

ÉCOLE NORMALE SUPÉRIEURE DE LYON

UNIVERSITÉ BORDEAUX - LABRI



RAPPORT DE STAGE DE L3

VERS UNE CLASSIFICATION DES “BIJECTIONS CATALANES” ?



Etudiant :
Maxime CAUTRÈS

Encadrant :
Yvan LE BORGNE

31 Mai 2021 - 9 Juillet 2021

Table des matières

| | |
|--|-----------|
| 1 Concepts et outils techniques | 2 |
| 1.1 Histoire des nombres de Catalan | 2 |
| 1.2 Motivation | 3 |
| Reformulation en informatique des $q, t - Catalan$: | 4 |
| 1.3 Le formalisme de départ | 5 |
| 1.4 Les objectifs | 10 |
| 2 Les grammaires à deux règles | 11 |
| 2.1 Adaptation du formalisme initial | 11 |
| 2.2 Recherche de grammaires bijectives à 2 règles | 11 |
| 3 Étude approfondie de ech_0 | 16 |
| 3.1 Retour sur la définition de ech_0 sur les arbres | 16 |
| 3.2 Le passage aux branches | 17 |
| 3.3 L'ordre de ech_0 sur $1\dots 1$ | 17 |
| 3.4 L'ordre de ech_0 | 18 |
| 3.5 L'ordre de ech_0 sur toute branche, par les automates cellulaires | 18 |
| 3.6 Une nouvelle involution | 19 |
| 4 Conclusion | 20 |
| 5 Appendice | 21 |
| A Démonstration de la bijectivité de ech_0 avec ech_0^{-1} | 21 |
| B Démonstration de $\forall G_0 \equiv_{pic} G_1, \forall s \in \mathcal{ADS}, \forall i \in \{0, 1\}, \Phi_{G_0, G_1}(\text{ech}_0 s) = \Phi_{\text{ech}_0^m(G_0, G_1)}(s)$ | 22 |
| C Illustration des transmissions de bijectivité pour la grammaire canonique | 25 |
| D Triangle de Sierpinski | 26 |
| E Démonstration de l'égalité entre le triangle de Pascal mod 2 et la trace des puissances de ech_0 sur $1\dots 1$ | 26 |
| F Démonstration de l'égalité entre le triangle de Sierpinski et le triangle de Pascal mod 2 | 27 |
| G Démonstration de $\forall b \in \{0, 1\}^{\mathbb{N}}, \text{ord}_{1b}(\text{ech}_0) = 2^{\lfloor \log_2 b \rfloor + 1}$ | 28 |
| H Démonstration du caractère involutif de $\mathcal{ADS}_n \rightarrow \mathcal{ADS}_n$ de ech_0^u pour tout $n \in \mathbb{N}$ | 30 |

Introduction

Dans le cadre de ma formation en L3 à l'École Normale Supérieure de Lyon. J'ai pu effectuer un stage de recherche de 6 semaines dans un laboratoire.

J'ai cherché mon stage dans le domaine de la combinatoire. Mes divers échanges de mails m'ont mené jusqu'au LaBRI et à mon encadrant Yvan Le Borgne. Nous avons échangé afin d'avoir un sujet qui nous intéressait et sur lequel j'ai eu grand plaisir de travailler. Je le remercie pour son aide, son encadrement et le temps qu'il m'a consacré sur ces 6 semaines.

Durant mon stage j'ai été très bien intégré à l'équipe Combinatoire et Interaction. Je remercie l'ensemble des membres du LaBRI que j'ai rencontrés pour les bons moments passés ensemble le midi et les discussions passionnantes.

Je tiens aussi à remercier Jérôme Cottenceau, vulgarisateur de la chaîne youtube Eljj qui m'a fait découvrir la combinatoire, les nombres de Catalan et qui plus généralement a su attiser ma curiosité tout le long de mon parcours scientifique.

Les sujets de recherche de l'Équipe Combinatoire et Interaction sont centrés autour de la recherche de résultats de nature mathématique décrivant les structures discrètes, les relations et les comportements d'objets combinatoires. Ces résultats ont des retombées en analyse d'algorithmes et en physique statistique. Mon travail sur les relations entre objets combinatoires s'inscrit donc parfaitement dans le thème.

Le but de mon stage, titré «Vers une classification des “bijections catalanes” ?», est de trouver de nouvelles structures à ces objets, à travers la généralisation d'un formalisme proposé par mon maître de stage dans sa thèse [3]. Je suis aussi à la recherche de nouvelles involutions candidates à une conjecture sur les q, t – Catalan datant d'une vingtaine d'années.

Je vais tout d'abord présenter les notions et le formalisme de départ. Puis je détaillerai mes deux principaux axes de recherche.

1 Concepts et outils techniques

Intéressons nous à des classes d'objets combinatoires qui partagent une décomposition récursive simple. Pour tout objet de taille n , il existe un découpage unique et non ambiguë, en un objet atomique de taille 1 et de deux sous objets indépendant de la même classe sauf pour la somme de leur taille qui vaut $n - 1$. Cette décomposition est naturelle et apparaît à de nombreuses reprises dans divers travaux. (plus d'une centaine recensée par [5, Stanley]). Dans chacun des cas le nombre d'objets de taille n est compté par le n -ème nombre de Catalan noté C_n et dont les premiers termes sont 1,1,2,5,14,42.

1.1 Histoire des nombres de Catalan

Leonhard Euler Tout commence avec Leonhard Euler au XVIII^{ème} siècle lorsque celui-ci s'intéressa au nombre de triangulations d'un polygone régulier à $n + 2$ côtés.

Définition 1. La triangulation d'un polygone est un ensemble maximal par l'inclusion des cordes du polygone (en rouge Figure 1) ne s'intersectant pas.

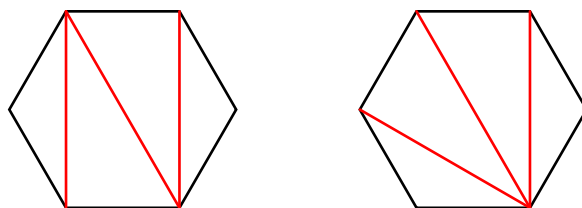


FIGURE 1 – Exemple de deux triangulations de l'hexagone

Il réussit alors à dénombrer le cardinal des triangulations des polygones du carré au décagone. Ce qui lui donna le tableau suivant :









| Nombre de côtés | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----------------|---|---|---|---|---|---|---|---|
| Représentation |  |  |  |  |  |  |  |  |
| Triangulations | 1 | 2 | 5 | 14 | 42 | 132 | 429 | 1430 |

FIGURE 2 – Nombre de triangulations en fonction de la taille du polygone

Eugène Charles Catalan est le premier à identifier la décomposition récursive en deux sous-objets sur une classe d'objets alors qu'il travaille sur les mots de parenthèse. Il remarque que tout mot de parenthèse non-vide se factorise sous la forme $(w)w'$ avec w et w' des mots de parenthèse (éventuellement vides). On a alors $C_n(n) = \frac{\binom{2n}{n}}{n+1}$. Cette expression est la solution de l'expression que l'on doit à la structure inductive de l'objet $C_n(n) = \sum_{k=0}^{n-1} C_n(k)C_n(n-1-k)$. On peut définir les mots de Dyck comme étant les mots de parenthèse avec la substitution suivante $' \rightarrow x$ et $' \rightarrow \bar{x}$. Si x (resp. \bar{x}) joue alors le rôle d'un pas montant (resp. descendant) dans un chemin alors le chemin obtenu est un chemin de Dyck.

Définition 2. Un arbre binaire désigne un arbre où chaque sommet peut avoir un fils gauche et un fils droit.

Définition 3. Un arbre planaire désigne un arbre dont la racine est sur la droite délimitant un demi plan qui contient l'arbre. Les sommets de celui-ci sont ordonnés.

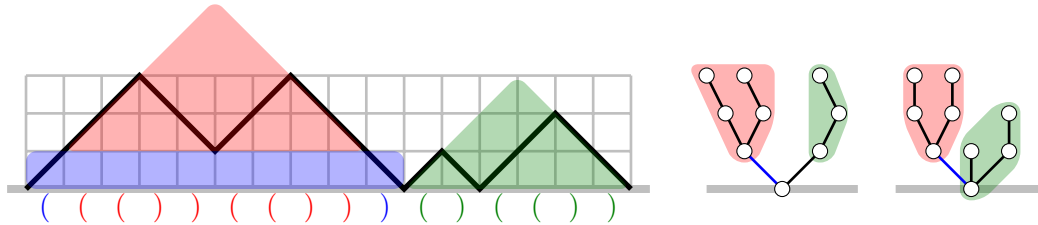


FIGURE 3 – Exemples d'objets combinatoires comptés par les nombres de Catalan de part leur structure avec les chemins de Dyck, les arbres binaires et les arbres planaires

Définition 4. Les Chemins de Dyck noté \mathcal{DW} désignent aussi des mots sur l'alphabet $\{x, \bar{x}\}$ où x (resp. \bar{x}) désigne un pas montant (resp descendant) dans un chemin de Dyck. La notation $|u|_x$ désigne le nombre de x dans les éléments composant u .

$$u \in \mathcal{DW} \iff |u|_x = |u|_{\bar{x}} \text{ et } \forall v \text{ suffixe } u, |v|_x \leq |v|_{\bar{x}}$$

Notation: On parlera de chemin de Dyck ou mot de Dyck pour désigner le même objet. On notera \mathcal{DW}_n la restriction de \mathcal{DW} aux chemins de longueur $2n$ qui sont pour nous de taille n .

1.2 Motivation

Les q,t-Catalan (Présentation Combinatoritienne) Les q,t-Catalans correspondent à une bi-statistique sur les chemins de Dyck. On peut effectivement s'intéresser à deux statistiques :

- **L'Aire sous la courbe** correspond au nombre de carrés entiers entre la courbe et l'axe des abscisses.

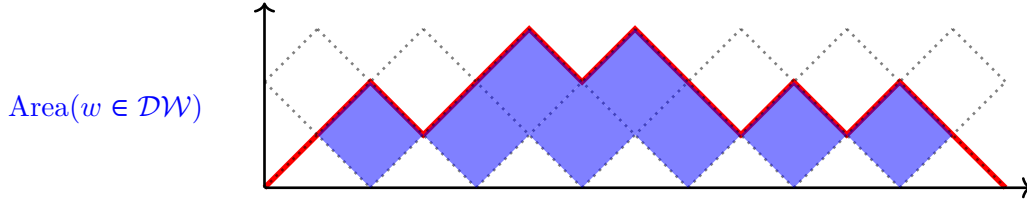


FIGURE 4 – L'Aire sous un chemin de Dyck

- **Le Dinv** correspond au nombre de couples de deux pas montants dont le second est après le premier dans le mot de Dyck et qui sont soit à la même hauteur, soit le second est un moins haut que le premier.

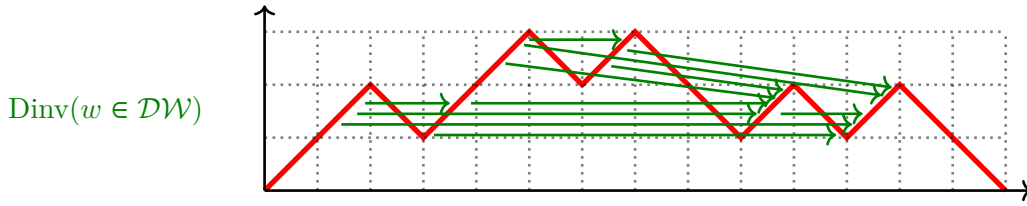


FIGURE 5 – Représentation du Dinv sur un chemin de Dyck

Définition 5. Les q, t – *Catalan* sont alors définis de la manière suivante :

$$\forall n \in \mathbb{N}^* \quad \forall (q, t) \in \mathbb{R}^2, C_n(q, t) := \sum_{w \in \mathcal{DW}_n} q^{\text{Area}(w)} t^{\text{Dinv}(w)}$$

Il s'agit d'un raffinement des nombres de Catalan, au lieu d'ajouter 1 pour chaque élément on compte un mot de Dyck w par un monôme $q^{\text{Area}(w)} t^{\text{Dinv}(w)}$. Dans le contexte hors du stage dans lequel ce raffinement est apparu cela correspond à deux statistiques pertinentes (formant une série de Hilbert pour certains espaces de polynômes).

Propriété: La Symétrie des q, t – *Catalan* :

$$\forall n \in \mathbb{N}^* \quad \forall (q, t) \in \mathbb{R}^2, C_n(q, t) = C_n(t, q)$$

Cette propriété des q, t – *Catalan* est directe par symétrie de la définition algébrique. Cependant il a été compliqué de montrer que la définition combinatoire décrite ici coïncidait avec la définition algébrique (voir les travaux de [1, Haiman, Garsia] et [2, Haglund]). La preuve de cette égalité brisant la symétrie initialement évidente, on ne dispose pas à ce jour d'involution sur les chemins de Dyck expliquant la symétrie connue des q, t – *Catalan*.

Reformulation en informatique des q, t – Catalan : Il y a un lien entre l'**Aire** (resp. le **Dinv**) d'un chemin de Dyck et la **somme des tailles de la pile** (resp. **file**) lors de la **découverte de chaque sommet** du parcours en **profondeur** (resp. **largeur**) de l'arbre planaire associé.

Conjecture 5.1. Une conjecture ouverte depuis 20 ans propose l'existence d'une involution φ telle que :

$$\exists \varphi \text{ Stat } \forall q, t, C(q, t) = \sum_{w \in \mathcal{DW}} q^{\text{Stat}(w)} t^{\text{Stat}(\varphi(w))}$$

Cette involution aurait la propriété d'inverser les sommes des piles ou files des parcours en profondeur et largeur sur les arbres planaires.

1.3 Le formalisme de départ

Dans sa thèse [3] Yvan Le Borgne a travaillé sur cette question à l'aide d'un formalisme précis. Il s'est intéressé aux symétries naturelles de divers objets combinatoires comptés par les nombres de Catalan pour voir si ces symétries induisaient une involution compatible avec les q, t -Catalan. On peut représenter la démarche de cette manière.

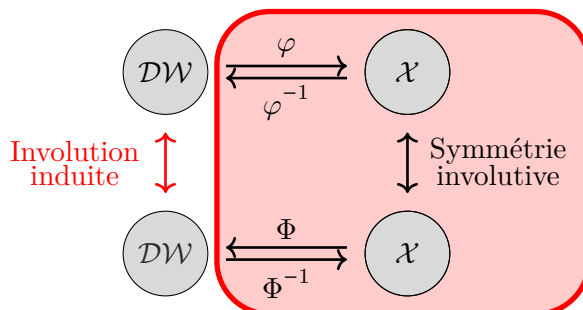


FIGURE 6 – Soit \mathcal{X} un objet combinatoire et $\varphi : \mathcal{DW} \rightarrow \mathcal{X}$ et $\Phi : \mathcal{X} \rightarrow \mathcal{DW}$ une application bijective

L'objet \mathcal{X} qu'il a choisi d'utiliser est la suite quasi décroissante que l'on note \mathcal{ADS} .

Définition 6. \mathcal{ADS} : Soit $a \in \mathbb{N}^{\llbracket 0, n \rrbracket}$, on a alors que

$$(a_i)_{i \in \llbracket 0, n \rrbracket} \in \mathcal{ADS}_n \iff a_0 = 0 \text{ et } \forall k \in \llbracket 0, n-1 \rrbracket, a_{k+1} \in \llbracket 0, a_k + 1 \rrbracket$$

Observation: Une structure arborescente : Soit $s \in \mathcal{ADS}_n$, s peut s'écrire de la manière suivante $0 \underbrace{1 \dots 1}_{t_1 \oplus 1} \underbrace{0 \dots 0}_{t_2}$ avec t_1 qui correspond aux valeurs entre les 2 premiers zéros de s que l'on décrémente de 1 et t_2 à la fin de s à partir du second 0 inclus. t_1 et t_2 sont eux aussi des \mathcal{ADS} . Avec les couleurs on retrouve le schéma vu Figure 3. Toutes les suites presque croissantes suivent ce schéma, ce qui crée une bijection avec les parenthésages et donc les nombres de Catalan. On note alors $s = 0, t_1 \oplus 1, t_2$. Voici un exemple Figure 7.

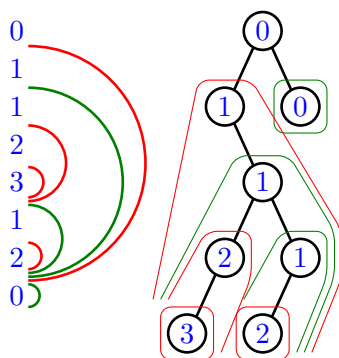


FIGURE 7 – Arbre représentant l'ADS
0, 1, 1, 2, 3, 1, 2, 0.

Construction de l'arbre binaire :

On a $s = 0, t_1 \oplus 1, t_2$ avec t_1 et t_2 des \mathcal{ADS} éventuellement vide.

L'arbre a 0 pour racine et a 2 sous arbres.

- Le sous arbre gauche est celui associé à t_1 où l'on a ajouté 1 à toutes les étiquettes.
- Le sous arbre droit est celui associé à t_2 . Les étiquettes ne changent pas.
- Si t_1 ou t_2 sont vides il n'y a pas de fils gauche ou droit.

Remarque: Si les \mathcal{ADS} ont été choisis en particulier, c'est que la statistique d'Aire des $q, t - \text{Catalan}$ est facile à définir sur les \mathcal{ADS} . L'Aire du chemin de dyck associée à une suite quasi décroissante est exactement la somme des termes de la suite. Ces suites deviennent donc particulièrement intéressantes.

1. On définit \oplus comme ceci : $(b_0, b_1, \dots, b_n) \oplus k := (b_0 + k, b_1 + k, \dots, b_n + k)$

Définition 7. La première bijection φ : On peut rapidement se convaincre que cet objet est en bijection avec les mots de Dyck (\mathcal{DW}) en associant à chaque pas montant sa hauteur (Ce qui nous donne la définition de φ). Elle sera aussi appelée bijection canonique dans la suite. Voici un exemple Figure 8

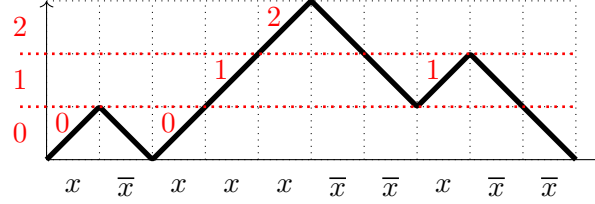


FIGURE 8 – 00121 est l' \mathcal{ADS} associé au \mathcal{DW} $x\bar{x}xx\bar{x}\bar{x}\bar{x}\bar{x}$

φ^{-1} peut s'écrire comme l'insertion d'un pic (un facteur $x\bar{x}$) sur le sommet, dans la dernière descente du chemin de Dyck, dont la hauteur vaut la valeur actuellement lue dans l' \mathcal{ADS} . Pour Φ on va généraliser φ^{-1} en s'inspirant du formalisme de réécriture des grammaires algébriques. On s'intéresse particulièrement à ce type de système car une généralisation d'un système comme celui-ci (En autorisant de permuter circulairement les marques) permet d'obtenir la statistique de dinv du chemin de dyck associé à une suite quasi décroissante comme la somme des termes de la suite.

Définition 8. Un mélange de deux chaînes de caractères notées \sqcup : Le mélange de deux chaînes de caractères ici consistera en l'entrelacement des caractères, c'est-à-dire, une permutation des lettres des deux mots telle que si deux lettres du mot obtenu appartaient au même mot initialement, alors elles doivent être dans le même ordre dans le mot initial et celui obtenu après mélange. Dans la suite, ' $'$ ' désignera la chaîne de caractères contenant une virgule.

Définition 9. La seconde bijection Φ :

Symboles et règles autorisés :

- Les symboles non terminaux (aussi appelé **marque**) : $\{n, \bar{n} \mid n \in \mathbb{N}\}$, en nombre infini contrairement aux grammaires algébriques usuelles.
- Les symboles terminaux : $\{x, \bar{x}\}$
- Les mot atteignable par la grammaire sur l'alphabet des terminaux et non terminaux appartiennent au \mathcal{MDW} pour **Mot de Dyck Marqués**. Plus précisément, ils sont de la forme $\mathcal{MDW} := \{w \sqcup v \mid w \in \mathcal{DW} \wedge v \in \bigcup_{n \in \mathbb{N}} \{ \sqcup_{k \in [0, n]} k\bar{k} \}\}$
- Les grammaires autorisées à une règle (noté \mathcal{GR}^*) :

$$\mathcal{GR}^* := \{(n, \bar{n}) \rightarrow (m_R, m_F) \mid n \in \mathbb{N}, m_R, m_F \in (x, \bar{x} \sqcup n\bar{n} \sqcup (n+1)\overline{n+1})\}$$

- Les grammaires à une règle paramétrable autorisées (noté \mathcal{GR}) :

$$\mathcal{GR} := \left\{ \left\{ \begin{array}{l} \mathbb{N} \rightarrow \\ A \mapsto \end{array} \right. (A, \bar{A}) \rightarrow (m_R, m_F) \mid m_R, m_F \in (x, \bar{x} \sqcup A\bar{A} \sqcup B\bar{B}) \right\}$$

Par exemple $A \mapsto (A, \bar{A}) \rightarrow (x\bar{B}\bar{B}, \bar{x}A\bar{A})$ est une grammaire valide contrairement à $A \mapsto (A, \bar{A}) \rightarrow (\bar{B}A\bar{B}, x\bar{x}\bar{A})$.

Règles d'évolutions :

- **Initialisation :**
 - On choisit une unique règle paramétrable $\mathcal{R} \in \mathcal{GR}$.
 - On commence avec la graine $0\bar{0}$.
- **Évolutions :**
 - On **choisi** quel couple de marque i, \bar{i} on souhaite réécrire.
 - On **supprime toutes les marques strictement plus grandes que i** .
 - On **obtient donc la règle** à appliquer avec $\mathcal{R}(i) \in \mathcal{GR}^*$. Cela revient à remplacer A par i et B par $i + 1$.
 - $\mathcal{R}(i)$ est sous la forme $(i, \bar{i}) \rightarrow (m_R, m_F)$, on **substitue** alors m_R (resp m_F) à la marque i (resp \bar{i}).
- **Terminaison :**
 - Lorsque l'on souhaite s'arrêter, on supprime toutes les marques restantes.
 - On obtient alors un mot de Dyck (DW).

Dans ce formalisme on ne parle pas donc simplement de Φ mais bien de Φ_G avec $G \in \mathcal{GR}$.

Propriété: Toute ADS peut piloter l'insertion via la grammaire, d'où Φ_G une application de $ADS \rightarrow DW$, Une illustration est fournie Figure 9

Démonstration. Par induction

- **Hypothèse d'Induction :** Les marques de $0, \bar{0}$ à $m + 1, \overline{m + 1}$ sont présentes exactement une fois avec m le dernier entier lu de l' ADS , on lit toujours i avant \bar{i} dans le MDW .
- **Initialisation :** On insère uniquement des marques $0, 1, \bar{0}, \bar{1}$ en ayant lu un 0.
- **Induction :** La prochaine valeur m' de l' ADS appartient à $\llbracket 0, m + 1 \rrbracket$. Or par induction, toutes les marques de 0 à $m + 1$ sont dans le mot. Donc on peut réécrire en m' ce qui laissera des marques entre 0 et $m' + 1$. De plus les marques insérées seront dans le bon ordre, i.e. m' avant $\bar{m'}$ et $m' + 1$ avant $\overline{m' + 1}$.
- **Conclusion :** Par induction, on a bien que l'on peut piloter l'algorithme d'insertion avec les ADS .

□

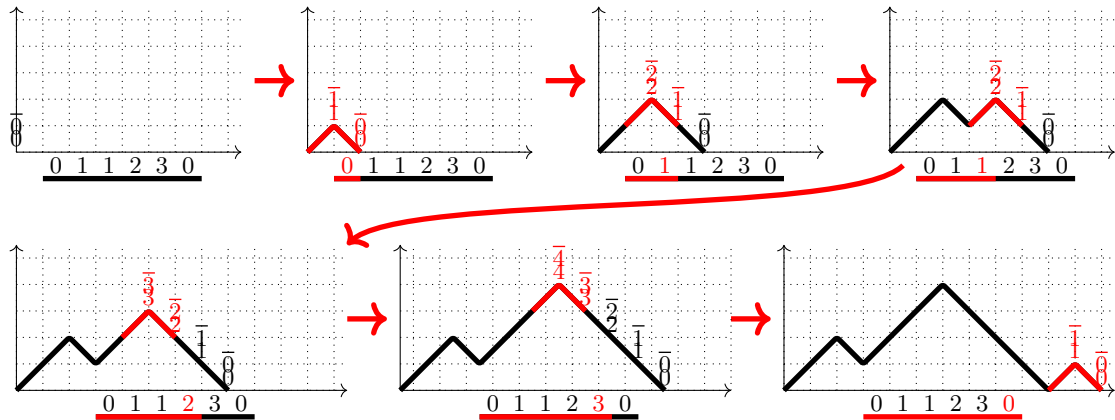


FIGURE 9 – Affichage des étapes de l'algorithme sur 011230 avec la règle $(A, \bar{A}) \rightarrow (xB\bar{B}, \bar{x}A\bar{A})$

On rend compte que cette règle $((A, \bar{A}) \rightarrow (x\bar{B}\bar{B}, \bar{x}A\bar{A}))$ est (une de) celle(s) qui engendre la bijection réciproque de φ .

Théorème 9.1. *On a l'égalité suivante $\forall w \in \mathcal{DW}, \Phi_{(A, \bar{A}) \rightarrow (x\bar{B}\bar{B}, \bar{x}A\bar{A})}(\varphi(w)) = w$*

Démonstration. La preuve se fait par induction, avec l'hypothèse d'induction suivante : Les marques sont uniquement sur la dernière descente (i.e. sur le plus long suffixe de la forme $\bar{x}...\bar{x}$) et sont sous la forme $i\bar{i}$ avec i la hauteur du sommet. \square

Remarque: Φ n'est pas toujours bijective.

Effectivement, prenons l'exemple de la règle $A \mapsto (A, \bar{A}) \rightarrow (xA\bar{A}, B\bar{B}\bar{x})$.

$$\begin{array}{ccccccc} 0\bar{0} & \xrightarrow{\text{En } 0} & x0\bar{0}1\bar{1}\bar{x} & \xrightarrow{\text{En } 0} & xx0\bar{0}1\bar{1}\bar{x}\bar{x} & \xrightarrow{\text{On stop}} & xxx\bar{x}\bar{x} \\ 0\bar{0} & \xrightarrow{\text{En } 0} & x0\bar{0}1\bar{1}\bar{x} & \xrightarrow{\text{En } 1} & x0\bar{0}x1\bar{1}2\bar{2}\bar{x}\bar{x} & \xrightarrow{\text{On stop}} & xxx\bar{x}\bar{x} \end{array}$$

Cela implique donc qu'il faille trouver les grammaires telles que Φ soit bijective. Dans sa thèse, Yvan Le Borgne a détaillé la classification des grammaires.

Combien de grammaires différentes existe-t-il ?

Lemme 9.1. *Si u et v n'ont aucune lettre en commun, $\#(u \sqcup v) = \binom{|u|+|v|}{|u|}$*

Démonstration. Soit u, v deux mots de n et m lettres toutes distinctes. Le nombre de mélanges conservant les ordres des lettres de u et v correspond au nombre de façon de choisir n emplacements dans un mot de $n + m$ lettres, d'y mettre celle de u puis de compléter avec les lettres de v . D'où $\#(u \sqcup v) = \binom{n+m}{n}$ \square

Théorème 9.2. $\#\mathcal{GR} = \#(x, \bar{x} \sqcup A\bar{A} \sqcup B\bar{B}) = 210$:

Démonstration. Cette égalité vient de la définition de \mathcal{GR} où l'on retrouve $x, \bar{x} \sqcup A\bar{A} \sqcup B\bar{B}$.

$$\begin{aligned} \#\mathcal{GR} &= \#(x, \bar{x} \sqcup A\bar{A} \sqcup B\bar{B}) = \sum_{u \in (x, \bar{x} \sqcup A\bar{A} \sqcup B\bar{B})} 1 = \sum_{u \in (x, \bar{x} \sqcup A\bar{A})} \left(\sum_{v \in u \sqcup B\bar{B}} 1 \right) \\ \#\mathcal{GR} &= \sum_{u \in (x, \bar{x} \sqcup A\bar{A})} \binom{3+2+2}{2} = \binom{3+2}{2} \binom{7}{2} = 10 \times 21 = 210 \end{aligned}$$

\square

Quelles sont les grammaires qui induisent des bijection pour Φ

Cette question a été traitée par Yvan Le Borgne dans sa thèse. Cependant nous nous attellerons à une généralisation de cette tâche, il est donc nécessaire de présenter les différents concepts.

Dans la suite on utilisera le raccourci grammaire bijective pour désigner une grammaire qui induit une application bijective pour Φ .

Partie expérimentale J'ai choisi de refaire de zéro la partie expérimentale de la thèse. Cela m'a permis d'obtenir une base solide de code, facilement modelable pour la généralisation, le tout en me familiarisant avec les concepts.

La démarche consiste à générer l'ensemble des \mathcal{ADS} de taille n ainsi que l'ensemble des règles des grammaires. Pour chaque grammaire on vérifie si l'algorithme d'insertion engendre une application injective : Soit c'est le cas alors on considère que la grammaire est une potentiel grammaire bijective à cette taille d'entrée. Soit on enregistre simplement un contre exemple à l'injectivité.

- Sur les 210 grammaires, à partir d' \mathcal{ADS} de taille 3, 178 ne sont pas bijectives.
- Il reste donc 32 candidats. Pour tout les \mathcal{ADS} de tailles inférieures à 16, ces 32 candidats restent bijectifs.
- Ces 32 candidats semblent engendrer 12 bijections différentes des \mathcal{ADS} sur \mathcal{DW}

Partie théorique Plutôt que de démontrer les 32 bijectivités une part une, on va plutôt s'intéresser à les regrouper en fonction de similitudes ou de symétries plus ou moins naturelles.

Deux catégories principales ont été exhibées :

- **Les équivalences** : Il y en a trois. Si $G_0 \equiv_{pic} G_1 \vee G_0 \equiv_{mont} G_1 \vee G_0 \equiv_{desc} G_1$ alors $\Phi_{G_0} = \Phi_{G_1}$
 - \equiv_{pic} : Si G_0 et G_1 de la forme $G_i = (A, \bar{A}) \rightarrow (m_R^i, m_F^i)$ avec $A\bar{A}$ et $B\bar{B}$ non sécables dans G_0 (G_0 est appelée **un mode pics**) et G_1 alors $G_0 \equiv_{pic} G_1 \Leftrightarrow m_R^0 m_F^0 = m_R^1 m_F^1$
 - \equiv_{mont} : Si G_0 et G_1 de la forme $G_i = (A, \bar{A}) \rightarrow ([xj, jx]_i, m_F^i)$ avec $j \in A, B$ alors $G_0 \equiv_{mont} G_1$.
 - \equiv_{pic} : Si G_0 et G_1 de la forme $G_i = (A, \bar{A}) \rightarrow (m_R^i, [\bar{x}j, j\bar{x}]_i)$ avec $j \in A, B$ alors $G_0 \equiv_{desc} G_1$.
- **Les symétries** : Il y en a deux qui agissent différemment :
 - **Échange** : Cette symétrie consiste à inverser le rôle de A et B dans la substitution.
Exemple : $(A, \bar{A}) \rightarrow (xB\bar{B}, \bar{x}A\bar{A}) \xleftrightarrow{\text{ech}^m} (A, \bar{A}) \rightarrow (xA\bar{A}, \bar{x}B\bar{B})$
 - **Miroir** : Cette symétrie consiste à inverser le sens de lecture de la règle ainsi que les barres.
Exemple : $(A, \bar{A}) \rightarrow (xB\bar{B}, \bar{x}A\bar{A}) \xleftrightarrow{\text{mir}^m} (A, \bar{A}) \rightarrow (A\bar{A}x, B\bar{B}\bar{x})$

Théorème 9.3. *Relation pour les symétries :*

Ces symétries induisent de nouvelles symétries sur l'entrée ou la sortie qui vérifie les propriétés suivantes :

$$\Phi_{\text{ech}^m(G)} = \Phi_G \circ \text{ech} \quad (1)$$

$$\Phi_{\text{mir}^m(G)} = \text{mir} \circ \Phi_G \quad (2)$$

Leurs définitions sont les suivantes :

- ech est définie inductivement sur la décomposition récursive canonique des \mathcal{ADS} :

$$\text{ech} := \begin{cases} \text{ech}(\varepsilon) & = \varepsilon \\ \text{ech}(0, t_1 \oplus 1, t_2) & = 0, \text{ech}(t_2) \oplus 1, \text{ech}(t_1) \end{cases}$$

- mir : Soit $w \in \mathcal{DW}_n$, on défini mir de la manière suivante :

$$\text{mir} := \begin{cases} \text{mir}(\varepsilon) = \varepsilon \\ \text{mir}(au) = \text{mir}(u)\bar{a} & \text{si } a \in x \cup \bigcup_{i \in \mathbb{N}} i \\ \text{mir}(\bar{a}u) = \text{mir}(u)a & \text{si } \bar{a} \in \bar{x} \cup \bigcup_{i \in \mathbb{N}} \bar{i} \end{cases}$$

Remarque: Ces relations sont très sympathiques car si les symétries induites sont bijectives, la symétrie conserve la possible bijectivité de la règle.

Théorème 9.4. *Bijectivité des symétries : ech (resp mir) est bijective sur les \mathcal{ADS} (resp \mathcal{DW}).*

Toutes les démonstrations ont été faites dans la thèse. Elles permettent de dire que les symétries transmettent le caractère bijectif d'une grammaire et de même pour les équivalences qui induisent la même application. Cela permet donc de réunir les 32 règles dans 3 catégories comme montré dans la Figure 10

Il ne reste donc plus que 3 démonstrations de bijectivités (Faites dans la thèse).

$$\begin{array}{ccc}
\left\{ \begin{array}{l} \equiv_{pic} (xB\bar{B}, \bar{x}A\bar{A}) \\ \equiv_{pic} (x, B\bar{B}\bar{x}A\bar{A}) \\ \equiv_{mont} (xB, \bar{B}\bar{x}A\bar{A}) \\ \equiv_{mont} (Bx, \bar{B}\bar{x}A\bar{A}) \end{array} \right\} & \xleftrightarrow{\text{mir}^m} & \left\{ \begin{array}{l} (A\bar{A}x, B\bar{B}\bar{x}) \\ \equiv_{pic} (A\bar{A}x, B\bar{B}\bar{x}) \\ \equiv_{pic} (A\bar{A}xB, \bar{B}\bar{x}) \\ \equiv_{desc} (A\bar{A}xB, \bar{x}\bar{B}) \end{array} \right\} \\
\updownarrow \text{ech}^m & & \updownarrow \text{ech}^m \\
\left\{ \begin{array}{l} (xA\bar{A}, \bar{x}B\bar{B}) \\ \equiv_{pic} (x, A\bar{A}\bar{x}B\bar{B}) \\ \equiv_{pic} (xA, \bar{A}\bar{x}B\bar{B}) \\ \equiv_{desc} (Ax, \bar{A}\bar{x}B\bar{B}) \end{array} \right\} & \xleftrightarrow{\text{mir}^m} & \left\{ \begin{array}{l} (B\bar{B}x, A\bar{A}\bar{x}) \\ \equiv_{pic} (B\bar{B}x, A\bar{A}\bar{x}) \\ \equiv_{pic} (B\bar{B}xA, \bar{A}\bar{x}) \\ \equiv_{mont} (B\bar{B}xA, \bar{x}\bar{A}) \end{array} \right\} \\
\left\{ \begin{array}{l} (AB\bar{B}x, \bar{x}\bar{A}) \\ \equiv_{desc} (AB\bar{B}x, \bar{A}\bar{x}) \end{array} \right\} & \xleftrightarrow{\text{mir}^m} & \left\{ \begin{array}{l} (Ax, \bar{x}B\bar{B}\bar{A}) \\ \equiv_{mont} (xA, \bar{x}B\bar{B}\bar{A}) \end{array} \right\} \\
\updownarrow \text{ech}^m & & \updownarrow \text{ech}^m \\
\left\{ \begin{array}{l} (BA\bar{A}x, \bar{x}\bar{B}) \\ \equiv_{desc} (BA\bar{A}x, \bar{B}\bar{x}) \end{array} \right\} & \xleftrightarrow{\text{mir}^m} & \left\{ \begin{array}{l} (Bx, \bar{x}A\bar{A}\bar{B}) \\ \equiv_{mont} (xB, \bar{x}A\bar{A}\bar{B}) \end{array} \right\} \\
\left\{ \begin{array}{l} (BA\bar{B}x, \bar{x}\bar{A}) \\ \equiv_{desc} (BA\bar{B}x, \bar{A}\bar{x}) \end{array} \right\} & \xleftrightarrow{\text{mir}^m} & \left\{ \begin{array}{l} (Ax, \bar{x}B\bar{A}\bar{B}) \\ \equiv_{mont} (xA, \bar{x}B\bar{A}\bar{B}) \end{array} \right\} \\
\updownarrow \text{ech}^m & & \updownarrow \text{ech}^m \\
\left\{ \begin{array}{l} (AB\bar{A}x, \bar{x}\bar{B}) \\ \equiv_{desc} (AB\bar{A}x, \bar{B}\bar{x}) \end{array} \right\} & \xleftrightarrow{\text{mir}^m} & \left\{ \begin{array}{l} (Bx, \bar{x}A\bar{B}\bar{A}) \\ \equiv_{mont} (xB, \bar{x}A\bar{B}\bar{A}) \end{array} \right\}
\end{array}$$

FIGURE 10 – Classification des différentes grammaires en fonction de leurs symétries et équivalences

Une solution à notre problème ouvert ? Malheureusement, aucune de ces symétries engendrent l'involution recherchée. On souhaite donc modifier le formalisme pour essayer d'analyser de nouvelles symétries. Une piste proposée dans la thèse suggère d'utiliser des grammaires à deux règles.

1.4 Les objectifs

L'exploration de l'extension du formalisme : Nous souhaitons classifier de la même manière que dans le cas des grammaires à une règle les grammaires à deux règles. Le but serait de trouver toutes les grammaires bijectives de \mathcal{GR}^2 . On partira donc à la recherche de symétries et équivalences pour se simplifier la tâche.

La recherche d'involution pour la conjecture : Une fois que le formalisme sera amplement étudié, on pourra s'atteler à tester si les involutions sont des candidates valides à la distribution des $q, t - Catalan$.

On pourrait aussi s'intéresser aux symétries d'autres objets : Effectivement, Stanley fournit de nombreux objets avec leurs symétries propres. Il serait intéressant de voir comment elles s'induisent sur les mots de Dyck.

2 Les grammaires à deux règles

2.1 Adaptation du formalisme initial

Explicitons l'extension : A partir de maintenant, nous allons utiliser deux règles pour chaque grammaire. Le changement majeur sera une fois que l'on a choisi la marque sur laquelle on souhaite réécrire. Selon la parité de la marque sélectionnée on choisira la première ou la seconde règle pour la réécriture. Ce formalisme est apparu naturellement sur un exemple de la thèse pour interpréter combinatoirement une variante à la méthode du noyau. Cependant l'étude systématique n'avait pas été menée.

Règles d'évolutions :

- **Initialisation :**
 - On choisi **deux** règles paramétrables $\mathcal{R}_0, \mathcal{R}_1 \in \mathcal{GR}$.
 - On commence avec la graine $0\bar{0}$.
- **Évolutions :**
 - On **choisi** quel couple de marque i, \bar{i} on souhaite réécrire.
 - On **supprime toutes les marques strictement plus grandes que i** .
 - On **obtient donc la règle** à appliquer avec $R_{i \bmod 2(i)} \in \mathcal{GR}^*$.
 - $R_{i \bmod 2(i)}$ est sous la forme $(i, \bar{i}) \rightarrow (m_R, m_F)$, on **substitue** alors m_R (resp m_F) à la marque i (resp \bar{i}).
- **Terminaison :**
 - Lorsque l'on souhaite s'arrêter, on supprime toutes les marques restantes.
 - On obtient alors un mot de Dyck (\mathcal{DW}).

Dans ce formalisme on ne parle pas de Φ mais de Φ_G avec $G \in \mathcal{GR}^2$.

Adaptation des symétries : Que ce passe-t-il si on applique une symétrie uniquement sur l'une des deux règles? D'une manière générique on pose la notation suivante pour n'importe quelle symétrie noté Sym^m :

$$\begin{cases} Sym_{01}^m(G_0, G_1) & := (Sym^m(G_0), Sym^m(G_1)) \\ Sym_0^m(G_0, G_1) & := (Sym^m(G_0), G_1) \\ Sym_1^m(G_0, G_1) & := (G_0, Sym^m(G_1)) \end{cases}$$

2.2 Recherche de grammaires bijectives à 2 règles

Petits mots sur les algorithmes Comme il y a 210 règles, cela nous donne 44100 grammaires à tester. J'ai codé tout les algorithmes et ce sont des modifications de ceux utilisés pour retrouver les résultats de la thèse. Il y a des algorithmes d'énumérations par force brute ainsi qu'un programme interactif qui permet de jouer avec le modèle pour pouvoir essayer et ensuite exporter directement en \LaTeX les expérimentations qui étaient concluantes.

Partie expérimentale Les études expérimentales que j'ai menées vont jusqu'à des \mathcal{ADS} de taille 16, il est cependant important de signaler que toutes les données présentées ici sont constantes à partir de $n = 5$.

Voici ici un tableau (Figure 11) récapitulatif des diverses expérimentations :

| Nombre de sous règles bijectives | Nombre de grammaires bijectives | ech_{01}^m | ech_0^m | ech_1^m | mir_{01}^m | mir_0^m | mir_1^m |
|----------------------------------|---------------------------------|---------------------|------------------|------------------|---------------------|------------------|------------------|
| 0 | 68 | 0 | 0 | 0 | 68 | 28 | 28 |
| 1 | 116 | 0 | 16 | 16 | 116 | 4 | 4 |
| 2 | 602 | 472 | 528 | 528 | 602 | 324 | 324 |
| total | 786 | 472 | 544 | 544 | 786 | 356 | 356 |

FIGURE 11 – Données obtenue par simulation

Analyse des résultats empiriques

- Il y a **43314 grammaires non bijectives** pour lesquelles il existe des contre-exemples à l'injectivité de taille inférieure à 5.
- On **conjecture** l'existence de **786 grammaires bijectives**.
- Donc il existe des paires de deux règles bijectives qui forment des grammaires non bijectives.
- Réciproquement, il semble y avoir des couples de 2 règles dont au moins une non bijective qui forment des grammaires bijectives.
- On voit aussi que pour toutes les symétries sauf mir^m (conjecture) aucune ne conserve réellement la bijectivité dans tous les cas.

Cela signifie qu'on ne pourra pas aussi simplement travailler pour transmettre les bijectivités. Le cas par cas sur de petits ensembles semble obligatoire. Ce fût une petite déception car nous pensions initialement que tout serait facilement regroupable.

On va tout de même essayer de développer au maximum la méthode des symétries.

Petits retours sur les symétries induites

- $\text{mir}^m \rightarrow \text{mir}$ **pour une règle** : On fait une rapide induction, supposons que mir^m induise une symétrie axiale d'axe vertical sur la sortie. Lorsque l'on applique la règle en miroir, au lieu d'obtenir $AxB\bar{x}C$ que l'on aurait eu avec la règle originale avec A, B, C des mots de marques on obtient $\text{mir}(C)x\text{mir}(B)\bar{x}\text{mir}(A)$ par l'hypothèse d'induction pour A, B et C et le miroir de la règle. Ce qui vérifie bien l'hypothèse d'induction.

On a donc bien la relation suivante vérifiée :

$$\forall G \in \mathcal{GR}, \Phi_{\text{mir}^m(G)} = \text{mir} \circ \Phi_G$$

Cette relation est toujours vérifiée pour $\text{mir}_{01}^m : \mathcal{GR}^2 \rightarrow \mathcal{GR}^2$. Effectivement, si chacune des deux règles est inversée, alors le résultat au global sera aussi inversé d'où de nouveau la relation suivante :

$$\forall G \in \mathcal{GR}^2, \Phi_{\text{mir}_{01}^m(G)} = \text{mir} \circ \Phi_G$$

De plus, $\text{mir}^m : \mathcal{DW} \rightarrow \mathcal{DW}$ est une involution donc on a bien que $\text{mir}_{01}^m : \mathcal{GR}^2 \rightarrow \mathcal{GR}^2$ conserve le caractère bijectif d'une grammaire à deux règles. Ce que l'on avait remarqué dans le tableau.

- mir_0^m et mir_1^m : L'étude n'a pas été plus amplement menée car aucun pattern n'a été trouvé.

- $\text{ech}^m \rightarrow \text{ech}$: On a vu que l'on peut poser la structure $0, t_1 \oplus 1, t_2$ à n'importe quelle éléments $s \in \mathcal{ADS}$. Lorsque l'on écrit avec le premier 0, on introduit deux marques A et B . Les marques B sont utilisées pour réécrire avec la sous-séquence $t_1 \oplus 1$, alors que les marques A sont utilisées pour t_2 . Si l'on échange A et B dans la règle, cela veut dire si l'on souhaite obtenir le même résultat qu'il faille inverser t_1 et t_2 . D'où la définition récursive de ech .

Cette définition vérifie bien $\Phi_{\text{ech}^m(G)} = \Phi_G \circ \text{ech}$, la preuve technique est faite dans la thèse. Cependant, pour $\text{ech}_{01}^m : \mathcal{GR}^2 \rightarrow \mathcal{GR}^2$ cette relation ne marche plus, comme le montre le tableau. Il faudra donc trouver des cas sur lesquels on puisse le démontrer.

- ech_0 et ech_1 : Lorsque l'on regarde la définition récursive de ech sur les \mathcal{ADS} , on remarque qu'elle travaille directement sur la structure d'arbre. Et qu'elle effectue des rotations des fils de tous les nœuds dans l'arbre binaire associé. Voici une illustration Figure 12.

On sait aussi qu'appliquer ech_0^m à une grammaire revient à inverser uniquement la règle qui sera utilisée avec les marques paires, donc à inverser le rôle de A et B uniquement pour les marques paires et donc permuter les sous arbres d'un sommet uniquement si l'étiquette du nœud avant transformation est paire. On illustre cette remarque Figure 12, les flèches bleues (ech) ou rouges (ech_0) indiquant les sommets où les sous-arbres sont échangés.

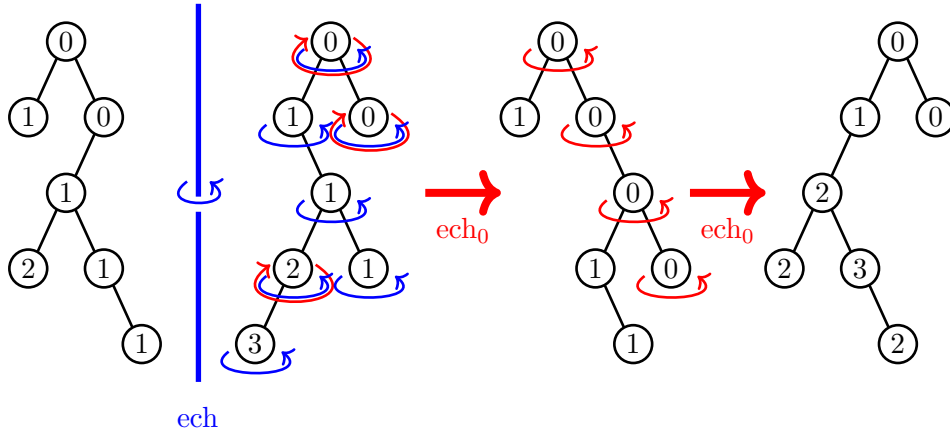


FIGURE 12 – Application de ech et ech_0 sur l'arbre correspondant à l' \mathcal{ADS}

Observation: En jouant un peu avec la transformation, on voit que l'on peut pour tout $s \in \mathcal{ADS}$ propager certaines informations dans l'arbre de $\text{ech}_0(s)$ pour retrouver où avait eu lieu les rotations. Ce procédé permet alors de réappliquer les rotations et de retrouver l'arbre original. Le fonctionnement est le suivant :

- On ajoute une valeur i sur la racine que l'on initialise à 0.
- on lance un parcours de l'arbre, pour tout sommet au fur et à mesure du parcours, on calcule la somme de son i et de son étiquette :
 - Si la somme est paire. Cela signifie que ce sommet était pair dans l'arbre de base donc il faut faire une rotation, on attribue alors $i + 1$ au nouveau fils gauche et $i - 1$ au nouveau fils droit. Puis on explore le sous arbre gauche et le sous arbre droit.
 - Si la somme est impaire, alors dans l'arbre original il n'y avait pas eu de rotation donc il n'est pas nécessaire d'en faire une. On se contente de transmettre i aux fils sans le modifier.

Pour ces diverses définitions, on va introduire un compteur, j , contenant un entier déduit de la branche parente des sous-arbres.

Définition 10. Expression de ech_i .

En retravaillant l'expression de ech on peut définir ech_{01} et $\text{ech}_{01}^t | \forall s \in \mathcal{ADS}, \text{ech}_{01}(s) = \text{ech}_{01}^t(s, 0)$:

$$\text{ech}_{01}^t((k, t1, t2), j) := \begin{cases} \varepsilon & \text{si } k = \varepsilon \\ k + j, \text{ech}_{01}^t(t2, j + 1), \text{ech}_{01}^t(t1, j - 1) & \text{sinon} \end{cases}$$

Il devient dès lors relativement facile d'exprimer ech_i via $\text{ech}_i^t | \forall s \in \mathcal{ADS}, \text{ech}_i(s) = \text{ech}_i^t(s, 0)$:

$$\text{ech}_i^t((k, t1, t2), j) := \begin{cases} \varepsilon & \text{si } k = \varepsilon \\ k + j, \text{ech}_{ti}^t(t2, j + 1), \text{ech}_i^t(t1, j - 1) & \text{sinon si } k \equiv i[2] \\ k + j, \text{ech}_i^t(t1, j), \text{ech}_i^t(t2, j) & \text{sinon} \end{cases}$$

Remarque: ech_0 n'est pas involutive. Effectivement, dans la Figure 12, on voit bien que la double application de ech_0 n'est pas involutive. Cela complexifie beaucoup les raisonnements car on souhaite travailler sur des involutions. On va donc passer par l'application réciproque pour montrer sa bijectivité.

Définition 11. ech_0^{-1} On a donc un protocole récursif pour définir une possible application réciproque. On la définit de telle sorte :

$$\text{ech}_i^{t,-1}((k, t1, t2), j) := \begin{cases} \varepsilon & \text{si } k = \varepsilon \\ k + i, \text{ech}_i^{t,-1}(t2, i + 1), \text{ech}_i^{t,-1}(t1, j - 1) & \text{sinon si } k + j \equiv i[2] \\ k + i, \text{ech}_i^{t,-1}(t1, i), \text{ech}_i^{t,-1}(t2, j) & \text{sinon} \end{cases}$$

De même que pour ech_0 , on a : $\text{ech}_0^{-1}(s) = \text{ech}_0^{t,-1}(s, 0)$.

Théorème 11.1. ech_i est bijective et sa bijection réciproque est ech_i^{-1}

Démonstration. La démonstration est en Appendice A pour $i = 0$. La démonstration étant très similaire pour $i = 1$ elle n'a pas été intégrée. \square

Trouvons maintenant un cas tel que l'on ait $\Phi_{\text{ech}_i^m(G)} = \Phi_G \circ \text{ech}$ avec $i \in \{0, 1, 01\}$ Avec les simulations on observe un petit groupe de règles pour lesquelles les égalités du dessus semblent fonctionner. Ce petit groupe est difficile à décrire mais les couples de la forme $(G_0, G_1) | G_0 \equiv_{pic} G_1$ semblent conserver leur supposées bijectivités.

Théorème 11.2. $\forall G_0 \equiv_{pic} G_1, \forall s \in \mathcal{ADS}, \forall i \in \{0, 1, 01\}, \Phi_{G_0, G_1}(\text{ech}_i s) = \Phi_{\text{ech}_i^m(G_0, G_1)}(s)$

Démonstration. La preuve est détaillée pour $i = 0$ en Appendice B. Les démonstrations étant très similaires pour $i \in \{1, 01\}$ elle n'ont pas été intégrées. \square

Théorème 11.3. Pour $i \in \{01, 0, 1\}$, l'application $\text{ech}_i^m : \mathcal{GR}^2 \rightarrow \mathcal{GR}^2$ transmet le caractère bijectif d'une paire de règles \equiv_{pic} à la paire image.

Démonstration. Pour tout $i \in \{0, 1, 01\}$ comme on a $\text{ech}_i : \mathcal{ADS} \rightarrow \mathcal{ADS}$ bijective et $\Phi_{G_1, G_2} \circ \text{ech}_i = \Phi_{\text{ech}_i(G_1, G_2)}$, alors si Φ_{G_1, G_2} est bijective on a $\Phi_{G_1, G_2} \circ \text{ech}_i$ bijective par composition et donc $\Phi_{\text{ech}_i(G_1, G_2)}$ est bijective. \square

Conclusion: On a donc réussi à montrer que si les deux règles de la grammaire sont \equiv_{pic} alors le caractère bijectif de la grammaire est induit à la grammaire image par ech_i^m pour $i \in \{0, 1, 01\}$ et mir^m .

Proposition 11.1. Les grammaires de la forme (G, G) avec G bijective $\in \mathcal{GR}$ sont bijectives

Démonstration. Si les deux règles de la grammaires sont identiques, la règle appliquée en fonction de la marque devient une constante, on retombe donc dans l'étude de la thèse dans le cas à une règle. Si la règle engendre une grammaire à une règle bijective alors la grammaire engendrée par deux fois cette règle l'est aussi. \square

Théorème 11.4. *Les équivalences \equiv_{pic}^i pour $i \in \{0, 1, 01\}$ transmettent la bijectivité des grammaires dont les deux règles sont des modes pics.*

Démonstration. Procédons par induction

Hypothèse d'induction: Dans le mot de dyck marqués, les marques d'un même couple sont insécables

Initialisation: Pour $n = 0$, le mot est $0\bar{0}$

Induction: On suppose l'Hypothèse Induction valide après les k premières insertions.

Les deux règles sont \equiv_{pic} donc toutes deux des modes pics. Or les modes avec leurs équivalents pics ont cette propriété vérifiée :

$$G_i \equiv_{pic} G_i' \Leftrightarrow m_R^i m_F^i = m_R^i m_F^i$$

Elle est très intéressante car elle montre que si l'on réécrit en i et que $i\bar{i}$ est présent dans le mot de dyck marqué (**HI**) alors on remplacera toujours par le même motif. Qui insérera $i\bar{i}$ et $(i+1)\overline{i+1}$ ce qui respecte l'hypothèse d'induction.

Conclusion: On a donc que \equiv_{pic}^i pour $i \in \{0, 1, 01\}$ transmettent plus que la bijectivité car elle induisent la même application si G_0 et G_1 sont des modes pics. \square

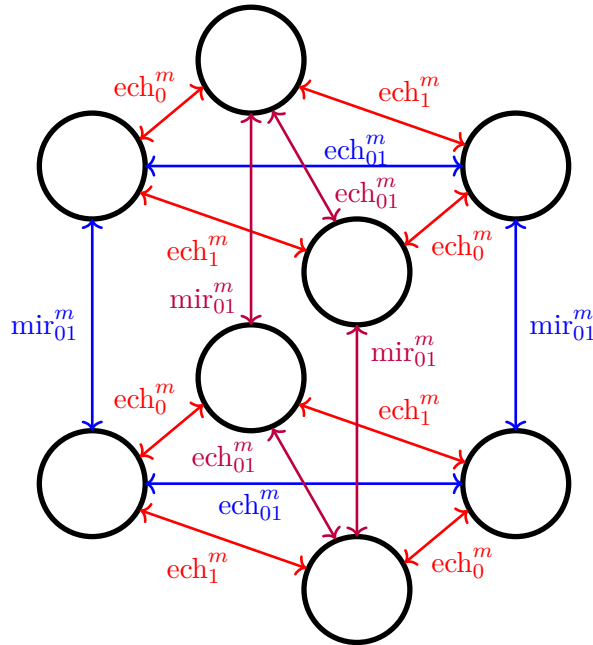


FIGURE 13 – Représentation des transmissions de bijectivités entre objets

On peut donc illustrer tout ceci dans ce schéma des transmissions de bijectivité avec la Figure 10 Dans ce diagramme on peut placer 82 grammaires en partant de la grammaire à 2 règles suivante $(A, \bar{A}) \rightarrow (x\bar{B}\bar{B}, \bar{x}A\bar{A})$ et $(A, \bar{A}) \rightarrow (x\bar{B}\bar{B}, \bar{x}A\bar{A})$. On peut observer le schéma correspondant à ces règles en Figure C.

3 Étude approfondie de ech_0

Nous avons détaillé la bijectivité de ech_0 dans la partie précédente. Cependant nous sommes à la recherche d'involution et le fait que celle-ci n'en soit pas une nous intrigue. On s'intéresse donc à calculer son ordre, voir s'il existe et si oui construire une involution à partir de ses itérées.

Définition 12. L'ordre d'une application

- Sur une entrée :

$$\text{ord}_s(f) = \begin{cases} \min(n \in \mathbb{N} | f^n(s) = s) & \text{si } s \neq \emptyset \\ +\infty & \text{sinon} \end{cases}$$

$$\text{ord}(f) = \begin{cases} \min(n \in \mathbb{N} | f^n = \text{Id}) & \text{si } s \neq \emptyset \\ +\infty & \text{sinon} \end{cases}$$

3.1 Retour sur la définition de ech_0 sur les arbres

$$\text{ech}_i((k, t_1, t_2), j) := \begin{cases} \varepsilon & \text{si } k = \varepsilon \\ k + j, \text{ech}_i(t_2, j + 1), \text{ech}_i(t_1, j - 1) & \text{sinon si } k \equiv i[2] \\ k + j, \text{ech}_i(t_1, j), \text{ech}_i(t_2, j) & \text{sinon} \end{cases}$$

On rappelle que la définition de ech_0 exprime la rotation des fils si l'étiquette du nœud est paire.

Intéressons nous à la répartition des étiquettes. On rappelle la structure arborescente d'une \mathcal{ADS} qui se détaille $s = 0, t_1 \oplus 1, t_2$. Le fils droit d'un nœud a la même étiquette que le parent car t_2 commence par un 0. Et le fils gauche a l'étiquette valant celle du père plus 1 car $t_1 \oplus 1$ commence avec un 1. Donc l'étiquette d'un sommet correspond au nombre de arêtes menant à un fils gauche empruntées sur la branche allant de la racine au sommet que l'on souhaite étiqueter. Cela induit donc des couches d'étiquettes sur l'arbre. Or ech_0 ne travaille que sur les étiquettes, donc on peut isoler les branches pour calculer les images uniquement sur les branches puis reconstituer un arbre avec les branches images obtenues. On peut alors calculer l'ordre sur chaque branche pour trouver une borne supérieure à l'ordre sur l'arbre. De plus ech_0 envoie une branche sur une branche car elle n'effectue que des rotations de sommets.

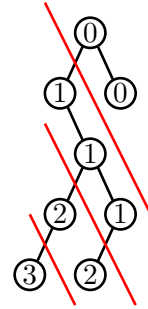


FIGURE 14 – Illustration des couches de la structure arborescente des \mathcal{ADS}

Théorème 12.1. $\forall s \in \mathcal{ADS}, \text{ord}_s(\text{ech}_0) \leq \max_{b \text{ branche} \in s} \text{ord}_b(\text{ech}_0)$ et le cas d'égalité existe

Démonstration. Une branche de taille n appartient au groupe des branches de tailles n . Il est de cardinal 2^n . Toute application de cet ensemble dans lui même engendre un sous groupe dont le cardinal divise celui du groupe (Lagrange). Donc l'ordre de chaque branche est un diviseur de $2^{\text{taille(branche)}}$ donc une puissance de 2 plus petite. Or le PGDC de puissances de 2 vaut le maximum. Cependant, si l'on a plusieurs branches de même taille, on peut ne pas avoir besoin d'appliquer $\text{ord}_b(\text{ech}_0)$ fois ech_0 car les branches peuvent permutationner en utilisant moins de fois ech_0 . On a l'égalité si s est une branche par exemple. \square

On peut donc dès maintenant travailler sur les branches.

3.2 Le passage aux branches

Représentation d'une branche : L'unique information importante sur une branche, qui est nécessaire pour retrouver l'étiquetage est l'emplacement des fils gauches. On va donc choisir de représenter une branche avec un mot binaire obtenu en lisant les arêtes à partir de la racine et en leur associant 0 (resp 1) si l'arête mène vers un fils droit (resp gauche). On indicera les nombres binaires à partir de 1 pour la suite.

ech_0 sur les mots binaires :

Définition 13. Définition induite de ech_0 sur les branches :

$$\text{ech}_0(b)_i = b_i + 1 + \underbrace{\sum_{k \in \llbracket 1, i-1 \rrbracket} b_k}_{*} \mod 2$$

* ici correspond au fait de compter le nombre de fils gauche dans le chemin jusqu'au sommet puis d'inverser. Or l'arête est inversée uniquement si la somme est paire, or $\text{inverser} \mod 2$ c'est ajouter 1 d'où le +1. Cela revient à ajouter une arête fictive qui rend la racine fils gauche, on note $1b$ la branche après l'ajout de l'arête fictive, elle est indicée à partir de 0. On remarque aussi la redéfinition suivante de $\text{ech}_0(1b)$

$$\text{ech}_0(1b)_i = b_i + \underbrace{\left(b_{i-1} + \sum_{k \in \llbracket 0, i-2 \rrbracket} b_k \mod 2 \right)}_{\text{ech}_0(1b)_{i-1}} \mod 2 = b_i + \text{ech}_0(1b)_{i-1} \mod 2$$

3.3 L'ordre de ech_0 sur $1\dots 1$

Les simulations disent que l'ordre sur l'entrée $\underbrace{1\dots 1}_n$ croît par palier suivant $\underbrace{1\dots 1}_n \mapsto 2^{\lfloor \log_2 n \rfloor + 1}$ comme on le voit Figure 15. Cette entrée semble donc être un bon candidat pour montrer que l'ordre est infini. Analysons la trace des puissances successives de ech_0 sur $1b := 1\underbrace{1\dots 1}_{n-1}$.

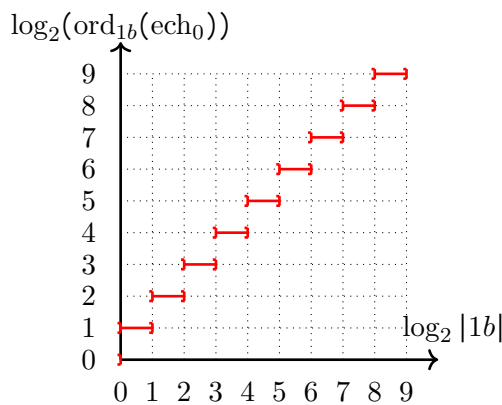


FIGURE 15 – Évolutions de l'ordre en fonction de n avec $b = \underbrace{1\dots 1}_{n-1}$

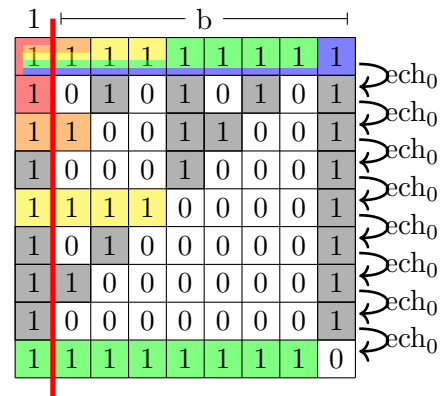


FIGURE 16 – Trace de $1b = 1\underbrace{1\dots 1}_{n-1}$ par les puissances de ech_0 dessinant le triangle de Sierpinski

On est fort surpris la première fois lorsque la simulation nous affiche Figure 16 ce qui ressemble au triangle de Sierpinski (Figure ??). Mais on est aussi très content que ce soit le cas car il s'agit d'un objet bien connu.

Théorème 13.1. *La trace des puissances de ech_0 sur $1b$ dessine le triangle de pascal modulo 2.*

Démonstration. La démonstration par induction est faite en Appendice E □

Théorème 13.2. *Le triangle de pascal modulo deux dessine le triangle de sierpinski [6].*

Démonstration. La preuve est faite en Appendice F □

Théorème 13.3. $\text{ord}_{1b}(\text{ech}_0) = 2^{\lfloor \log_2 |b| \rfloor + 1}$ si b du sous la forme $1\dots 1$.

Démonstration. L'hypothèse d'induction de la dernière preuve est la clef de cette démonstration. Effectivement on cherche la première ligne telle que l'entrée se répète. Soit $n := |1b|$ notre entrée. On a $1b' := \underbrace{1\dots 1}_{2^{\lfloor \log_2(n) \rfloor}}$ préfixe de $1b$. Via l'hypothèse d'induction on a que $\text{ord}_{1b'}(\text{ech}_0) = 2^{\lfloor \log_2(n) \rfloor}$. D'où on a $2^{\lfloor \log_2(n) \rfloor} | \text{ord}_{1b}(\text{ech}_0)$. Or il n'y a pas l'égalité car $\text{ech}_0^{2^{\lfloor \log_2(n) \rfloor}}(1b)_{2^{\lfloor \log_2(n) \rfloor}} = 0$. Cependant si l'on passe à la puissance de deux du dessus on recopie le carré initial en dessous donc aussi l'entrée. D'où $\text{ord}_{1b}(\text{ech}_0) = 2^{\lfloor \log_2(n-1) \rfloor + 1}$ □

3.4 L'ordre de ech_0

Théorème 13.4. *Infinité de l'ordre On a donc $\forall n, \exists k | b = \underbrace{1\dots 1}_k \wedge \text{ord}_{1b}(\text{ech}_0) \geq n$. Ce qui implique que $\text{ord}(\text{ech}_0) = +\infty$.*

Démonstration. $\forall n \in \mathbb{N}, b = \underbrace{1\dots 1}_{n-1} \implies \text{ord}_{1b}(\text{ech}_0) = 2^{\lfloor \log_2(n-1) \rfloor + 1} \geq n$ □

3.5 L'ordre de ech_0 sur toute branche, par les automates cellulaires

On rappelle que l'objectif est de trouver de nouvelles involutions. Nous avons l'espoir de pouvoir en obtenir si elle avait eu un ordre fini. Cependant, ce n'est pas le cas. On peut en dernier recours voir si l'on arrive à calculer l'ordre de ech_0 pour chaque élément.

Pour le moment, nous n'avons regardé que la trace de $1\dots 1$. Nous pouvons le faire pour n'importe quelle entrée mais le travail est très fastidieux. Essayons de trouver une nouvelle représentation de notre problème plus appropriée.

Un automate cellulaire ? Effectivement, depuis le début nous propageons les données lignes par lignes de la gauche vers la droite. Cependant ce n'est pas l'unique manière de procéder. On peut aussi les propager diagonale par diagonale (une diagonale monte vers la droite) et incliner le regard de $\frac{\pi}{4}$ vers la gauche. On remarque alors que la relation d'induction revient à faire la somme des deux cases supérieures adjacentes le tout mod 2. Cela rappelle les automates cellulaires à une dimension de [4, Wolfram]. Voyons comment on peut s'y ramener avec l'illustration Figure 17.

Pour obtenir cette représentation j'ai utilisé la règle d'évolution suivante :

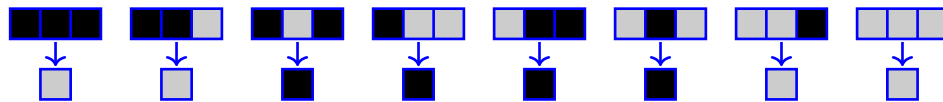


FIGURE 18 – Règle 60 de Wolfram

Les cases blanches dans l'automate cellulaire sont les cases qui ne nous intéressent pas. La principale difficulté de cette représentation est qu'il n'est pas évident de générer l'entrée. Effectivement, on voit que l'entrée $1\dots 1$ est naturellement écrite par l'automate. Il faut donc trouver un moyen de paramétrer l'entrée de l'automate pour que les valeurs au niveau de l'entrée

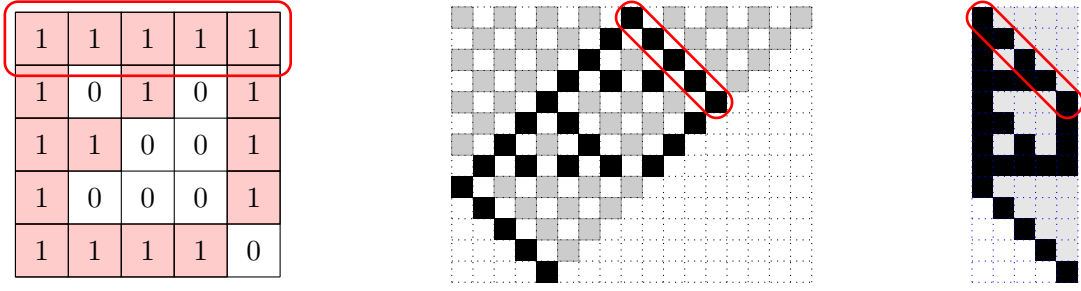


FIGURE 17 – Représentation du même objet en Triangle de Pascal et en Automate de Wolfram avec les règles 90 et 60.

de ech_0 soient correctes. Pour cela on définit Π l'application qui à une entrée de ech_0 paramètre correctement l'automate cellulaire.

De plus, cette règle 60 est dite additive [4]. Elle calcule la somme $\text{mod } 2$ de 2 cellules adjacentes pour obtenir l'état d'une troisième sous la seconde. Elle a donc la propriété que l'on peut découper l'entrée, faire les calculs sur chaque morceaux obtenus, puis recombinaison simplement en faisant la somme modulo 2. Or c'est très intéressant car on comprend très bien le cas où il y a une seule cellule activée. Pour calculer la trace on utilise Π pour trouver les cellules de l'entrée de l'automate, puis on somme des triangles de Sierpinski pour chaque cellule activée de l'entrée.

On a donc maintenant les outils pour trouver l'ordre pour tout éléments.

Théorème 13.5.

$$\forall b \in \{0, 1\}^{\mathbb{N}} \text{ord}_{1b}(\text{ech}_0) = 2^{\lfloor \log_2 |b| \rfloor + 1}$$

Démonstration. La preuve est en Appendice G. □

3.6 Une nouvelle involution

Grâce à l'ordre on peut maintenant construire une nouvelle involution :

Définition 14. ech_0^u :

Ici on appellera profondeur d'un arbre enraciné le nombre d'arêtes du chemin le plus court de la racine à la feuille la plus profonde.

$$\text{ech}_0^u : \begin{cases} \mathcal{ADS} & \rightarrow \mathcal{ADS} \\ s & \mapsto \underbrace{\text{ech}_0 \circ \dots \circ \text{ech}_0(s)}_{2^{\lfloor \log_2(\text{profondeur}(s)) \rfloor}} \end{cases}$$

Cette expression n'est pas satisfaisante, nous avons donc tabulé ech_0^u et remarqué un pattern. Le symbole \mathbb{M} dans la suite désignera la reconstruction d'un arbre à partir des images des branches, décrites à l'intérieur du \mathbb{M} , données en indice.

Conjecture 14.1.

$$\text{ech}_0^u(s) = \mathbb{M}_{b \text{ branche} \in s} \left\{ \begin{array}{l} b + (-1)^{b_{n-\lfloor \log_2 n \rfloor}} + \sum_{k \in [0, \lfloor \log_2 n \rfloor]} b_{\lfloor \log_2 n \rfloor + k} (-1)^{b_k} 2^k \\ b \end{array} \right. \quad \begin{array}{l} \text{avec } b = \sum_{k \in [0, n]} b_k 2^k \\ \text{si } |b| \geq 2^{\lfloor \log_2(\text{profondeur}(s)) \rfloor} \\ \text{sinon} \end{array}$$

Un exemple s'impose pour clarifier le fonctionnement du pattern.

$$\text{ech}_0^u(101101) = \begin{array}{c} 10 | 1101 \\ \quad \quad \quad \textcolor{blue}{1} 10 \end{array} = 10 | 1011$$

ech^u trouve le plus grand suffixe de taille puissance de deux, décale ensuite le préfixe restant pour aligner les derniers bit du suffixe et du préfixe, ajoute un 1 devant le préfixe. L'image par ech^u s'obtient en concaténant le préfixe avec la version du suffixe où l'on aurait inversé les bits surplombant un 1.

Pour faire la preuve de cette égalité je pense qu'il est nécessaire de comprendre comment fonctionne l'application Π et non simplement de savoir quelle existe. Je n'ai malheureusement pas réussi à exhiber quelque chose de probant, ce qui laisse cette expression en conjecture.

Théorème 14.1. *Cette définition de ech_0^u est une définition d'une involution des \mathcal{ADS}_n dans les \mathcal{ADS}_n .*

Démonstration. La preuve est faite en Appendice H. □

Je n'ai malheureusement pas eu le temps de faire les programmes pour tester si cette involution est celle des $q, t - \text{Catalan}$. Cependant nous doutons qu'il s'agisse de la bonne involution.

4 Conclusion

Durant mon stage, le travail que j'ai mené n'a pas permis de classer l'ensemble des bijections du nouveau formalisme. Au contraire, nous nous sommes heurté à des objets avec un comportement beaucoup plus chaotique qu'espéré. Cela a rendu la tâche de classification très complexe mais nous avons tout de même réussi à démontrer la bijectivité d'un petit nombre de grammaires. Cependant le fait que ech_0 ne soit pas involutive a engendré la résolution de problèmes aux démonstrations élégantes qui lient de multiples domaines de l'informatique. On reste cependant sur notre faim car on ne réussit pas à montrer l'expression exacte de ech_0^u et je n'ai pas eu le temps de développer les codes pour tester si une involution était une bonne candidate pour les $q, t - \text{Catalan}$.

Cependant, le travail a montré qu'il y avait d'autres angles d'attaque sur le problème de recherche d'involution sur les \mathcal{ADS} . Effectivement, un système de tables de vérité permettrait de générer des involutions sur les branches ainsi qu'une exploration exhaustive des involutions sur les branches. Exploration que l'on pourrait interrompre sur certaines branches d'involutions si elles ne suivent plus les $q, t - \text{Catalan}$. On peut aussi se demander si un système d'automate écrivant à chaque transition, appelé transducteur, pourrait être un moyen compact de représenter ces involutions.

Ce stage m'a permis de découvrir la vie en laboratoire, ayant eu une expérience de recherche en distanciel dans le passé, l'esprit et l'ambiance qui régnait au LaBRI était motivante et créait une véritable émulation dans laquelle j'ai pu m'épanouir.

Références

- [1] A.M. Garsia and M. Haiman. A remarkable q, t -catalan sequence and q -lagrange inversion. *Journal of Algebraic Combinatorics*, 5, 07 1996.
- [2] J. Haglund. Conjectured statistics for the q, t -catalan numbers. *Advances in Mathematics*, 175 :319–334, 10 2000.
- [3] Yvan Le Borgne. *Variations combinatoires sur des classes d'objets comptées par la suite de Catalan*. PhD thesis, Bordeaux 1, 2004.
- [4] Olivier Martin, Andrew Odlyzko, and Stephen Wolfram. Algebraic properties of cellular automata. *Communications in Mathematical Physics*, 93, 06 1984.
- [5] Richard P Stanley. Catalan addendum, 2008.
- [6] Ian Stewart. Les fractales de pascal. *Pour la science*, (129), 1988.

5 Appendice

Le Cadre du stage

J'ai eu la chance de pouvoir faire mon stage au LaBRI en présentiel. J'avais un bureau que je partageais avec un autre stagiaire de l'ENS Paris-Saclay. Nous avions le même maître de stage et nos sujets bien que différents, ce rejoignaient justement sur la conjecture des $q, t - Catalan$. Nous nous sommes très bien entendus et cela à beaucoup contribué au bon déroulement du stage. Nous pouvions venir tous les jours de la semaine au laboratoire, sans horaire précis, cela permettait de pouvoir faire de bonnes sessions de travail sans avoir à tout le temps changer de lieu. Yvan Le Borgne, mon maître de stage passait en moyenne 3 fois par semaine, les lundis, mercredis et vendredis. Nous faisions alors des après-midis tableaux. Il y a eu de vrais échanges et débats durant ces après midi qui ont fait grandement avancer le stage. Durant les pauses le midi, nous mangions avec les autres chercheurs du laboratoire, laboratoire qui était plutôt vide du fait des restrictions sanitaires pour les chercheurs. Cependant, il y avait énormément de stagiaires de tous horizons, avec qui nous mangions, jouions et faisions du sport. Les chercheurs souhaitais nous présenter de multiples sujets de recherche, cela c'est fait par le biais des groupes de travail de l'équipe et de diverses conférences. Ils ont aussi instauré des sortes d'exposés "posters" sur les mur de l'atrium où 3 fois par semaine des chercheurs de différentes équipes venait présenter leurs travaux, dans un cadre très convivial nous mettant plus à l'aise pour poser des questions. Dans ce cadre, en fin de stage, nous avons présenté nos travaux sous forme d'exposés "posters" aux autres stagiaires. Le stage s'est terminé sur un exposé de fin de stage où j'ai présenté avec plaisir en 25 minutes mes recherches aux autres membres de l'équipe Combinatoire et Interaction.

A Démonstration de la bijectivité de ech_0 avec ech_0^{-1}

Lemme 14.1. $\forall t \in \mathcal{ADS}, \forall j \in \{-1, 1\}, \text{ech}_0^{-1}(\text{ech}_0(t, j), -j) = t$

Démonstration. On raisonne par induction :

$$\text{Rappel : } \forall t \in \mathcal{ADS}, \exists k, t1, t1', t2/t = k, t1' \oplus (k + 1), t2 \wedge t1 = t1' \oplus k + 1$$

Hypothèse d'induction: $\mathcal{P}(t) := \forall j \in \{-1, 1\}, \text{ech}_0^{-1}(\text{ech}_0(t, j), -j) = t$

Initialisation: Pour $|t| = 0$ c'est bon car $\text{ech}_0(\varepsilon, 0) = \text{ech}_0^{-1}(\varepsilon, 0) = \varepsilon$. Pour $|t| = 1$ on a $\exists k \in \mathbb{N}/t = k$ c'est bon car $\text{ech}_0(k, 0) = \text{ech}_0^{-1}(k, 0) = k$

Induction: On suppose $\forall t', |t'| < |t| \Rightarrow \mathcal{P}(t')$

$$\begin{aligned} \text{ech}_0^{-1}(\text{ech}_0(t, i), -i) &= \text{ech}_0^{-1}(\text{ech}_0((k, t1, t2), i), -i) \\ \text{ech}_0^{-1}(\text{ech}_0(t, i), -i) &= \text{ech}_0^{-1}((k + i, \text{ech}_0(t1, i), \text{ech}_0(t2, i)), -i) \\ \text{ech}_0^{-1}(\text{ech}_0(t, i), -i) &= ((k + i) - i, \text{ech}_0^{-1}(\text{ech}_0(t1, i), -i), \text{ech}_0^{-1}(\text{ech}_0(t2, i), -i)) \\ \text{ech}_0^{-1}(\text{ech}_0((k, t1, t2), i), -i) &= (k, \text{ech}_0^{-1}(\text{ech}_0(t1, i), -i), \text{ech}_0^{-1}(\text{ech}_0(t2, i), -i)) \end{aligned}$$

Par hypothèse d'induction sur $t1$ et $t2$

$$\text{ech}_0^{-1}(\text{ech}_0(t1, i), -i) = (k, t1, t2) = t$$

Conclusion: Par induction on en déduit $\forall t \in \mathcal{ADS}, \mathcal{P}(t)$ □

Théorème 14.2. *L'application ech_0 est une bijection et l'on a que ech_0^{-1} est sa bijection réciproque*

Démonstration.

$$\text{Rappel : } \forall t \in \mathcal{ADS}, \exists k, t1, t1', t2/t = k, t1' \oplus (k + 1), t2 \wedge t1 = t1' \oplus k + 1$$

On pose alors $t = 0, t1, t2$.

$$\begin{aligned} \text{ech}_0^{-1}(\text{ech}_0(t, 0), 0) &= \text{ech}_0^{-1}(\text{ech}_0((0, t1, t2), 0), 0) \\ \text{ech}_0^{-1}(\text{ech}_0(t, 0), 0) &= \text{ech}_0^{-1}((0, \text{ech}_0(t2, 1), \text{ech}_0(t1, -1)), 0) \\ \text{ech}_0^{-1}(\text{ech}_0(t, 0), 0) &= 0, \text{ech}_0^{-1}(\text{ech}_0(t1, -1), 1), \text{ech}_0^{-1}(\text{ech}_0(t2, 1), -1) \end{aligned}$$

Ce qui d'après le Lemme (14.1) nous donne

$$\text{ech}_0^{-1}(\text{ech}_0(t, 0), 0) = 0, t1, t2 = t$$

Conclusion: On a donc bien le fait que ech_0 est une bijection par l'existence de sa bijection réciproque ech_0^{-1} □

B Démonstration de $\forall G_0 \equiv_{pic} G_1, \forall s \in \mathcal{ADS}, \forall i \in \{0, 1\}, \Phi_{G_0, G_1}(\text{ech}_0 s) = \Phi_{\text{ech}_0^m(G_0, G_1)}(s)$

Définition 15. (L'application ϕ) Pour la démonstration nous aurons besoin de définir des relation et application sur les mots de Dyck marqués \mathcal{MDW} . On définit \mathcal{ADSS} comme $\{s \oplus k / \forall (s, k) \in \mathcal{ADS} \times \mathbb{N}\}$. Soit ϕ paramétré par deux grammaire $(G_0, G_1) \in \mathcal{GR}^2$ et définie de la manière suivante.

$$\phi_{G_0, G_1} : \begin{cases} \mathcal{ADSS} \times \mathcal{MDW} & \rightarrow \\ s, d & \mapsto \begin{cases} \mathcal{MDW} & \text{si } s = \varepsilon \\ d & \\ \varphi_{G_0, G_1}(s[-1], \phi_{G_0, G_1}(s[: -1], d)) & \text{sinon} \end{cases} \end{cases}$$

On peut maintenant définir l'équivalence "pas" que l'on note \equiv_{pas} comme l'égalité de $\forall (d_1, d_2) \in \mathcal{MDW}^2$ sans leurs marques.

Démonstration. Procédons par induction sur la structure inductive naturelle de \mathcal{ADSS} . Dans toute la démonstration, on a G_0, G_1 choisi de telle sorte que $G_0 \equiv_{pic} G_1$. On ne prendra plus la peine de le préciser.

Notation: On notera \dagger l'application qui à un mot w de \mathcal{MDW} et un entier k de \mathbb{N} associe w' de \mathcal{MDW} qui coïncide avec w sur les pas montants et descendants et dont ses marques sont celles de w auxquelles on additionne k .

Hypothèse d'induction:

$$\mathcal{P}(s) := \begin{cases} \text{a)} & \phi_{G_0, G_1}(\text{ech}_i(s), 0\bar{0}) \equiv_{pas} \phi_{\text{ech}_0^m(G_0, G_1)}(s, 0\bar{0}) \\ \text{b)} & \text{les marques } j\bar{j} \text{ sont toujours adjacentes dans } \phi_{\text{ech}_0^m(G_0, G_1)}(s, 0\bar{0}) \end{cases}$$

Initialisation: Pour $s = \varepsilon$

$$\begin{aligned} \phi_{G_0, G_1}(\text{ech}_i(\varepsilon), 0\bar{0}) &= \phi_{G_0, G_1}(\varepsilon, 0\bar{0}) \\ \iff \phi_{G_0, G_1}(\text{ech}_i(\varepsilon), 0\bar{0}) &= 0\bar{0} \\ \iff \phi_{G_0, G_1}(\text{ech}_i(\varepsilon), 0\bar{0}) &= \phi_{\text{ech}_i^m(G_0, G_1)}(\varepsilon, 0\bar{0}) \end{aligned}$$

Ces égalité nous permettent d'en déduire **a** et **b** pour s .

Induction: On suppose $\forall t/|t| < |s| \implies \mathcal{P}(t)$
On commence pour $i = 0$

$$\begin{aligned} R &:= \phi_{G_0, G_1}(\text{ech}_0(s), 0\bar{0}) = \phi_{G_0, G_1}(\text{ech}_0((0, t_1, t_2), 0), 0\bar{0}) \\ \iff \phi_{G_0, G_1}(\text{ech}_0(s), 0\bar{0}) &= \phi_{G_0, G_1}((0, \text{ech}_0(t_2, 1), \text{ech}_0(t_1, -1)), 0\bar{0}) \\ \iff \phi_{G_0, G_1}(\text{ech}_0(s), 0\bar{0}) &= \phi_{G_0, G_1}(\text{ech}_0(t_1, -1), \phi_{G_0, G_1}(\text{ech}_0(t_2, 1), \phi_{G_0, G_1}(0, 0\bar{0}))) \end{aligned}$$

Comme les règles sont \equiv_{pic} elles sont les deux des modes pics, on a donc $\phi_{G_0, G_1}(0, 0\bar{0})$ qui respecte **b**. Cela nous dit que l'insertion de tout ce qu'il faut en $1\bar{1}$ laisse invariant $0\bar{0}$. Grâce à cela on peut calculer séparément $\phi_{G_0, G_1}(\text{ech}_0(t_2, 1), 1\bar{1})$ et $\phi_{G_0, G_1}(\text{ech}_0(t_1, -1), 0\bar{0})$ et ensuite substituer dans $\phi_{G_0, G_1}(0, 0\bar{0})$

$$\begin{aligned} \iff R &= \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{G_0, G_1}(\text{ech}_0(t_2, 1), 1\bar{1})][0\bar{0} \rightarrow \phi_{G_0, G_1}(\text{ech}_0(t_1, -1), 0\bar{0})] \\ \iff R &= \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{G_0, G_1}(\text{ech}_0(t_2, 1), 1\bar{1})][0\bar{0} \rightarrow \phi_{G_1, G_0}(\text{ech}_1(t_1 \ominus 1, 0), 0\bar{0})] \end{aligned}$$

Ici on peut maintenant appliquer l'hypothèse d'induction sur le cas $0\bar{0}$

$$\implies R \equiv_{pas} \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{G_0, G_1}(\text{ech}_0(t_2, 1), 1\bar{1})][0\bar{0} \rightarrow \phi_{\text{ech}_1^m(G_1, G_0)}(t_1 \ominus 1, 0\bar{0})]$$

Le but est de se ramener à des ech_0^m , pour cela on va inverser et modifier les autres valeur pour que cela ne change pas le résultat

$$\implies R \equiv_{pas} \phi_{\text{ech}_0^m(G_0, G_1)}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{\text{ech}_0^m(G_0, G_1)}(t_1, 1\bar{1})][0\bar{0} \rightarrow \phi_{G_0, G_1}(\text{ech}_0(t_2, 0), 0\bar{0})]$$

Ici on peut maintenant appliquer l'hypothèse d'induction sur le cas

$$\implies R \equiv_{pas} \phi_{\text{ech}_0^m(G_0, G_1)}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{\text{ech}_0^m(G_0, G_1)}(t_1, 1\bar{1})][0\bar{0} \rightarrow \phi_{\text{ech}_0^m(G_0, G_1)}(t_2, 0\bar{0})]$$

On peut donc maintenant recombinaer les compositions car on utilise toujours la même grammaire

$$\begin{aligned} \implies R &\equiv_{pas} \phi_{\text{ech}_0^m(G_0, G_1)}((0, t_1), 0\bar{0})[0\bar{0} \rightarrow \phi_{G_0, G_1}(\text{ech}_0(t_2, 0), 0\bar{0})] \\ \implies R &\equiv_{pas} \phi_{\text{ech}_0^m(G_0, G_1)}((0, t_1, t_2), 0\bar{0}) \end{aligned}$$

On a donc bien la propriété **a**). Pour la propriété **b**). Il suffit de remarquer que la dernière substitution en $0\bar{0}$ efface alors toutes les marques présentes dans $\phi_{\text{ech}_0^m(G_0, G_1)}((0, t_1), 0\bar{0})$ sauf le $0\bar{0}$ qui sera alors remplacé. On a on bien par l'hypothèse d'induction que $\phi_{G_0, G_1}(\text{ech}_0(t_2, 0), 0\bar{0})$ a ses marques qui respecte **b**) donc les nouvelles marques de $\phi_{\text{ech}_0^m(G_0, G_1)}((0, t_1, t_2), 0\bar{0})$ respecteront elles aussi **b**)

Pour $i = 1$

$$\begin{aligned}
R &:= \phi_{G_0, G_1}(\text{ech}_1(s), 0\bar{0}) = \phi_{G_0, G_1}(\text{ech}_1((0, t_1, t_2), 0), 0\bar{0}) \\
&\iff \phi_{G_0, G_1}(\text{ech}_0(s), 0\bar{0}) = \phi_{G_0, G_1}((0, \text{ech}_1(t_1, 0), \text{ech}_1(t_2, 0)), 0\bar{0}) \\
&\iff \phi_{G_0, G_1}(\text{ech}_0(s), 0\bar{0}) = \phi_{G_0, G_1}(\text{ech}_1(t_2, 0), \phi_{G_0, G_1}(\text{ech}_1(t_1, 0), \phi_{G_0, G_1}(0, 0\bar{0}))) \\
&\iff \phi_{G_0, G_1}(\text{ech}_0(s), 0\bar{0}) = \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{G_0, G_1}(\text{ech}_1(t_1, 0), 1\bar{1})][0\bar{0} \rightarrow \phi_{G_0, G_1}(\text{ech}_1(t_2, 0), 0\bar{0})]
\end{aligned}$$

Ici on peut maintenant appliquer l'hypothèse d'induction sur le cas $0\bar{0}$

$$\begin{aligned}
&\implies R \equiv_{pas} \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{G_0, G_1}(\text{ech}_1(t_1, 0), 1\bar{1})][0\bar{0} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_2, 0\bar{0})] \\
&\implies R \equiv_{pas} \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{G_1, G_0}(\text{ech}_0(t_1 \ominus 1, 0), 0\bar{0})\dagger 1][0\bar{0} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_2, 0\bar{0})]
\end{aligned}$$

Ici on peut maintenant appliquer l'hypothèse d'induction sur le cas $1\bar{1}$

$$\begin{aligned}
&\implies R \equiv_{pas} \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{\text{ech}_0^m(G_1, G_0)}(t_1 \ominus 1, 0\bar{0})\dagger 1][0\bar{0} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_2, 0\bar{0})] \\
&\implies R \equiv_{pas} \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_1, 1\bar{1})][0\bar{0} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_2, 0\bar{0})] \\
&\implies R \equiv_{pas} \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{\text{ech}_0^m(G_1, G_0)}(t_1 \ominus 1, 0\bar{0})\dagger 1][0\bar{0} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_2, 0\bar{0})] \\
&\implies R \equiv_{pas} \phi_{G_0, G_1}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_1, 1\bar{1})][0\bar{0} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_2, 0\bar{0})] \\
&\implies R \equiv_{pas} \phi_{\text{ech}_1^m(G_0, G_1)}(0, 0\bar{0})[1\bar{1} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_1, 1\bar{1})][0\bar{0} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_2, 0\bar{0})] \\
&\implies R \equiv_{pas} \phi_{G_0, G_1}((0, t_1), 0\bar{0})[0\bar{0} \rightarrow \phi_{\text{ech}_1^m(G_0, G_1)}(t_2, 0\bar{0})] \\
&\implies R \equiv_{pas} \phi_{G_0, G_1}((0, t_1, t_2), 0\bar{0})
\end{aligned}$$

D'où on en déduit

$$\phi_{G_0, G_1}(\text{ech}_0(s), 0\bar{0}) \equiv_{pas} \phi_{G_0, G_1}((0, t_1, t_2), 0\bar{0})$$

On a donc bien la propriété **a)** et pour les mêmes raisons que sur le dessus nous avons aussi la propriété **b)**.

Conclusion: Par induction on peut donc déduire que l'on a $\forall s \in \mathcal{ADS}, \mathcal{P}(s)$. Or on a que la relation binaire \equiv_{pas} sur \mathcal{MDW} est équivalente à $=$ sur \mathcal{DW} via l'application qui supprime les marques. D'où on a bien que

$$\forall G_1 \equiv_{pic} G_0 \in \mathcal{GR}, \forall s \in \mathcal{ADS}, \forall i \in \{0, 1\}, \Phi_{G_0, G_1} \circ \text{ech}_i(s) = \Phi_{\text{ech}_i^m(G_0, G_1)}(s)$$

□

C Illustration des transmissions de bijectivité pour la grammaire canonique

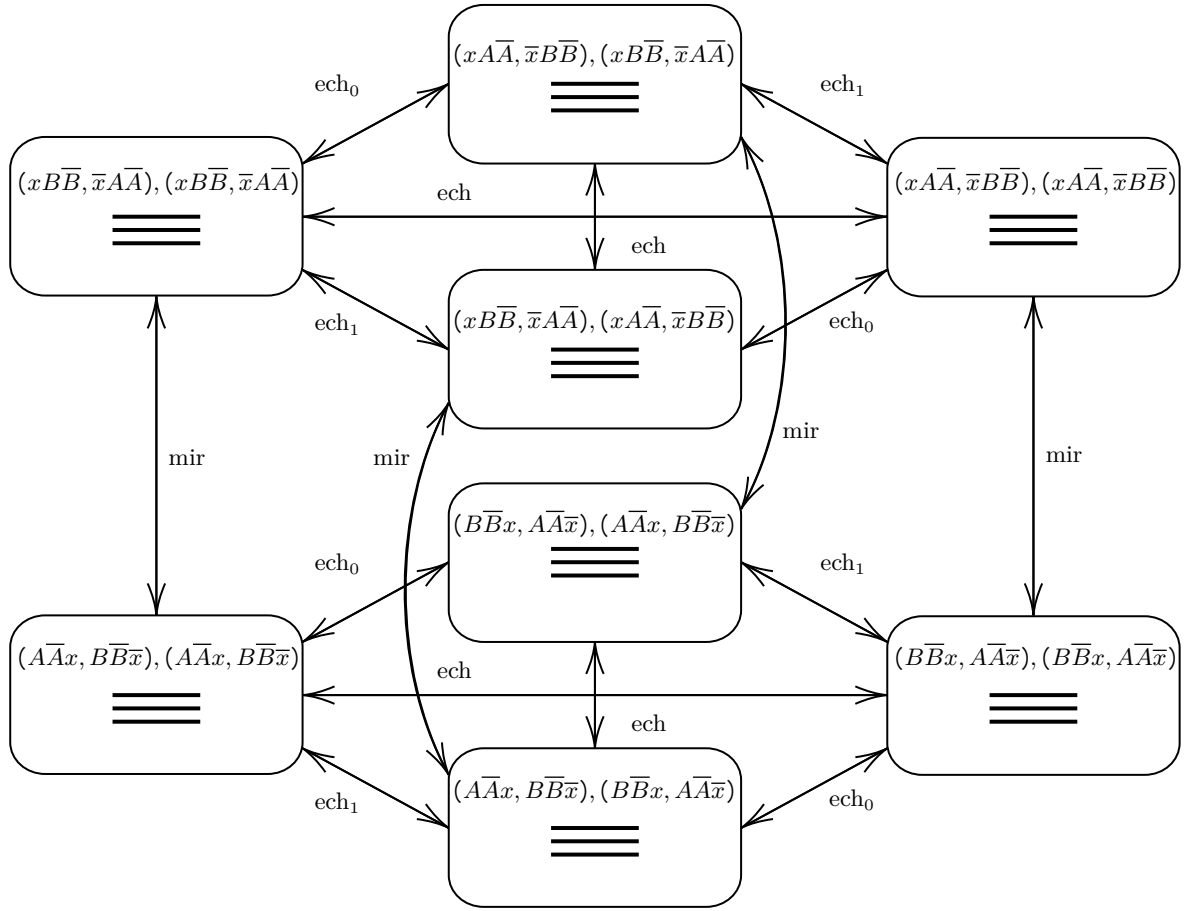
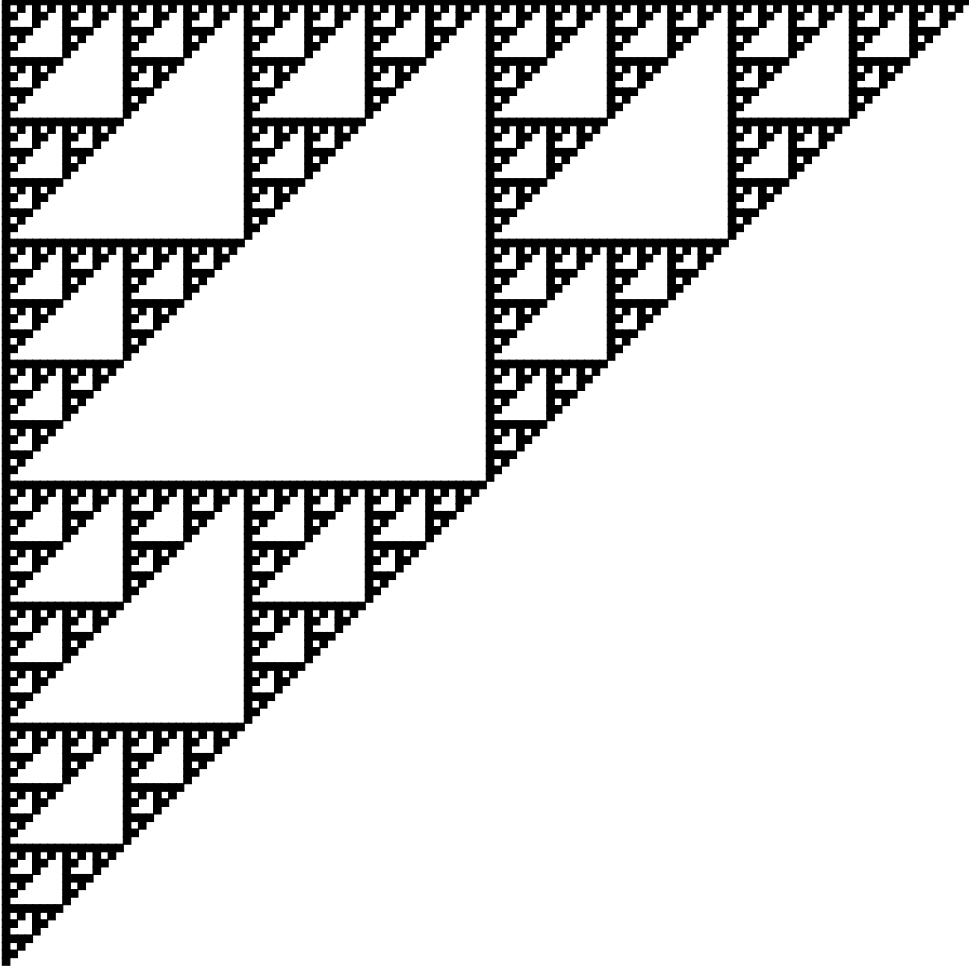


FIGURE 19 – Spécification du diagramme des transmissions de bijectivité en partant de la règle canonique

D Triangle de Sierpinski



E Démonstration de l'égalité entre le triangle de Pascal mod 2 et la trace des puissances de ech_0 sur $1\dots 1$

Démonstration. On défini P comme ceci $\forall n, k \in \mathbb{N}, P(n, k) := \binom{n+k}{k} \mod 2$. Le but est de montrer que $\forall n, k \in \mathbb{N}, P(n, k) \equiv \text{ech}_0^n(1 \underbrace{1\dots 1}_{k-1})_k$.

- **Initialisation** On a les mêmes conditions aux bornes (par convention on les pose identiques où elles ne sont pas censées être définies), i.e.

$$\forall n, k \in \mathbb{N} \cup \{-1\}, (k = -1 \vee n = -1) \implies P(n, k) = \text{ech}_0^n(1 \underbrace{1\dots 1}_{k-1})_k = 0$$

$$P(0, 0) = \text{ech}_0^0(1)_0 = 1$$

- **Hérédité** On a aussi la même relation d'induction. Pour Pascal on a que

$$\binom{j+1}{i+1} = \binom{j}{i} + \binom{j}{i+1} \implies \binom{j+1}{i+1} \mod 2 = \left(\binom{j}{i} + \binom{j}{i+1} \right) \mod 2$$

Ce qui nous donne en P les relations suivantes.

$$\begin{aligned} P(j-i, i+1) \mod 2 &= (P(j-i, i) + P(j-(i+1), i+1)) \mod 2 \\ \implies P(j^i, i^i+1) \mod 2 &= (P(j^i, i^i) + P(j^i-1, i^i+1)) \mod 2 \end{aligned}$$

Qui est la même relation d'induction que celle pour la trace de ech_0 :

$$\text{ech}_0^{j'}(1 \underbrace{1 \dots 1}_{i'})_{i'+1} \mod 2 = \left(\text{ech}_0^{j'}(1 \underbrace{1 \dots 1}_{i'})_{i'} + \text{ech}_0^{j'-1}(1 \underbrace{1 \dots 1}_{i'})_{i'+1} \right) \mod 2$$

Conclusion: On a la même initialisation ainsi que la même induction. On génère donc le même tableau. \square

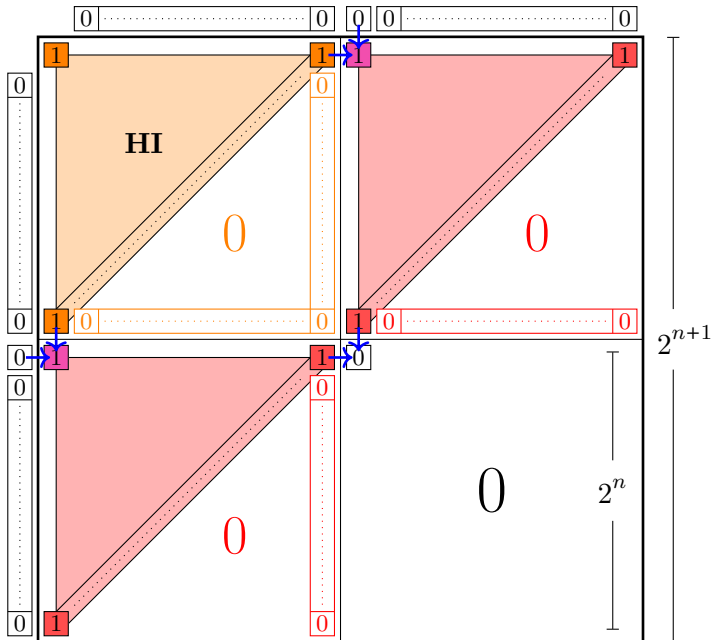
F Démonstration de l'égalité entre le triangle de Sierpinski et le triangle de Pascal $\mod 2$

Démonstration. Par induction :

Hypothèse d'induction: On a bien égalité entre les triangle de Sierpinski de côté 2^n et le triangle de Pascal $\mod 2$ de même taille. De plus on a la caractéristique d'avoir la diagonale de $(0, 2^n - 1)$ à $(2^n - 1, 0)$ que de 1 et aucune case à 1 sous cette diagonale dans le carré de taille $2^n \times 2^n$.

Initialisation: Pour $n = 0$ Les deux sont une unique case à 1. La configuration vérifie bien l'Hypothèse d'Induction.

Induction: Supposons que pour n on ait l'hypothèse d'induction montrons que l'on à $n + 1$ Quand on passe à la taille supérieure, i.e. de 2^n à 2^{n+1} dans Sierpinski on va simplement dupliquer le premier carré à droite, en dessous et ne rien mettre en dessous à droite. On montre donc Figure 20 que l'induction de Pascal est équivalente.



On commence avec, en orange, l'hypothèse d'induction. Ensuite, en magenta, les deux 1 que l'on peut facilement induire avec l'**HI**. On se rend alors compte que les conditions initiales que l'on retrouve pour l'**HI** se retrouve pour le carré en haut à droite et celui en bas à gauche. Comme on utilise la même règle d'induction que pour le carré en haut à gauche, on peut dupliquer le motif, en Rouge. On peut alors induire le zéro, en blanc, du dernier carré. De la les conditions initiales impliquent que l'on ne va écrire que des zéros. L'objet obtenu vérifie bien l'**HI**.

FIGURE 20 – Illustration que l'induction de Pascal engendre l'induction de Sierpinski

Conclusion: Par induction on peut donc en déduire que le triangle de Pascal $\mod 2$ est bien égal au triangle de Sierpinski. \square

G Démonstration de $\forall b \in \{0, 1\}^{\mathbb{N}}, \text{ord}_{1b}(\text{ech}_0) = 2^{\lfloor \log_2 |b| \rfloor + 1}$

Démonstration. But : montrer que l'ordre de ech_0 sur une entrée est la première puissance de deux strictement plus grande que ça taille.

Nous avons déjà vu que c'est le cas pour les entrées du type 1...1. Montrons que cela reste vrai pour les autres entrées. Pour cela nous utiliserons le formalisme des automates cellulaires 1d de Wolfram.

Nous travaillons avec la règle 60 de wolfram que je rappelle ici :

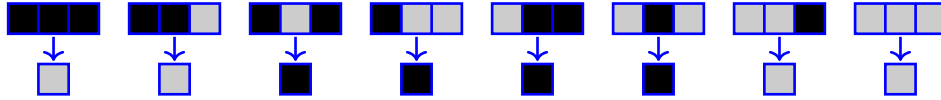


FIGURE 21 – Règle 60 de Wolfram

Cette règle a l'avantage important d'être additive, i.e. pour calculer son déroulé, on peut simplement calculer le déroulé pour chaque cellule activée de l'entrée et ensuite sommer. Voici un petit exemple :

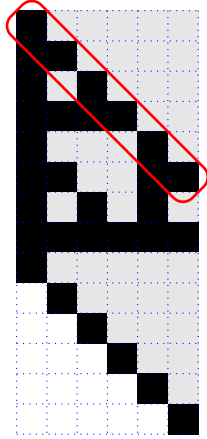


FIGURE 22 – Une unique cellule activée

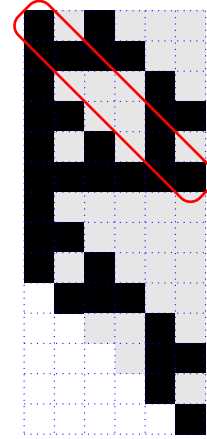


FIGURE 23 – Deux cellules activées

La difficulté cependant que nous avons avec ce modèle d'automate cellulaire est que l'entrée correspondant à l'entrée de ech_0 se situe sur la diagonale entourée en rouge Figure 22. Or ce n'est en rien l'entrée de l'automate. Nous avons donc définie la notation Π pour la fonction qui a une entrée de ech_0 associe le bon paramétrage de l'entrée de l'automate afin que celui-ci génère sur la diagonale l'entrée de ech_0 . Cependant nous n'avons pas montré son existence.

Théorème 15.1. *La fonction Π est bien définie pour tout mot binaire*

Démonstration. On va faire remonter l'information dans l'automate afin de retrouver l'entrée. Pour cela on va inverser la règle 60 pour obtenir un nouveau automate cellulaire qui aurait pour entrée l'entrée de ech_0 et qui en se déroulant écrira l'entrée de l'automate initial. La règle est décrite Figure 24.

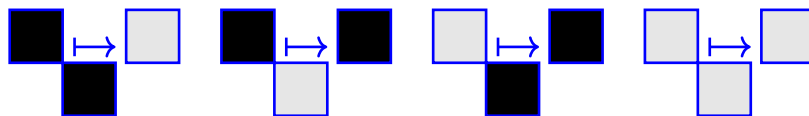


FIGURE 24 – La règle π pour faire remonter l'entrée

Cette règle est compatible avec la règle 60 d'origine. Elle permet de générer une par une les diagonales parallèles à celle de l'entrée de ech_0 vers l'entrée de l'automate de Wolfram. On peut donc à toute entrée de ech_0 associer l'évolution d'un automate utilisant la règle π et en extraire le paramétrage de l'entrée de l'automate de Wolfram recherché. On a l'existence pour chaque mot binaire b en entrée de ech_0 d'un paramétrage de l'entrée de l'automate tel que sur la diagonale il y ait b . D'où l'existence de Π . Un exemple illustratif de Π est donné Figure 25

□

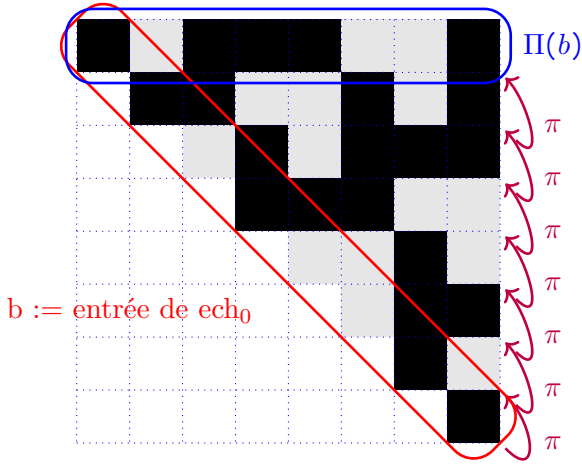


FIGURE 25 – Évolution de la règle π

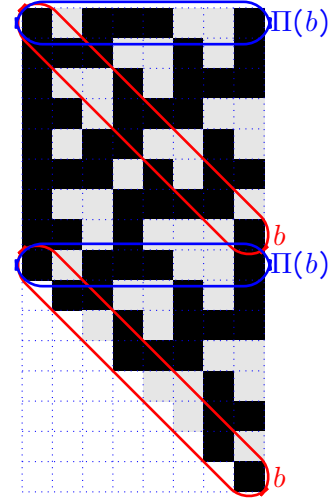


FIGURE 26 – Illustration du passage de la réflexion sur les diagonales à sur les lignes

Maintenant que l'on a l'existence de Π . On sait que l'on peut représenter n'importe quelle entrée de ech_0 et donc utiliser ce modèle pour calculer la trace des puissances de ech_0 sur n'importe quelle entrée. Ce changement est très important, car pour le moment on travaillait sur des diagonales dans l'automate ce qui n'est pas naturelle. Or Π nous donne la possibilité de travailler sur les lignes, qui sont bien plus naturelles pour les automates cellulaires. On cherche donc maintenant l'indice de la première ligne où le motif de l'entrée de l'automate se répète. On illustre ceci Figure 26

Une borne supérieure Commençons par montrer que notre conjecture est une borne supérieure. On commence par remarquer que si l'on stoppe les calculs à la couche de profondeur $2^{\lfloor \log_2(n) \rfloor + 1}$ pour l'entrée $1 \underbrace{0 \dots 0}_{n-1}$, on retrouve une copie de l'entrée d'après le Théorème 13.5. Donc on retrouve une unique cellule activée qui est à la même abscisse.

On va maintenant utilisé l'argument d'additivité, on calcule la propagation séparément pour chaque cellule activée de l'entrée de l'automate. Ensuite on somme modulo 2 pour retrouver la propagation avec l'entrée complète. Cependant, tout les calculs pour pour chaque cellule activée de l'entrée sont les mêmes. Ils sont équivalent à copier et décaler la propagation pour l'entrée $1 \underbrace{0 \dots 0}_{n-1}$ de l'automate. Or le décalage que l'on applique sur l'entrée s'applique à toute les couches

et donc à la couche de profondeur $2^{\lfloor \log_2(n) \rfloor + 1}$. Ce qui a pour conséquence d'obtenir une copie de l'entrée sur la couche de profondeur $2^{\lfloor \log_2(n) \rfloor + 1}$. Pour mieux comprendre voici la Figure 27. Or la copie de la ligne d'entrée de l'automate à pour conséquence directe le fait que l'entrée de ech_0 soit elle aussi recopiée, $2^{\lfloor \log_2(n) \rfloor + 1}$ lignes plus basses. On a donc bien l'inégalité suivante :

$$\forall b \in \{0, 1\}^{\mathbb{N}}, \text{ord}_{1b}(\text{ech}_0) \leq 2^{\lfloor \log_2 |b| \rfloor + 1}$$

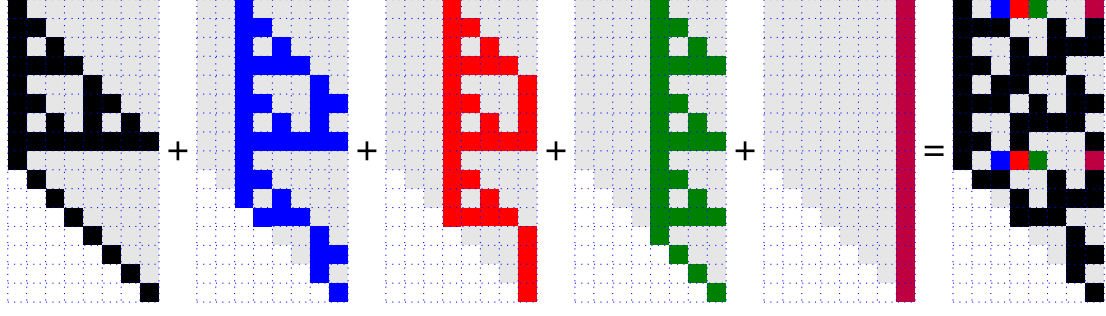
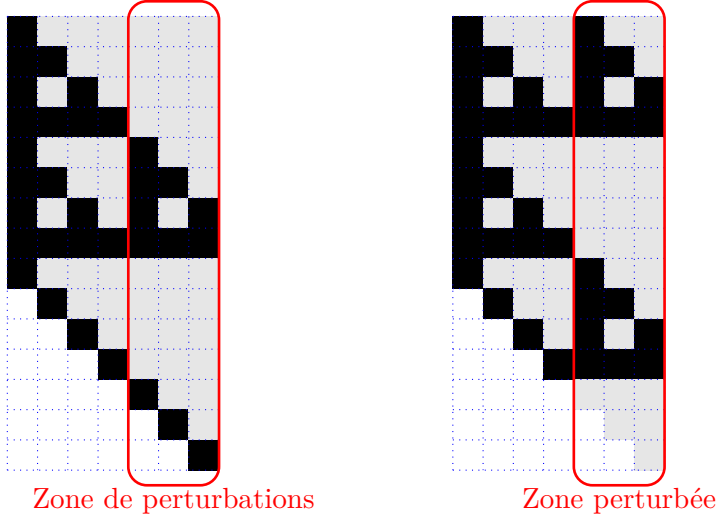


FIGURE 27 – Illustration de la somme des développements indépendants des cellules de l'entrée donne le développement de l'entrée.

Une borne inférieur Montrons maintenant que cette borne est stricte. Pour cela on va analyser rapidement comment se propagent les perturbations du premier 1 imposé à l'avant de l'entrée dans le reste du développement et se rendre compte qu'il impose son ordre.



Prenons une entrée $1b$ avec $b \in \bigcup_{n \in \mathbb{N}} \{0, 1\}^{\llbracket 1, n \rrbracket}$. Comme on le voit, le premier 1 imposé induit une perturbation sur la zone de propagation du reste des cellules actives (b). Or cette perturbation est très bien définie et est une restriction à un cylindre infini du triangle de pascal, la relation du triangle de pascal induit donc que cette perturbation est périodique de période $2^{\lfloor \log_2 |b| \rfloor + 1}$.

On a donc le résultat suivant, en posant b' comme $b = 0...01b'$.

$$\begin{aligned} \text{ord}_{1b}(\text{ech}_0) &= \text{PGCD} \left(2^{\lfloor \log_2 |b| \rfloor + 1}, \text{ord}_{1b'}(\text{ech}_0) \right) \\ \text{ord}_{1b}(\text{ech}_0) &= \text{PGCD} \left(2^{\lfloor \log_2 |b| \rfloor + 1}, \text{un diviseur de } 2^{\lfloor \log_2 |b'| \rfloor + 1} \right) \\ \text{ord}_{1b}(\text{ech}_0) &= 2^{\lfloor \log_2 |b| \rfloor + 1} \end{aligned}$$

On en déduit donc le théorème, i.e. $\forall b \in \{0, 1\}^{\mathbb{N}}, \text{ord}_{1b}(\text{ech}_0) = 2^{\lfloor \log_2 |b| \rfloor + 1}$. \square

H Démonstration du caractère involutif de $\mathcal{ADS}_n \rightarrow \mathcal{ADS}_n$ de ech_0^u pour tout $n \in \mathbb{N}$

On commence par l'involutivité : Pour cela on la montre sur les branches. On peut découper la branche b de taille n en deux parties pour ech_0^u : Un suffixe de taille $2^{\lfloor \log_2(n) \rfloor}$ et un préfixe. On remarque que la partie préfixe joue le rôle de commande pour les flips ayant lieu sur le suffixe. Or le préfixe est invariant par l'application donc appliquer 2 fois ech_0^u revient à faire 2 fois les mêmes flips et donc à ne rien faire. D'où l'involutivité sur les branches. Si maintenant on considère une \mathcal{ADS} plus complexe avec plusieurs branches, chaque branche indépendamment

des autres se vera appliquer 2 fois ech_0^u qui d'après le point précédent est involutive, soit laisse constant deux fois la branche. On a de nouveaux une involutivité. Cette fois si dans le cadre général.

L'application ech_0^u laisse invariante la taille de l'entrée : Il ne reste plus qu'à montrer que l'on ne change pas le nombre de sommets de l'arbre lorsque l'on applique ech_0^u .

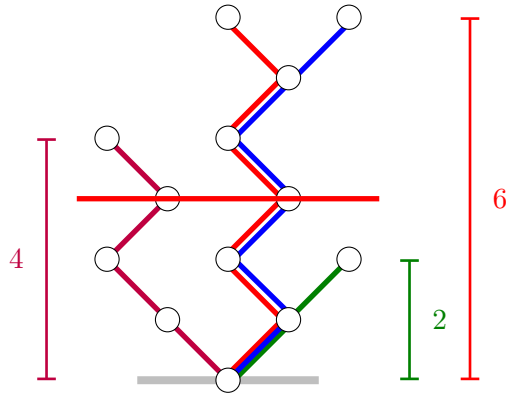


FIGURE 28 – Illustration de l'application de ech_0^u sur une \mathcal{ADS} représentative

On a ici 4 branches intéressantes pour ech_0^u , on pose $n = \lfloor \log_2 \text{profondeur}(s) \rfloor$:

Démonstration. • La branche **verte**, sa taille est inférieure strictement à 2^n donc il reste constant.

- Entre la branche **rouge** et la branche **violet** : Il n'y a pas de risque de collisions. Effectivement, si la branche est assez longue pour être impactée par ech_0^u , alors on a la propriété suivante : $b = b_1 b_2$ avec $|b_2| = 2^n$. Seuls les $|b_1| + 1$ derniers bits seront modifiés ce qui laisse constant les $|b| - |b_1| - 1 = |b_2| + |b_1| - |b - 1| - 1 = 2^n - 1$ premiers bits. Notre application est donc constante pour tout les débuts de branche jusqu'à la profondeur 2^n exclue. Il n'y a donc pas de risque que les bases des branches fusionnent ce qui entraînerait une diminution de la taille.

- Entre la branche **rouge** et la branche **bleu** : Pour le début de la branche jusqu'à la profondeur 2^n il n'y a pas de problème car ech_0^u est constante, après cependant il y a des rotations. Cependant, ces rotations sont les mêmes sur la partie commune des deux branches donc l'image de la sous branche commune des deux est toujours une sous branche commune aux deux images. Il n'y a donc pas de création de sommet et donc pas d'augmentation de la taille.

On peut donc enfin garantir que l'expression conjecturée de ech_0^u est involutive de \mathcal{ADS}_n dans \mathcal{ADS}_n pour tout $n \in \mathbb{N}$. \square