On XPath Dialects with Variables

Emmanuel Filiot, Joachim Niehren, Jean-Marc Talbot, Sophie Tison

INRIA Futurs, Lille, Mostrare project

October 24, 2006

Plan

- Logical queries in trees
- Standard XML query languages and Core XPath 2.0
- First-order expressiveness
- NP-completeness
- An efficient FO-expressive fragment

XML Documents

- standard format for data exchange
- parse into trees
  - unranked, sibling-ordered
  - nodes contain data values (text)
  - unordered attributes
- ignore details of data model and values in this talk

Unranked Sibling-Ordered Trees

labels \( a \in \Sigma \)

trees \( t \in T_\Sigma := a(t_1, \ldots, t_n) \) where \( n \geq 0 \)

example
N-ary Queries in Trees

function $q$ mapping trees to sets of $n$-tuples of nodes

$$\forall t \in T_{\Sigma} : q(t) \subseteq \text{nodes}(t)^n$$

- $n=0$: Boolean queries are tree languages
- $n=1$: monadic queries select nodes
- $n=2$: binary queries select pairs of nodes

Trees as Logical Structures

- domain is $\text{nodes}(t)$
- binary relations: $\text{child}^*, \text{nextsib}^*$
- unary relations: $\text{lab}_a$ for all $a \in \Sigma$
- enough relations for FO, but more relations are needed for constraints.

Logical Languages

First-order logic (FO)

$$\phi ::= \text{child}^*(x,y) | \text{nextsib}^*(x,y) | \neg \phi | \exists x \phi$$

Monadic second-order logic (MSO)

$$\phi ::= \ldots | x \in X | \exists X. \phi$$

Logical Queries

- let $\phi$ be a logical formulas
- let $\vec{x} = (x_1, \ldots, x_n)$ be a sequence of $n$ variables
- they define an $n$-ary queries such that for all trees $t \in T_{\Sigma}$:

$$q_{\phi, \vec{x}}(t) = \{(\alpha(x_1), \ldots, \alpha(x_n)) | t, \alpha \models \phi\}$$
Parametrized Logical Languages

Fix set of queries $\Gamma$ as parameter, for instance:
- $\{\text{child, nextsib, root, leaf, child}^*, \text{nextsib}^*\}$
- all FO-definable binary queries
- all MSO-definable binary queries

Positive propositional formulas

$$\phi ::= r(x_1, \ldots, x_n) | x = y | \phi \land \phi' | \phi \lor \phi' | \exists x \phi$$

where $r \in \Gamma$

Constraints (conjunctive queries)

$$\phi ::= r(x_1, \ldots, x_n) | x = y | \phi \land \phi' | \exists x \phi$$

where $r \in \Gamma$

Core XPath 2.0

$s ::= \text{child} | \text{nextsib} | \text{child}^{-1} | \text{nextsib}^{-1}$

$p ::= s | s^* | x | a | p/p' | p \cup p' | [p] | \neg p$

Navigational language inspired by modal logics
- define navigation paths from start nodes to end nodes
- binary queries: pairs (start-node, end-node)
- monadic queries: end-nodes when starting at the root

Variables to capture nodes as in hybrid logic (Blackburn 95)
- define n-ary queries by navigations paths with n-variables
- not available in XPath 1.0

Example

select all author-title-pairs in books of a bibliography:

Core XPath 2.0

$\text{child}^*/\text{book}/$

$[\text{child}/\text{author}/\text{child}/x]/$

$[\text{child}/\text{title}/\text{child}/y]$

FO

$$\exists z_1, z_2, z_3, z_4$$

$$(\text{child}^*(z_1, z_2) \land \text{lab}_{\text{book}}(z_2)) \land$

$$\text{child}(z_2, z_3) \land \text{lab}_{\text{author}}(z_3) \land \text{child}(z_3, x)$$

$$\text{child}(z_2, z_4) \land \text{lab}_{\text{title}}(z_4) \land \text{child}(z_4, y)$$

Standard XML Query Languages

- XPath 1.0
  - select nodes in trees (monadic queries)
  - no variables
- XPath 2.0
  - select n-tuples of nodes in trees (n-ary queries)
  - with variables
- XQuery 1.0 / XSLT 2.0
  - tree-to-tree transformations
  - use XPath 2.0
FO-Expressiveness

Theorem

All FO-queries can be expressed in the Core XPath 2.0, and vice versa.

Proof.

- Binary FO-queries can be expressed in Core XPath 2.0 (Marx 2005)
- Propositional connectives can be expressed in XPath 2.0.
- Quantifier elimination (Schwentick 2000):
  - Ehrenfeucht-Fraisse games
  - Shelah’s composition method

Query Non-Emptiness is NP-hard

Booleans

- Tree $t = b(0, 1)$
- Define $\text{child}_0 = \text{child} / [0]$ and $\text{child}_1 = \text{child} / [1]$.

Encode 3-Sat

- Example: $(x \lor y) \land (\neg x \lor z)$
- Core XPath 2.0 encoding:

$$[lab_0] / [\text{child}_1 / x \cup \text{child}_1 / y] / [\text{child}_0 / x \cup \text{child}_1 / z]$$

Query on $t$ is non-empty iff all clauses are satisfiable.

Query Expressiveness and Efficiency

Overview

Core XPath 2.0 $= \text{FO} < \text{MSO} = \text{tree automata}$

PSpace $\leq$ PSpace $\leq$ PSpace $\leq$ P

Question

Are there FO-expressive fragments of Core XPath 2.0 which allow for efficient query answering?

Only partial answers so far

- Gottlob & Koch & Pichler (2004) present efficient algorithms for monadic queries in Core XPath 1.0 (without variables).
- Marx (2005) presents an efficient algorithm for binary queries in Core XPath 2.0.
Efficient Fragment of Core XPath 2.0

Restrictions

- **NVS(/)** no sharing of variables between the path on two sides of composition operators
- **NV(¬)** no variable below negation

Query answering algorithm

- valid more generally
- for a parametrized path language without negation

Parametrized Path Composition Language

Let \( \Gamma \) be a set of binary queries.

Composition language \( C(\Gamma) \)

\[
\psi \in C(\Gamma) ::= r \mid \psi/\psi' \mid x \mid [\psi] \mid \psi \cup \psi' \quad \text{where } r \in \Gamma
\]

If \( \text{child}^* \in \Gamma \) then \( C(\Gamma) \) has the same expressiveness as positive propositional formulas over \( \Gamma \) up to linear time conversions:

\[
\begin{align*}
 r(x,y) & \Rightarrow \text{child}^*/x/r/y \\
 \psi \land \psi' & \Rightarrow [\psi]/[\psi']
\end{align*}
\]

Lemma (Relation to Core XPath 2.0)

\[
C(\text{bin(Core XPath 2.0)}) = \text{Core XPath 2.0} \cap \text{NV(¬)}
\]

Restriction \( \text{NVS}(/) \) is identical on both sides.

Answering \( C(\Gamma) \) Queries satisfying \( \text{NVS}(/) \)

Ideas of the Algorithm

- test emptiness of all subqueries at all nodes of the tree.
- needs a new sharing trick, to treat \( \cup \) on the left of \( / \)
- process queries recursively:
  - always filter unsatisfiable cases
  - eliminate duplicates
  - memoize auxiliary values

Theorem

Computing \( q_\psi, \pi(t) \) for paths \( \psi \) of Core XPath 2.0 satisfying \( \text{NVS}(/) \) and \( \text{NV(¬)} \) on trees \( t \) is in time \( O(|\psi|^2 |t|^2 (1 + |q_\psi, \pi(t)|)) \).

Comparison with Acyclic Conjunctive Queries

- Paths in \( C(\Gamma) \) without \( \cup \) and satisfying \( \text{NVS}(/) \) can be translated to acyclic conjunctive queries
- With \( \cup \) a different translation is needed, which fails to map to disjunctions of acyclic conjunctive queries.
FO-Expressiveness

Theorem
The restriction of Core XPath 2.0 satisfying NV(¬) and NVS(/) is FO-expressive (and permits efficient query answering).

Proof.
- all binary FO-queries can be expressed (Marx 05)
- paths of the composition language \( C(bin(FO)) \) restricted by \( NVS(/) \) can be expressed.
- new quantifier elimination theorem concludes.

Theorem (Quantifier elimination)
\[ C(bin(FO)) = FO. \] Compositions restricted by \( NVS(/) \) are enough, too.

Future Work
- efficient enumeration algorithms for fragments of Core XPath 2.0 and \( C(\Gamma) \).
- towards XML transformations.

Conclusions
- Core XPath 2.0 is FO-expressive
- Efficient FO-expressive fragments exist
- Path perspective is essential to define restriction \( NVS(/) \).
- Otherwise, propositional formulas would be good enough.