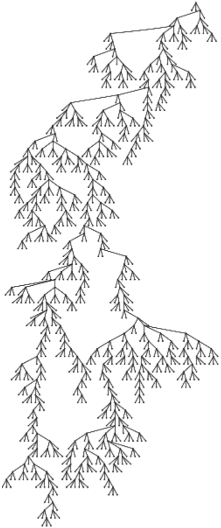
Génération Aléatoire et Modèle de Boltzmann
Expressivité et Effectivité

Michèle Soria
LIP6 – UPMC

Journées ALEA 2011

CIRM, Marseille, 7 Mars 2011



Génération Aléatoire

- Experimenter $\rightarrow$ Analyser et Conjecturer
  Test de conjectures, estimation d'ordres de grandeur
  visualisation de formes et propriétés limites.
- Validation de modèles
  Test de modèles (graphes de terrain), signification de phénomènes observés (séquences biologiques).
- Tests logiciels
  correction, benchmark, robustesse
  couverture typique, génération intensive et automatique
- Passage à l'échelle $\rightarrow$ Efficacité
  structures de grande taille : $10^6 - -10^9$
  mots, arbres, graphes, pavages. . .

Méthodes génériques et automatiques

Classe combinatoire $\mathcal{C}$

$\mathcal{C}$: Spécification Combinatoire $\longrightarrow$ Objet aléatoire $\gamma \in \mathcal{C}$

Random Number Gen.

- Généricité : Classes Combinatoires $\rightarrow$ Spécifications, Spécifications $\rightarrow$ Équations fonctionnelles de SG,
- Effectivité : Générateur compilé à partir des spécifications, algorithmes probabilistes simples

Méthode récursive

Flajolet, Van Cutsem, Zimmerman

- Taille fixée, Génération uniforme
- $\Gamma\mathcal{C}(n)$; $\mathbb{P}(\gamma_n) = 1/c_n$
- Calcul et stockage des coef.
- Complexité de génération : $O(n \log n)$

Méthode de Boltzmann

Duchon, Flajolet, Louchard, Schaeffer

- Taille = v.a. ; mais unif. vs taille
- $\Gamma\mathcal{C}(x)$; $\mathbb{P}(\gamma) \propto e^{|\gamma|}/C(x)$
- Calcul de valeurs de SG $C(x)$
- Complexité linéaire pour génération en taille approchée

Cadre : Combinatoire Analytique (FlSe09)

Classe combinatoire $\mathcal{C}$, objets étiquetés
Série génératrice exponentielle $C(z) = \sum_{\gamma \in \mathcal{C}} \frac{z^{|\gamma|}}{|\gamma|!} = \sum_{n \geq 0} \frac{c_n z^n}{n!}$

SPÉCIFICATION DE $\mathcal{C}$ $\rightsquigarrow$ EQUATION FONCTIONNELLE S.G. $C(z)$

Construction	Notation	SG. exponentielle
ε	1	1
atome	$\mathcal{Z}$	z
Union	$\mathcal{C} = \mathcal{A} \cup \mathcal{B}$	$C(z) = A(z) + B(z)$
Produit	$\mathcal{C} = \mathcal{A} \times \mathcal{B}$	$C(z) = A(z) \times B(z)$
Séquence	$\mathcal{C} = \mathrm{SEQ}(\mathcal{A})$	$C(z) = (1 - A(z))^{-1}$
Ensemble	$\mathcal{C} = \mathrm{SET}(\mathcal{A})$	$C(z) = \exp(A(z))$
Cycle	$\mathcal{C} = \mathrm{CYC}(\mathcal{A})$	$C(x) = -\mathrm{Log}\,(1 - A(z))$

Analogue pour classes non étiquetées, avec séries génératrices ordinaires.

Générateurs de Boltzmann (DuFlLoSc04)

Générateur d'objets étiquetés : $\mathbb{P}_x(\gamma) = \frac{x^{|\gamma|}}{|\gamma|! C(x)}$, avec $C(x) = \sum \frac{x^{|\gamma|}}{|\gamma|!}$ $(x < \rho)$

SPÉCIFICATION DE $\mathcal{C}$ $\rightsquigarrow$ ALGORITHME DE GÉNÉRATION $\Gamma\mathcal{C}(x)$

Construction	*Algorithme de génération*
$\mathcal{C} = \emptyset$ or $\mathcal{Z}$	$\Gamma\mathcal{C}(x) := \varepsilon$ or atom
$\mathcal{C} = \mathcal{A} \cup \mathcal{B}$	$\Gamma\mathcal{C}(x) :=$ Bernouilli $\left(\frac{A(x)}{A(x)+B(x)}\right)$; return $\Gamma\mathcal{A}(x)$ or $\Gamma\mathcal{B}(x)$
$\mathcal{C} = \mathcal{A} \times \mathcal{B}$	$\Gamma\mathcal{C}(x) :=$ return $< \Gamma\mathcal{A}(x); \Gamma\mathcal{B}(x) >_{etiq}$ *appels indépendants*
$\mathcal{C} = \text{SEQ}(\mathcal{A})$	$\Gamma\mathcal{C}(x) :=$ $k \leftarrow$ Geometric$(A(x))$; return $< \underbrace{\Gamma\mathcal{A}(x), \ldots, \Gamma\mathcal{A}(x)}_{k \text{ fois}} >_{etiq}$
$\mathcal{C} = \text{SET}(\mathcal{A})$	$\Gamma\mathcal{C}(x) :=$ $k \leftarrow$ Poisson$(A(x))$; return $< \underbrace{\Gamma\mathcal{A}(x), \ldots, \Gamma\mathcal{A}(x)}_{k \text{ fois}} >_{etiq}$
$\mathcal{C} = \text{CYCLE}(\mathcal{A})$	$\Gamma\mathcal{C}(x) :=$ $k \leftarrow$ Logarithmic$(A(x))$; return $< \underbrace{\Gamma\mathcal{A}(x), \ldots, \Gamma\mathcal{A}(x)}_{k \text{ fois}} >_{etiq}$

$\alpha \in \mathcal{A} \cup \mathcal{B} \to \mathbb{P}_x(\alpha) = \frac{A(x)}{A(x)+B(x)} \frac{x^{|\alpha|}}{|\alpha|! A(x)} = \frac{x^{|\alpha|}}{|\alpha|! A(x)}$

$\gamma \in \mathcal{C} = \mathcal{A} \times \mathcal{B} \to \mathbb{P}_x(\gamma) = \frac{x^{|\alpha|}}{|\alpha|! A(x)} \cdot \frac{x^{|\beta|}}{|\beta|! B(x)} \cdot (\alpha + \beta)! = \frac{x^{|\alpha|+|\beta|}}{(|\alpha|+|\beta|)! A(x) B(x)} = \frac{x^{|\gamma|}}{|\gamma|! C(x)}$

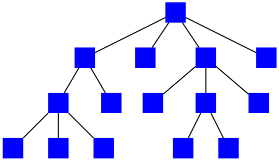
$\gamma = (\alpha_1, \cdots, \alpha_k) \in \text{SEQ}(\mathcal{A}) \to \mathbb{P}_x(\gamma) = A(x)^k (1 - A(x)) \cdot \frac{x^{|\alpha_1|}}{|\alpha_1|! A(x)} \cdots \frac{x^{|\alpha_k|}}{|\alpha_k|! A(x)} \cdot (\alpha_1 + \cdots + \alpha_k)! = \frac{x^{|\gamma|}}{|\gamma|! C(x)}$

Exemples

Arbres binaires :

$$\mathcal{B} = \mathcal{Z} + \mathcal{B} \times \mathcal{B}$$

$$B(z) = z + B(z)^2 = \frac{1-\sqrt{1-4z}}{2}$$



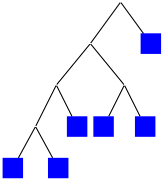
Algorithm : $\Gamma\mathcal{T}(x)$

if Bernoulli $\frac{x}{B(x)}$ **then**
  Return $\blacksquare$
else
  Return $\langle\, \Gamma\mathcal{B}(x)\ ,\ \Gamma\mathcal{B}(x)\, \rangle_{etiq}$;
end if

Arbres planaires :

$$\mathcal{T} = \mathcal{Z} \times \textsc{Seq}(\mathcal{T})$$

$$T(z) = \frac{z}{1-T(z)} = \frac{1-\sqrt{1-4z}}{2}$$



Algorithm : $\Gamma\mathcal{T}(x)$

$k \leftarrow \mathrm{Geom}(T(x))$;
$forest \leftarrow \langle\ \underbrace{\Gamma\mathcal{T}(x), \ldots, \Gamma\mathcal{T}(x)}_{k\ times}\ \rangle_{etiq}$;
Return $\blacksquare \times forest$;

Générateurs de Boltzmann (DuFlLoSc04)

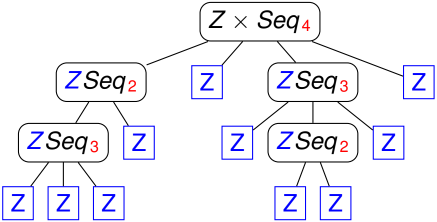
- Génération libre, paramètre x
- Algorithmes probabilistes : *choix aléatoire selon lois de paramètre $A(x)$*
- Évaluation numérique des séries génératrices : *calcul de l'oracle*
- Générateur aléatoire uniforme [0,1] + manipulation de réels

Théorème (Complexité de la génération libre – DuFlLoSc04)

Le générateur de Boltzmann $\Gamma C(x)$ d'une classe $\mathcal{C}$, spécifiée récursivement sur les opérateurs ε , Z, $\cup$, $\times$, Seq, Set, Cycle, a une complexité arithmétique linéaire en la taille du résultat (sous hypothèse oracle en $O(1)$).

Algorithm : Geom-Poisson-Loga(x)

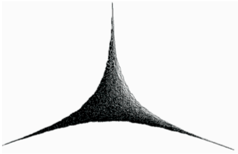
$U \leftarrow uniform()$; $S \leftarrow 0$; $k \leftarrow 0$;
while $U < S$ **do**
$S \leftarrow S + p_k$; $k \leftarrow k + 1$; **od**
Return k ; $O(k+1)$



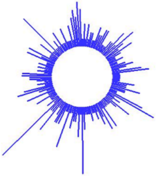
Générateurs de Boltzmann non étiqueté (FlFuPi07)

Théorème (FlFuPi07)

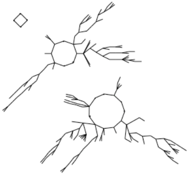
Pour une classe $\mathcal{C}$ non étiquetée, spécifiée recursivement par des opérateurs ε, Z, $\cup$, $\times$, Seq, Set, Cycle, le générateur de Boltzmann $\Gamma C(x)$ a une complexité arithmétique linéaire en la taille du résultat.



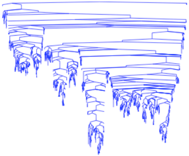
Partition plane



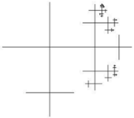
Composition circulaire



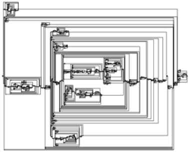
Graphe fonctionnel



Arbre non planaire



Alcool acyclique



Circuit série-parallèle

Thanks to Carine Pivoteau

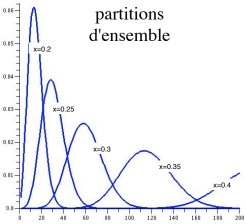
Distribution de la taille

- Générateur libre $\rightarrow$ taille objet engendré = v. aléatoire N

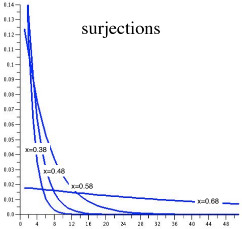
$$\mathbb{P}_x(N = n) = \frac{c_n x^n}{n!\,C(x)} \qquad \text{et} \qquad \mathbb{E}_x(N) = x\frac{C'(x)}{C(x)}$$

  mais uniformité à taille donnée : $\mathbb{P}_x(\gamma_n) = \frac{x^n}{n!\,C(x)}$

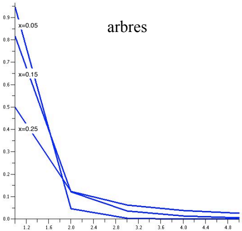
- Distribution taille dépend de x, $0 < x < \rho$
  et du type de singularité SG



$C(x) = e^{(e^x-1)}$



$C(x) = \frac{1}{2-e^x}$



$C(x) = \frac{1-\sqrt{1-4x}}{2}$

Contrôle de la taille

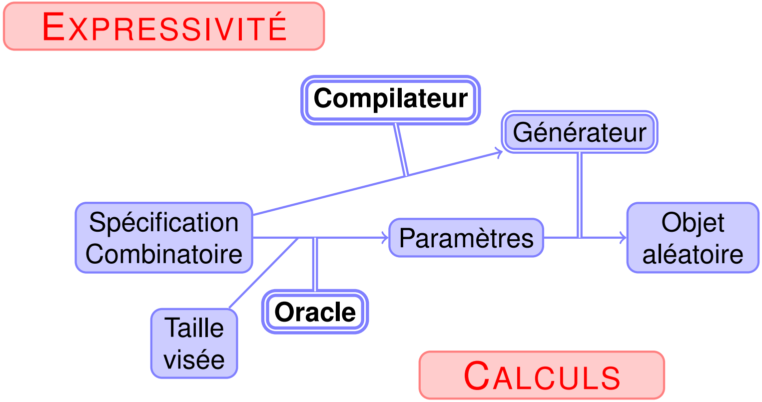
- Contrôle de la taille : générer objets de taille exacte n, ou de de taille approchée $[n(1-\varepsilon), n(1+\varepsilon)]$,
- Rejet : si la taille ne convient pas, rejeter et lancer une autre génération
- Taille approchée : complexité moyenne (essentiellement) linéaire

Complexité de la génération contrôlée (DuFlLoSc04, FlFuPi07)

Soit $\mathcal{C}$ une classe combinatoire, spécifiée recursivement à l'aide des opérateurs ε, Z, $\cup$, $\times$, Seq, Set, Cyc. Le générateur avec rejet $\Gamma C(x, n, \varepsilon)$ a une complexité moyenne en $\mathcal{O}(n)$:

- Distributions en cloche : 1 essai en moyenne pour engendrer un objet de taille $\in [n(1-\varepsilon), n(1+\varepsilon)]$
- Distributions plates : $O(1)$ essais en moyenne pour engendrer un objet de taille $\in [n(1-\varepsilon), n(1+\varepsilon)]$
- Distributions piquées : $O(n)$ essais mais pointage pour se ramener à distribution plate

- Mais $O(n^2)$ pour taille exacte ($O(n)$ pour distributions en cloche).


EXPRESSIVITÉ
Compilateur
Générateur
Spécification Combinatoire
Paramètres
Objet aléatoire
Taille visée
Oracle
CALCULS

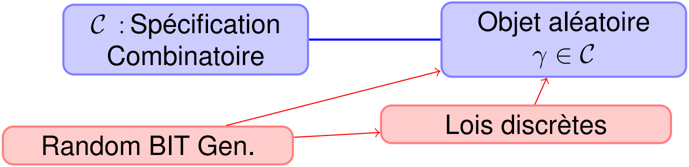
Expressivité

- Extensions à d'autres opérateurs combinatoires
  - Cycle pointé non-étiqueté(*BoFuKaVi07, PaSt07*)
  - Langages Réguliers avec Shuffle (*DaPaRoSo10*)
  - Produit Boite (*RoSo09*)
  - Opérateurs différentiels (Roussel, *BoSoRo11*)
  - Produit d'Hadamard $O(n\sqrt{n})$(*BoGaRo10*)
  - ...
- Extensions à d'autres univers combinatoires
  - Semi-étiquetés et colorés (*BoJa09*)
  - Classes multiplicatives Boltzmann-Dirichlet (*BoBa11*)
  - ...
- Génération Biaisée
  - Boltzmann multicritère (*BoPo10*)
  - Boltzmann "uniforme biaisé"
  - ...

Calculs, certification, implantation

- Calculs de séries
  - Oracle pour systèmes combinatoires récursifs et différentiels (Salvy, *PiSaSo11*)
  - Rayon de convergence ?
  - Paramètre en fonction de la taille visée ?
  - Systèmes combinatoires multivariés ?
  - ...
- Certification
  - arithmétique réelle :impact des erreurs (Duchon)
  - arithmétique discrète : générateurs de Boltzman binaires (*FlPeSo11*)
  - ...
- Implantation
  - Générateurs automatiques
  - Bibliothèques dédiées
    bio, tests, automates, arbres, réseaux
  - ...

I - Modèle de Boltzmann Discret



- Généricité et Effectivité
- Complexité moyenne booléenne en $O(|\gamma|)$

- Simulateur parfait de v.a. discrètes
  - X : Géométrique, Poisson, Logarithmique
  - $\mathbb{P}(X = k)$ nécéssite $O(k)$ bits en moyenne

Bernoulli et Géométrique

Algorithm : Bernoulli(p)

$i \leftarrow 1 + Geom(1/2)$;
Return $Bit_i(p)$;

$\mathbb{P}(\#bits = i) = 2^{-(i+1)}$, $\mu = 2$
\# bits indépendant de p
$Bit_i(p)$: accès avec proba $\frac{1}{2^{(i+1)}}$

Algorithm : Geométrique(p)

$k \leftarrow 0$;
do $i \leftarrow 1 + Geom(1/2)$;
if $Bit_i(p) = 1$ **then** Return k ;
else $k \leftarrow k + 1$;
fi ;**od** ;

O(k+1) bits pour Geom(p)= k

Poisson et Logarithmique (Machine de Buffon)

SGE ($\mathcal{P}$) : $P(x) = \sum P_n \frac{x^n}{n!}$

Algorithm : Schema Von Neuman $(x, \mathcal{P})$

do
$N \leftarrow Geom(x)$;
Tirer $U_1, \ldots, U_N$ v.a. indep. unif. $[0, 1]$;
if $\sigma(U_1, \ldots, U_N) \in \mathcal{P}$ **then** Return N ;
od ;

$\mathbb{P}\,(VN(x, \mathcal{P}) = n) = \frac{1}{P(x)} \frac{P_n x^n}{n!}$ *Boltz*

- $U_1 < \ldots < U_n$
  $P_n = 1 \rightarrow P(x) = e^x$
  $\Longrightarrow$ Poisson
- $U_1 < \{U_2, \ldots, U_n\}$
  $P_n = (n-1)! \rightarrow P(x) = \log \frac{1}{1-x}$
  $\Longrightarrow$ Logarithmique

Générateur discret (*FlPeSo11*) : $U_i = \sum_{j>0} b_j 2^{-j}, \;\; b_j \in \{0, 1\}$

- Générer $(U_1, \ldots, U_N)$ avec le minimum de bits pour les différencier
- Construction d'un trie :
  - LC = nombre de bits consommés
  - 1 seul registre

Complexité binaire Schéma Von Neuman

Théorème (FlPeSo11)

– $\mathcal{P}$ classe de permutations et x paramètre, Schema von Neumann →
exactement v.a. discrete $\mathbb{P}(N = n) = \dfrac{1}{P(\lambda)} \dfrac{P_n \lambda^n}{n!}$.

– Nombre moyen d'itérations $\frac{1}{s}$, $s = (1 - x)P(x)$, *et distrib.* $1 + Geom(s)$.

– S.G. probabilités du nombre C de flips consommés par l'algorithme :

$$E(q^C) = \frac{qH^+(x,q)}{1 - qH^-(x,q)},$$

$H^+(z,q) = (1-z)\sum_{n=0}^{\infty} \frac{P_n}{n!} h_n(q) z^n, \qquad H^-(z,q) = (1-z)\sum_{n=0}^{\infty} \left(1 - \frac{P_n}{n!}\right) h_n(q) z^n.$

– Queues de distribution exponentielles.

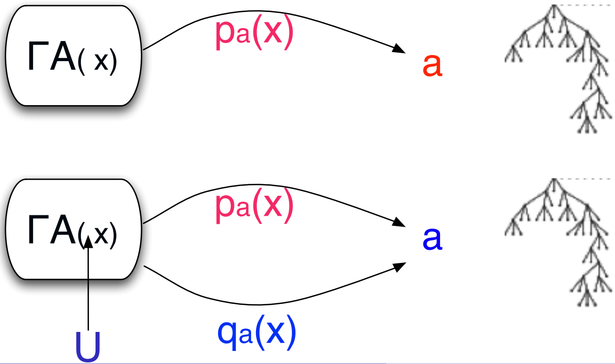
En cours :

- Extension autres lois
- Intégration dans générateur de Boltzmann

II - Boltzmann Uniforme Biaisé

Générateur de Boltzmann Biaisé : $\Gamma\mathcal{A}(x) \to \Gamma\mathcal{A}(ux)$

Soit $\Gamma\mathcal{A}(x)$ gen. Boltzmann engendre α avec probabilité $p_\alpha(x)$.
Soit $\delta_{A,x}(t)$ densité de probabilité sur σ.
Alors le générateur biaisé $\Gamma\mathcal{A}(ux)$, avec u tiré selon $\delta_{A,x}(t)$, engendre α avec probabilité $q_\alpha(x) = \int_\sigma p_\alpha(tx)\delta_{A,x}(t)\,dt$.



Générateur exponentiel $\widehat{\Gamma} \to$ Générateur ordinaire Γ

$\Gamma\mathcal{A}(x) = \widehat{\Gamma}\mathcal{A}(ux)$

- Générateur exponentiel : $\widehat{\Gamma}\mathcal{A}(x)$
  $$p_\alpha(x) = \frac{x^{|\alpha|}}{|\alpha|!}\frac{1}{A_E(x)}, \quad \text{avec } A_E(x) = \sum_{\alpha\in\mathcal{A}} \frac{x^{|\alpha|}}{|\alpha|!}$$
- Générateur ordinaire : $\Gamma\mathcal{A}(x)$
  $$q_\alpha(x) = \frac{x^{|\alpha|}}{A_O(x)}, \quad \text{avec } A_O(x) = \sum_{\alpha\in\mathcal{A}} x^{|\alpha|}$$
- Biais : $\delta_{A,x}(t) = \frac{A_E(xt)}{A_O(x)} e^{-t}$ sur $[0, \infty)$

$$\begin{aligned}
\mathbb{P}\left(\Gamma\mathcal{A}(x) = \alpha_{/n}\right) &= \int_0^\infty \mathbb{P}\left(\widehat{\Gamma}\mathcal{A}(xt) = \alpha\right) . \delta_{A,x}(t)dt \\
&= \int_0^\infty \frac{(xt)^n}{n!A_E(xt)} \cdot \frac{A_E(xt)}{A_O(x)} e^{-t}dt \\
&= \frac{x^n}{A_O(x)} \int_0^\infty \frac{t^n}{n!} \cdot e^{-t}dt \\
&= \frac{x^n}{A_O(x)}
\end{aligned}$$

Générateur pointé $\dot\Gamma \rightarrow$ Générateur de base Γ *(sgE et sgO)*

$\Gamma\mathcal{A}(x) = \dot\Gamma\mathcal{A}(ux)$

- Générateur pointé $\dot\Gamma\mathcal{A}(x)$: $p_\alpha(x) = \frac{\alpha x^{|\alpha|}}{xA'(x)}$
- Générateur non pointé $\Gamma\mathcal{A}(x)$: $q_\alpha(x) = \frac{x^{|\alpha|}}{A(x)}$
- Biais $\delta_{A,x}(t) = \frac{xA'(xt)}{A(x)-A(0)}$ sur $[0,1]$

$$\begin{aligned}
\mathbb{P}\left(\Gamma\mathcal{A}(x) = \alpha_{/n>0}\right) &= \int_0^1 \frac{n(xt)^n}{xtA'(xt)} \cdot \frac{xA'(xt)}{A(x)-A(0)} dt \\
&= \frac{x^n}{A(x)-A(0)} \int_0^1 nt^{n-1} du = \frac{x^n}{A(x)-A(0)}
\end{aligned}$$

- Générateur base $\Gamma \rightarrow$ Générateur intégré $\bar\Gamma\mathcal{A}(x) = \Gamma\mathcal{A}(ux)$

$p_\alpha(x) = \frac{x^{|\alpha|}}{A(x)}$, $\quad q_\alpha(x) = \frac{x^{|\alpha|+1}}{(|\alpha|+1)\int_0^x A(v)}$, $\quad \delta_{A,x}(t) = \frac{xA(xt)}{\int_0^x A(v)}$ sur $[0,1]$

Exemple : Arbres de Cayley

- Enracinés : $\mathcal{A} = \mathcal{Z} \times \textsc{Set}(\mathcal{A}) \quad A(z) = \sum n^{n-1} \frac{z^n}{n!}$

$$\begin{aligned} \Gamma A(x) &:= k \leftarrow \mathit{Poisson}(A(x)), \\ &\quad \textbf{return } \mathcal{Z} \times \langle \underbrace{\Gamma A(x), ..., \Gamma A(x)}_{k} \rangle \end{aligned}$$

- Non enracinés $B(z) = \sum n^{n-2} \frac{z^n}{n!} \rightarrow zB'(z) = A(z)$

$$\begin{aligned} \Gamma B(x) &:= \text{tirer } u \text{ sur } [0,1] \text{ selon densité } \tfrac{A(tx)}{tB(x)}\ ; \\ &\quad \textbf{return } \Gamma A(ux) \end{aligned}$$

- Enracinés croissants $\mathcal{C} = \mathcal{Z}^{\square} \times \textsc{Set}(\mathcal{C}) \rightarrow C'(z) = e^{C(z)}$
  $\rightarrow C(z) = -\mathrm{Log}\,(1-z)$

$$\begin{aligned} \Gamma C(x) &:= \text{tirer } u \text{ sur } [0,1] \text{ selon densité } \tfrac{xC'(tx)}{C(x)} \\ &\quad k \leftarrow \mathit{Poisson}(C(ux)), \\ &\quad \textbf{return } \mathcal{Z} \times \langle \underbrace{\Gamma C(ux), ..., \Gamma C(ux)}_{k} \rangle \end{aligned}$$

Nouveaux opérateurs

Combinaison de GB Biaisés : $(\Gamma\mathcal{B}(ux), \Gamma\mathcal{C}(ux)) \to \Gamma\mathcal{A}(x)$

$\Gamma\mathcal{B}(x)$ et $\Gamma\mathcal{C}(x)$ générateurs de Boltzmann pour $\mathcal{B}$ et $\mathcal{C}$,
$\delta_{B,C,x}(t)$ densité de probabilité sur σ ; $\mathcal{A} = \mathcal{B} \star \mathcal{C}$
u tiré selon $\delta_{B,C,x}(t)$, $(\Gamma\mathcal{B}(ux), \Gamma\mathcal{C}(ux)) \to \Gamma\mathcal{A}(x)$.

- Produit Boîte :

$$\mathcal{A} = \mathcal{B}^{\square} \times \mathcal{C} \quad \to \quad A(z) = \int_0^z B'(t)C(t)dt$$

$\Gamma\mathcal{A}(x) :=$ tirer u selon $\delta(t) = \frac{xB'(tx)C(tx)}{A(x)}$, sur [0,1]
$\beta \leftarrow \Gamma B'(xu)$; $\gamma \leftarrow \Gamma C(xu)$
return $\alpha = \mathit{ProduitBoite}(\beta, \gamma)$

Généralisation :
GB pour équation différentielle du 1er ordre
$\mathcal{T}' = \mathcal{F}(\mathcal{Z}, \mathcal{T})$ (BoRoSo11)

- Produit ordonné :

$$\mathcal{A} = \mathcal{B} \bullet \mathcal{C} \quad \to \quad A(z) = \int_0^z B'(z-t)C(t)dt$$

$\Gamma\mathcal{A}(x) :=$ tirer u selon $\delta(t) = \frac{B'(x-t)C(t)}{A(x)}$, sur [0,x]
$\beta \leftarrow \Gamma B'(x-u)$; $\gamma \leftarrow \Gamma C(u)$
return $\alpha = \mathit{ProduitOrdonne}(\beta, \gamma)$

Application : GB pour langages réguliers avec Shuffle (DaPaRoSo10)