XPath

Reference

- XPath leashed, Michael Benedikt and Christoph Koch, TR, 2006

Expressivity of XPath

Formal setting

- XPath interpreted in a logical structure *t* with a *finite* set of labels and a *finite* set of Attributes @Ai (functions from nodes to integers)
- Navigational XPath:
  - p ::= step | p/p | p ⋁ p
  - step ::= axis | step[q]
  - q ::= lab() = L | p | q ⋀ q | q ⋁ q | **not q**
- Semantics:
  - [[p]]*t* : Node -> P(Node) (= NodeSet)
  - [[q]]*t* : Node -> Bool

FO-XPath

- We add:
  - id(p/@A): {<m,n> | m p/@A m' and n/@ID = m' }
  - p/@A RelOp i: existential semantics
  - p/@A RelOp q/@B: existential semantics
- Integers i are just constants

AggXPath

- Integers are extended with aggregates and arithmetic:
  - i ::= 'c' | i+i | i*i | count(p) | sum(p/@A)
- Comparisons are extended with i RelOp j
- AggXPath with positions (OrdXPath):
  - We add position() and last():
    i ::= … | position() | last()
  - Qualifiers are evaluated wrt to a context enriched with the position of the current element and the length of its sequence

Restrictions:

- P-X-XPath: no negation or disequality
- Conjunctive query: positive, no disjunction, no union

Expressiveness

- NavXPath can be translated in linear time as FO over Lab_L, R_axis where axis in: child, next-sibl, desc, foll-sibl:

  (x,y) in book[title]/author:

  ∃z,w. child(x,z) /\ Lab_book(z) /\ child(z,w) /\ <title>(w) /\ child(z,y) /\ <author>(y)

  (x,y) in parent::(book)/child::author:

  ∃z. child(z,x) /\ <book>(z) /\ child(z,y) /\ <author>(y)

NavXPath vs. FO

- FO is more expressive:
  - Exists a subsequence C-B*-C?
- NavXPath = FO^2 :
  - qualifiers in NavXPath corresponds to FO^2 (2-variables FO) with one free variable
  - NavXPath paths have a linear normal form

NavXPath and FO^2

- XPNF:
  - $\exists z_2 \ldots \exists z_{n-1}.\ \rho_1(z_1) \wedge \chi_1(z_1, z_2) \wedge \rho_2(z_2) \wedge \ldots \wedge \chi_{n-1}(z_{n-1}, z_n) \wedge \rho_n(z_n)$
  - ρ_i are FO^2 formulas, and the $\chi_{i-1}(z_{i-1}, z_i)$ are unions of binary atomic formulas over predicates from child, next-sibl, desc, foll-sibl
- Theorem:
  - NavXPath *filters* correspond to FO^2 formulas
  - NavXPath *relations* correspond to expressions in XPNF
- Key observation: any boolean combination of steps, equality, inequality can be reduced to a union of steps

Proof

- Key case: translate ∃y β(x, y), where β is in FO2 into qualifiers
- Bring β in DNF; every disjunct contains some binary axes (including equality), maybe negated, and two unary FO2 formulas
- Since axes are mutually exclusive, we can assume that every disjunct is just:
  - φi(x) /\ Rχi (x, y) /\ ψi(y)
- Which becomes
  - self[T(φi)]/χi[T(ψi)]

Closure of NavXPath

- NavXPath includes union
- NavXPath is closed under intersection:
  - A NavXPath query is conjunctive
  - Conjunctive queries are intersection-closed
  - Conjunctive queries over trees can be transformed into unions of acyclic conjunctive queries
  - These can be expressed by NavXPath

Closure of NavXPath

- NavXPath predicates are closed under complement
- NavXPath relations are not closed under complement
- Proof sketch:
  - with complement we can express Until (actually, all of FO)
  - NavXPath cannot express Until
- A until B (where /\ and not are relational):
  - desc[lab = B] /\ not(desc[lab != A]/desc)

NavXPath and tree patterns

- Tree patterns: node- and edge-labeled trees
- Edges are labeled with forward axes
- Nodes are labeled with either L or *
- Boolean TP: one *context* node
- Unary TP: context node + selected node

Matching a tree pattern

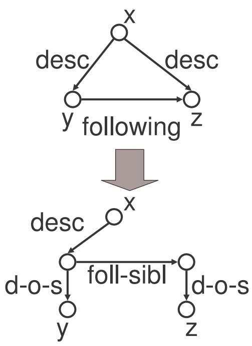
- Boolean: a homomorphism from the pattern to the tree, that maps the *context* into the node
- Unary: *context* is mapped into the first node, *selected* into the second
- Finite set of TPs: take the union of the results

TPs and NavXPath

- The following are equally expressive:
  - P-NavXPath binary queries
  - Sets of unary patterns
  - Exists+ FO with child, next-sibl, desc, following-sibl
- (1) and (2) into (3) is immediate
- TP to XPath: every edge is a step
- FO to TP: form the formula graph, then remove the cycles (non trivial!)

From Ex+ FO to TP

- Ex+ FO is the same as union of (cyclic) conjunctive queries:
  - ∃y.desc(x,y), desc(x,z), following(y,z)
- Every cycle can be rewritten out



Some rules

- d-o-s(x,z),d-o-s(y,z) ->
  - d-o-s(x,z),d-o-s(y,x) \/ d-o-s(x,y),d-o-s(y,z)
  - Same for foll-sibl
- child(x,z),d-o-s(y,z) ->
  - (child(x, z) /\ y = z) \/ (child(x, z) /\ d-o-s(y, x))
  - Same for next-sibl / foll-sibl
- next-sibl(x,z),d-o-s(y,z)
  - (next-sibl(x,z) /\ y = z) \/ (next-sibl(x, z) /\ desc(y, x))
  - Same for NS+, NS*

TP, Ex+, and P-NavXPath

- From the previous theorem, a couple of nice corollaries about P-NavXPath:
  - Using EX-+: P-NavXPath is closed under …?
  - Using TP: only forward axes are needed for positive root-queries (Olteanu et al 2002)

Extending XPath to FO

- Add path complement
- Add Until

Back to FO-XPath

- We add:
  - id(p/@A): i nodi n tali che n/@ID = p/@A
  - i RelOp i
  - p/@A RelOp i: existential semantics
  - p/@A RelOp q/@B: existential semantics
- Easy to translate in FO with the obvious signature (Ai-Comp-Aj(x,y) + trans-navigation)
- Is FO-XPath complete for FO?

Weakness of FO-XPath

- Navigational query: does not depend on attributes, but just on the tree structure
- FO-XPath expresses the same navigational queries as NavXPath

Back to Agg-XPath

- Integers are extended with aggregates and arithmetic:
  - i ::= 'c' | i+i | i*i | count(p) | sum(p/@A)
- Count can express Until
- Hence: FO complete
- Until(E2,E1) (where desc is not reflexive):
  - desc[E2] and
    count(desc[not E1]/desc[E2]) != count(desc[E2])

Complexity of evaluation

Complexity: reminder

- Some classes I may name, and their relationship
  - LOGSPACE ⊆ PTIME ⊆ PSPACE ⊆ EXPTIME
  - LOGSPACE ⊆ NLOGSPACE ⊆ P(TIME) ⊆ NP(TIME) ⊆ PSPACE ⊆ EXPTIME
  - P ⊆ co-NP ⊆ PSPACE
- Non-elementary: not bounded by 2^(2^…(2^n))

Data complexity and combined complexity

- Assume that the evaluation of a query Q on a structure T costs: O(|T|^|Q|)
- How bad is that?
  - Data complexity: it is in PTime: O(|T|^n)
  - Query complexity: ExpTime: O(n^|Q|)
  - Combined complexity: ExpTime: O(|In|^|In|)
- MSO: data is linear, query is PSpace

Data complexity of XPath

- *Unary* NavXPath has linear *data* complexity
  - Proof: boolean MSO is linear on trees
- MSO does not help much with combined complexity:
  - MSO over trees is PSpace-complete for combined complexity

Combined complexity

- NavXPath is PTime-hard
- Full XPath 1.0 is in O(|Data|^5 * |Query|^2)

Satisfiability

- FO over trees is decidable, but is non-elementary
- Satisfiability for NavXPath and for unnested NavXPath is ExpTime complete:
  - Reduction to Deterministic Propositional Dynamic Logic with Converse shows that NavXPath is in ExpTime (Marx – EDBT 04)
  - Hardness follows by hardness of containmens (Neven-Schwentick – ICDT 03)
  - An O(2^n) algorithm has been recently described, based on translation on mu-calculus with converse
- Satisfiability for NavXPath with intersection is NExpTime complete
  - Etessami Vardi Wilke: FO2 can encode Unary Temporal Logic

XPath fragments

- P-NavXPath: no negation, and = is the only relation
- Benedikt – Fan – Geerte (PODS05:
  - PNavXPath with downard axes: every expression is satisfiable
  - If we add upward, or sibling, or a DTD: NP-complete
  - P-FOXPath is still NP-complete
- However (Geerts-Fan, DBPL05):
  - Sat for FOXPath is undecidable
    - Reduction from halting of two-register machines
- Borders of decidability are not well understood