

Probabilistic XML: Modeling with RMCs and Querying with MSO

Evgeny Kharlamov

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

Michael Benedikt

Oxford University

Dan Olteanu

Oxford University

Pierre Senellart

Télécom ParisTech

KRDB Lunch Seminar, June 2010

Outline

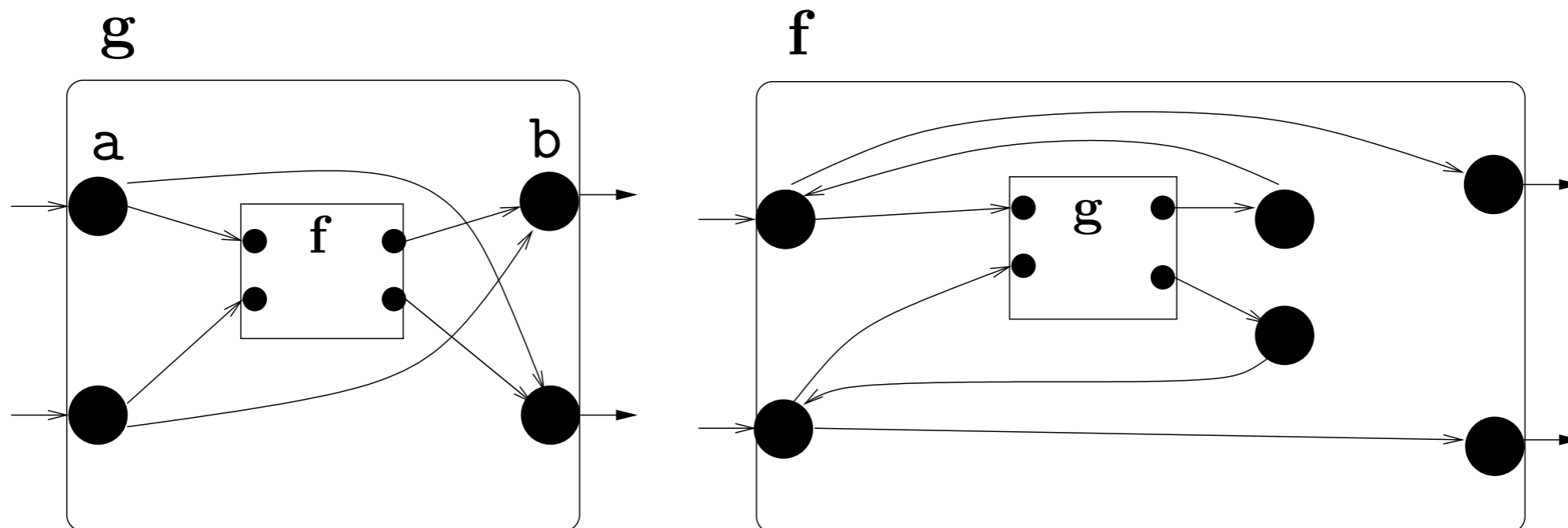
1. Recursive Markov Chains (RMCs)

[Etessami&Yannakakis'05, Etessami'06, Etessami&Yannakakis'09]

2. RMCs for Probabilistic XML

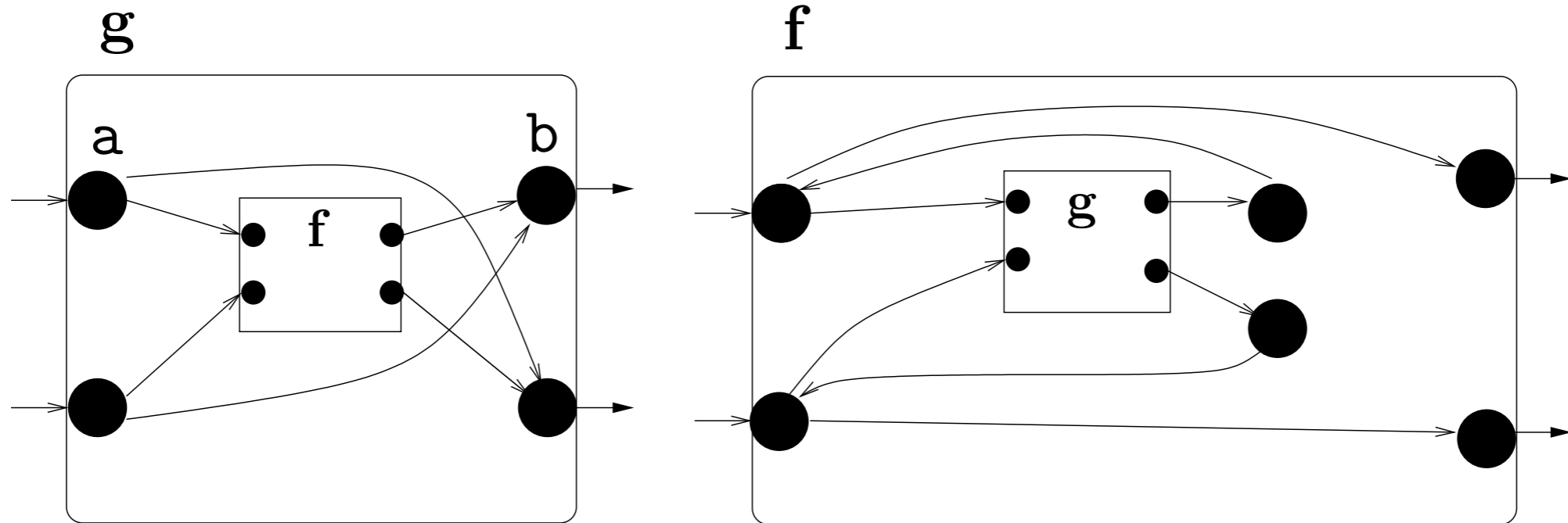
[Benedikt&al'10]

Recursive Graphs



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

Recursive Graphs

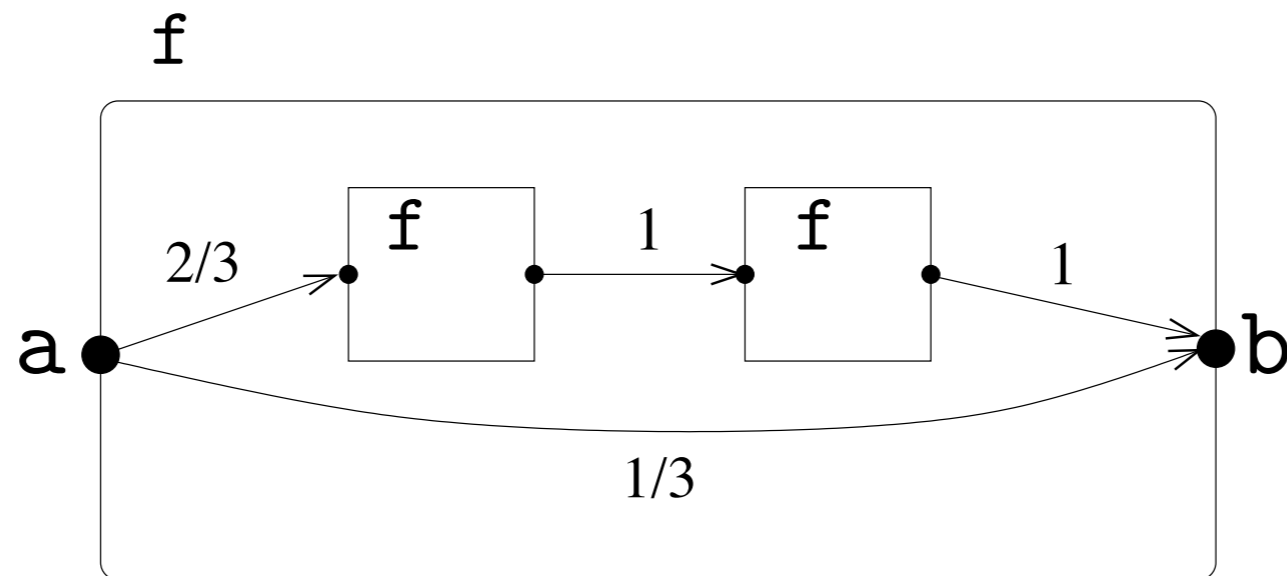


- Is it **possible** to reach **b** from **a**?
- Can be checked in
 - **cubic** time in general
 - **linear** time when either $\#entries$ or $\#exits$ is bounded in each component

Yes, but **not** always.

How to **measure** the frequency?

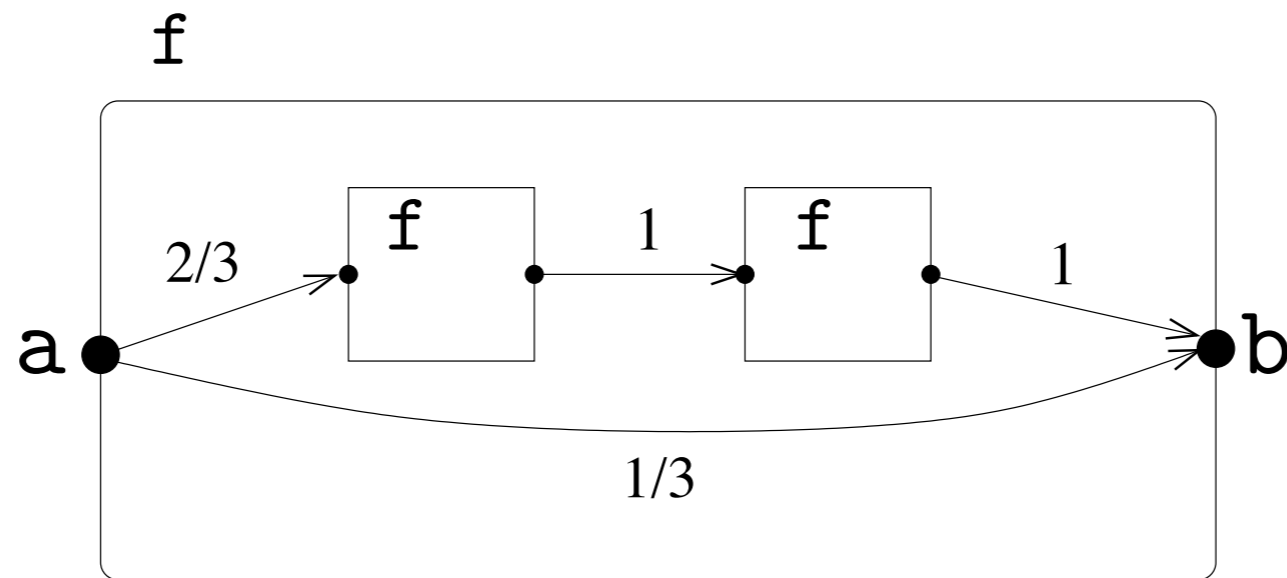
Recursive Markov Chains



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

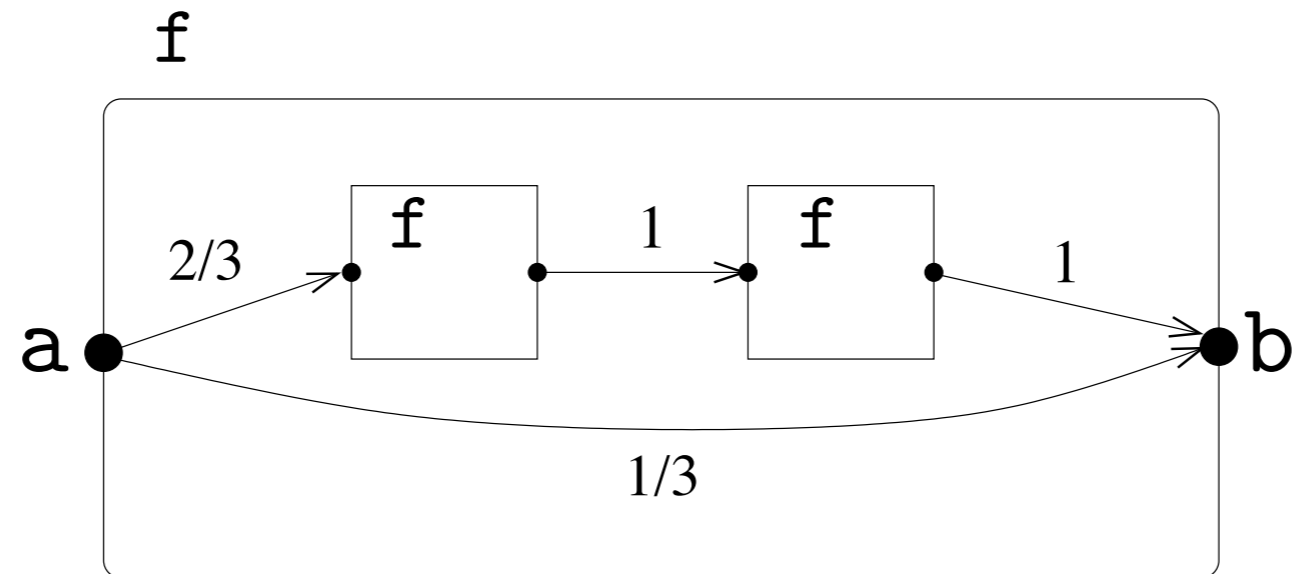
[Kwiatkowska'03]

Recursive Markov Chains



- Recursive graphs
- With **probabilities** on edges
- For every node,
the probabilities of outgoing edges sum up to 1

Recursive Markov Chains



- What is the **probability** of eventually reaching **b** from **a**?

Theorem [Etessami&Yannakakis'09] :
Least solution is the right answer

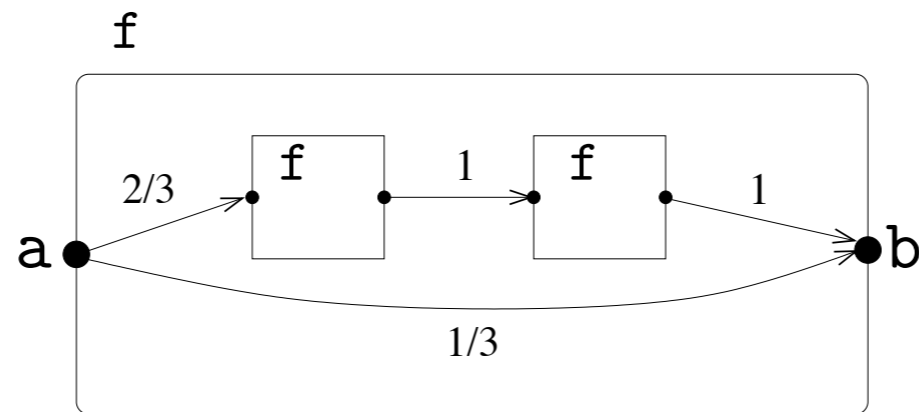
- Can be computed using non-linear equations:

$$x = 2/3 \cdot x^2 + 1/3 \quad \Rightarrow \quad x = 1/2 \text{ or } x = 1$$

- upper path: $2/3 \cdot x \cdot 1 \cdot x \cdot 1$
- lower path: $1/3$

x - probability of
going through **f**
termination probability

Computing Termination Probabilities



- Sum probabilities across different paths

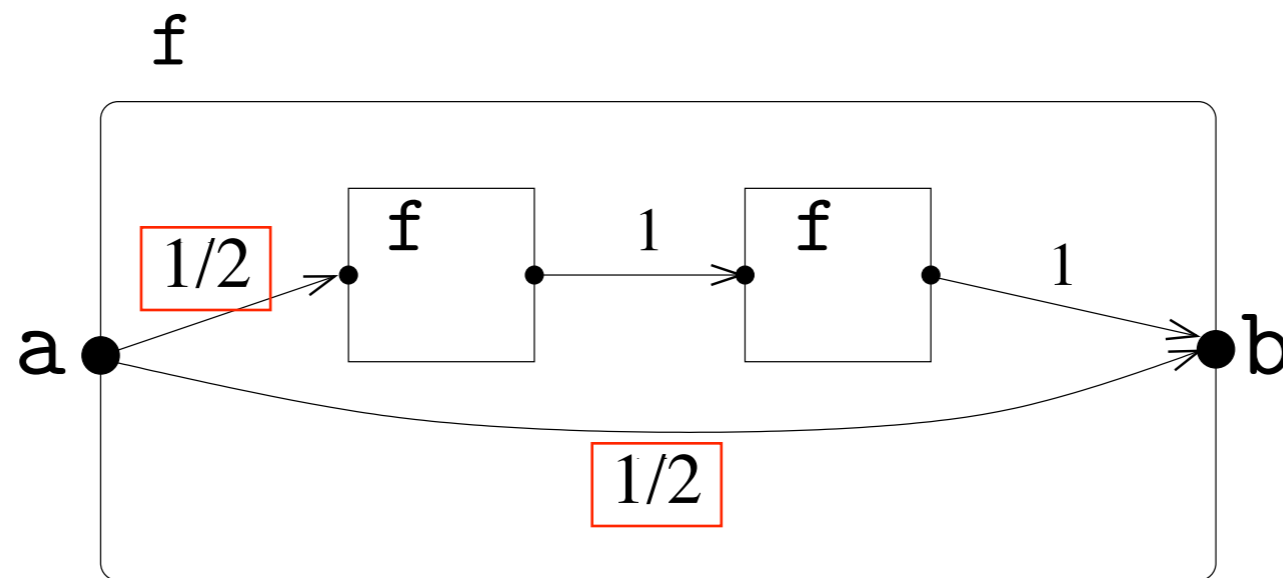
$$\Pr^*(a, b) = \sum_{(aa_1 \dots a_n b)\text{-path from } a \text{ to } b} \Pr(aa_1 \dots a_n b)$$

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

$$\Pr(aa_1 \dots a_n b) = \Pr(aa_1) \times \dots \times \Pr(a_n b)$$

- **Properties of termination probabilities**

Almost Sure Termination



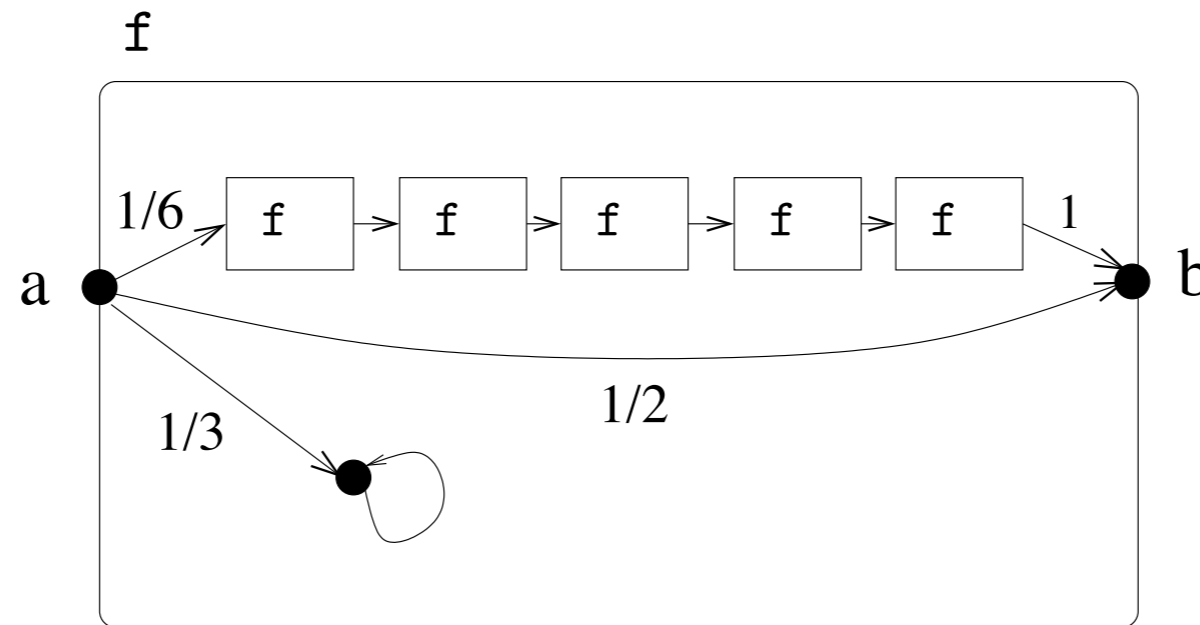
- Can infinite loops prevent from termination? **Not always**

$$x = 1/2 \cdot x^2 + 1/2 \quad \Rightarrow \quad x = 1 \text{ or } x = 0$$

- upper path: $1/2 \cdot x \cdot 1 \cdot x \cdot 1$
- lower path: $1/2$

In some cases we
almost surely
reach the exit

Irrational Termination Prob.s



- Are probabilities always rational? **No**

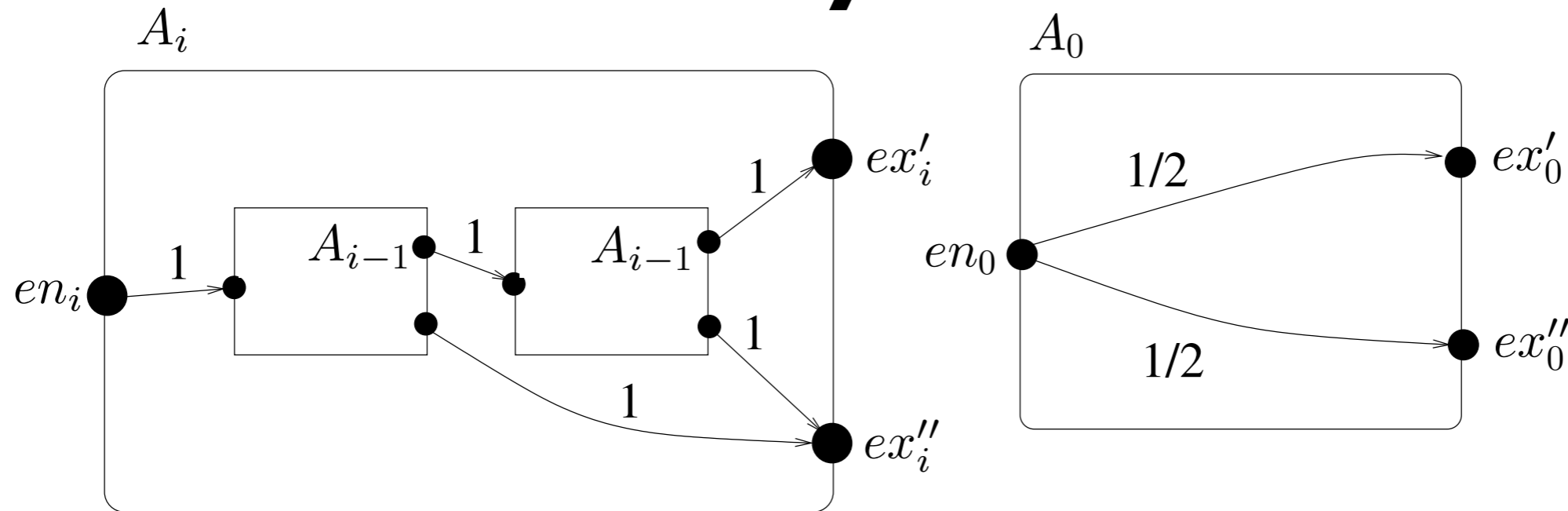
- Not:

$$x = 1/6 \cdot x^5 + 1/2 \quad \Rightarrow \quad x \sim 0.50550123$$

- not solvable by radicals
- For **finite Markov chains** reachability prob.s are **rational**

In some cases prob.s
are **irrational**

Reachability Probability

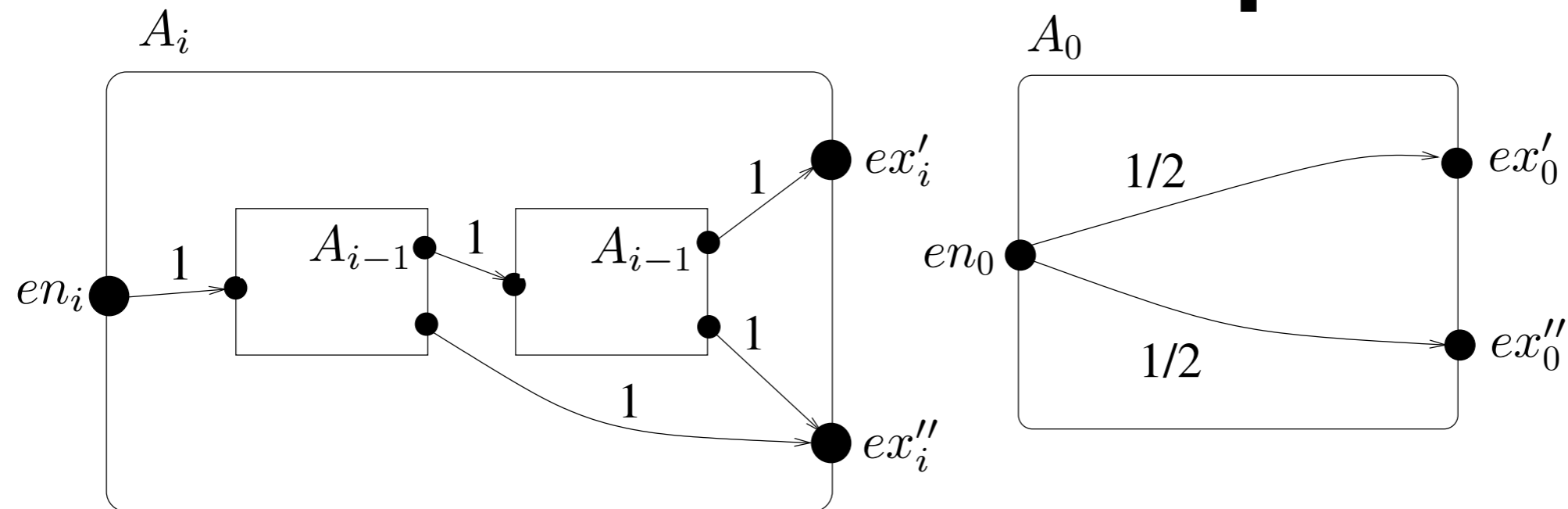


- Probability of reaching a specific exit of a component
- **Generalization** of termination probability

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

- A_i - given component
- en_i - starting point
- ex'_i - ending point

Double EXP Computation



- If prob.s are rational, is it always easy to compute? **No**

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

- 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

[Etessami&Yannakakis'09]

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

Qualitative

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

- For a class of **one exit** RMCs and $p \geq 0$
SQRT-SUM is PTIME reducible to deciding

Quantitative

$$\Pr(A \text{ terminates}) \geq p$$

Complexity of SQRT-SUM:
long standing **open** problem

SQRT-SUM: For k and d_1, \dots, d_n in \mathbb{N} , decide $\sum_i \sqrt{d_i} \geq k$

Hardness of Approximations

- **Theorem:**

[Etesami, 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

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

⇒ 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

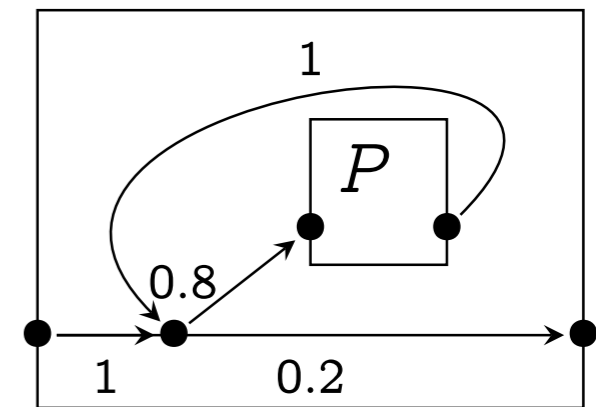
<!ELEMENT directory (person*)>

<!ELEMENT person (name,phone*)>

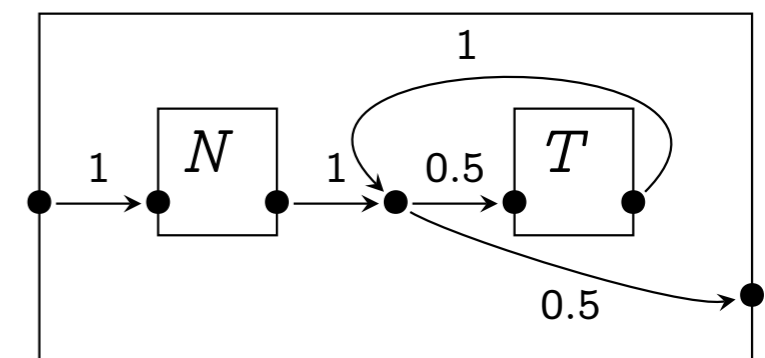
- Document d: <directory>
</directory> $\frac{Pr = 1 \cdot 0.2}{Pr(d) = 1 \cdot 0.2}$
- <directory>
 <person> $Pr = 1 \cdot 0.8$
 <name> $Pr = 1$
 </name> $Pr = 1$
 <phone> $Pr = 1 \cdot 0.5$
 </phone> $Pr = 1$
 </person> $Pr = 1 \cdot 0.5$
</directory> $Pr = 1 \cdot 0.2$

$$Pr(d) = 0.2 \cdot 0.8 \cdot 0.5 \cdot 0.2$$

D: directory

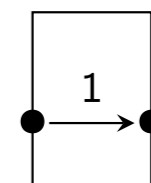
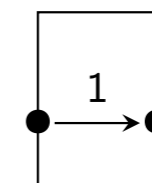


P: person



N: name

T: phone



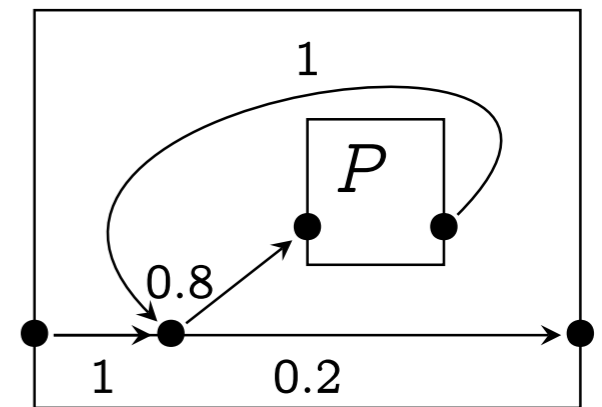
RMCs as Prob. DTDs

```
<!ELEMENT directory (person*)>
```

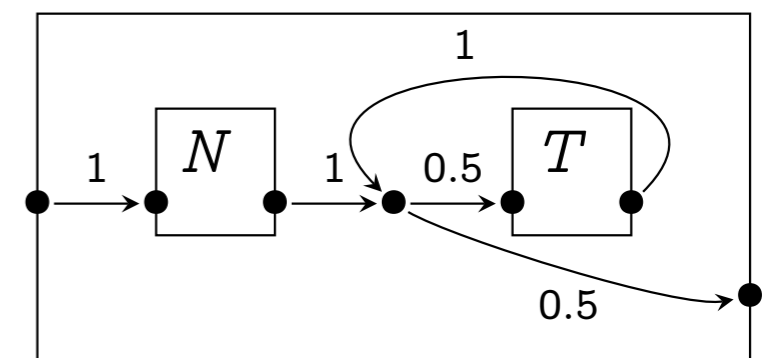
```
<!ELEMENT person (name,phone*)>
```

- 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

D : directory

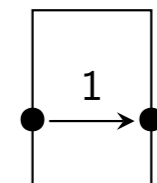
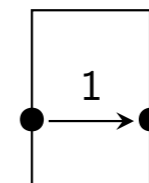


P : person



N : name

T : phone



RMCs as Prob. DTDs

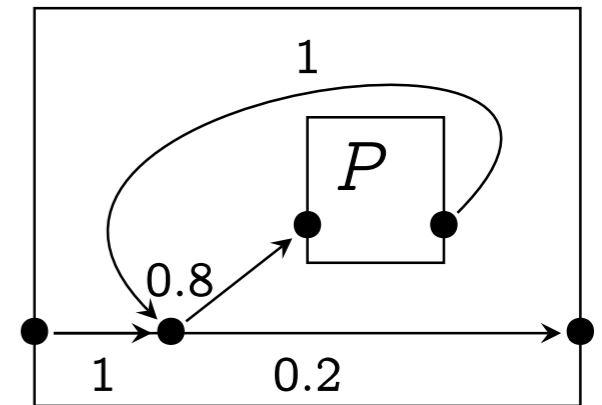
<!ELEMENT directory (person*)>

<!ELEMENT person (name,phone*)>

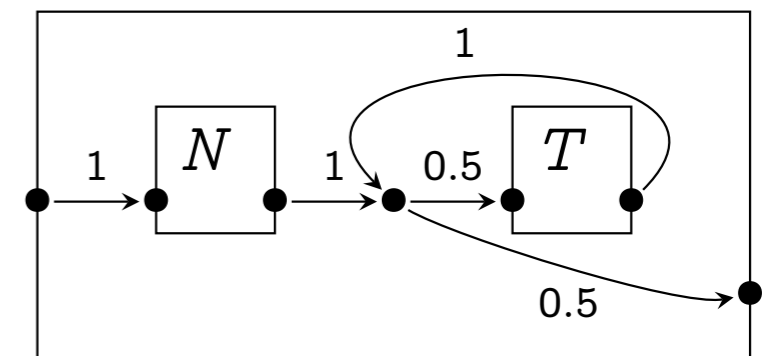
- Entering a component labeled L = generating an **opening** tag <L>
- Exiting a component labeled L = generating a **closing** tag </L>
- RMC A corresponds to a **probabilistic space** (px-space) of documents

$$[[A]] = \{(\text{srt}_1, p_1), \dots, (\text{srt}_n, p_n)\}$$

D: directory

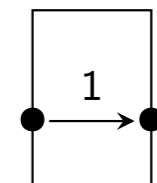
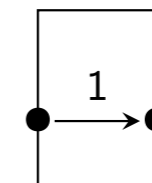


P: person



N: name

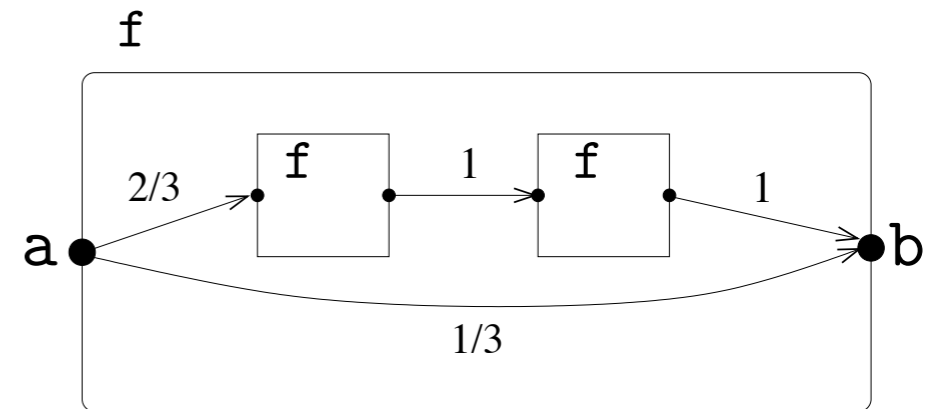
T: phone



Properties of PX-Spaces Generated by RMCs

1. Unbounded depth of generated trees

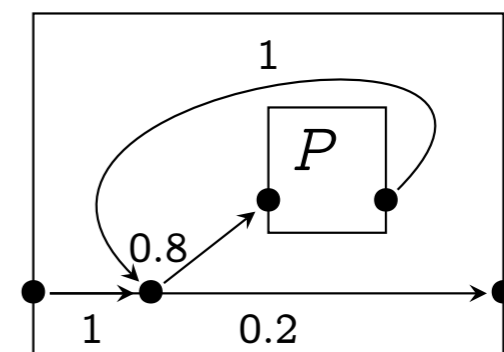
- **recursive** call of boxes
- recursive call \approx nesting in XML



2. Unbounded width of generated trees

- **looping** within boxes
- looping \approx siblings in XML

D: directory



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

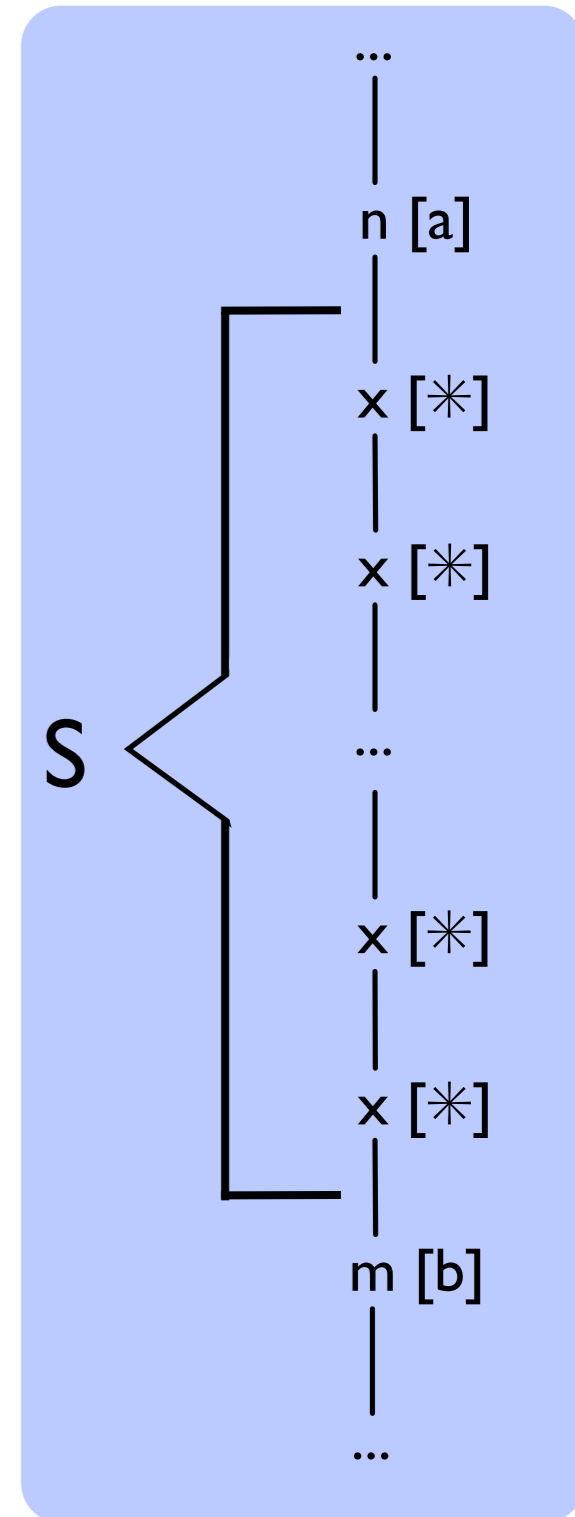
- $\exists n, m, k. \text{Child}(n, m) \wedge \text{Label}_a(n) \wedge \text{Label}_b(m) \wedge (m < k)$

- a/b[following-sibling(*)]
- no label predicate for a node k
~ wildcard for k

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

- a//b

- transitive closure of *Child* relations



MSO for XML: Example

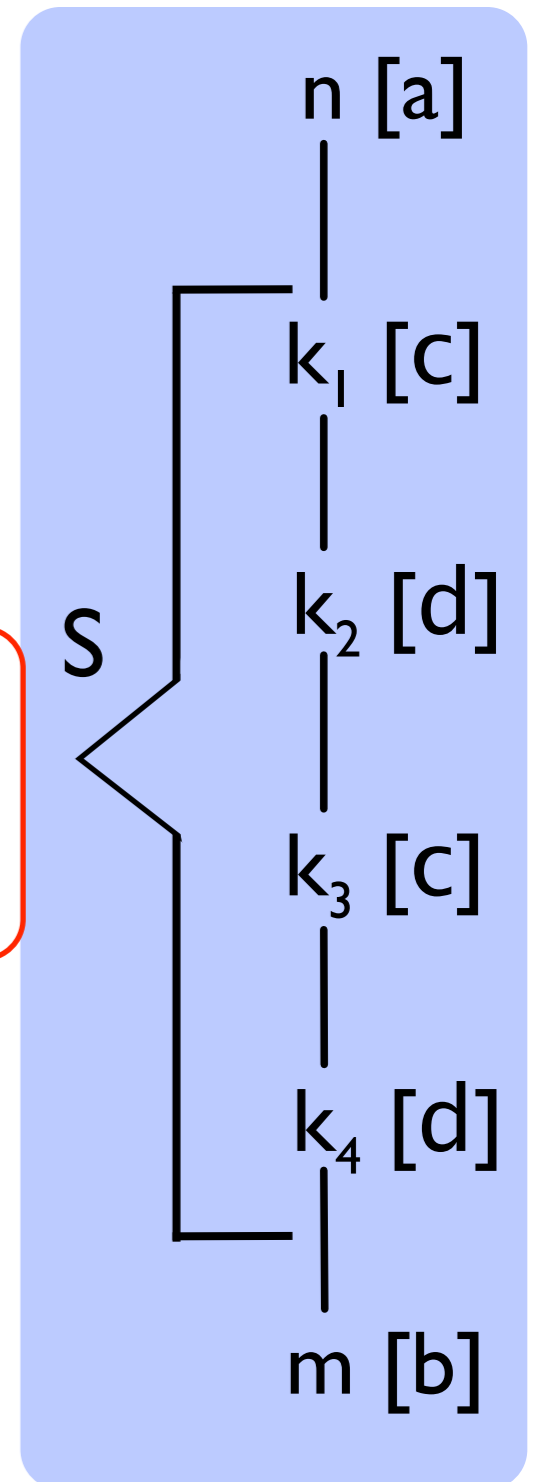
$$\begin{aligned} & \exists n, m. \text{Label}_a(n) \wedge \text{Label}_b(m) \wedge (\text{Child}(n, m) \\ & \vee (\exists S. \neg S(n) \wedge \neg S(m) \\ & \quad \wedge \forall x. S(x) \rightarrow \\ & \quad \quad \text{Child}(n, x) \\ & \quad \vee \text{Child}(x, m) \\ & \quad \vee \exists k. (\text{Child}(x, k) \vee \text{Child}(k, x)) \wedge S(k))) \end{aligned}$$

$$\begin{aligned} & \forall x. S(x) \rightarrow (\text{Child}(n, x) \rightarrow \text{Label}_c(x)) \wedge \\ & \quad (\text{Label}_c(x) \rightarrow \exists y. S(y) \wedge \text{Child}(x, y) \wedge \text{Label}_d(y)) \\ & \quad (\text{Label}_d(x) \wedge \exists y. S(y) \wedge \text{Child}(x, y) \rightarrow \text{Label}_c(y)) \end{aligned}$$

1. $a//b$ and

2. labels follow the pattern:

$$S_0 \rightarrow a (c d)^* b$$



MSO Queries for XML

- Variables: **node IDs**, unary **predicates**
- Unary predicates for labels: $Label_a(n)$, $Label_b(n)$, ...
- Navigation in XML docs
 - vertical navigation: $Child(n, m)$
 - horizontal navigation: $n < m$

Combined via Boolean operators and first and second-order quantifiers $\exists n$ and $\exists S$

Deciding MSO over XML

Theorem:

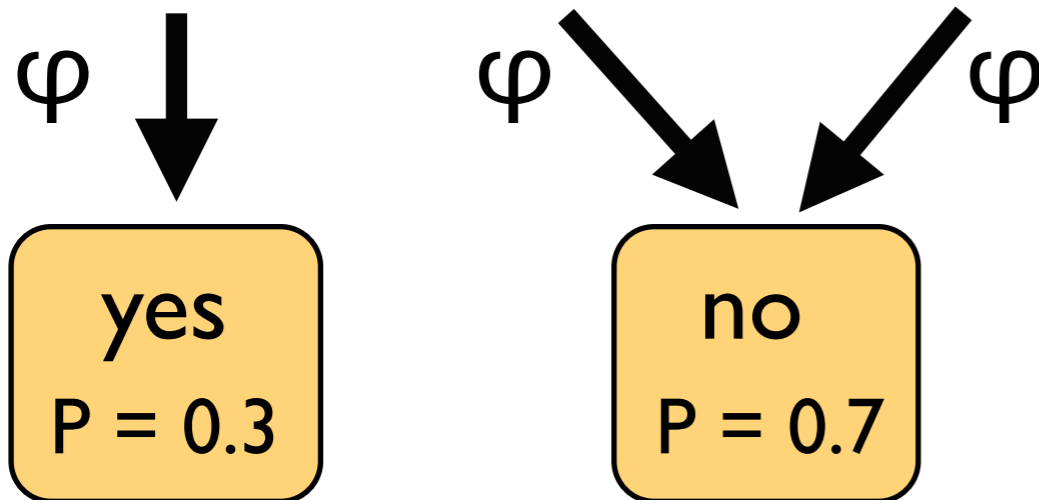
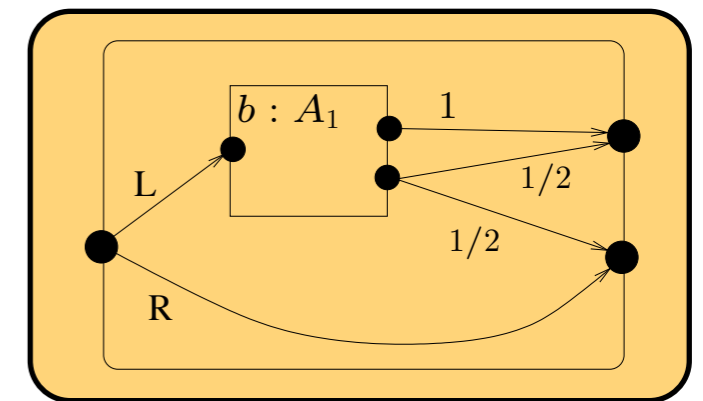
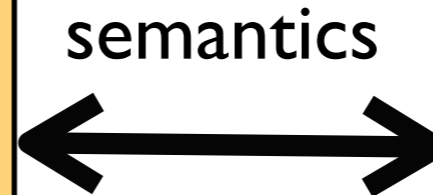
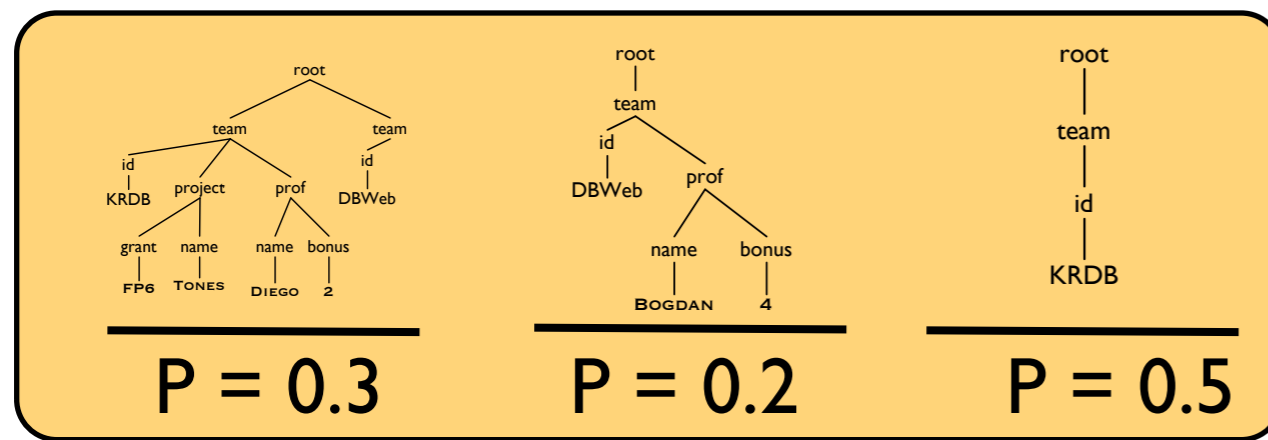
MSO properties over trees are decidable
in **Linear time** in **data complexity**

- 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**

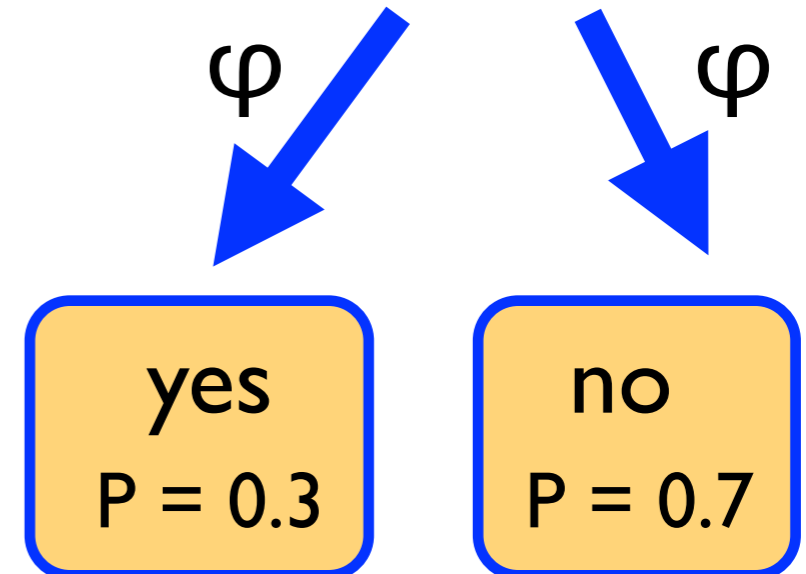
MSO Queries for RMCs

px-space of XML docs

RMC



distribution for φ



distribution for φ

Verifying MSO for RMCs

- **Theorem:** For RMCs verifying MSO properties

- Is in PSPACE

- SQRT-SUM is PTIME reducible to deciding

$$\Pr(\phi \text{ is true}) \geq p$$

- Approximation of the probability is also hard

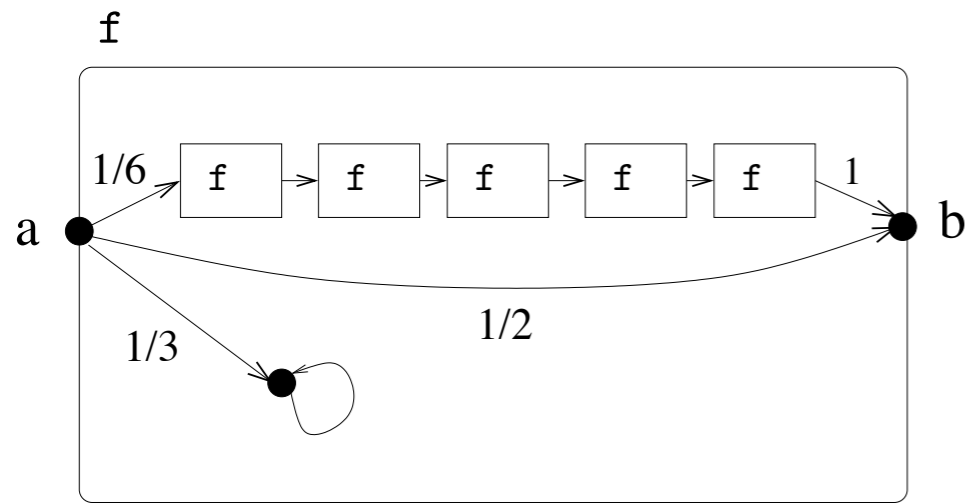
Tractable RMC Subclasses

- 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**

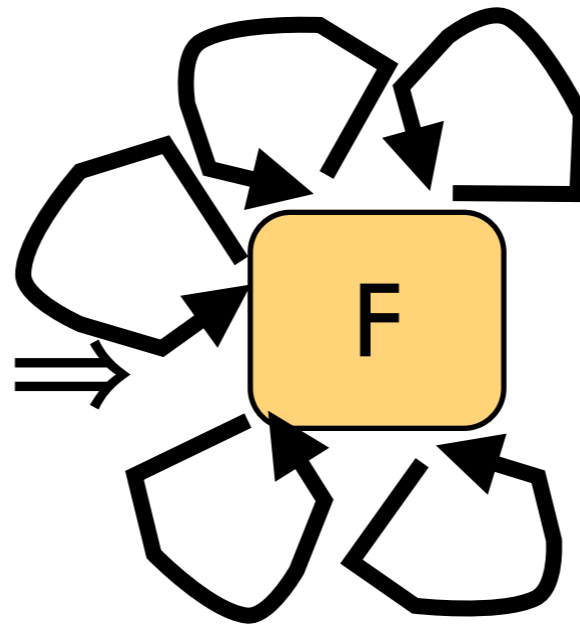
Goal is to achieve
“tree shapeness” of RMCs

RMC Subclasses

RMC:



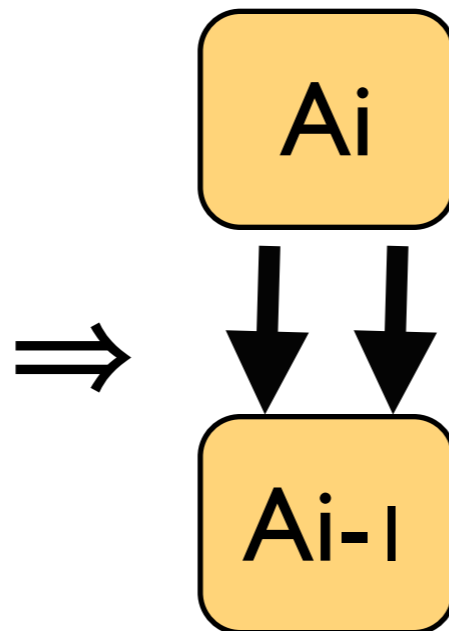
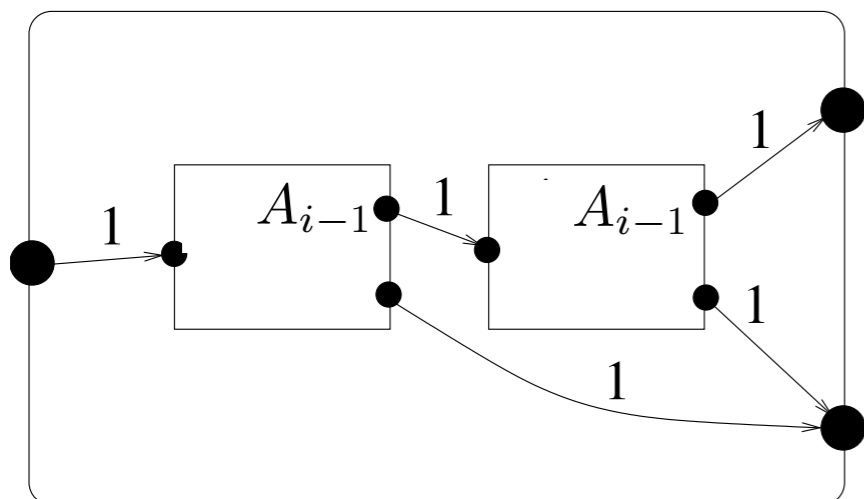
Call hyper-graph:



Cycles \sim not HMC
Sharing \sim not TLMC

There is an MSO query with **irrational probability**

A_i



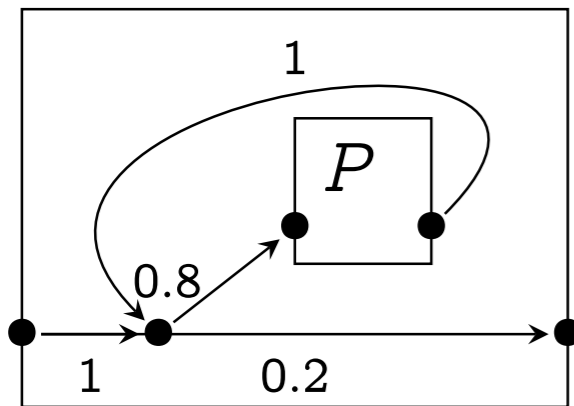
No cycles \sim HMC
Sharing \sim not TLMC

There is an MSO query with **double exp. small** prob.

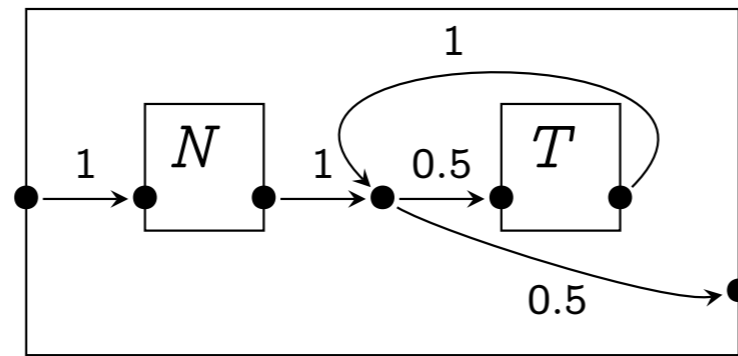
RMC Subclasses

RMC:

D: directory

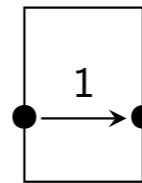
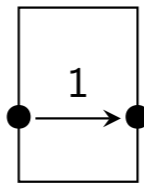


P: person

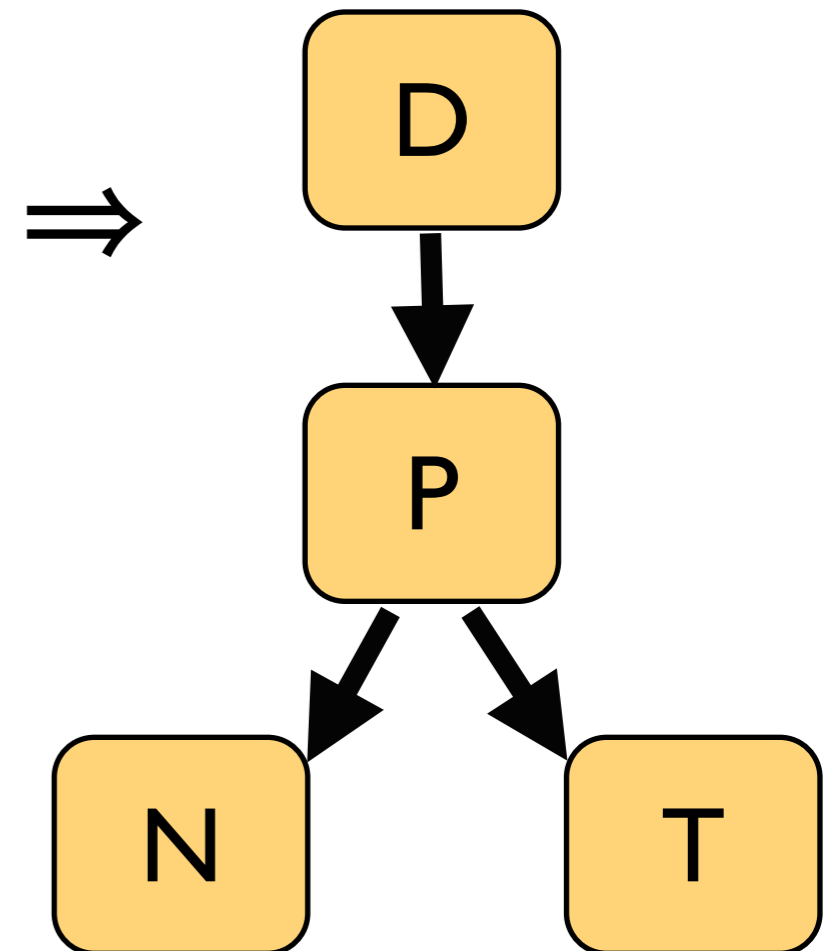


N: name

T: phone



Call hyper-graph:



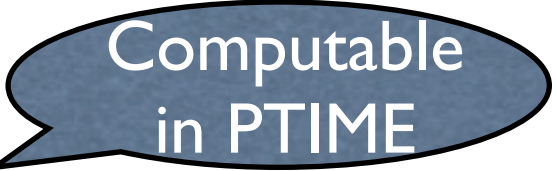
No cycles ~ HMC
No Sharing ~ TLMC

Every MSO query can be evaluated in **PTIME**

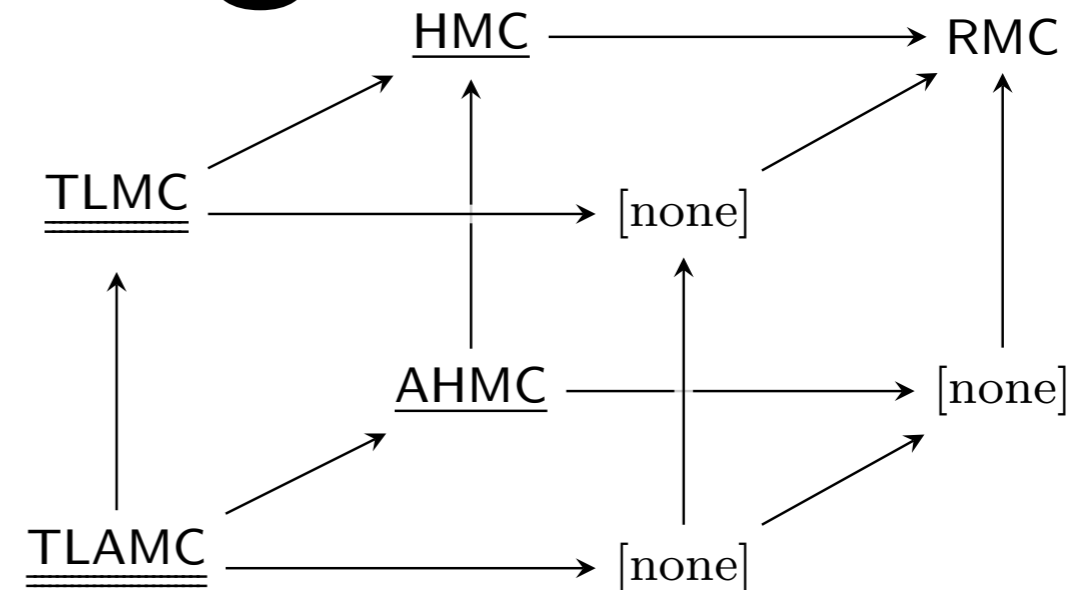
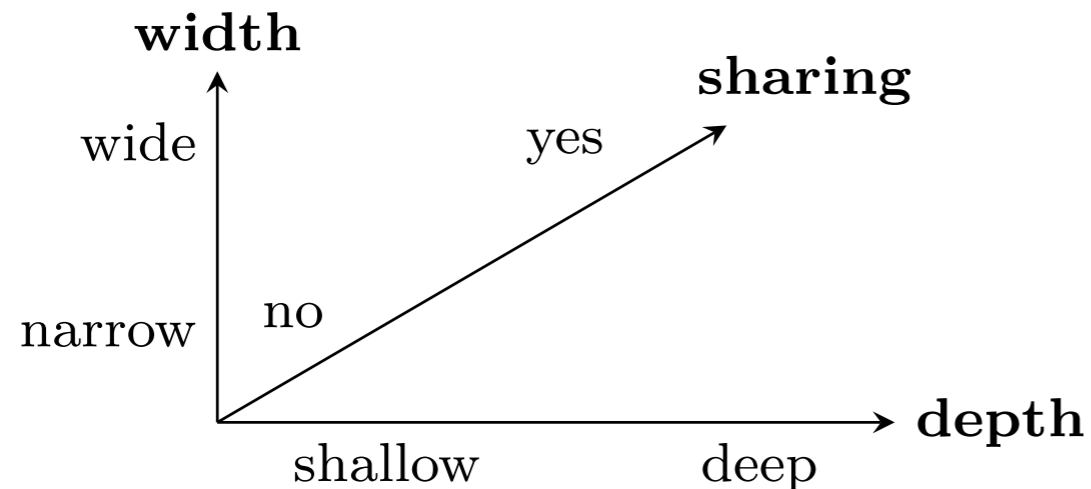
Tractability of MSO

- 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 φ
 - TLMC $A \Rightarrow$ probabilistic push-down automat. (PPDA) B
Linear
 - MSO $\varphi \Rightarrow$ MSO φ' over the stack alphabet of B
 - MSO $\varphi' \Rightarrow$ tree automaton B' (det. streaming tree aut.)
PTIME
 - Construct $B \times B'$ - PPDA corresponding to a TLMC
 - $\Pr(A \models \phi) = \Pr(B \times B' \text{ terminates})$ 

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

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/](http://www.lina.univ-nantes.fr/projets/DataRing/)



ONTORULE Project:

ONTologies Meets Business RULEs

<http://ontorule-project.eu/>

Thank you

References

- [\[Abiteboul&al'10\]](#) - S. Abiteboul, T-H. H. Chan, E. Kharlamov, W. Nutt, and P. Senellart, Aggregate Queries for Discrete and Continuous Probabilistic XML. ICDDT 2010
- [\[Alur&al'01\]](#) - R. Alur, K. Etessami, M. Yannakakis: Analysis of Recursive State Machines. CAV 2001
- [\[Benedikt&al'01\]](#) - M. Benedikt, P. Godefroid, T.W. Reps: Model Checking of Unrestricted Hierarchical State Machines. ICALP 2001
- [\[Benedikt&al'10\]](#) - M. Benedikt, E. Kharlamov, D. Olteanu, P. Senellart. Probabilistic XML via Markov Chains. Preprint
- [\[Cohen&Kimelfeld'10\]](#) - S. Cohen and B. Kimelfeld. Querying parse trees of stochastic context-free grammars. ICDDT 2010

References

- [\[Cohen&al'10\]](#) - S. Cohen, B. Kimelfeld, Y. Sagiv: Running tree automata on probabilistic XML. PODS 2009
- [\[Doner'70\]](#) - J. Doner: Tree Acceptors and Some of Their Applications. J. Comput. Syst. Sci. 4(5): 406-451. 1970
- [\[Etessami&Yannakakis'05\]](#) - K. Etessami, M. Yannakakis: Recursive Markov Chains, Stochastic Grammars, and Monotone Systems of Nonlinear Equations. STACS 2005
- [\[Etessami'06\]](#) - Slides of talks at Dagstuhl. Available at http://homepages.inf.ed.ac.uk/kousha/etessami_wamt_tutorial.pdf
- [\[Etessami&Yannakakis'09\]](#) - K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. JACM, 56(1), 2009.

References

- [\[Kimelfeld&al'07\]](#) - B. Kimelfeld, Y. Sagiv: Matching Twigs in Probabilistic XML. VLDB 2007
- [\[Kwiatkowska'03\]](#) - M. Z. Kwiatkowska: Model checking for probability and time: from theory to practice. LICS 2003
- [\[Senellart&al'07\]](#) - P. Senellart, S. Abiteboul: On the complexity of managing probabilistic XML data. PODS 2007
- [\[ThatcherWright'68\]](#) - J. W. Thatcher, J. B. Wright: Generalized Finite Automata Theory with an Application to a Decision Problem of Second-Order Logic. Mathematical Systems Theory 2(1): 57-81 (1968)