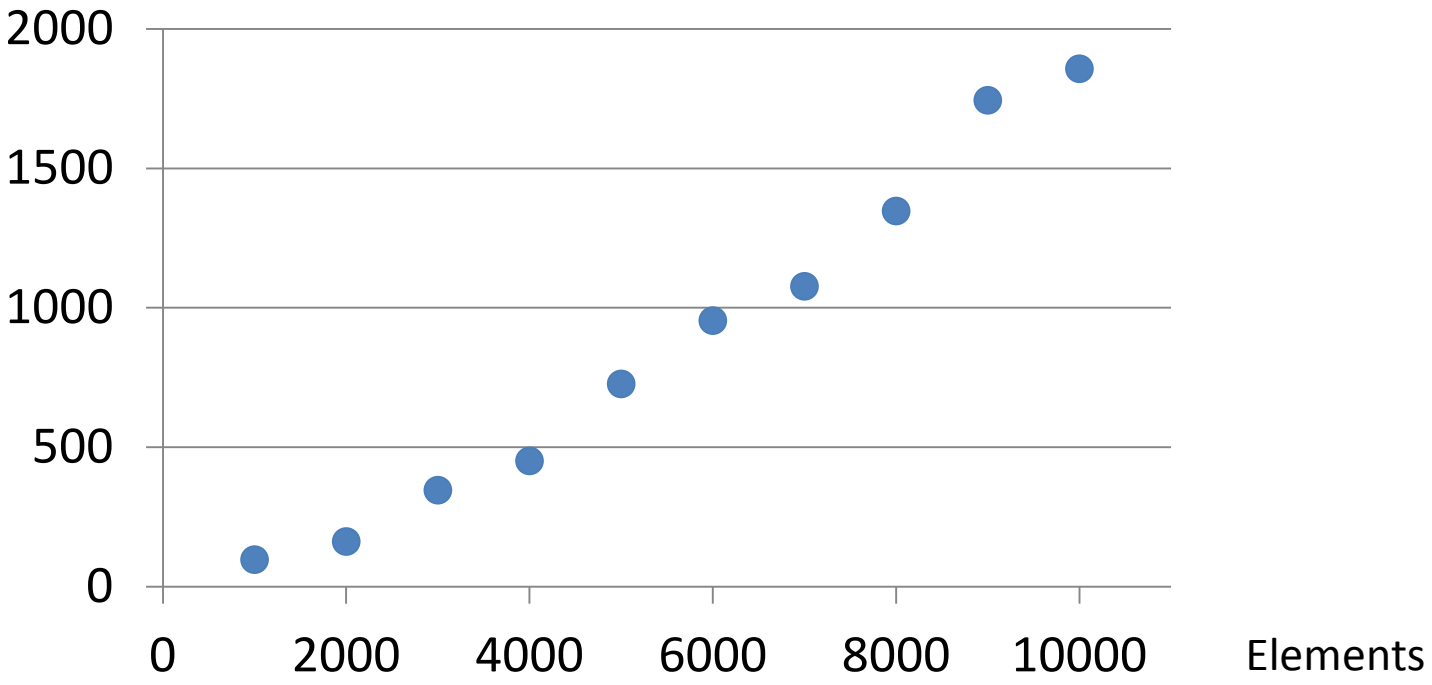
Experiments

- Number of correction intents



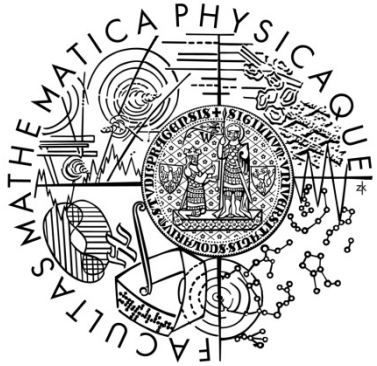
Experiments

- Execution time in miliseconds



Conclusion

- Main contributions
  - Single type tree grammars
  - Always all minimal repairs
  - Efficient algorithm proposal
- Other advantages
  - Extended edit operations
  - Compact repair structure
  - Prototype implementation



Thank you for your attention…