Detecting XML Query-Update Independence

Federico Ulliana

joint work with

Dario Colazzo and Nicole Bidoit

LRI, Université Paris-Sud 11 - INRIA Leo Team

WTS 2010, October 11th, Nancy

XML Query-Update Independence

Informally

Determine whether the result of a query is never affected by an update instruction, over any possible database instance.

Problem Statement

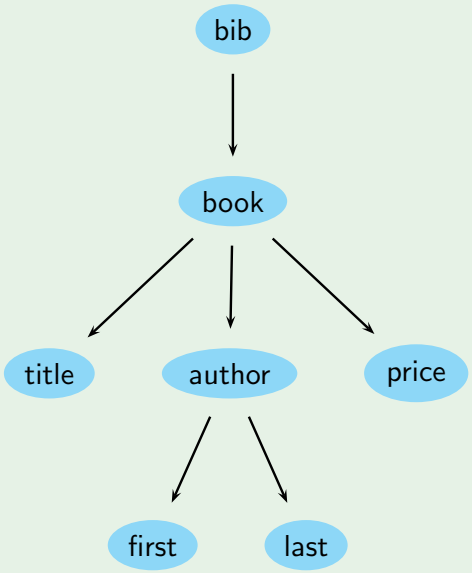
Provided a query $\mathtt{Q} \in XQ$ and an update $\mathtt{U} \in XQUF$, determine whether

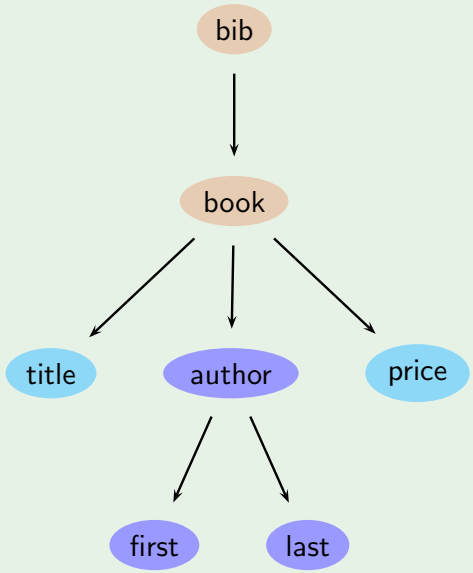
$$\mathtt{Q}(\mathtt{D}) \simeq \mathtt{Q} \circ \mathtt{U}(\mathtt{D})$$

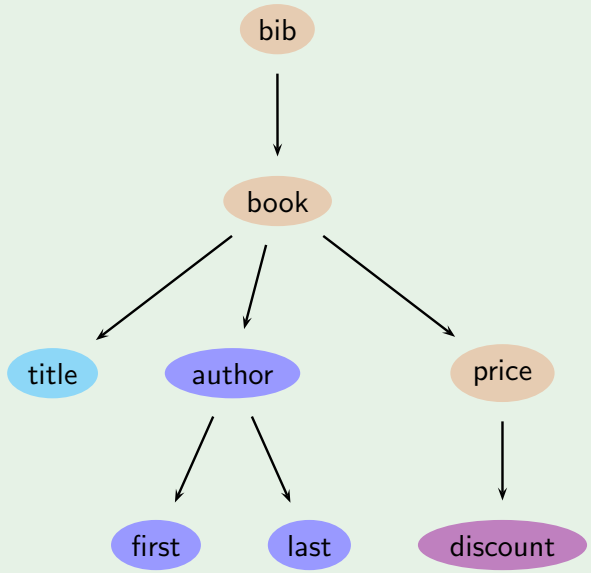
for every XML document D.

Independence Examples

Example





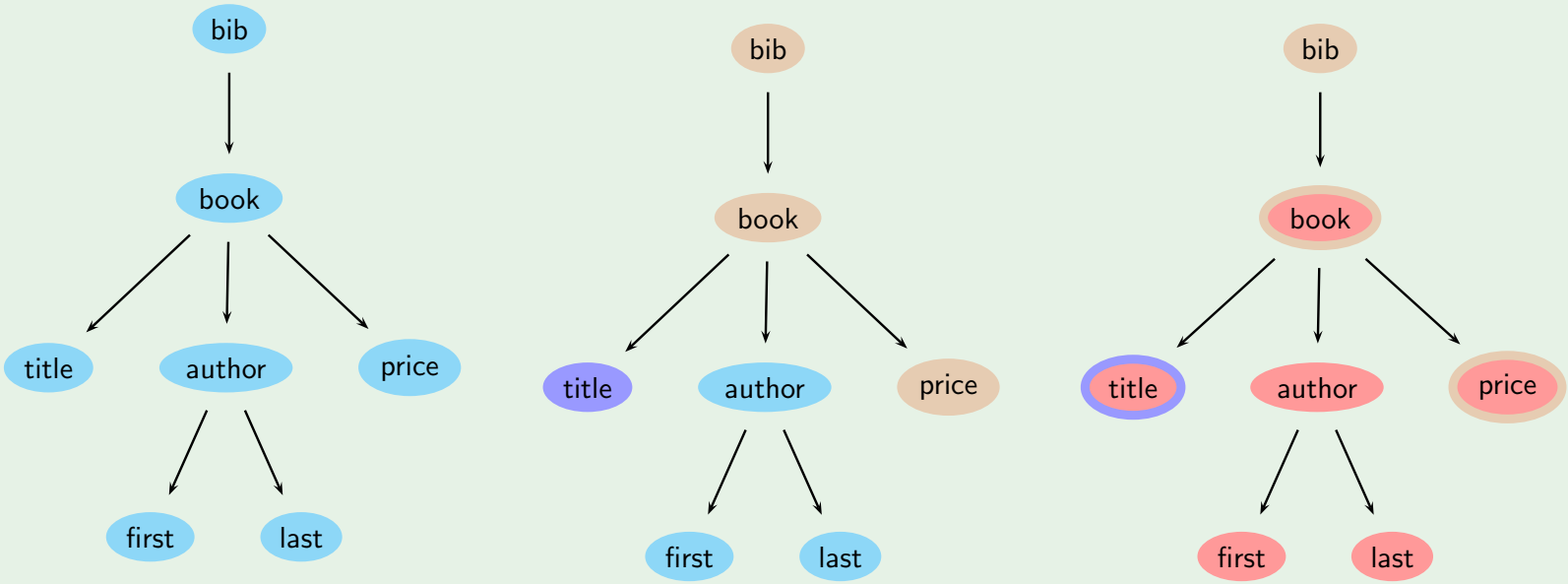


$Q_1 = //book/author$ $\quad (\perp) \quad$ $U_1 =$ `for x in` $//book/price$ `insert` $<discount/>$ `into x`

- Update does not erase data needed by the query.
- Query does not read data added by the update.

Independence Examples

Example



$Q_1 = //book[not(price)]/title$ $(\not\perp)$ $U_3 =$ `for x in //book delete x`

- Update is erasing returned nodes.
- Update is erasing used nodes.

Outline

Outline

Motivations

Applicative Contexts

- View mainteinance
  - Can avoid view re-materialization after updates occurence.
- Concurrent Updates Execution
  - Can ensure isolation between update expression, thus parallelism.
- Optimization based on rewritings
  - Can explore different evaluation orders (e.g. Join reordering)
- Access control
  - Can verify access policies.

Outline

Complexity results

- Static query-update independence testing is undecidable for XQuery.
- Decidability for *navigational XPath* and *navigational XQuery*.
  - [Benedikt, Bonifati, Flesca and Vyas. XIME-P 2005]
- NP-C for *downward* XPath based query and update with predicate. Polynomial algorithm for *linear downward* XPath.
  - [Raghavachari and Shmueli, EDBT 2006]

We should settle for a static analysis that conservatively approximates the result.

State of the Art - Projection-based techniques

General strategy

1. Abstract over set of nodes accessed by the query.
2. Abstract over set of nodes impacted by the update.
3. Check for an empty intersection among the two sets.

Query Data-Projection

- The set of document nodes needed for query execution.
  - s.t. executing a query over a data-projection does not alter query semantics.
- Projection for the XML case
  - [Marian and Simeon, VLDB03] Projection paths
  - [Benzaken, Castagna, Colazzo and Nguyen, VLDB06] Type-projectors.

State of the Art - Projection-based techniques

Update Commutativity [Ghelli, Rose, Siméon, TODS 08]

- Considers *iterative* language, Schema-less analysis.
- Notion of accessed/updated nodes in terms of paths.
- Infers paths traced by expressions, and checks for a non-overlapping.

Independence Query - Update [Benedikt, Cheney, VLDB 09]

- Considers *snapshot* language. Schema-based analysis.
- Notion of accessed/updated nodes in terms of schema types.
- Type inference for expressions, and checking for type-set disjunction.

State of the Art - Other approach

Destabilizers and Independence [Benedikt and Cheney, VLDB 10]

- Inference of a finite set of update expressions that destabilize query result.
- Disjunction between destabilizer and update, after translation into FO.
- More general framework, applied in XML case with a Schema-less analysis.

Outline

Chain Inference and Independence Detection

Our solution

- Schema-based analysis.
- Accessed/updated nodes in terms of type chains (i.e., sequences of types)
  - Chain for a query : type of result + context + visiting order.
- Comports simplification of the deduction system.
- Basis for an extremely precise analysis.

Contributions

1. A type system for inferring chains.
2. Characterization of a sound and finite set of chains for Independence Analysis

Chain Inference and Independence Detection

Process Overview

1. Chain Inference for Query
   - Used chains
   - Returned chains
   - Element chains
2. Chain Inference for Update
   - Update chains
3. Query-Update Independence Test
   - Update chains checked against Return chains
   - Update chains checked against Used chains

Chain Inference and Independence Detection

$\mathtt{Q} = //book[editor]/title$

$\mathtt{U} =$ `for x in` $//book$ `insert` $<author/>$ `into x`

Path-based analysis [GRS 08] (Schema-less)

- $\{//book/editor, //book/title\} \perp^{?} \{//book/author\}$

Type-based analysis [BC 09] (Schema-based)

- $\{bib, book, editor, title\} \perp^{?} \{book, [\![book]\!]_{desc}\}$

Both the techniques exclude independence!

Chain Inference and Independence Detection

$Q = //book[editor]/title$

$U =$ `for x in` $//book$ `insert` $<author/>$ `into x`

Chain Inference (Schema-based)

- bib.book.title# is a return chain
- bib.book.editor is a used chain
- $\diamond$ author is an element chain
- bib.book : author is an update chain

Independence Cheking

- is bib.book.title# a prefix of bib.book : author ? No
- is bib.book : author a prefix of bib.book.title# ? No
  - author is crucial for detecting independence
- is bib.book : author a prefix of bib.book.editor ? No

Therefore we can conclude independence!

Chain Inference and Independence Detection

$\mathtt{Q} = //editor//name$

$\mathtt{U} =$ `for x in` $//publisher$
`delete x`$//name$

Path-based analysis [GRS 08](Schema-less)

- $\{//editor//name\} \perp^{?} \{//publisher//name\}$

Type-based analysis [BC 09] (Schema-based)

- $\{ bib, book, editor, name \} \perp^{?} \{ name \}$

Both the techniques exclude independence!

Chain Inference and Independence Detection

$Q = //editor//name$

$U =$ `for x in` $//publisher$
`delete x`$//name$

Chain Inference (Schema-based)

- bib.book.editor.name# is the return chain
- bib.book.publisher : name is the update chain

Independence Cheking

- is bib.book.editor.name# a prefix of bib.book.publisher : name ? No
- is bib.book.publisher : name a prefix of bib.book.editor.name# ? No

Context distinguish the two types : we conclude independence!

A Finite Characterization of chains sound for Independence

Which is the length of the chain we infer for Schemas and Expressions classes?

Recursive	Path lenght	∞
Non-recursive	Path length	Schema depth
	//-free	non //-free

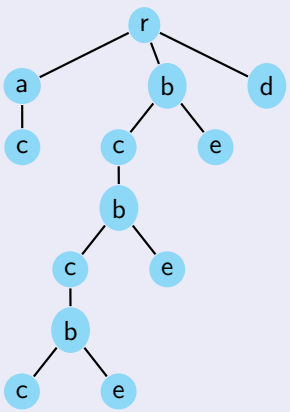
A Finite Characterization of chains sound for Independence

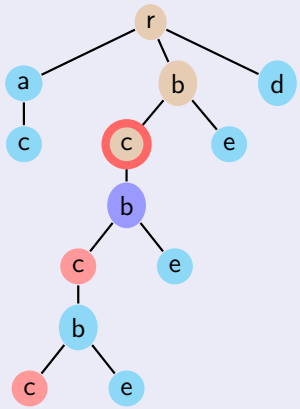
When the Schema is recursive, infinite chains could satisfy a //-navigation.

Characterize a minimal and finite set of chains sound for Independence Detection

Idea

A recursive type does not need to be completely unfolded.





$$\texttt{Q} \quad = \quad /r/b/c/b \qquad \rightsquigarrow \quad \{\, r.b.c.b\# \,\}$$

$$\texttt{U} \quad = \quad \texttt{delete}\ //b/c \quad \rightsquigarrow \quad \{\, r.b : c,\ r.b.c.b : c,\ r.b.c.b.c.b : c,\ \vdots \,\}$$

Sound chain set for determining independence

- accessed data chains that traverse an r, c element and two b elements
- updated data chains that traverse one b and one c element

A Finite Characterization of chains sound for Independence

$$k_{\mathtt{P}} \;:=\; \mathit{max\_tag\_freq}(\mathtt{P})$$

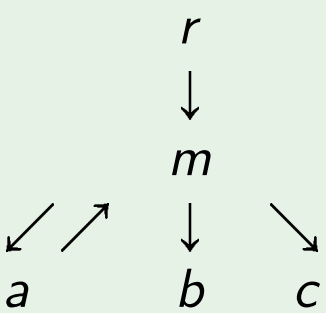
Fixed P, we restrict our analysis on chains s.t. label frequency is bounded by $k_{\mathtt{P}}$.

Example (Tag Frequency)

1. fr(b, /r/b/c/b) = 2 fr(r, /r/b/c/b) = 1 $k_{\mathtt{P}} = 2$
2. fr(b, /r/node()/b) = 2 fr(r, /r/node()/b) = 2 $k_{\mathtt{P}} = 2$
3. fr(c, //c/ancestor :: c) = 2 $k_{\mathtt{P}} = 2$

A Finite Characterization of chains sound for Independence

Example



$$\mathcal{C}_S^0 = \varnothing$$

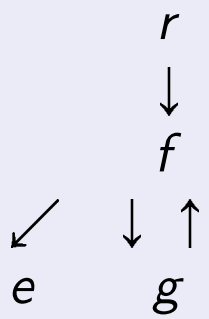
$$\mathcal{C}_S^1 = \{\, r.m.a\,,\ r.m.b\,,\ r.m.c\,\}$$

$$\mathcal{C}_S^2 = \{\, r.m.a\,,\ r.m.b\,,\ r.m.c\,,\ r.m.a.m.a\,,\ r.m.a.m.b\,,\ r.m.a.m.c\,\}$$

$$\mathcal{C}_S^3 = \{\, r.m.a\,,\ r.m.b\,,\ r.m.c\,,\ r.m.a.m.a\,,\ r.m.a.m.b\,,\ r.m.a.m.c\,,\ r.m.a.m.a.m.a\,,\ r.m.a.m.a.m.b\,,\ r.m.a.m.a.m.c\,\}$$

$$\vdots$$

Finite Chain Inference



$$\mathtt{Q}_1 = //e \quad \rightsquigarrow \quad \{r.f.e\}$$
$$\mathtt{U}_1 = \mathtt{delete}\ //f/g \quad \rightsquigarrow \quad \{r.f : g\}$$

SOUND *UNSOUND*!

Need to explore all of the descendants without repetition of a recursive type!

This can be done, adding $+1$ for each // navigation in the path.

Finite Chain Inference

$$k_{\mathtt{P}} \quad := \quad \mathit{max\_tag\_freq}(\mathtt{P}) \quad + \quad \#\mathtt{desc}(\mathtt{P})$$

Theorem (Soundness of Finite Approximation)

Provided a schema S, a query $\mathtt{Q}$ and an update $\mathtt{U}$ are deemed as independent for the analysis based on the set of all chains of S iif they are deemed as independent for the analysis based on the chains of S with tag frequency bounded by

$$k \quad := \quad \mathit{max}(\, k_{\mathtt{Q}},\, k_{\mathtt{U}}\,)$$

Finite Chain Inference

Cost of Chain Inference

Exponential when

- fully mutually recursive schemas
- and paths that uses // + `node()` filtering.

$$\begin{array}{rcl} r & \rightarrow & (a, b*, c) \\ a, b, c & \rightarrow & (a, b, c \mid string)^{+} \end{array}$$

$$//\texttt{node}()//\texttt{node}() \ldots //\texttt{node}()$$

This it is quite far from real cases.

- Recursion is quiet limited in schemas.
  - XMark Doc Schema, only 5/76 types are fully mutually recursive.
- Path expressions tends to be highly selective.
- Small tag frequency values for paths.
  - For XMark, XPathMark queries k is generally in $\{1, 2\}$ and never > 3.

The analysis is fast in practice!

Outline

Experimental setting

Prototyping

- Implemented chain inference using a prefix tree.
- Checking independence reduces to tree simulation.

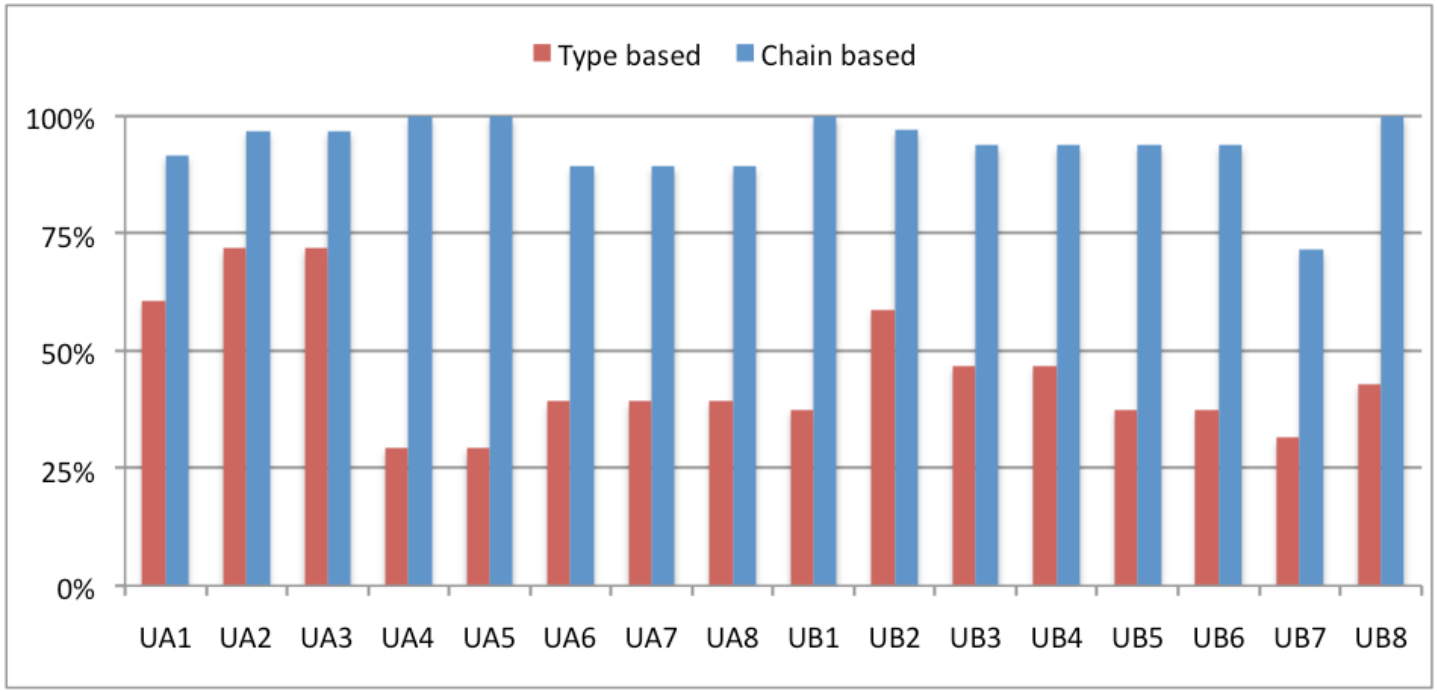
View management framework

- a set of 36 views defined by XMark and XPathMark expressions.
- a set of 16 updates defined using XPathMark expressions.

Features

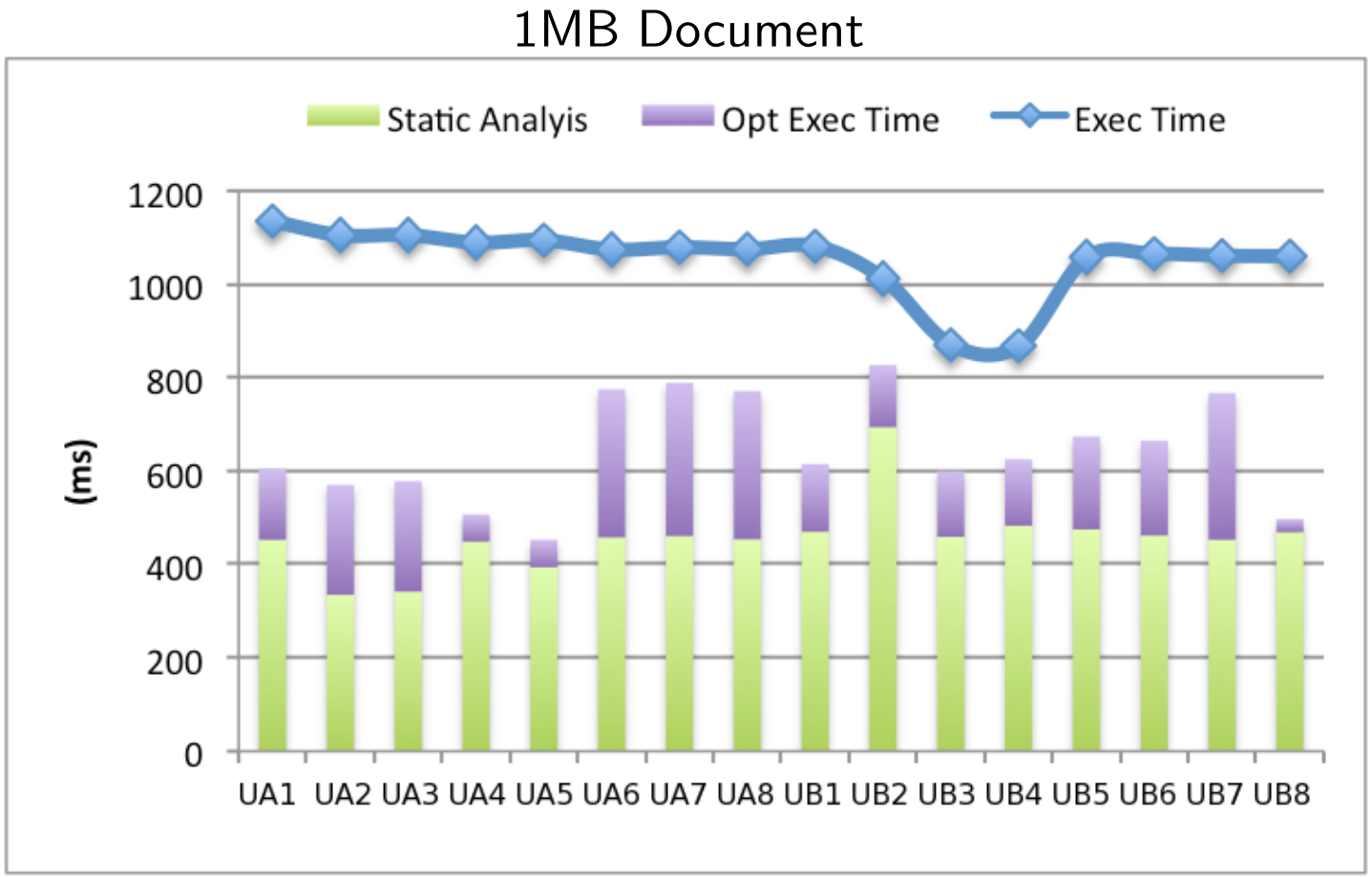
- Precision: How many independencies can we predict statically?
- Time consumption: How long does it takes to predict an independence ?

Experimental results - Precision



static independencies detected: type-based 44% vs. chain-based 94%

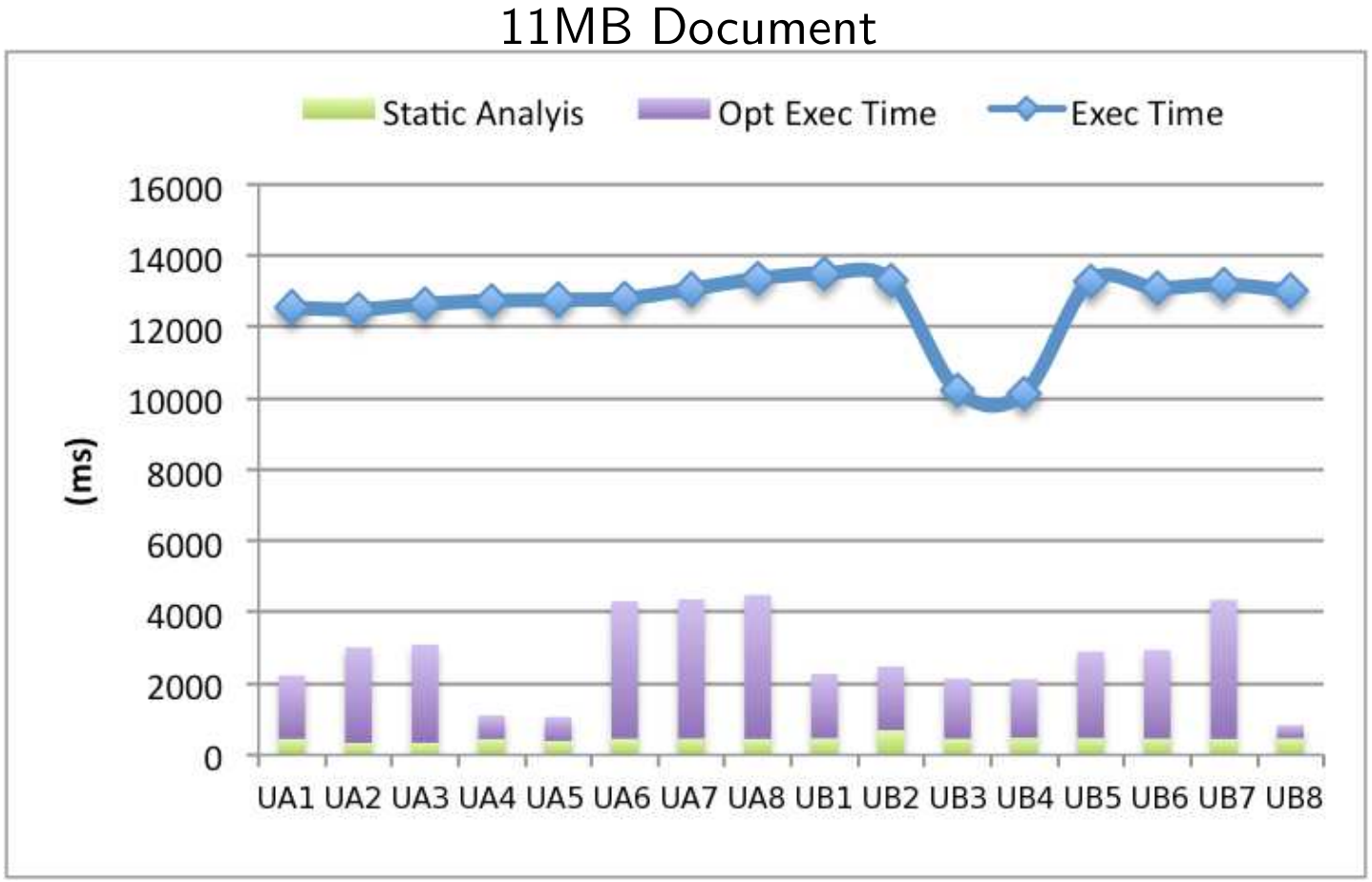
Experimental results - Time consumption



- 18% to 59% savings for 1.1MB doc
- 67% to 78% savings for 11MB doc

up to 25% for the previous technique with both docs

Experimental results - Time consumption



- 18% to 59% savings for 1.1MB doc
- 67% to 78% savings for 11MB doc

up to 25% for the previous technique with both docs

Outline

Future Works

Completeness

Find classes of schemas and queries for which the technique is complete.

Schema-less documents.

Data-guides instead of schemas.
We expect most of the precision will be maintained.

Access Control

Chain for security policies and queries/updates.

Thank you!