# Structures arborescentes complexes : analyse combinatoire, génération aléatoire et applications

Alexis Darrasse

26 janvier 2010







### Complex tree-like structures combinatorial analysis generating functions singularity analysis "real world" graphs Boltzmann model random sampling applications uniform & efficient tree-like data struct.

### "Real world" graphs



### "Real world" graphs



### "Real world" graphs



**Combinatorial Analysis** 

### Tree-like data structures

in the operatory defined 30 element \_chainey text optimizer alaan valaan 1. yyaaanny dad ina 10 yelaanny juul any attu danta, valaan yelainy (shuiny ) daan valaan 1. yyaaanny dad ina 10 yelaanny juul any attu danta, valaan yebaine y worker = //grammer/define180/alament\_boolaaryattrikuta\_valaur/doolca// is ine forest ine 5. Nyemmen (defind 132) olament\_invents ineft in it ine/yrmp) ette inste\_forest inv(dete 1 e 5. Nyemmen (defind 192) olament pet invjette inste (deter, volane/deter) s / invocation if is it into no bible row = '/grammar/define[3]/element\_resibile/ettribute\_rom/dete/// d'het wet verhaur 1 - Vyrmaner / del deal Cl/alament - Plet twet/at tri danta \_verhaur / dels // -chronent herde in the forest inn 1 - Vyrmaner / del trid / al/anaert \_ investat innie in it boa/group/atter danta\_forest invyd eperates) Start velans = -ygreener/definel@yelanert\_flottert/sttribute\_velans/dets/j Lanv velans = -ygreener/definel@yelanert\_boolans/stribute\_velans/denins/j dooleen veleur • typreser/definel@relevent.booleenvettribute.veleur/choice\*/ Converting the intervention of the second se Second sec ent in voluer 2. //granne/def ind 6/alaant\_art in/attrikuts.ordan/data// محتصادي همه (محتصان فحصاء / 12 كمت أطد/ محصرور، محد nen 🔹 oppressengelef kost Stysbesent, ver isbbegette ibste, nengelete 🕫 influences 8 - Vgranner/def inst 80/alterent\_our influents insta\_ver/dete V-instance 8 - Vgranner/def inst 92/alterent\_our instanter/ettrikate\_vellaur/dete na far vel ar F. Vernans/daf fail SValanast, art far/stir Bara, salasr/dats/ na far vel ar F. Vernans/daf fail SValanast, art far/stir Bara, salasr/dats/ na far vel ar F. Vernans/daf fail SValanast, art far/stir Bara, salasr/dats/ dealant value: 1 //grouns/definal@/almost\_bealanyattributa\_value//twise// where I forement defined Birelenert beckentritteinen velaurich in a second s (growner/definel 11/al.mart., theirs/text./define/ effettert value 1. Vermansidefical Gislament flettertrettrikets valuer/dets/i work in: while the set of parameter (and share and particles parts a basis parameter); see while we fit (pressure ) and parts (b) a meeter parameter (b) a part (b) part (b) a set while we fit (pressure) and parts (b) a meet parameter (b) and part (b) and part (b). محتملهان المعاريجة فليتراج المعطاء (12 كمنا أطار محمد دوار when a represent defined the least part is rate rises, when relations expension in the second s معتملين بعده بعثماني ومعتدان العنا أطار محمد وارتب anna -tart valar - Varanaardel (nd d) alarant, flattartzatz (bata, adarzedata V) - Conseine genetar - Varanazziel (nd tel tel alaren, genet (adarzegen) et i date, genetarziele (a ne weine hood war, wellwar, R., Sygermann / dat insel M/ allaward "bool wary at its iboot a journey of the inse 1/ s cent for values # //ground//def (rel 50/allegent\_ent (er/attribute\_values/deta//)

#### XMI Documents

Software Testing The first interaction of a part of the start with a transit, do show the start of the start o a site di lance was, lance ante descrite information, trimmeniation di distriction, canta da information et lan distriction di un descrite Contaction of State of Street, or other the other of the other oth State of the second state and design additions in 1 Mills The set of the second second of the second s A second And the Argentine Constraints of the Argentin • And Press descriptions of the set of th ne en la ser de la s La ser de la the provide state of the state

Random Sampling

the R. Landar, Million and

A R LOUIS MANY OF THE AREA PARTY.



### Tree-like data structures



### Plan



Thanks to H.-K. Hwang for this figure

### **First part**

## Analysis of *k*-trees bijection between planar 3-trees and ternary trees

bijection between planar 3-trees and ternary trees estimating distances in planar 2-trees BFS-profile of general *k*-trees

### Apollonian networks (deterministic)



[Andrade et al. 05]

- Properties similar to "real world" graphs
- Inspired from the apollonian packings



### Apollonian networks (deterministic)



The (trivial) bijection with ternary trees can be used to study distances to an external vertex.

### Apollonian networks (deterministic)



The (trivial) bijection with ternary trees can be used to study distances to an external vertex.

### Random Apollonian networks (Planar 3-trees, Stack triangulations)



### Random Apollonian networks (Planar 3-trees, Stack triangulations)





### Random Apollonian networks (Planar 3-trees, Stack triangulations)





























 $K_d(z, u) = \sum_{n,m} r_{n,m} u^m z^n$  $r_{n,m}$ : # of rooted k-trees with n total vertices and *m* vertices at distance *d*  $\frac{\partial}{\partial u}K_d(z,u)|_{u=1} = \sum_n r_n z^n$  $r_n$ : # of vertices at distance d in all rooted k-trees with n total vertices  $K_d(z, u) = z^2 T_{d-1,1}(z, u)$  $T_{d,1}(z, u) = 1 + zT_{d,1}(z, u)T_{d-1,0}(z, u)$  $T_{d,0}(z,u) = 1 + zT_{d,1}^2(z,u)$  $T_{0,1}(z, u) = 1 + zT_{0,1}(z, u)T(z)$ 





### Generalization to planar k-trees





### Generalization to (non-planar) k-trees



### k-trees

#### Definition [Beineke, Pippert 69] (k = 2 [Harary, Palmer 68])

A k-tree is:

- either a k-clique,
- or a *k*-tree with one of its *k*-cliques connected to a new vertex.



Some graph theory NP-hard problems have linear time algorithms on partial *k*-trees [*Arnborg, Proskurowksi* 89]

### Generating function for distances

$$K(z) = \sum_{n} K_{n} z^{n}$$
$$K(z) = z^{k} T(z)$$
$$T(z) = \exp(zT^{k}(z))$$

 $K_d(z, u) = \sum_{n,m} r_{n,m} u^m z^n$ 

$$\begin{split} & \mathcal{K}_{d}(z, u) = z^{k} T_{d,1}(z, u) \\ & T_{d,i}(z, u) = \exp(z T_{d,i}^{k-i}(z, u) T_{d,i+1}^{i}(z, u)) \\ & T_{d,d}(z, u) = T_{d-1,0}(z, u) \end{split}$$

$$T_{0,i}(z, u) = \exp(uzT_{0,i}^{k-i}(z, u)T_{0,i+1}^{i}(z, u))$$
  
$$T_{0,d-1}(z, u) = \exp(uzT_{0,d-1}^{k-1}(z, u)T(z))$$

$$\frac{1}{nK_n}[z^n] \left. \frac{\partial}{\partial u} K_d(z, u) \right|_{u=1} =$$
mean # of vertices at distance *d*



### Generating function for distances

 $K(z) = \sum_{n} K_{n} z^{n}$   $K(z) = z^{k} T(z)$   $T(z) = \exp(zT^{k}(z))$   $K_{d}(z, u) = \sum_{n,m} r_{n,m} u^{m} z^{n}$   $K_{d}(z, u) = exp(zT^{k-i}_{d,i}(z, u) T^{i}_{d,i+1}(z, u))$   $T_{d,i}(z, u) = T_{d-1,0}(z, u)$ 

$$T_{0,i}(z, u) = \exp(uzT_{0,i}^{k-i}(z, u)T_{0,i+1}^{i}(z, u))$$
  
$$T_{0,d-1}(z, u) = \exp(uzT_{0,d-1}^{k-1}(z, u)T(z))$$

 $\frac{1}{nK_n}[z^n] \left. \frac{\partial}{\partial u} K_d(z, u) \right|_{u=1} =$ mean # of vertices at distance *d* 

$$\frac{\frac{\partial}{\partial u} K_d(z, u)|_{u=1} = H^{d-2}(z) \frac{\partial}{\partial u} K_2(z, u)|_{u=1}}{\mathcal{H}}$$

$$\frac{\mathcal{H}}{\mathcal{H}} = \frac{\mathcal{H}}{\mathcal{H}}$$
interpretation

$$H(z) = k!(zT^{k}(z))^{k} \prod_{i=1}^{k-1} \frac{1}{1 - izT^{k}(z)}$$
  
= 1 - c\_{k}  $\sqrt{2(1 - kez)} + O(1 - kez).$ 

Semi-large power theorem for calculating  $[z^n]H^{d-2}(z)$  in the range  $x\sqrt{n}$ .

ŀ

### Main Result

#### Theorem [A.D., Soria 09] (RAN [Bodini, A.D., Soria 08])

Given a rand. *k*-tree *G* over *n* vert., the distance between the root vertex *r* and a random vertex *v* of *G* has asymptotic mean value of order  $\sqrt{n}$  and is Rayleigh distributed in the range  $x\sqrt{n}$ :

$$\sqrt{n} \cdot \lim_{n \to \infty} \mathbb{P}(D(r, v) = \lfloor x \sqrt{n} \rfloor) = c_k^2 x e^{-\frac{(c_k x)^2}{2}}, \text{ with } c_k = k \sum_{i=1}^k \frac{1}{i}$$



### Main Result

#### Theorem [A.D., Soria 09] (RAN [Bodini, A.D., Soria 08])

Given a rand. *k*-tree root vertex *r* and a r value of order  $\sqrt{n}$  ar

What about the distance between 2 random vertices?

ance between the as asymptotic mean d in the range  $x\sqrt{n}$ :

$$\sqrt{n} \cdot \lim_{n \to \infty} \mathbb{P}(D(r, v) = \lfloor x \sqrt{n} \rfloor) = c_k^2 x e^{-\frac{(c_k x)^2}{2}}$$
, with  $c_k = k \sum_{i=1}^{k} \frac{1}{i}$ 



### Main Result

#### Theorem [A.D., Soria 09]

Given a rand. *k*-tree *G* over *n* vert., the distance between the two random vertices *v*,*w* of *G* has asymptotic mean value of order  $\sqrt{n}$  and is Rayleigh distributed in the range  $x\sqrt{n}$ :

$$\sqrt{n} \cdot \lim_{n \to \infty} \mathbb{P}(D(v, w) = \lfloor x \sqrt{n} \rfloor) = c_k^2 x e^{-\frac{(c_k x)^2}{2}}, \text{ with } c_k = k \sum_{i=1}^{k} \frac{1}{i}$$

١.



### Second part

### Sampling tree structures generation of random data for software testing with the Boltzmann method

### Tree-like data structures



### Tree-like data structures



Generic oracle in Maple by Pivoteau Optimized oracle for tree structures in C

### Algebraic Data Types

type expression =
 Const of float
 Var of string
 Sum of expression \* expression
 Prod of expression \* expression

 $\mathcal{T} = \mathcal{Z} + \mathcal{Z} + \mathcal{T} \star \mathcal{T} + \mathcal{T} \star \mathcal{T}$ 

Boltzmann sampler (a = 0.25, b = 0.5, c = 0.75)

```
let rec rand_exp = function (a,b,c) as v ->
let r = Random.float 1.0 in
if r < a then Const 4.2
else if r < b then Var "x"
else if r < c then Sum (rand_exp v, rand_exp v)
else Prod (rand_exp v, rand_exp v)
;;</pre>
```

### Boltzmann sampling

[Duchon, Flajolet, Louchard, Schaeffer 04]

#### Simple translation from specification to probabilistic algorithm

construction	Boltzmann sampler with param. x				
$\mathcal{C} = \mathcal{I}$	$\Gamma C(x) := \varepsilon$				
$\mathcal{C}=\mathcal{Z}$	$\Gamma C(x) := z$				
$\mathcal{C} = \mathcal{A} + \mathcal{B}$	$\Gamma C(x) := \operatorname{Bern} \frac{A(x)}{C(x)} \longrightarrow \Gamma A(x) \mid \Gamma B(x)$				
$\mathcal{C} = \mathcal{A}  imes \mathcal{B}$	$\Gamma C(x) := \langle \ \Gamma A(x) \ ; \ \Gamma B(x) \ \rangle$				



### Boltzmann sampling

[Duchon, Flajolet, Louchard, Schaeffer 04]

 $\mathcal{C}$  set of objects  $\gamma$  with size function  $|\cdot|: \mathcal{C} \to \mathbb{N}$ 

Boltzmann model, parameter x

$$\mathbb{P}_x(\gamma) = rac{x^{|\gamma|}}{C(x)}$$
 where  $C(x) = \sum_{\gamma \in \mathcal{C}} x^{|\gamma|}$ 

"Oracle" for computing C(x) for  $x < \rho_C$ [*Pivoteau, Salvy, Soria* 08]

#### Properties

- Guarantees uniformity (between objects of the same size)
- Output size is random, but can be tuned with parameter x
- Linear complexity of sampling (for approximate size)  $\rightarrow$  ability to generate huge objects

### Boltzmann sampling of trees

[Duchon, Flajolet, Louchard, Schaeffer 04]



Ceiled rejection  $\Rightarrow$  still linear

### Boltzmann sampling of trees



A

Very natural translation from type definition to combinatorial specification.

$$\phi(\text{int}), \phi(\text{bool}), \dots = \mathcal{Z}$$
  

$$\phi(t) = \mathcal{X}_t$$
  

$$\phi(t \text{ list}) = \mathcal{Z} \star \text{SEQ}(\mathcal{Z} \star \phi(t))$$
  

$$\phi(t_1 \star \dots \star t_n) = \mathcal{Z} \star \phi(t_1) \star \dots \star \phi(t_n)$$
  

$$\phi(\text{A1 of } t_1 \mid \dots \mid \text{An of } t_n) = \mathcal{Z} \star (\phi(t_1) + \dots + \phi(t_n))$$



### XML documents following a RELAX NG grammar

Very large specifications, interesting for testing and benchmarking the oracle.

Grammar	nb	max	nb	Generic <sup>1</sup>	Tree optim. <sup>2</sup>
	eqs	deg	sols	C(x)	ho and $C( ho)$
rss	10	5	2	0.02s	0.02s
PNML	22	4	4	0.05s	0.26s
xslt	40	3	10	0.4s	2.5s
relaxng	34	4	32	0.4s	1.8s
xhtml-basic	53	3	13	1.2s	4s
mathml2	182	2	18	3.7s	18s
docbook	407	11		67.7s	66s

<sup>1</sup>Evaluation of C(x) using Pivoteau's Maple oracle, made by Salvy <sup>2</sup>Estimation of  $\rho$  and evaluation of  $C(\rho)$  using optimized oracle in C

### Meta-models [Mougenot, A.D., Blanc, Soria 09]

Initiated and developed by people in the application's domain.





### Perspectives



#### k-trees

Another random model: increasing trees. (joint work with Bodini, Hwang, Soria) Extension to chordal graphs.

#### **Random Sampling**

Go from proof of concept to finished product. Multi-parameter sampling. (using work by Bodini, Ponty) Add random generating feature to the Encyclopedia of Combinatorial Structures. (ECS by INRIA Algorithms)