Recognizable Tree Series and Probability distributions for Trees

Rémi Gilleron

Inria Lille - Nord Europe & LIFL & Univ Lille

8 septembre 2009

Overview

Why ?

- Lecture in ENS Cachan : Learning queries
- Joachim was insistent 8-)

Motivations

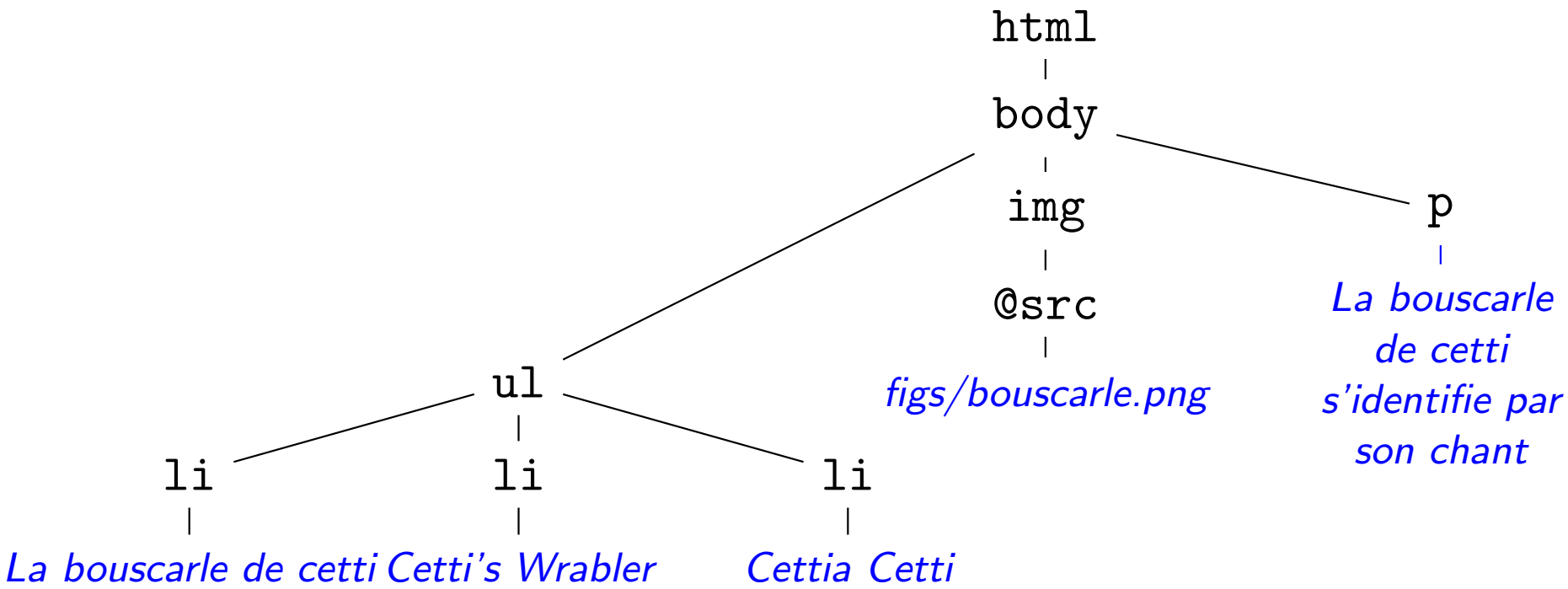
- Approximate learning for tree structured data
- Weighted tree automata as a tool

Objectives

- Recognizable tree series : automata and algebraic view
- Tree series over $\mathbb{R}$ and probabilistic tree automata
- What we are doing

Plan

XML Data



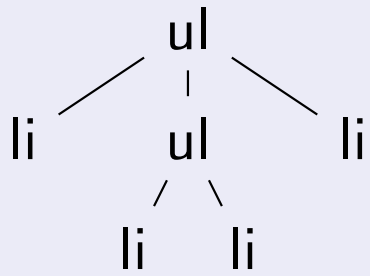
XML Trees

are sibling-ordered unranked labeled trees

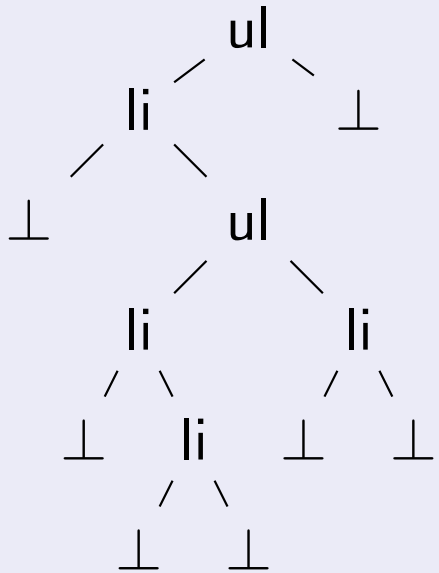
First-Child-Next-Sibling Encoding

allows to encode sibling-ordered unranked labeled trees into ranked trees.

An unranked tree



Its FCNS encoding

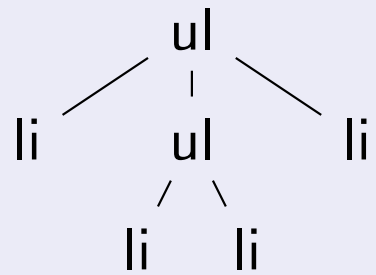


Curried Encoding of Unranked Trees

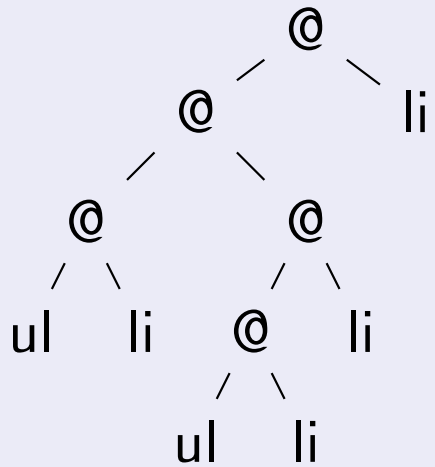
also allows to encode sibling-ordered unranked labeled trees into ranked trees in an algebraic way :

$$f(t_1, \ldots, t_p)@t' = f(t_1, \ldots, t_p, t')$$

An unranked tree



Its Curried encoding



Curried Encoding of Unranked Trees

also allows to encode sibling-ordered unranked labeled trees into ranked trees in an algebraic way :

$$f(t_1, \ldots, t_p)@t' = f(t_1, \ldots, t_p, t')$$

An unranked tree

ul

Its Curried encoding

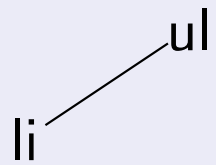
ul

Curried Encoding of Unranked Trees

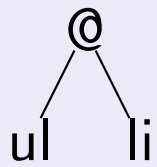
also allows to encode sibling-ordered unranked labeled trees into ranked trees in an algebraic way :

$$f(t_1, \ldots, t_p) @ t' = f(t_1, \ldots, t_p, t')$$

An unranked tree



Its Curried encoding

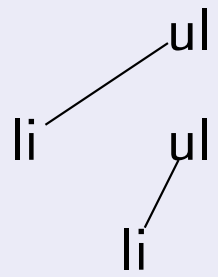


Curried Encoding of Unranked Trees

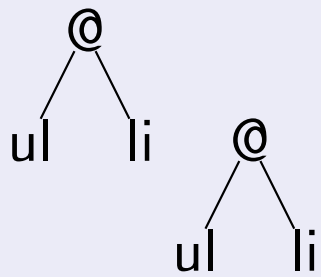
also allows to encode sibling-ordered unranked labeled trees into ranked trees in an algebraic way :

$$f(t_1, \ldots, t_p)@t' = f(t_1, \ldots, t_p, t')$$

An unranked tree



Its Curried encoding

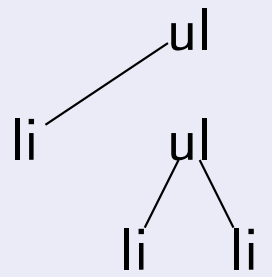


Curried Encoding of Unranked Trees

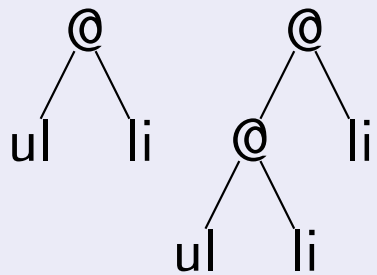
also allows to encode sibling-ordered unranked labeled trees into ranked trees in an algebraic way :

$$f(t_1, \ldots, t_p)@t' = f(t_1, \ldots, t_p, t')$$

An unranked tree



Its Curried encoding

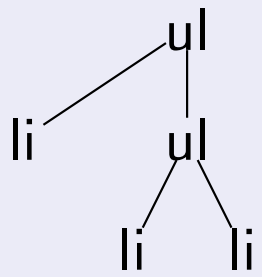


Curried Encoding of Unranked Trees

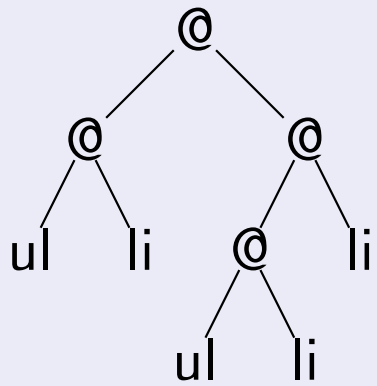
also allows to encode sibling-ordered unranked labeled trees into ranked trees in an algebraic way :

$$f(t_1, \ldots, t_p) @ t' = f(t_1, \ldots, t_p, t')$$

An unranked tree



Its Curried encoding

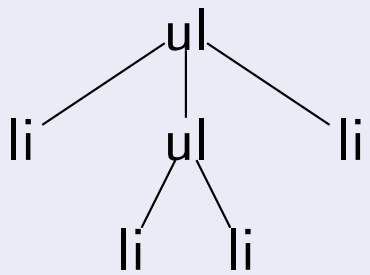


Curried Encoding of Unranked Trees

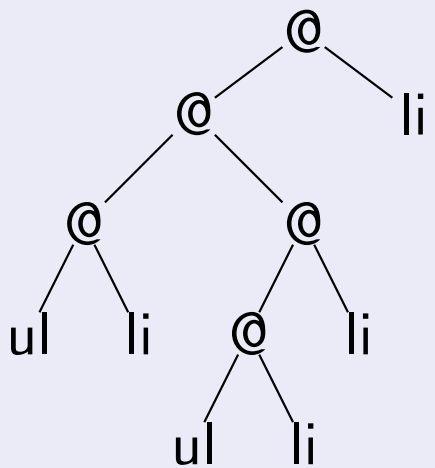
also allows to encode sibling-ordered unranked labeled trees into ranked trees in an algebraic way :

$$f(t_1, \ldots, t_p)@t' = f(t_1, \ldots, t_p, t')$$

An unranked tree



Its Curried encoding



Tree automata for unranked trees

See [TATA], chapter 8

Different Models

- Hedge Automata [Thatcher 67, Bruggemann-Klein 98, ...] defined on unranked trees
- *via* first-child-next-sibling encoding
- *via* Curried encoding [Carme et al 04]

All defining regular languages

- Closure properties, MSO definability
- The minimization problem, complexity results depend on the model
- Connection between XML and automata [Neven 02, ...]

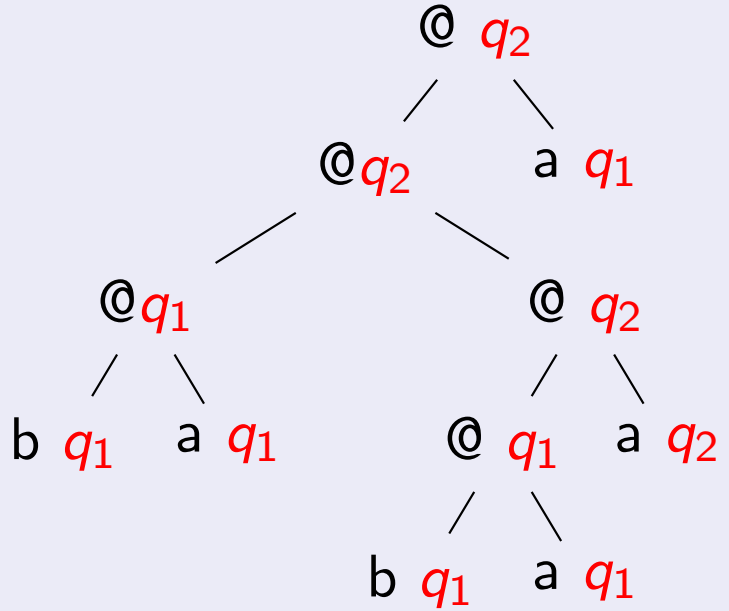
Thus, we mainly consider binary tree automata in the sequel.

Tree Automata by Example

A tree automaton for trees with an occurrence of a

$\mathcal{A} = (\Sigma, Q, Q_f, \Delta)$ where
$\Sigma = \{@(,), a, b\}$;
$Q = \{q_1, q_2\}$;
$Q_f = \{q_2\}$
$\Delta = \{a \to q_1$;
$a \to q_2$;
$b \to q_1$;
$@(q_1, q_1) \to q_1$;
$@(q_1, q_2) \to q_2$;
$@(q_2, q_1) \to q_2\}$.

A (successful) run r on a tree t



Hint : Guess one occurrence of a. There exists a successful run on a tree t if t contains one a.

Tree Automata

Bottom-up Tree Automata

- A bottom-up tree automaton (TA) $\mathcal{A}$ over a ranked alphabet Σ is a tuple (Q, Q_f, Δ) where Q is a finite set of states, $Q_f \subseteq Q$ is a set of final states, Δ is a finite set of rules : $@(q_1, q_2) \to q$
- A run r of $\mathcal{A}$ over t is a tree such that $\mathrm{nodes}(r) = \mathrm{nodes}(t)$ and r is compatible with Δ
- a run r is successful or accepting if $r(\epsilon) = q \in Q_f$
- The language $L(\mathcal{A})$ is the set of trees $t \in T(\Sigma)$ such that there exists a successful run of $\mathcal{A}$ over t
- A tree language L is regular or recognizable if $L = L(\mathcal{A})$ for some $\mathcal{A}$
- An automaton is deterministic (DTA) if there is no two rule with the same left-hand side

Properties of Tree Automata

Carry over from String Automata to Tree Automata

Closure properties, pumping lemma, determinization, minimization, Myhill-Nerode Theorem, decision results.

But

- Deterministic top-down automata (one initial state ; for every q, only one rule of the form $q \longrightarrow @(q_1, q_2))$ are strictly less powerful. Consider the recognizable (finite) language $L = \{@(a, b), @(b, a)\}$
- Regular tree languages are not closed under homomorphism because of non linear morphisms. For instance, consider $g(x) \longrightarrow f(x, x)$, and $\{f(t, t) \mid t \in T(\Sigma)\}$ is not recognizable ;
- Combinatorics and complexity results.

Plan

Beyond Tree Languages

Tree Automata as acceptors

i.e. they compute $\{0,1\}$-valued functions. Which functions can they compute ?

Over $\{@(,), a, b\}$, a tree automaton

- can check whether the number of a's is even
- can check whether the size of the tree is 2 modulo 3
- can not check whether the number of a's is equal to the number of b's

But

- For non deterministic automata, why do not use all runs ?
- Why do not consider more general functions ?

Tree automata to compute functions

Counting functions

Over $\{@(,), a, b\}$, can we define an automaton to compute using sums and products

- the size of a tree over $\mathbb{N}$?
- the height of a tree over $\mathbb{N}$?
- the probability of a tree over $\mathbb{R}$ w.r.t. a probability distribution over $T(\Sigma)$?

And more generally

- K-valued functions where $(K, +, \times, 0, 1)$ is a semiring ?
- The height of a tree over some semiring ?
- Tree transformations over a semiring ?

Counting the number of a's

A real-weighted tree automaton (WTA)

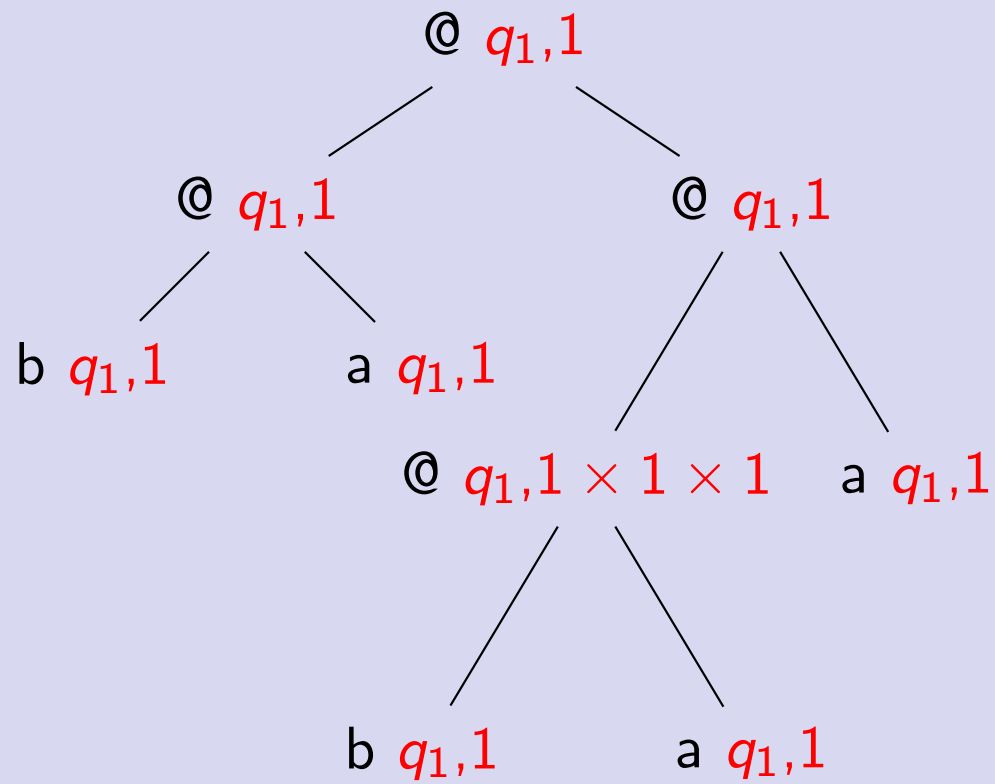
$\Sigma = \{@(,), a, b\}$;
$Q = \{q_1, q_2\}$;
$\Delta = \{a \to q_1(1)$;
$a \to q_2(1)$;
$b \to q_1(1)$;
$@(q_1, q_1) \to q_1(1)$;
$@(q_1, q_2) \to q_2(1)$;
$@(q_2, q_1) \to q_2(1)\}$.
final : $q1(0)$; $q_2(1)$

Counting the number of a's

A real-weighted tree automaton (WTA)

$\Sigma = \{@(,), a, b\}$;
$Q = \{q_1, q_2\}$;
$\Delta = \{a \to q_1(1)$;
$a \to q_2(1)$;
$b \to q_1(1)$;
$@(q_1, q_1) \to q_1(1)$;
$@(q_1, q_2) \to q_2(1)$;
$@(q_2, q_1) \to q_2(1)\}$.
final : $q1(0)$; $q_2(1)$

First Run : always choose q_1



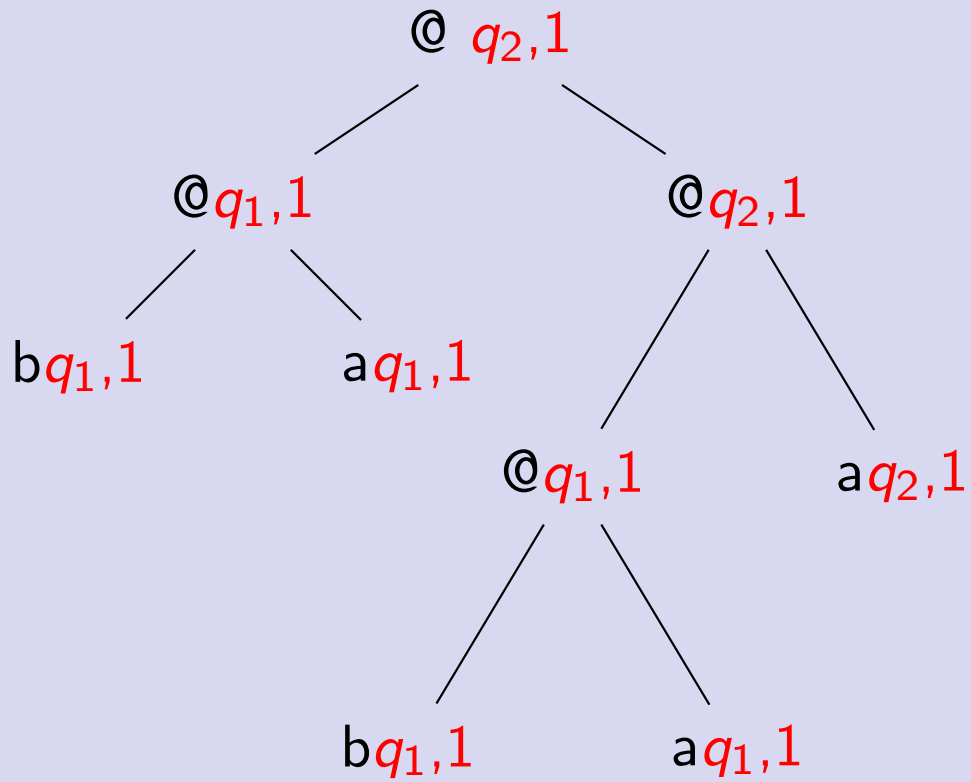
The final weight of this run on t is 1×0

Counting the number of a's

A real-weighted tree automaton (WTA)

$\Sigma = \{@(,), a, b\}$;
$Q = \{q_1, q_2\}$;
$\Delta = \{a \to q_1(1)$;
$a \to q_2(1)$;
$b \to q_1(1)$;
$@(q_1, q_1) \to q_1(1)$;
$@(q_1, q_2) \to q_2(1)$;
$@(q_2, q_1) \to q_2(1)\}$.
final : $q1(0)$; $q_2(1)$

Now choose q_2 for one of the a's



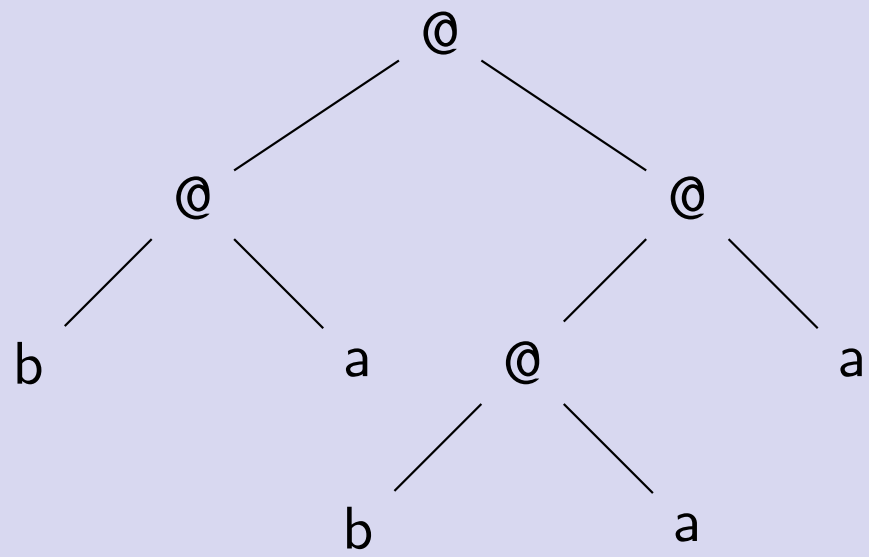
The final weight of this run on t is 1×1

Counting the number of a's

A real-weighted tree automaton (WTA)

$\Sigma = \{@(,), a, b\}$;
$Q = \{q_1, q_2\}$;
$\Delta = \{a \to q_1(1)$;
$a \to q_2(1)$;
$b \to q_1(1)$;
$@(q_1, q_1) \to q_1(1)$;
$@(q_1, q_2) \to q_2(1)$;
$@(q_2, q_1) \to q_2(1)\}$.
final : $q1(0)$; $q_2(1)$

All runs are considered



$1 \times 0 + 1 \times 1 + 1 \times 1 + 1 \times 1$

total weight : $\#_a(t)$!

Weighted Tree Automata

A WTA $\mathcal{A}$ over $(K, +, \times, 0, 1)$

- is (Σ, Q, F, Δ) where F is a function from Q into K, and Δ is a set of rules : $(f, q_1, \ldots, q_p, q, w)$
- bottom-up WTA : $f(q_1, \ldots, q_p) \longrightarrow q\ (w)$,
- top-down WTA : $q \longrightarrow f(q_1, \ldots, q_p)\ (w)$,
- bottom-up deterministic if one rule for fixed $f, q_1, \ldots, q_p$

Computing weights

- A run of $\mathcal{A}$ an a tree t is a tree r such that $\mathrm{nodes}(t) = \mathrm{nodes}(r)$, r has labels in Q, r is consistent with rules in Δ
- The weight $\Phi(t, r)$ is $\prod_{(f,q_1,\ldots,q_p,q,w)\in r} w \times F(r(\epsilon))$
- The weight $\Phi(t)$ is $\sum_{r\in \mathrm{runs}(t)} \Phi(t, r)$

Recognizable Tree Series

Tree Series

- A tree series is a mapping $r : T(\Sigma) \to K$ where $(K, +, \times, 0, 1)$ is a semiring. Why series ? $r = \Sigma_{t \in T(\Sigma)} r(t)t$.
- The support of a series is the set $\mathrm{Supp}(r) = \{t \in T(\Sigma) \mid r(t) \neq 0\}$
- The set of tree series is a vector space with $r_1 + r_2$ defined by : $\forall t$, $(r_1 + r_2)(t) = r_1(t) + r_2(t)$, and αr defined by : $\forall t$, $(\alpha r)(t) = \alpha \times r(t)$

Recognizable Tree Series

- a tree series is recognizable if it can be defined by a weighted tree automaton, or
- if it is a rational tree series via an algebraic definition

[Berstel and Reutenauer, Kuich and Vogler, Borchard, Maletti, ...]

Rational Tree Series

An algebraic view of WTA

A weighted tree automaton

$Q = \{q_1, q_2\}$;
$\Delta = \{a \to q_1(1)$;
$a \to q_2(1)$;
$b \to q_1(1)$;
$@(q_1, q_1) \to q_1(1)$;
$@(q_1, q_2) \to q_2(1)$;
$@(q_2, q_1) \to q_2(1)\}$.
final : $q1(0)$; $q_2(1)$

Algebraic presentation (V, μ, λ)

(e_1, e_2) basis of V ;
$\mu(a) = e_1 + e_2$;
$\mu(b) = e_1$;
$\mu(@)(e_1, e_1) = e_1$;
$\mu(@)(e_1, e_2) = e_2$;
$\mu(@)(e_2, e_1) = e_2$;
$\mu(@)(e_2, e_2) = 0$;
$\lambda(e_1) = 0$; $\lambda(e_2) = 1$

Definitions

V is a finite dimensional vector space over K, μ maps every symbol of arity p into a p-linear mapping from V^p into V, λ is a linear form.

Rational Tree Series

An algebraic view of WTA

A weighted tree automaton

$Q = \{q_1, q_2\}$;
$\Delta = \{a \to q_1(1)$;
$a \to q_2(1)$;
$b \to q_1(1)$;
$@(q_1, q_1) \to q_1(1)$;
$@(q_1, q_2) \to q_2(1)$;
$@(q_2, q_1) \to q_2(1)\}$.
final : $q1(0)$; $q_2(1)$

Algebraic presentation (V, μ, λ)

(e_1, e_2) basis of V ;
$\mu(a) = e_1 + e_2$;
$\mu(b) = e_1$;
$\mu(@)(e_1, e_1) = e_1$;
$\mu(@)(e_1, e_2) = e_2$;
$\mu(@)(e_2, e_1) = e_2$;
$\mu(@)(e_2, e_2) = 0$;
$\lambda(e_1) = 0$; $\lambda(e_2) = 1$

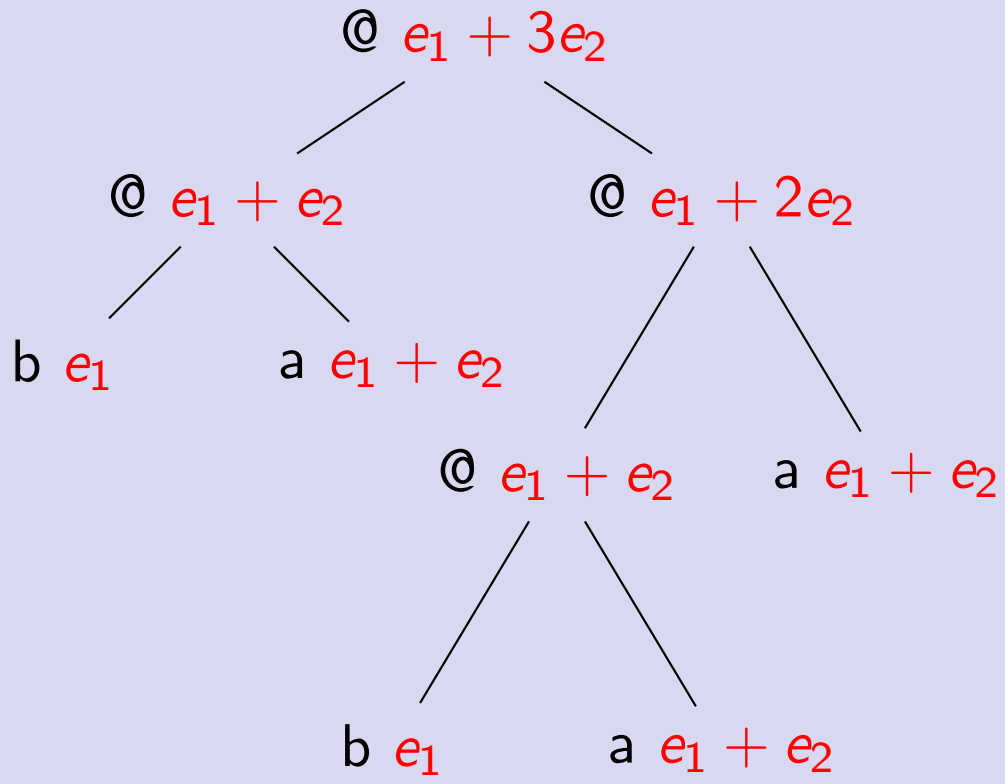
μ extends uniquely to a morphism

$$\begin{aligned}\mu(@(a, @(a, b))) &= \mu(@)(\mu(a), \mu(@(a, b))) \\ &= \mu(@)(\mu(a), \mu(@)(\mu(a), \mu(b))) \\ &= \mu(@)(e_1 + e_2, \mu(@)(e_1 + e_2, e_1))\end{aligned}$$

Computing $r(t)$ with an algebraic presentation

Algebraic presentation (V, μ, λ)

(e_1, e_2) basis of V ;
$\mu(a) = e_1 + e_2$;
$\mu(b) = e_1$;
$\mu(@)(e_1, e_1) = e_1$;
$\mu(@)(e_1, e_2) = e_2$;
$\mu(@)(e_2, e_1) = e_2$;
$\mu(@)(e_2, e_2) = 0$;
$\lambda(e_1) = 0$; $\lambda(e_2) = 1$



$$\mu(t) = e_1 + 3e_2$$
$$r(t) = \lambda(\mu(t)) = 3$$

Definition

r is a rational tree series if there exists (V, μ, λ) where V is a finite dimensional vector space over K, μ maps every symbol of arity p into a p-linear mapping from V^p into V, and λ is a linear form, such that, $\forall t \in T(\Sigma)$, $r(t) = \lambda(\mu(t))$.

Theorem

- For a commutative semiring, a tree series is recognizable if and only it is rational, i.e.
- summation at each step (algebraic view) and summation over all runs (weighted automata view) coincide.

Plan

Operations on tree series

Theorem

The set of recognizable tree series is a subvector space of the vector space of tree series. Also, the hadamard product of recognizable tree series is recognizable.

Proof

- If (V, μ, λ) is an algebraic representation of r, then $(V, \mu, \alpha\lambda)$ is a representation of αr
- Let assume that r_1 and r_2 are recognizable, and let (V_i, μ_i, λ_i) be a representation of S_i. We define
  - $V = V_1 \times V_2$,
  - $\mu(@)((v_1^1, v_1^2), (v_2^1, v_2^2)) = (\mu_1(@)(v_1^1, v_2^1), \mu_2(@)(v_1^2, v_2^2))$
  - $\lambda(v^1, v^2) = \lambda_1(v^1) + \lambda_2(v^2)$

  μ is well-defined, μ is p-linear, and we get
  $\lambda(\mu(t)) = \lambda_1(\mu_1(t)) + \lambda_2(\mu_2(t)) = r_1(t) + r_2(t) = (r_1 + r_2)(t)$

Recognizable series and recognizable languages

Characteristic series

r_L of a tree language L is defined by $r_L(t) = 1$ if $t \in L$, and 0 otherwise.
The characteristic series of a regular tree language is recognizable

Proof

Let $\mathcal{A}$ be a deterministic bottom-up automaton recognizing L. Consider the vector space $\mathbb{R}^Q$ with the canonical base $(e_q)_{q \in Q}$. Then

- μ is defined by $\mu(@)(e_{q_1}, e_{q_2}) = e_q$ where $@(q_1, q_2) \to q$ is a rule of $\mathcal{A}$
- $\lambda(e_q) = 1$ if q is final, and 0 otherwise

Now prove that $r_L(t) = 1$ if and only if $t \in L = L(\mathcal{A})$.
Note that if $\mathcal{A}$ is non deterministic, replace, in the definition of μ, e_q by $\sum e_q$. This defines a tree series which takes into account the multiplicity of acceptance of a tree.

And conversely ?

Computing the size

Size is a recognizable tree series over $(\mathbb{R}, +, \times, 0, 1)$

Proof

For every $f \in \Sigma$, the tree series $\#_f$ is recognizable (construction of a weighted automaton as shown before). Then, as $\mathrm{Size} = \sum \#_f$, use the closure by $+$

Warning

The support of a recognizable tree series may be a non recognizable tree language.

Proof

The series $\#_a - \#_b$ is recognizable but its support is not a recognizable tree language. Note that we use subtraction.

Which semiring ? Deterministic automata ?

The series Size

- is a recognizable tree series over $(\mathbb{R}, +, \times, 0, 1)$,
- is not deterministically recognizable over $(\mathbb{R}, +, \times, 0, 1)$,
- is deterministically recognizable over the tropical (or Min-Plus) semiring $(\mathbb{N} \cup \{+\infty\}, \min, +, +\infty, 0)$.

The series Height

- is not a recognizable tree series over $(\mathbb{R}, +, \times, 0, 1)$,
- is recognizable over the artic (or Max-Plus) semiring $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$,
- is not deterministically recognizable over the artic semiring.

Deterministically recognizable tree series

Deterministic weighted tree automata

A weighted automaton is deterministic if $\forall(q_1, q_2)$, there exists at most one rule $@(q_1, q_2) \to q(w)$ with $w \neq 0$.

The power Set Construction [Borchard2004]

for weighted tree automata only if $\{\mu(t) \mid t \in T(\Sigma)\}$ is finite. This is the case if the semiring is locally finite (every finitely generated subsemiring is finite). A pumping lemma can be designed.

Myhill-Nerode Theorem [Borchard2003], [Maletti2008]

Let us suppose that K is a semifield

- The congruence $\equiv_r$ is defined by : $u \equiv_v v$ if, there exists α in $K - \{0\}$ such that, for every C, $r(C[u]) = \alpha r(C[v])$.
- r is deterministically recognizable if and only if $\equiv_r$ has finite index.

.

Example of tree transformation

Tree pattern matching

Let $\Sigma = \{@(,), a\}$ and consider the semiring $(2^{\mathbb{N}^*}, \cup, \circ, \emptyset, \{\epsilon\})$
Let us consider a pattern $P = @(@(a, a), x)$. The automaton
$Q = \{q_\perp, q_a, q_{f(a,a)}, q_P\}$; and
$\Delta = \{a \to q_\perp\ (\{\epsilon\})$; $\quad a \to q_a\ (\{\epsilon\})$;
$@(q_\perp, q_\perp) \to q_\perp\ (\{\epsilon\})$; $\quad @(q_a, q_a) \to q_{f(a,a)}\ (\{\epsilon\})$;
$@(q_{f(a,a)}, q_\perp) \to q_P\ (\{\epsilon\})$;
$\forall q \neq q_P,\ @(q_P, q) \to q_P\ (\{1\})$; $\quad @(q, q_P) \to q_P\ (\{2\})\}$;
and final weights : $q_P\ (\{1\})$, $q\ (\{\epsilon\})$ for $q \neq q_P$.
computes, for every tree t, the set of paths leading to pattern P in t.

Tree series transducers

generalize over tree transducers. See [Maletti's thesis 2006].

Plan

Tree Series and Probability Distributions

Probability distributions over $T(\Sigma)$

A probability distribution over $T(\Sigma)$ is a mapping p from $T(\Sigma)$ into $\mathbb{R}$ such that

- $\forall t \in T(\Sigma),\ 0 \leq p(t) \leq 1$,
- $\sum_{t \in T(\Sigma)} p(t) = 1$.

Stochastic Tree Series

A (recognizable) tree series is stochastic if it defines a probability distribution over $T(\Sigma)$. Questions :

- Tree series over the field $(\mathbb{R}, +, \times, 0, 1)$ or commutative semifields $(\mathbb{R}_+, +, \times, 0, 1)$ or $([0, 1], +, \times, 0, 1)$?
- What is $\sum_{t \in T(\Sigma)} r(t)$?
- Is it decidable whether a recognizable tree series is stochastic ?
- Expressiveness w.r.t. properties of weighted tree automata ?

Convergence of Tree Series

Definitions

An $\mathbb{R}$-tree series r is convergent if the the sequence $\left(\sum_{\mathrm{Size}(t)\leq n} r(t)\right)_{n\geq 1}$ has a finite limit which is denoted by $\sum_{t\in T(\Sigma)} r(t)$;
An $\mathbb{R}$-tree series r is absolutely convergent if $|r|$ is convergent ;
An $\mathbb{R}$-tree series r is strongly convergent if $\mathrm{Size}(t) \times r$ is convergent.

Properties

- alsolute convergence implies convergence
- strong convergence implies convergence
- robust definitions w.r.t. Curried encoding.

Probabilistic Tree Automata (PTA)

Idea : introduce normalizing conditions to define probability distributions

Definition

Probabilistic tree automata are weighted tree automata

- over $([0,1], +, \times, 0, 1)$, and such that
- $\sum_{q \in Q} F(q) = 1$
- $\forall q \in Q$, $\sum_{(f, q_1, \ldots, q_p, q, w) \in \Delta} w = 1$

Example

A family of probabilistic tree automata $\mathcal{A}_{\alpha,\beta,\gamma}$ parameterized by α, β, γ in $[0,1]$ with rules $\{A \to a\,(1-\alpha)\,;\ A \to @(A,B)\,(\alpha)\,;$
$B \to b\,(1-\beta)\,;\ B \to @(B,B)\,(\beta)\,;$
and final weights $A\,(\gamma)$, $B\,(1-\gamma)$

In the string case, probabilistic automata define stochastic series. And for trees ?

A PTA may not define a stochastic series

A counter-example

adapted from [Wetherell80] for probabilistic context-free grammars.
Consider $\gamma = 0$ leading to set of rules $B \to b\,(1-\beta)$; $B \to @(B,B)\,(\beta)$, and final weight B (1). This defines series r_β.

r_β is stochastic if and only if $\beta \leq \frac{1}{2}$

Proof

Let $s_n = \sum_{\mathrm{Height(t)} \leq n} r_\beta(t)$. It is easy to show :

$$\forall n, s_{n+1} = \beta \times s_n^2 + (1-\beta), \text{ and } s_n \leq \min\{1, \frac{1-\beta}{\beta}\}.$$

Thus, $\forall \beta \in [0,1]$, r_β is (absolutely) convergent and
$\sum_{t \in T(\Sigma)} r_\beta(t) = 1$ if and only if $\beta \leq \frac{1}{2}$.
For $\beta > \frac{1}{2}$, $\sum_{t \in T(\Sigma)} r_\beta(t) = \frac{1-\beta}{\beta} < 1$.
For $\beta = \frac{1}{2}$, the series is not strongly convergent.

The stochasticity problem

Stochasticity of weighted tree automata

Instance : a weighted tree automaton $\mathcal{A}$ over $\mathbb{R}$.
Answer : "yes" if and only if the recognizable tree series $r_{\mathcal{A}}$ is stochastic.
is undecidable. It can be proved to be a consequence of the undecidability of the emptiness problem for (string) weighted automata over $\mathbb{R}$.

Stochasticity of probabilistic tree automata

Instance : a probabilistic tree automaton $\mathcal{A}$.
Answer : "yes" if and only if the recognizable tree series $r_{\mathcal{A}}$ is stochastic.
is decidable by a result of [EtessamiYannakakis05]. Recall that the problem is nonsense in the string case.

Deciding stochasticity (1)

Notations

Let $\mathcal{A}$ be a probabilistic tree automaton. Let us consider the square matrix M indexed by the state set Q of $\mathcal{A}$ and m_{ij} is the expected number of occurences of q^j when applying a rule with left-hand side q^i.
Example : $\mathcal{A}_{\alpha,\beta}$ with rules $\{A \to a\,(1-\alpha)\,;\ A \to @(A,B)\,(\alpha)\,;$ $B \to b\,(1-\beta)\,;\ B \to @(B,B)\,(\beta)$. Then

$$M_{\alpha,\beta} = \begin{pmatrix} \alpha & \alpha \\ 0 & 2\beta \end{pmatrix}$$

Definition

A probabilistic tree automaton $\mathcal{A}$ is said to be strongly consistent if $\rho(M) < 1$, where $\rho(M)$ is the spectral radius of M, i.e. the maximum among the modulus of the eigenvalues of M.

Deciding stochasticity

Easy case : $\rho(M) < 1$

- It is decidable using linear algebra results.
- The series $r_{\mathcal{A}}$ is stochastic.
- M^n converges to 0 and $M^n = (I - M)^{-1}$ gives the expected size of trees according to $r_{\mathcal{A}}$.

Difficult case : $\rho(M) = 1$

- The series $r_{\mathcal{A}}$ is not necessarily stochastic.
- Etessami and Yannakakis proved that it is decidable in polynomial time whether the series is stochastic.

Example continued

- $\rho(M_{\alpha,\beta}) = \max\{\alpha, 2\beta\}$
- $\mathcal{A}_{\alpha,\beta}$ is strongly consistent if $\alpha < 1$ and $\beta < \frac{1}{2}$.

Expressiveness

Main results

- $\mathcal{S}(PTA) = \mathcal{S}(WTA_+)$: rules with nonnegative weights can be normalized if the series is known to be stochastic, but ...
- $\mathcal{S}(WTA_+) \subset \mathcal{S}(WTA)$: there are more probability distributions when allowing negative weights. Already true in the string case.
- $\mathcal{S}(PTA) \subset \mathcal{N}(PTA)$: a PTA does not always define a probability distribution.
- $\mathcal{R}(WDTA) \subset \mathcal{R}(WTA)$ and $\mathcal{S}(PDTA) \subset \mathcal{S}(PTA)$, i.e. determinism is a restriction.

Plan

What have we done ? (automata)

Weighted unranked tree automata

- Weighted hedge automata and weighted stepwise automata are equivalent, also [Hogberg et al 09].
- Also holds for probabilistic automata, i.e. with normalization conditions.
- Notions of convergence robust w.r.t. Curried encodings.

Stochastic Tree Series over $\mathbb{R}$

- A class of normalized weighted tree automata with normalization condition but weights can be negative.
- A normalization algorithm for strongly consistent weighted tree automata

[Denisetal08]

What have we done ? (learning)

Learning stochastic tree series

- Learn using algebraic presentation of $\mathbb{R}$-tree series following [Denisetal06]
- What can we do when the series has negative weights and is not stochastic ?

Experiments

- Piccata ([FerieletaI]) : Programming Interface for effiCient Computations and Approximation on multiplicity Tree Automata
- experiments with sets of XML documents and synthetic datasets.

Perspectives

Weighted tree automata for labeling tasks

- A run is a labeling of a tree.
- Learn weights when the (non deterministic) weighted automaton is given
- Learn to label

Weighted tree automata for tree transformations

- Learn tree transformations defined by weighted tree automata