Probabilistic XML: Modeling with RMCs and Querying with MSO

Evgeny Kharlamov
Free University of Bozen-Bolzano, INRIA Saclay – Île-de-France

Michael Benedikt Dan Olteanu Pierre Senellart
Oxford University Oxford University Télécom ParisTech

KRDB Lunch Seminar, June 2010
Outline

1. Recursive Markov Chains (RMCs)  
   [Etessami&Yannakakis’05, Etessami’06, Etessami&Yannakakis’09]

2. RMCs for Probabilistic XML  
   [Benedikt&al’10]
Recursive Graphs

- Another name: Recursive State Machines
- Natural abstract model of procedural programs with potential recursion
- Used in verification and program analysis
Recursive Graphs

- Is it possible to reach b from a?
- Can be checked in
  - cubic time in general
  - linear time when either #entries or #exits is bounded in each component

Yes, but not always.

How to measure the frequency?
Recursive Markov Chains

- Natural when we introduce randomness into Recursive Graphs
- Generalizes finite Markov Chains used in verification and model checking

[Etessami&Yannakakis’09]
Recursive Markov Chains

- Recursive graphs
- With probabilities on edges
- For every node, the probabilities of outgoing edges sum up to 1
What is the probability of eventually reaching \( b \) from \( a \)?

Can be computed using non-linear equations:

\[
x = \frac{2}{3} \cdot x^2 + \frac{1}{3} \implies x = \frac{1}{2} \text{ or } x = 1
\]

- upper path: \( \frac{2}{3} \cdot x \cdot 1 \cdot x \cdot 1 \)
- lower path: \( \frac{1}{3} \)

Theorem [Etessami & Yannakakis’09]: Least solution is the right answer.
Computing Termination Probabilities

- Sum probabilities across different paths

\[
\Pr^*(a, b) = \sum_{(aa_1 \ldots a_n b)\text{-path from } a \text{ to } b} \Pr(aa_1 \ldots a_n b)
\]

- Probability of a path = product of probabilities of transitions in the path

\[
\Pr(aa_1 \ldots a_n b) = \Pr(aa_1) \times \cdots \times \Pr(a_n b)
\]
• Properties of termination probabilities
Almost Sure Termination

- Can infinite loops prevent from termination? **Not always**

\[ x = \frac{1}{2} \cdot x^2 + \frac{1}{2} \quad \Rightarrow \quad x = 1 \text{ or } x = \frac{1}{2} \]

- upper path: \( \frac{1}{2} \cdot x \cdot 1 \cdot x \cdot 1 \)

- lower path: \( \frac{1}{2} \)

In some cases we almost surely reach the exit
Irrational Termination Prob.s

• Are probabilities always rational?  No

• Not:
  \[ x = \frac{1}{6} \cdot x^5 + \frac{1}{2} \quad \Rightarrow \quad x \approx 0.50550123 \]
  • not solvable by radicals

• For finite Markov chains
  reachability prob.s are rational

In some cases prob.s are irrational
Reachability Probability

- Probability of reaching a specific exit of a component
- Generalization of termination probability

\[
\Pr_{A_i}(en_i, ex'_i)
\]

- \( A_i \) - given component
- \( en_i \) - starting point
- \( ex'_i \) - ending point
If prob.s are rational, is it always easy to compute? No

$$\Pr_{A_n}(en_n, ex'_n) = \frac{1}{2^{2n}}$$

$$\Pr_{A_n}(en_n, ex''_n) = 1 - \frac{1}{2^{2n}}$$

For finite Markov chains, reachability prob.s are PTIME

In some cases prob.s are hard to compute
Reachability Probabilities

Can be

• less than one
• almost sure
• irrational
• double-exponentially small or big

Computation of reach. prob. is not a trivial problem
• Deciding and approximating termination probabilities
Problems for Termination Probabilities

• Qualitative decision problem: decide which of the 3 options holds
  • \( \Pr(A \text{ terminates}) = 1 \)
  • \( \Pr(A \text{ terminates}) = 0 \)
  • \( \Pr(A \text{ terminates}) \in (0, 1) \)

• Quantitative
  • decision problem \( \Pr(A \text{ terminates}) \geq p \)
  • approximation problem \( \Pr(A \text{ terminates}) \)
Deciding Termination Probabilities

- **Upper bound**: Termination probability can be decided in PSPACE

- **Lower bound**: reduction from SQRT-SUM problem
  - For a class of two exits RMCs and for every $\varepsilon > 0$
    SQRT-SUM is PTIME reducible to deciding whether
    \[
    \Pr(A \text{ terminates}) \leq \varepsilon \quad \text{or} \quad \Pr(A \text{ terminates}) = 1
    \]
  - For a class of one exit RMCs and $p \geq 0$
    SQRT-SUM is PTIME reducible to deciding
    \[
    \Pr(A \text{ terminates}) \geq p
    \]

**SQRT-SUM**: For $k$ and $d_1, \ldots, d_n$ in $\mathbb{N}$, decide
\[
\sum_i \sqrt{d_i} \geq k
\]
Hardness of Approximations

- **Theorem:** [Etessami, Yannakakis’09]
  
  Decomposed version of multivariate Newton’s Method converges monotonically to termination probabilities

- For 2-exit RMCs convergence is slow due to reduction of SQRT-SUM to decision whether

\[
\Pr(A \text{ terminates}) \leq \varepsilon \quad \text{or} \quad \Pr(A \text{ terminates}) = 1
\]

\[\Rightarrow\ \text{approximation with any nontrivial constant error is as hard as deciding SQRT-SUM}\]
Outline

1. Recursive Markov Chains (RMCs)
   [Etessami&Yannakakis'05, Etessami'06, Etessami&Yannakakis'09]

2. RMCs for Probabilistic XML
   [Benedikt&al'10]
RMCs as Prob. DTDs

<!ELEMENT directory (person*)>
<!ELEMENT person (name,phone*)>

- Document d:  
  <directory>
  </directory>  
  \[ Pr(d) = 1 \cdot 0.2 \]
  \[ Pr = 1 \cdot 0.2 \]

- <directory>
  <person>
  <name>
  <phone>
  <name>
  <phone>
  </directory>
  \[ Pr(d) = 0.2 \cdot 0.8 \cdot 0.5 \cdot 0.2 \]

On such simple RMCs representing trees, MSO queries are tractable.
RMCs as Prob. DTDs

- Extension of RMCs by adding labels on RMC components
- Main component $A_0$: root of the tree
- Probabilistic run through $A_0$ = probabilistic generation of a (string that encodes a) tree
- Prob run through a box labeled $L$ = probabilistic generation of a subtree rooted at $L$
RMCs as Prob. DTDs

• Entering a component labeled L = generating an opening tag <L>
•Exiting a component labeled L = generating a closing tag </L>
• RMC A corresponds to a probabilistic space (px-space) of documents

\[[A] = \{(srt_1, p_1), \ldots, (srt_n, p_n)\}\]
Properties of PX-Spaces Generated by RMCs

1. **Unbounded depth** of generated trees
   - recursive call of boxes
   - recursive call $\approx$ nesting in XML

2. **Unbounded width** of generated trees
   - looping within boxes
   - looping $\approx$ siblings in XML

Example: prob. DTDs via rec. Markov chains

```
<!ELEMENT directory (person*)>
<!ELEMENT person (name,phone*)>
```

$D$: directory

$P$: person

$N$: name

$T$: phone

On such simple RMCs representing trees, MSO queries are tractable!
Properties of PX-Spaces Generated by RMCs

3. Reaching the main exit $\approx$ generation of a tree
   $\Rightarrow$
   Reachability probability $\approx$ probability of a tree

4. Irrational probabilities of generated trees

5. Probabilities of generated trees
double exponentially close to 0 or 1

Is reachability interesting for XML? NO
• Checking MSO properties for RMCs
MSO for XML: Example

- \( \exists n, m, k. \text{Child}(n, m) \land \text{Label}_a(n) \land \text{Label}_b(m) \land (m < k) \)

- \( \text{a/b[following-sibling(\star)]} \)
- no label predicate for a node \( k \)
  \( \sim \) wildcard for \( k \)

- \( \exists n, m. \text{Label}_a(n) \land \text{Label}_b(m) \land \)
  \( ( \text{Child}(n, m) \lor (\exists S. \neg S(n) \land \neg S(m)) \\
  \land \forall x. S(x) \rightarrow \\
  \text{Child}(n, x) \lor \text{Child}(x, m) \\
  \lor \exists k. (\text{Child}(x, k) \lor \text{Child}(k, x)) \land S(k)) \)

- \( \text{a//b} \)
- \text{transitive closure of Child relations}
∃n, m. Label_a(n) \land Label_b(m) \land (Child(n, m) 
\lor (\exists S. \neg S(n) \land \neg S(m) 
\land \forall x. S(x) \rightarrow 
Child(n, x) 
\lor Child(x, m) 
\lor \exists k. (Child(x, k) \lor Child(k, x)) \land S(k) )) 

\forall x. S(x) \rightarrow (Child(n,x) \rightarrow Label_c(x)) \land 
(\forall y. S(y) \land Child(x,y) \land Label_d(y)) 
(\forall y. S(y) \land Child(x,y) \rightarrow Label_c(y))

1. a//b and

2. labels follow the pattern:
   S_0 \rightarrow a (c d)* b

MSO for XML: Example
MSO Queries for XML

- Variables: node IDs, unary predicates
- Unary predicates for labels: $\text{Label}_a(n)$, $\text{Label}_b(n)$, ...
- Navigation in XML docs
  - vertical navigation: $\text{Child}(n, m)$
  - horizontal navigation: $n < m$

Combined via Boolean operators and first and second-order quantifiers $\exists n$ and $\exists S$
Deciding MSO over XML

Theorem:
MSO properties over trees are decidable in Linear time in data complexity

• For every MSO tree-property one can compute an equivalent tree automaton in PTIME

• Acceptance of a tree by a tree automaton decidable in Linear time
MSO Queries for RMCs

px-space of XML docs

px-space of XML docs

P = 0.3
P = 0.2
P = 0.5

RMC

distribution for $\varphi$

distribution for $\varphi$

$\varphi$

$\varphi$

$\varphi$

$\varphi$

$\varphi$

yes
P = 0.3

no
P = 0.7

yes
P = 0.3

no
P = 0.7
Verifying MSO for RMCs

- **Theorem**: For RMCs verifying MSO properties is in PSPACE.
- **SQRT-SUM** is PTIME reducible do deciding
  \[ \Pr(\phi \text{ is true}) \geq p \]
- Approximation of the probability is also hard.
Tractable RMC Subclasses

- Intractability: unrestricted components interact
- Hierarchical RMCs (HMC): A component can not (eventually) call itself
- Tree-like Markov chains (TLMC): Every component can be called in one place only
- Subclasses can be formalized using call hyper-graphs:
  - nodes are components of A
  - there is an edge from A to B if A calls B
- HMC ~ acyclic call graph, TLMC ~ call graphs without sharing

Goal is to achieve “tree shapeness” of RMCs
RMC Subclasses

RMC:

Call hyper-graph:

Cycles ~ not HMC
Sharing ~ not TLMC

There is an MSO query with irrational probability

No cycles ~ HMC
Sharing ~ not TLMC

There is an MSO query with double exp. small prob.
RMC Subclasses

RMC:

\[ D: \text{directory} \]

\[ P: \text{person} \]

\[ N: \text{name} \]

\[ T: \text{phone} \]

No cycles \sim HMC

No Sharing \sim TLMC

Every MSO query can be evaluated in PTIME

Call hyper-graph:

\[ D \rightarrow P \rightarrow N \rightarrow T \]
Tractability of MSO

• **Theorem:**
  HMC is *ra-tractable* for MSO (in data complexity)

• **ra-tractability:**
  • tractability in case of fixed-cost rational arithmetic
  • all arithmetic operations over rationals take unit time, no matter how large the numbers
Tractability of MSO

• **Theorem:** TLMC is tractable for MSO (in data complexity)

• **How it works:** Given TLMC A and MSO $\varphi$
  
  • TLMC A $\Rightarrow$ probabilistic push-down automat. (PPDA) B
  
  • MSO $\varphi$ $\Rightarrow$ MSO $\varphi'$ over the stack alphabet of B
  
  • MSO $\varphi'$ $\Rightarrow$ tree automaton B' (det. streaming tree aut.)
  
  • Construct B $\times$ B' - PPDA corresponding to a TLMC

  \[
  \Pr(A \models \varphi) = \Pr(B \times B' \text{ terminates})
  \]
Putting All Together

- **Expressiveness:** ability to generate
docs of any width ~ **wide**
docs of any depth ~ **deep**
docs with double EXP many leaves ~ **sharing**

- **Tractability for MSO:**
  - **double** underline ~ tractable
  - **single** underline ~ ra-tractable
  - **no** underlining ~ SQRT-SUM hard
Related Work

• All previous work on PXML is for shallow and narrow models [Kimelfeld et al.'07] [Senellart et al.'07]

• Query answering for these models
  • Tree Patterns with Joins [Kimelfeld et al.'07] [Senellart et al.'07]
  • MSO Queries [Cohen et al.'10]
  • Aggregate queries [Abiteboul et al.'10]

• Related notion was studied in [Cohen, Kimelfeld'10]
Conclusion

- A **very general RMC model** for PXML is adopted. It extends all the previous approaches by allowing
  - Deep models
  - Wide model
- MSO query answering is studied for RMC
  - “Intractability” is detected
- Tractable (**TLMC**) and ra-tractable (**HMC**) classes are isolated
Webdam Project: Foundations of Data Management
http://webdam.inria.fr

DataRing Project: P2P Data Sharing for Online Communities
http://www.lina.univ-nantes.fr/projets/DataRing/

ONTORULE Project: ONTOlogies Meets Business RULes
http://ontorule-project.eu/

Thank you
References


• [Cohen&Kimelfeld’10] - S. Cohen and B. Kimelfeld. Querying parse trees of stochastic context-free grammars. ICDT 2010
References


References

- [Kwiatkowska’03] - M. Z. Kwiatkowska: Model checking for probability and time: from theory to practice. LICS 2003