Proof Systems for Data Queries

http://projects.lsv.ens-cachan.fr/prodaq

Mid-project Review

October 5th, 2016
CONSORTIUM

Partner Laboratoire Spécification et Vérification
École Normale Supérieure Cachan

Current Participants

David Baelde  Anthony Lick  Sylvain Schmitz
Life of the Project

May–Jul. 2015  MSc. internship of Simon Lunel
Apr.–Jul. 2016  MSc. internship of Anthony Lick
Jun.–Jul. 2016  BSc. internship of Manoj Kilaru
**GENERAL CONTEXT**

- **XML format**
- **data trees**
- **XPath queries**
- **Satisfiability**

```xml
<catalog>
  <artists>
    <a id="a1">John Coltrane</a>
    <a id="a2">Miles Davis</a>
    <a id="a3">Oscar Peterson</a>
  </artists>
  <records>
    <genre name="jazz">
      <genre name="studio">
        <r id="r1" artist="a2">Birth of Cool</r>
        <r id="r2" artist="a1">Blue Train</r>
        <r id="r3" artist="a2">Kind of Blue</r>
      </genre>
    </genre>
    <genre name="compilation">
      <r id="r4" artist="a2">The Complete Jack Johnson Sessions</r>
    </genre>
  </records>
</catalog>
```
GENERAL CONTEXT

- XML format
- data trees
- XPath queries
- Satisfiability
GENERAL CONTEXT

- XML format
- data trees
- XPath queries

“artists with at least one record”

//a[@id='//r/@artist']

- Satisfiability
GENERAL CONTEXT

- XML format
- data trees
- XPath queries
- Satisfiability
  - optimization
  - verification (access control)
**OBJECTIVES:** Bridges and Tools

- Data Trees
- Data Logics
- Data Automata
- Substructural Logics
- Proof Systems
- Counter Systems

**Task A**
**Task B**
**Task C**
OBJECTIVES: BRIDGES AND TOOLS

Data Trees

Data Logics

Substructural Logics

Data Automata

Counter Systems

Proof Systems

Decidability complexity

Verification

Benchmarks

Proof search

Tool

Task A

Task B

Task C
A: Proof Systems for Data Logics

XPath in the literature: CoreDataXPath
[Benedikt & Koch, Bojánczyk et al.]

- **single** attribute: \( @d \)

- **full joins** after paths \( \pi, \pi' \):
  \[
  \pi/@d = \pi'/@d \quad \pi/@d \neq \pi'/@d
  \]

- navigation in paths \( \pi \) along several axes, e.g.:
Reasoning about CoreDataXPath

- undecidable satisfiability

- decidable fragments (subsets of axes)
  [Figueira]
  - different proofs for different fragments
  - model-theoretic approaches:
    - bound size of models if they exist
    - combinatorial algorithms: enumerate potential models

- in PRODAQ: proof systems
  - concrete proof search algorithms
  - witnesses of unsatisfiability (proofs)
Reasoning about CoreDataXPath

- undecidable satisfiability
- decidable fragments (subsets of axes) [Figueira]
  - different proofs for different fragments
  - model-theoretic approaches:
    - bound size of models if they exist
    - combinatorial algorithms: enumerate potential models

- in PRODAQ: proof systems
  - concrete proof search algorithms
  - witnesses of unsatisfiability (proofs)
Reasoning about CoreDataXPath

- undecidable satisfiability
- decidable fragments (subsets of axes) [Figueira]
  - different proofs for different fragments
  - model-theoretic approaches:
    - bound size of models if they exist
    - combinatorial algorithms: enumerate potential models
- in PRODAQ: proof systems
  - concrete proof search algorithms
  - witnesses of unsatisfiability (proofs)
Modal Logic on Data Trees

Publications: [MSc’15, CSL’16]; People: David Baelde, Simon Lunel, Sylvain Schmitz

- “descendant” axis of CoreDataXPath: DataKL
  - $\Diamond \equiv \varphi$ “\varphi holds in a descendant with the same data”
    - XPath syntax: $@d = \text{descendant::*}[\varphi]/*@d$
  - $\Diamond \neq \varphi$ “\varphi holds in a descendant with a different data”
    - XPath syntax: $@d \neq \text{descendant::*}[\varphi]/*@d$

- sound & complete hypersequent calculus
- optimal PSPACE proof search algorithm
  - no penalty from the use of proof systems!
Modal Logic on Data Trees

Publications: [MSc’15, CSL’16]; People: David Baelde, Simon Lunel, Sylvain Schmitz

- “descendant” axis of CoreDataXPath: DataKL

  ◊_≡_φ “φ holds in a descendant with the same data”
  XPath syntax: @d = descendant::*[φ]/@d

  ◊_≠_φ “φ holds in a descendant with a different data”
  XPath syntax: @d != descendant::*[φ]/@d

- sound & complete hypersequent calculus

- optimal PSpace proof search algorithm
  no penalty from the use of proof systems!
Modal Logic on Data Trees

Publications: [MSc’15, CSL’16]; People: David Baelde, Simon Lunel, Sylvain Schmitz

- “descendant” axis of CoreDataXPath: DataKL
  - $\Diamond \equiv \varphi$ “$\varphi$ holds in a descendant with the same data”
    - XPath syntax: $\varphi$ holds in a descendant with the same data
      
  - $\Diamond \neq \varphi$ “$\varphi$ holds in a descendant with a different data”
    - XPath syntax: $\varphi$ holds in a descendant with a different data

- sound & complete hypersequent calculus
- optimal PSPACE proof search algorithm
  - no penalty from the use of proof systems!
Modal Logic on Data Trees

Publications: [MSc’15, CSL’16]; People: David Baelde, Simon Lunel, Sylvain Schmitz

- “descendant” axis of CoreDataXPath: DataKL
  - $\diamond \equiv \varphi$ “$\varphi$ holds in a descendant with the same data”
    - XPath syntax: @d = descendant::*[\varphi]/@d
  - $\diamond \neq \varphi$ “$\varphi$ holds in a descendant with a different data”
    - XPath syntax: @d != descendant::*[\varphi]/@d

- sound & complete hypersequent calculus
- optimal PSPACE proof search algorithm
  - no penalty from the use of proof systems!
Modal Logic on Words

Publications: [MSc’16]; People: David Baelde, Anthony Lick, Sylvain Schmitz

- **trichotomous** (non-branching) XPath axes: “ancestor”, “following-sibling”, “preceding-sibling”

- “following-sibling” fragment of **CoreXPath**: KL.3

- sound & complete sequent calculus

- optimal coNP proof search
Modal Logic on Words

Publications: [MSc’16]; People: David Baelde, Anthony Lick, Sylvain Schmitz

- trichotomous (non-branching) XPath axes: “ancestor”, “following-sibling”, “preceding-sibling”
- “following-sibling” fragment of CoreXPath: KL.3
- sound & complete sequent calculus
- optimal coNP proof search
Modal Logic on Words

Publications: [MSc’16]; People: David Baelde, Anthony Lick, Sylvain Schmitz

- inverse XPath axes: “descendant” & “ancestor”, “following-sibling” & “preceding-sibling”
- “following-sibling” & “preceding-sibling”
- fragment of CoreXPath: $K_t4.3$
- sound & complete hypersequent calculus
- optimal coNP proof search
- need to force finite words: $K_tL.3$
- need to extend to Data$K_tL.3$
Modal Logic on Words

Publications: [MSc’16]; People: David Baelde, Anthony Lick, Sylvain Schmitz

- **inverse XPath axes**: “descendant” & “ancestor”, “following-sibling” & “preceding-sibling”
- “following-sibling” & “preceding-sibling”
- Fragment of CoreXPath: $K_t4.3$
- Sound & complete hypersequent calculus
- Optimal coNP proof search
- Need to force finite words: $K_tL.3$
- Need to extend to Data$K_tL.3$
**Modal Logic on Data Words**

Publications: [MSc’16]; People: David Baelde, Anthony Lick, Sylvain Schmitz

- **inverse XPath axes**: “descendant” & “ancestor”, “following-sibling” & “preceding-sibling”
- “following-sibling” & “preceding-sibling”
  fragment of CoreXPath: $K_t^{4.3}$
- sound & complete hypersequent calculus
- optimal coNP proof search
- need to force finite words: $K_t^{L.3}$
- need to extend to Data$K_t^{L.3}$
XPath Benchmark

Work in progress; People: David Baelde, Anthony Lick, Sylvain Schmitz

- actual XPath practice ≠ CoreDataXPath
- joins are local:

  <query file="/usr/share/xml/docbook/stylesheets/.../blocks2dbk.xsl"
    line="766" type="test"
    content="@rnd:style = preceding−sibling::dbk:emphasis/@rnd:style"/>

- nominal/hybrid constructs:

  <query file="/usr/share/xml/docbook/stylesheets/.../blocks2dbk.xsl"
    line="236" type="select"
    content="following−sibling::dbk:para[@rnd:style = current()]/@rnd:style"/>

  <query file="/usr/lib/python2.7/.../iso−schematron−xslt1/iso_dsd1_include.xsl"
    line="975" type="select"
    content="//[@xml:id = current()]/@xpointer | id(@xpointer)"/>
A: Outlook

- Enrich our hypersequent calculus
  - more axes
  - nested paths
  - hybrid constructs like \texttt{id()} and \texttt{current()}
- Compile an XPath benchmark
- Implement a proof search engine
B: Data Models for Substructural Logics

Separation logic:
[O’Hearn, Reynolds, et al.]

- \( M \models \varphi_1 \ast \varphi_2 \) if \( M = M_1 \ast M_2 \) with \( \forall i. M_i \models \varphi_i \)
- versatile: heaps, concurrent data structures, etc.

- in PRODAQ: data models
  - separation should be compatible with data values
  - a new viewpoint on data logics
**B: DATA MODELS FOR SUBSTRUCTURAL LOGICS**

Separation logic:

[O’Hearn, Reynolds, et al.]

\[ M \models \varphi_1 \ast \varphi_2 \text{ if } M = M_1 \ast M_2 \text{ with } \forall i. M_i \models \varphi_i \]

- versatile: heaps, concurrent data structures, etc.

- in PRODAQ: data models
  - separation should be compatible with data values
  - a new viewpoint on data logics
SEPARATION LOGIC ON DATA WORDS

Work in progress; People: Manoj Kilaru, Étienne Lozes, Sylvain Schmitz

- separating shuffle \( \otimes \) over data words:
  - \((a_1)(b_2)(c_1)(d_3) \in (a_1)(c_1) \otimes (b_2)(d_3)\)
  - \((a_1)(b_2)(c_1)(d_3) \not\in (a_1)(d_3) \otimes (b_2)(c_1) = \emptyset\)

- separating shuffle regular expressions (SSRE) on top of regular expressions \( E \):
  - \( \varphi ::= E \mid \varphi \lor \varphi \mid \neg \varphi \mid \varphi \cdot \varphi \mid \varphi \otimes \varphi \)
Separation Logic on Data Words

Work in progress; People: Manoj Kilaru, Étienne Lozes, Sylvain Schmitz

- separating shuffle \( \mathcal{\ast} \) over data words:
  \[
  \begin{align*}
  (a_1 b_2 c_1 d_3) \in (a_1 c_1) \mathcal{\ast} (b_2 d_3) \\
  (a_1 b_2 c_1 d_3) \not\in (a_1 d_3) \mathcal{\ast} (b_2 c_1) = \emptyset
  \end{align*}
  \]

- separating shuffle regular expressions (SSRE) on top of regular expressions \( \varphi \):
  \[
  \varphi ::= E \mid \varphi \lor \varphi \mid \neg \varphi \mid \varphi \cdot \varphi \mid \varphi \mathcal{\ast} \varphi
  \]
B: Outlook

- **undecidable** emptiness of SSRE

- extension with homomorphisms strictly contains $\text{EMSO}^2$
  
  [Bojánczyk et al.]

- negation-free SSRE strict fragment of $\text{EMSO}^2$, thus decidable emptiness

- is there a fragment equivalent to $\text{FO}^2$ to $\text{EMSO}^2$?
B: Outlook

- **undecidable emptiness of SSRE**

- extension with homomorphisms strictly contains $\text{EMSO}^2$
  [Bojánczyk et al.]

- negation-free SSRE strict fragment of $\text{EMSO}^2$, thus decidable emptiness

- is there a fragment equivalent to $\text{FO}^2$? to $\text{EMSO}^2$?
B: Outlook

- **undecidable emptiness of SSRE**

- extension with homomorphisms strictly contains $\text{EMSO}^2$
  
  [Bojánczyk et al.]

- negation-free SSRE strict fragment of $\text{EMSO}^2$, thus decidable emptiness

- is there a fragment equivalent to $\text{FO}^2$ to $\text{EMSO}^2$?
C: COUNTER SYSTEMS AND DATA

- data logics restricted to (dis)equality: counting
- counter systems: operational model
- complexity & algorithms
Complexity in Petri Nets

Publications: [LICS’15]; People: Jérôme Leroux, Sylvain Schmitz

- reachability decidable
  
  [Mayr, Kosaraju, Lambert]

- related to numerous formalisms on data words: EMSO$^2$, FO$^2$, modal-$\mu$-calculi, data automata, class memory automata, etc.

- also related to many problems in other fields (process algebra, verification of concurrent systems, …)

- first known upper bound: $F_{\omega^3}$

- 35 years old major open problem
Complexity in Petri Nets

Publications: [LICS’15]; People: Jérôme Leroux, Sylvain Schmitz

- reachability decidable
  [Mayr, Kosaraju, Lambert]

- related to numerous formalisms on data words: EMSO$^2$, FO$^2$, modal-μ-calculi, data automata, class memory automata, etc.

- also related to many problems in other fields (process algebra, verification of concurrent systems, ...)

- first known upper bound: $F_{\omega^3}$

- 35 years old major open problem
**Complexity in Petri Nets**

Publications: [LICS’15]; People: Jérôme Leroux, Sylvain Schmitz

- reachability decidable
  [Mayr, Kosaraju, Lambert]

- related to numerous formalisms on data words: EMSO\(^2\), FO\(^2\), modal-\(\mu\)-calculi, data automata, class memory automata, etc.

- also related to many problems in other fields (process algebra, verification of concurrent systems, ...)

- first known upper bound: \(F_{\omega^3}\)

- 35 years old major open problem
Petri Nets with Data

Publications: [FoSSaCS’16, LICS’16]; People: Piotr Hofman, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, Sylvain Schmitz, Patrick Totzke

- tokens carry data

- \( \nu \)-Petri nets: enforce fresh data
  - \( F_{\omega \cdot 2} \)-complete safety verification

- data Petri nets: data might be non-fresh
  - decidable boundedness problems (in \( F_{\omega \omega} \))
Petri Nets with Data

Publications: [FoSSaCS’16, LICS’16]; People: Piotr Hofman, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, Sylvain Schmitz, Patrick Totzke

- tokens carry data

- \( \nu \)-Petri nets: enforce fresh data
  - \( F_{\omega^2} \)-complete safety verification

- data Petri nets: data might be non-fresh
  - decidable boundedness problems (in \( F_{\omega^\omega} \))
**Petri Nets with Data**

Publications: [FoSSaCS’16, LICS’16]; People: Piotr Hofman, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, Sylvain Schmitz, Patrick Totzke

- Tokens carry data

\[
p_0 \xymatrix{ & xxy \ar[dr] & \cr \ar[ur] & t & \ar[ul] p_2 \cr \ar[dr] & y \ar[ur] & \cr & xz & \cr \ar[ur] & \ar[dr] p_1 \cr & x & \ar[ul]}
\]

- \(\nu\)-Petri nets: enforce fresh data
  - \(F_{\omega^2}\)-complete safety verification

- Data Petri nets: data might be non-fresh
  - Decidable boundedness problems (in \(F_{\omega^\omega}\))
C: Outlook

- deviation from program centered on branching extension of Petri nets
- towards verification of data-centric dynamic systems
  [Vianu, Deutsch, et al.]
Updated Schedule

Task A  delayed by roughly a year

Task B  currently on track

Task C  on track?
PUBLICATIONS


Image Credits

- Scientific Review Public Domain
- Internet1 CC BY-SA 3.0, by Rock1997