

## eatp et pbf

### A Ce dont on va parler

On va parler d'Expressions Algébriques Totalelement Parenthésées et de Parenthèses Bien Formées

Les pbf? ce sont les listes de  $n$  ( et de  $n$  ), avec pour chaque sous liste d'indices  $1..k$ , le nombre de ( qui est toujours au moins égal au nombre des )

Mais alors c'est juste comme les eatp ! PAS DU TOUT! il suffit de regarder l'exemple qui suit pour ... commencer à voir qu'on ne voit pas

Comme j'ai cru m'en être aperçu seul il y a une dizaine d'années, le nombre d'eatp avec  $n+1$  termes et  $n$  produits est égal au nombre de pbf avec  $n$  parenthèses ouvrantes. Les voici ci-dessous pour  $n=3$

$[[a*[a*a]]*a]$	$(( )) ( )$
$[[[a*a]*a]*a]$	$( )( )( )$
$[[a*a]*[a*a]]$	$( ) (( ))$
$[a*[a*[a*a]]]$	$(( ( ) ))$
$[a*[[a*a]*a]]$	$(( ) ( ))$

Une autre façon de voir que ce n'est pas pareil est de comparer les grammaires qui me semblent "canoniques"

$\text{eatp} == a \mid [\text{eatp} * \text{eatp}]$                        $\text{pbf} == \text{rien} \mid (\text{pbf}) \mid \text{pbf pbf}$

Ayant remarqué cela, j'ai cherché une bijection .... et j'ai trouvé une anomalie : une bijection qui ne marche que dans un sens (des détails viennent)

En cherchant à comprendre tout cela je correspondais (au sens e-mail) avec un ancien étudiant et une solution est sortie de nos échanges

### B Historique

1 En 1990, l'article [1], signale cette égalité de cardinaux et dit "qu'il y a peu de chances que l'on puisse trouver une bijection directe entre les deux ensembles"

Note : la date de cet article annule ma prétention à m'en être aperçu tout seul

2 En 1996, l'article [2] construit récursivement une famille de bijections entre  $\text{EATP}(n)$  et  $\text{PBF}(n)$

3 En 1999, l'article [3] fournit un algorithme pour passer des  $\text{EATP}(n)$  aux  $\text{PBF}(n)$

4 En 2000, l'article [4] fournit des explications "par arbres" de tout ce qui précède et montre que [2] et [3] avaient fait la même chose

**5** En 2000, l'article [5] "explique" tout en reliant cela aux partitions convexes d'un polygone régulier : comme dans l'algorithme du théorème des 4 couleurs selon le problème X 2000

**6** En 2003, selon [6], je vous raconte l'idée Goffinet-Grand

## C Des preuves

### 1 Conway & Guy

Ils ont remarqué le phénomène, constaté que les deux fois le nombre  $N$  de solutions vérifie ( $N(0)=1$ )  $N(1)=1$ ,  $N(2)=2$  et  $N(n)=\sum_{k=1}^{n-1} N(k)N(n-k)$  d'où l'égalité des cardinaux.

Pour les eatp :  $k$  est le cardinal du fils gauche de l'arbre représentant l'eatp, il est bien entre 1 et  $n-1$

Pour les pbf : on prouve l'unique existence d'une décomposition d'une pbf non vide en  $(pbf1)pbf2$  avec  $pbf1$  non vide, le nombre  $PBF(n)$  des pbf de longueur  $n$  vérifie donc

$$PBF(n) = PBF(n-1) + \sum_{k=1}^{n-2} PBF(n-k-1).PBF(k)$$

où  $k$  a désigné la longueur de  $pbf2$ ; c'est donc bien la même relation ... et cela s'est présenté presque pareil

### 2 Krusemeyer

Sa bijection est la reformulation de la preuve qui précède : il définit une famille d'applications  $F_n$  qui vont des  $PBF_n$  vers les  $EATP_n$  pour  $n=0$  c'est sans intérêt, pour  $n=1$  il transforme  $()$  en  $[a^*a]$ , il suppose avoir défini  $F_2, \dots, F_{n-1}$  et il prend une eatp avec  $n$  (, elle se décompose de façon unique en  $(pbf1)pbf2$  avec  $|pbf1| = k \geq 1$ , il définit alors

$$F_n\{(pbf1)pbf2\} = [\{F_k\{pbf1\}\} * \{F_{n-k}\{pbf2\}\}]$$

où les  $\{$  et  $\}$  sont de délimiteurs sans sens ici

Cette application est d'évidence injective (à partir d'un arbre on peut reconstituer les branches gauches et droites) comme les ensembles  $PBF_n$  et  $EATP_n$  ont même cardinal, elle est bijective

### 3 Penrice

L'idée (que j'ai un peu trafiquée) est celle de "la bijection qui ne marche que dans un sens". Vous partez d'une eatp, vous la lisez en partant de la gauche, quand vous rencontrez

[	on ne fait rien
a	on ne fait rien
*	on écrit (
]	on écrit )

Il reste à voir que quand on fait ce qui précède, le texte que l'on a écrit est une pbf, et à montrer que cette application est injective ... ce qui fait qu'elle sera aussi surjective

La preuve suit la définition : une eatp c'est **a** ou **[eatp\*eatp]**.

- pour **a** on obtient la pbf vide

- pour  $[eatp1*eatp2]$  on obtient  $pbf\_of\_eatp\{eatp1\} \ ( \ pbf\_of\_eatp\{eatp2\} \ )$  qui est bien une eatp.

L'injectivité résulte du lemme précédent sur l'unicité de l'écriture  $(pbf1)pbf2$  (à l'envers cette fois-ci)

Et la réciproque ? C'est l'idée de Penrice :

```
let eatp_of_pbf x =
  let rec boulot provisoire pile=function
    | []          ->      provisoire
    | "("::q ->      boulot "a" ((provisoire^"*")::pile) q
    | ")"::q ->      boulot ("["^(hd pile)^provisoire^"]") (tl pile) q
  in boulot "a" [] (list_of_string x);;
```

Un exemple complet d'aller retour

```
let z8=alea_eatp 8;; let p8=pbf_of_eatp (list_of_string z8);;
eatp_of_pbf p8;;
```

```
z8 : string = "[[a*a]*[[a*a]*[a*[[a*a]*a]]]"
```

```
p8 : string = "()((()((()())))")
```

```
string = "[[a*a]*[[a*a]*[a*[[a*a]*a]]]"
```

exécution de eatp\_of\_pbf "()((()((()())))")

	provisoire	pile	à traiter
a	.....		::::: ( ) ( ( ) ( ( ( ) ( ) ) ) )
a	.....	a*	::::: ) ( ( ) ( ( ( ) ( ) ) ) )
[a*a]	.....		::::: ( ( ) ( ( ( ) ( ) ) ) )
a	.....	[a*a]*	::::: ( ) ( ( ( ) ( ) ) ) )
a	.....	a*[a*a]*	::::: ) ( ( ( ) ( ) ) ) )
[a*a]	.....	[a*a]*	::::: ( ( ( ) ( ) ) ) )
a	.....	[a*a]* [a*a]*	::::: ( ( ) ( ) ) ) )
a	.....	a*[a*a]*[a*a]*	::::: ( ) ( ) ) ) )
a	.....	a*a*[a*a]*[a*a]*	::::: ) ( ) ) ) )
[a*a]	.....	a*[a*a]*[a*a]*	::::: ( ) ) ) )
a	.....	[a*a]*a*[a*a]*[a*a]*	::::: ) ) ) )
[[a*a]*a]	.....	a*[a*a]*[a*a]*	::::: ) ) )
[a*[[a*a]*a]]	.....	[a*a]*[a*a]*	::::: ) )
[[a*a]*[a*[[a*a]*a]]]	.....	[a*a]*	::::: )
[[a*a]*[[a*a]*[a*[[a*a]*a]]]]			

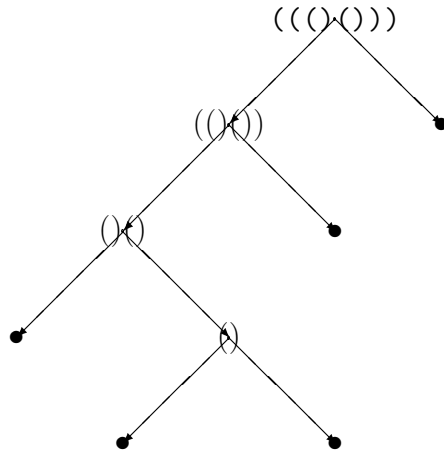
Quand je me suis attelé à la tâche de prouver que ces deux bijections étaient réciproques l'une de l'autre, j'ai fabriqué et détaillé l'exemple précédent, à partir de là je me suis lancé dans une preuve par induction structurée .... 2 pages après j'y étais formellement arrivé et je n'y avais toujours rien compris

## 4 Shen & Shult

En plus des ensembles  $PBF_n$  et  $EATP_n$ , ils introduisent  $ABC_n$  = les Arbres binaires complets à  $n$  nœuds et  $n+1$  feuilles et ils vont procéder en deux temps :  $PBF_n \leftrightarrow ABC_n \leftrightarrow EATP_n$

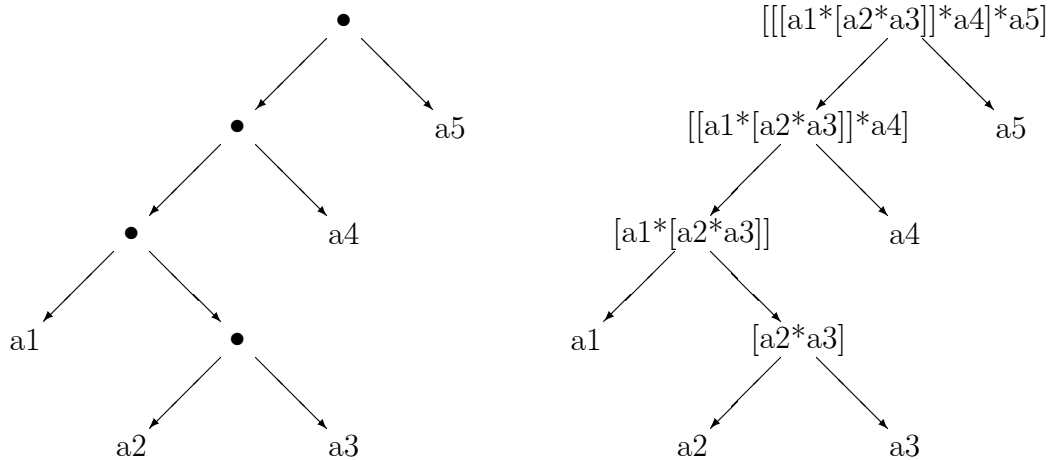
**a**  $PBF_n \leftrightarrow ABC_n$

Si on a rien comme pbf on lui associe Nil, si on a (pbf1)pbf2 on lui associe l'arbre dont la racine est étiquetée par (pbf1)pbf2, le fils gauche est l'arbre de pbf1 et le fils droit est celui de pbf2. L'arbre image est juste la structure de l'arbre obtenu, en enlevant toutes les étiquettes



**b**  $ABC_n \leftrightarrow EATP_n$

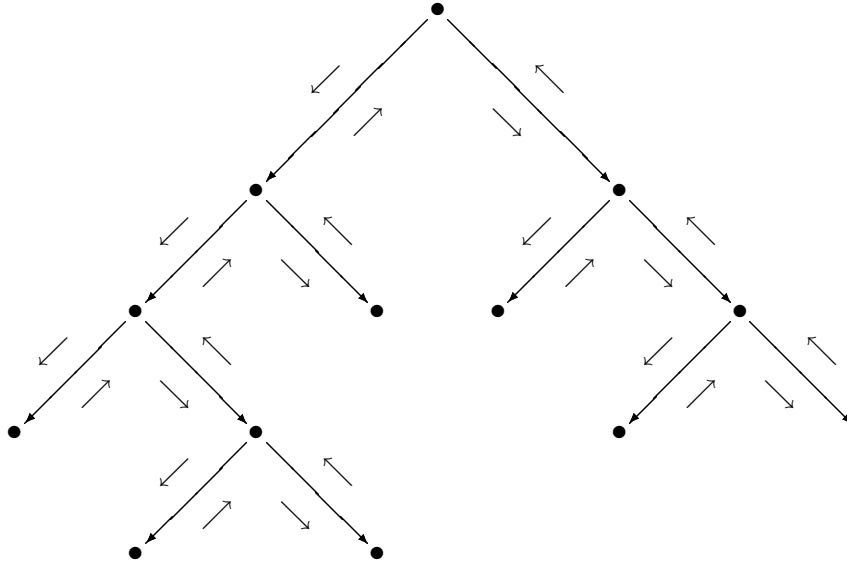
On part d'un  $ABC_n$  et on va en tirer une eatp. Pour cela on prend le squelette d'arbre et on étiquette toutes les feuilles par des  $a$  (ou des  $a_k$  pour s'y retrouver). Une fois cela fait on considère avoir un arbre syntaxique dont les nœuds des  $*$ , et on remonte récursivement en emballant les termes de  $[ \quad ]$  pour obtenir une eatp



### c La bijection vue comme parcours de l'écureuil

C'est le nom sous lequel j'ai rencontré la première fois le parcours infixe d'un arbre. Et Shen & Shult se sont aperçu que leur bijection  $ABC_n \rightarrow EATP_n$ , était simplement l'enregistrement du trajet suivi par l'écureuil

Précisons de quels enregistrements il s'agit : l'écureuil part de la racine et il va partir vers le fils gauche en gardant le bord de l'arbre à main gauche. Ce faisant il va rencontrer des branches gauches et des branches droites, son enregistrement va consister à inscrire des G (pour gauche ..) et des D *la première fois qu'il parcourt* ces branches, selon l'exemple qui suit :



La liste obtenue est G G G D G D D D G D G D ou, en en pbf ( ( ( ) ( ) ) ) ( ) ( )

### d l'écureuil n'a rien inventé

La bijection en question est la même !

## 5 Shapiro et Sulanke

Leur objectif n'est pas le même, ils veulent compter le nombre de parenthésages d'un mot, non nécessairement binaires : voici ce dont ils parlent pour le mot  $abcd$  :

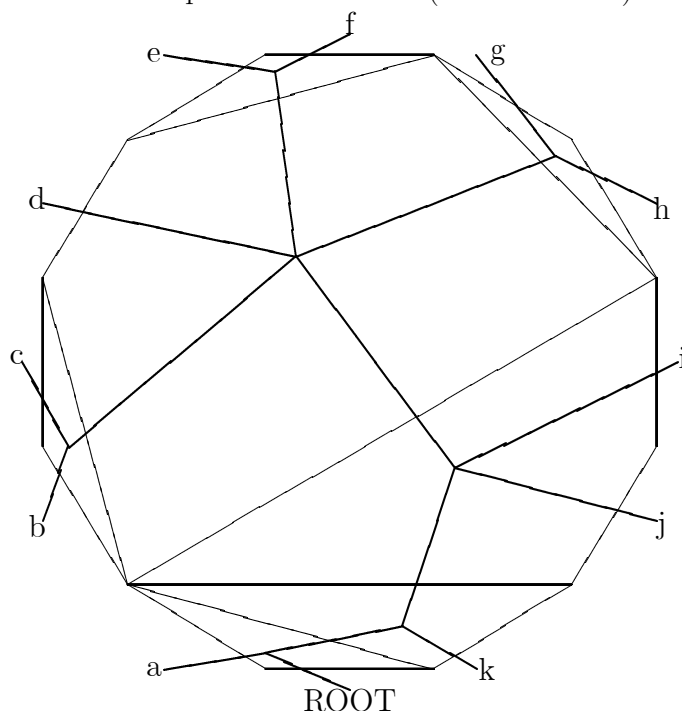
$$\begin{array}{cccccc}
 abcd & (ab)cd & (abc)d & a(bc)d & a(bcd) & ab(cd) \\
 (a(bc))d & (a(bc))d & a((bc)d) & a(b(cd)) & (ab)(cd) & 
 \end{array}$$

En re-faisant une preuve comme celle pour les nombres de parenthésages binaires on obtient la fonction génératrice des nombres de Schröder :

$$\sum_{n \geq 0} s_n x^n = \frac{1 + x - \sqrt{1 - 6x + x^2}}{4x}$$

et les auteurs de cet article fabriquent des chaînes de bijections entre de tels comptes en disant que "l'on peut réduire chacun de ces ensembles de configurations à celui énuméré par les nombres de Catalan" ... ce que je n'ai pas été capable de comprendre

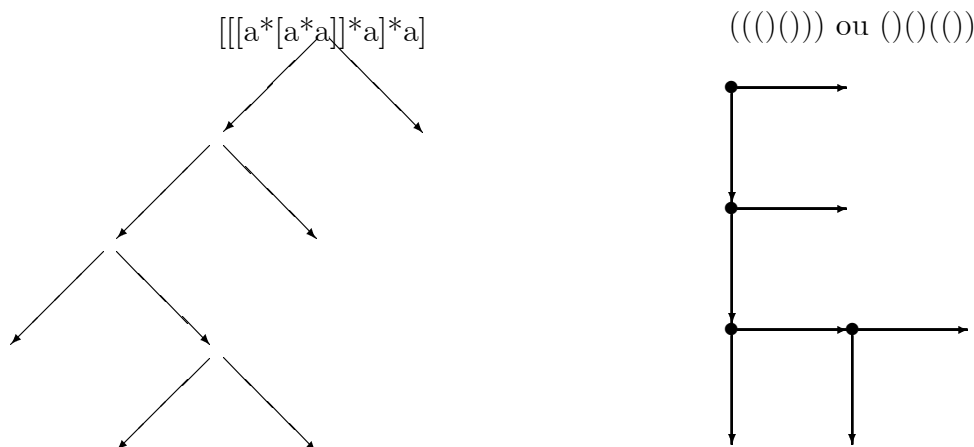
L'une de leurs idées vaut le détour : on y retrouve une généralisation des dissections d'un polygone régulier vues dans le problème X 2001 (les 4 couleurs)



## 6 G & G

Voici cette solution : les eatp avec  $n *$  ne sont qu'une version des arbres binaires avec  $n$  nœuds interne =  $(n+1)$  feuilles; les pbf avec  $n ($  ne sont qu'une version des arbres d'arité quelconque avec  $n$  nœuds. Il y a une bijection évidente entre les deux : tourner la tête de  $45^\circ$ , ce qui était fils gauche devient frère, et le fils droit devient fils, c'est la disposition habituelle dans l'interface graphique d'un explorateur de fichiers

Pour la programmer, on a facilement les deux équivalences  $\text{eatp} \leftrightarrow \text{arbre-binaire}$  et  $\text{pbf} \leftrightarrow \text{arbre-frère-fils}$  et il faut s'appliquer pour programmer "tourner la tête de  $45^\circ$ "



Ce n'est bien sûr qu'une autre présentation de la version Shen-Shult, et ce n'est donc pas différent de la bijection de Krusemeyer

## D Bibliographie

[1] R.K.Guy “Second strong law of small numbers”

Mathematics Magazine (1990) page 18

[2] Krusemeyer “A Parenthetical Note (to a paper of Guy)”

Mathematics Magazine (1996) 257-260

[3] Stephen Penrice “Stacks, Bracketings, an CG-Arrangements”

Mathematics Magazine (1999) 321-324

[4] Yujin Shen and Jiang Shult “Sailing Around Binary Trees and Krusemeyer’s Bijection”

Mathematics Magazine (2000) 216-220

[5] Louis Shapiro and Robert Sulanke “Bijections for the Schröder Numbers”

Mathematics Magazine (2000) 369-3760

[6] Christophe Grand : nombreux e-mails