

On the Correctness of Query Results in XML P2P Databases (Extended Abstract)

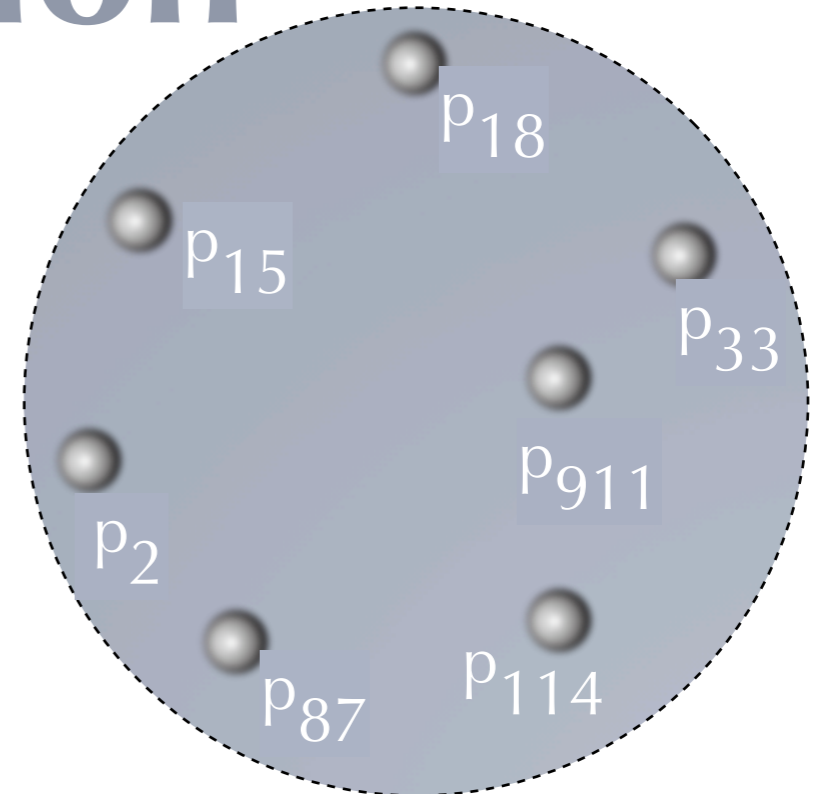
Carlo Sartiani

Dipartimento di Informatica - Università di Pisa

(This work has been funded by the GRID.IT Project and by Microsoft Corporation)

Introduction

```
for $p in input()//article
  $a in $b/author,
  $t in $b/title
return <author-title>
      {$a,$t}
      </author-title>
```



- $\{p_2, p_{15}, p_{18}, p_{33}, p_{87}, p_{114}, p_{911}\}$: complete & correct result
- $\{p_2, p_{15}, p_{18}, p_{87}, p_{911}\}$: incomplete but correct result

Introduction (bis)

```
for $a in input()//author
let $p_list := for $p in input()//article
                where $p/author = $a AND
                    $p/year = 2003
                return {$p}
where count($p_list) < 5
return <subIudice> {$a} </subIudice>
```

- $\{p_2, p_{15}, p_{18}, p_{33}, p_{87}, p_{114}, p_{911}\}$: complete & correct result
- $\{p_2, p_{15}, p_{18}, p_{87}, p_{911}\}$: incomplete & maybe incorrect result

Objectives

- to study the problem of correctness of query results in XML P2P databases in the presence of *incomplete* query plans
- to identify classes of statically correct queries
 - correctness enforced at compile-time

Outline

- background
- the nature of the problem
- the results
- future work

Background

Query Processing

- node p_1 submits the query Q to the overlay network
- the overlay network returns a query plan (algebraic expression + a set of operators-peers bindings)
- node p_1 executes Q

The Nature of the Problem

Issues

- algebraic query plans can be incomplete
 - sudden changes in the topology
 - data structures not up-to-date
- queries executed on a subset of relevant nodes

Consequences

- dangerous for set manipulations
 - set predicates
 - aggregation functions
 - `count($p_list) < 5`
- errors may propagate from inner queries to outer queries

Errors

- errors are generated by non-monotone operators of XQuery
 - where + set predicates
 - return + aggregation functions
- errors may propagate from inner queries to outer queries

Incompleteness Sources

- path expressions spanning on multiple sites
 - unguarded path expressions
 - `input()//book`
- nested queries

Results

Approach

- studying the monotonicity properties of XQuery
 - locating constructs originating incompleteness
- finding query classes for which correctness can be statically enforced
 - warnings from the compiler
- using syntactical information only

Class 1

- no nested queries returning incorrect results
- no set predicates or aggregation functions applied to incomplete sets
- semi-decidable

Class 2

- no nested queries returning incorrect results
- no set predicates or aggregation functions applied to the result of the evaluation of nested queries and unguarded path expressions
- statically enforced

Future Work

- using type information for refining the classification [2]
- extending XQuery type-checking to P2P environments [3]

Bibliography

- [1] C. Sartiani, G. Ghelli, P. Manghi, G. Conforti.: Xpeer: A Self-organizing XML P2P Database Systems. In Proceedings of the 1st International Workshop on Peer-to-peer Computing and Databases, 2004 (P2P&DB2004).
- [2] D. Colazzo, G. Ghelli, P. Manghi, C. Sartiani.: Types for Path Correctness of XML Queries. In Proceedings of ICFP 2004

Bibliography

- [3] D. Colazzo, C. Sartiani.: Typechecking Queries for Maintaining Schema Mappings in XML P2P Databases. In Proceedings of the 3th Workshop on Programming Language Technologies for XML (Plan-X), in conjunction with POPL 2005.