Classes de Valiant dans le cas non-associatif

Florent Capelli

17 novembre 2011

Résumé

Nous présentons ici l'article *Relationless completeness and separations* de P. Hrubes, A. Wigderson et A. Yehudayoff[5]. Des analogues aux classes de Valiant VP et VNP y sont définies dans un cas où la loi $\times$ n'est pas associative. Les auteurs exhibent alors une version du permanent dont ils prouvent la complétude pour cette version de VNP puis parviennent à séparer VP de VNP dans ce cas très particulier.

Introduction

La plupart des problèmes ouverts en complexité aujourd'hui concernent la séparation de deux classes de complexité présumées différentes. Le problème le plus emblématique est bien sûr la célèbre question de savoir si P est différent de NP. Face à la difficulté de la question, on a vu naître des questions proches fondées sur des modèles de calcul un peu différents. C'est le cas du modèle BSS[1] (et la question $\mathsf{P}_{\mathbb{C}} \neq \mathsf{NP}_{\mathbb{C}}$) qui généralise la question à une structure mathématique quelconque ou du modèle de Valiant[6] (et la question $\mathsf{VP} \neq \mathsf{VNP}$) qui reformule la question dans un cadre différent, plus proche des classes de complexité de comptage. Dans les deux cas, on espère qu'abandonner certaines propriétés de simplification sur les booléens (comme $x^2 = x$) mènera à des preuves plus simples ; on limiterait l'effet de phénomènes étranges et contre-intuitifs. La difficulté de trouver des bornes inférieures pour VNP a mené les auteurs de l'article *Relationless completeness and separations* de P. Hrubes, A. Wigderson et A. Yehudayoff[5] à réduire encore les propriétés mathématiques des structures considérées. La question de la non-commutativité ayant déjà été bien étudiée[4], les auteurs s'intéressent ici essentiellement au cas non-associatif (qu'il soit commutatif ou non). Cette simplification leur permet de séparer VP de VNP dans ce cas particulier.

Tout d'abord, les auteurs définissent un permanent non-associatif possédant des propriétés intéressantes qui en font un bon candidat pour être un problème complet. Ils passent pour cela par la définition d'arbres binaires universels, qui contiennent en mineur tous les arbres binaires d'une certaine taille. Si on représente un monôme non-associatif par un arbre binaire décrivant la façon de faire les produits, ces arbres universels peuvent simuler chaque monôme d'une taille donnée, propriété essentielle pour pouvoir projeter un polynôme quelconque sur ce permanent. La complétude de ce permanent en caractéristique différente de 2 est ensuite montrée en adaptant la preuve de G. Malod et N. Portier[3] pour montrer qu'il n'y a pas de différence de puissance entre les formules et les circuits dans VNP puis il s'inspire de la preuve classique de complétude du permanent qu'on pourra trouver dans le livre de Bürgisser [2]. Enfin, les auteurs exhibent un polynôme de VNP et donne une borne inférieure exponentielle sur la taille des circuits non-associatifs le calculant, séparant ainsi VP de VNP dans ce cas, et ce, en quelque caractéristique que ce soit.

Nous suivrons ici le même plan que l'article en insistant essentiellement sur les différences entre cette preuve et la preuve dans le cas commutatif et associatif. La première partie présente les différentes définitions spécifiques à cet article dont nous avons besoin pour la suite (arbres universels, permanent et déterminant dans le cas non-associatif). Dans la deuxième partie, nous expliquons les idées de la preuve de complétude du permanent non-associatif en essayant de les relier à la preuve originale. Finalement, dans une troisième partie, nous expliquons comment les auteurs parviennent à séparer les classes VP et VNP dans le cas non-associatif.

1 Définitions et notations

Classes de Valiante étendues On considère dans la suite, sauf mention explicite, que la loi $\times$ n'est pas associative. Les lettres A, $\bar{A}$, C et $\bar{C}$ en indice des noms de classes indiquent qu'on considère la classe avec ou sans l'associativité (A) ou la commutativité (C).

Exemple 1.1 $\mathsf{VP}_{\bar{A},\bar{C}}$ *est la classe des polynômes* (f_n) *qui ont un degré polynomialement borné et un circuit de taille polynomialement borné où les portes* $\times$ *ne sont ni associatives ni commutatives.*

Arbres binaires Pour un monôme commutatif et associatif donné, il existe un nombre exponentiel de monôme non-commutatif et non-associatif correspondant selon l'ordre dans lequel on effectue les opérations. On utilise des arbres binaires pour décrire l'ordre des opérations :

$$T \ ::= \ v \mid (T_1, T_2)$$

La taille $|T|$ d'un arbre binaire est son nombre de feuille. Si T est de taille n et que $(P_1, \ldots, P_n)$ sont des polynômes, alors on note : $\prod^T (P_1 \ldots P_n)$ le produit des polynômes selon T. Plus formellement, $\prod^v P = P$ et $\prod^{(T_1,T_2)}(P_1, \ldots, P_n) = \prod^{T_1}(P_1, \ldots, P_{n_1}) \times \prod^{T_2}(P_{n_1+1}, \ldots, P_n)$ où $|T_1| = n_1$. On décrit ainsi bien un monôme donné sans ambiguité sur la commutativité ou l'associativité.

On définit une opération sur les arbres qui à partir d'un arbre et d'un sous-ensemble de ses feuilles construit un nouvel arbre binaire. Soit T un arbre,

$$\kappa(T; \{v\}) = v$$
$$\kappa((T_1, T_2); V) = \begin{cases} \kappa(T_2; V_2) & \text{si } V_1 = \emptyset \\ \kappa(T_1; V_1) & \text{si } V_2 = \emptyset \\ (\kappa(T_1; V_1), \kappa(T_2; V_2)) & \text{sinon.} \end{cases}$$

où V_1 (respectivement V_2) est l'ensemble des feuilles de T_1 (resp. T_2) qui sont dans V. En fait, cela revient juste à enlever toutes les feuilles qui ne sont pas dans V et de ramener ça à un arbre, comme montré sur la figure 1. Si T est vu comme un monôme, $\kappa(T; V)$ peut-être vu comme un sous-monôme de T.

Arbres universels La notion d'arbre universel est très importante dans ce papier. En effet, un arbre $\mathcal{T}$ est t-universel si pour tout arbre T de taille t, il existe un sous-ensemble V des feuilles de $\mathcal{T}$ tel que $\kappa(\mathcal{T}; V) = T$. Un arbre t-universel est donc capable de simuler tous les monômes de taille t d'un certain ensemble de variable. On utilisera ce type d'arbre pour définir un permanent dans le cas non-associatif car si on veut obtenir des résultats de complétude, il doit être capable de simuler n'importe quel monôme pour qu'on soit en mesure de projeter nos polynômes dessus. Le théorème suivant rend effective cette notion en exhibant des arbres t-universels de taille $O(t^4)$.

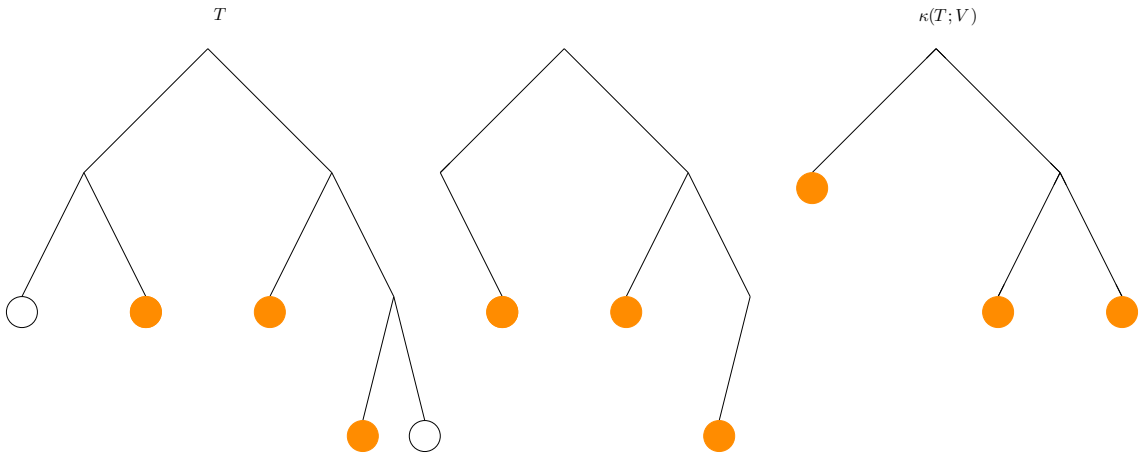

FIG. 1 – La fonction κ

Theorem 1.1 *Pour tout $t \geq 1$, il existe un arbre $\mathcal{T}(t)$ t-universel de taille $O(t^4)$ constructible en temps polynomial en t. De plus, si T est un arbre de taille t, on peut trouver V tel que $T = \kappa(\mathcal{T}; V)$ en temps polynomial en t.*

On définit alors le permanent et le déterminant en fonction d'un arbre :

$$PERM^T(M) = \sum_\sigma \prod^T (M_{1,\sigma(1)}, \ldots, M_{m,\sigma(m)})$$
$$DET^T(M) = \sum_\sigma (-1)^{sgn(\sigma)} \prod^T (M_{1,\sigma(1)}, \ldots, M_{m,\sigma(m)})$$

Finalement, on définit un permanent *intéressant* c'est-à-dire candidat aux résultats de complétude : pour un entier n, soit $\mathcal{T}_n$ l'arbre n-universel obtenu dans le théorème précédent. Cet arbre a m feuille. On définit, pour tout M de dimension $m \times m$:

$$PERM_n(M) = PERM^{\mathcal{T}_n}(M)$$
$$DET_n(M) = DET^{\mathcal{T}_n}(M)$$

Le gros avantage de cette définition est que $PERM_n$ peut simuler n'impote quel autre $PERM^T$ pour $|T| \leq n$. Les preuves de complétudes s'attacheront donc la plupart du temps à prouver qu'il existe un arbre T tel que $PERM^T$ a la propriété voulue, et on peut immédiatemment en déduire que $PERM_{|T|}$ la possède aussi !

Ce permanent et ce déterminant peuvent simuler n'importe quelle formule arithmétique. Nous allons présenter ce résultat dans la partie suivante et montrer comment les auteurs parviennent à un résultat de complétude pour le permanent défini ci-dessus dans le cas non-associatif.

2 Complétude

Pour montrer la complétude du permanent non-associatif, les auteurs suivent de près la preuve habituelle et la découpe en trois étapes :

– ils montrent en adaptant la preuve de G. Malod et N. Portier que $\mathsf{VNP}_{e_{\bar{A},C}} = \mathsf{VNP}_{\bar{A},C}$ où $\mathsf{VNP}_{e_{\bar{A},C}}$ désigne la classe $\mathsf{VNP}_{\bar{A},C}$ restreinte aux formules

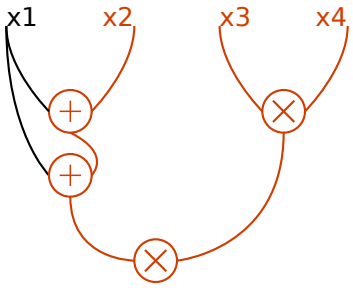

FIG. 2 – Le développement du monôme $x_2 \times (x_3 \times x_4)$

– ils montrent ensuite que toute formule peut être exprimée comme la projection d'un permanent ou d'un déterminant en construisant explicitement la matrice associée
– ils montrent enfin que le permanent simule efficacement les sommes booléennes. Ils réutilisent en fait pour cela la construction du cas associatif et commutatif et montrent qu'elle fonctionne toujours ici.

Nous détaillerons ces trois points dans trois sous-parties différentes en expliquant les démonstrations qui nous semblent les plus intéressantes. Nous essaierons aussi de donner quelques intuitions qui ne sont pas forcément décrites dans le papier original.

2.1 Développements de formules non-associatives

Cette partie a pour but de démontrer que $\mathsf{VNP}_{e_{\bar{A},C}} = \mathsf{VNP}_{\bar{A},C}$, étape essentielle de la preuve. En effet, cela permet de se limiter à la projection de formules dans le reste de la preuve, souplesse non-négligeable pour faire des constructions par induction.

La preuve est essentiellement la même que celle utilisant les développements (ou *parse trees*) sauf quand il s'agit de représenter une somme sur les développements par une somme booléenne, où il y a un tout petit peu plus de travail qu'avant. Nous détaillerons donc ici essentiellement ce point.

Il est assez aisé de se convaincre que la preuve construisant un circuit multiplicativement disjoint de VP à partir d'un circuit de VP fonctionne toujours dans le cas non-associatif, non-commutatif. On définit les développements exactement de la même façon que dans le cas original et on se convainc assez facilement qu'un développement T calcule toujours un monôme $\widehat{T}$ du circuit original pour peu qu'on respecte l'associativité et la commutativité induite par ce développement, comme l'illustre la figure 2 et que pour une circuit multiplicativement disjoint calculant le polynôme ϕ, on a bien : $\phi = \sum_T \widehat{T}$ où T décrit l'ensemble des développements du circuit.

Ce qui est moins évident c'est que cette somme sur des développements peut être vue comme une somme booléenne. Pour un circuit multiplicativement disjoint Ψ de taille s, on pose $\mathcal{T} = \mathcal{T}(s)$, l'arbre s-universel défini dans la première partie qui est de taille $O(s^4)$. Voici les variables booléennes que nous définissons et la signification qu'on veut leur donner :

(i) à chaque porte v de Ψ, on associe une variable booléenne a_v. On veut que $T = \{v \mid a_v = 1\}$ soit un développement de Ψ.

(ii) à chaque feuille u de $\mathcal{T}$, on associe une variable booléenne b_u. On veut que $V = \{u \mid b_u = 1\}$ soit telle que $T = \kappa(\mathcal{T}; V)$

(iii) à chaque feuille u de $\mathcal{T}$, à chaque entrée w de Ψ on associe la variable booléenne $c(u, w)$. On

veut que $c(u,w) = 1$ si et seulement si u est une feuille de V, w une entrée présente dans T et que u correspond à w.

Si a, b, c respectent ces trois conditions, on décrit bien un développement du circuit initial et sa projection sur l'arbre universel, ce qui nous permet de calculer le monôme de ce développement en posant :

$$L_u = 1 - b(u) + \sum_w c(u,w)\widehat{w}$$

où $\widehat{w}$ désigne l'étiquetage de l'entrée w. Si la feuille u ne fait pas parti de l'ensemble V, alors tous les $c(u,w)$ sont nuls et $L_u = 1$. Sinon, il existe un unique w tel que $c(u,w) = 1$, qui correspond à l'entrée de Ψ associé à u, et on a bien $L_u = \widehat{w}$. On remarque donc que :

$$\widehat{T} = \prod^{\mathcal{T}}(L_{u_1}, \ldots, L_{u_m})$$

Vérifier ces trois conditions sur a, b, c se fait en temps polynomial et donc peut être décrit par une formule booléenne $B((a,b,c),y)$ telle que : a, b, c vérifient ces trois conditions si et seulement si $\sum_y B((a,b,c),y) = 1$. On a donc bien :

$$\Psi = \sum_{a,b,c,y} B((a,b,c),y) \prod^{\mathcal{T}}(L_{u_1}, \ldots, L_{u_m})$$

2.2 Projection des formules sur le déterminant

On sait désormais que pour montrer la complétude du permanent, on peut se restreindre à la projection des formules. La preuve originale introduit ici le concept de branching programs qui permet d'associer une formule à un graphe orienté, donc à une matrice en considérant la matrice d'adjacence de ce graphe. La notion de branching program ne s'adapte pas vraiment au cas non-associatif et c'est pourquoi les auteurs ici se voient obligés de construire explicitement les matrices représentant la projection sur le permanent du polynôme initial. L'article traite le cas du permanent et mentionne qu'un raisonnement analogue permet de montrer que toute formule se projette aussi sur le déterminant. Ici, nous décrirons donc plus en détail le cas du déterminant.

Voici l'énoncé précis du théorème et sa démonstration :

Theorem 2.1 *Soit Φ une formule arithmétique de taille s. Alors il existe une matrice M de taille au plus $s+1 \times s+1$ avec des variables et des constantes comme coefficients et un arbre binaire T avec $s+1$ feuilles tels que : $\Phi = DET^T(M)$. De plus, M est telle que : pour tout $i \leq s$, $M_{i,i+1} = -1$ et pour $j > i+1$, $M_{i,j} = 0$.*

Cas initial $\Phi = x$ est une entrée. Dans ce cas, $M = \begin{bmatrix} 1 & -1 \\ 0 & x \end{bmatrix}$ et $T = (v,v)$ conviennent et M a la forme désirée.

Csa $\Phi = \Phi_1 \times \Phi_2$ Par induction, on possède les matrices M_1 et M_2 et les arbres T_1 et T_2. On pose $M = \begin{bmatrix} M_1 & E \\ 0 & M_2 \end{bmatrix}$ avec E ayant un -1 en bas à gauche, et des 0 partout ailleurs et $T = (T_1, T_2)$. Cette matrice a bien la forme voulue par induction et sa taille vaut $|\Phi_1| + |\Phi_2| \leq |\Phi| + 1$. De plus,

pour calculer son déterminant, on peut séparer les permutations selon différentes propriétés. En effet, si π est une permutation telle que pour un $i \leq s_1$, $\pi(i) > s_1 + 1$ alors $M_{i,\pi(i)} = 0$ donc elle ne participe pas à la valeur du déterminant. De plus, s'il existe $i \leq s_1$ tel que $\pi(i) = s_1 + 1$ alors par principe des tiroirs, on a forcément un $i \leq s_1$ tel que $\pi(i) > s_1+1$ et donc ce genre de permutation ne participe pas non plus à la valeur du déterminant. Finalement, les seules permutations participant à la valeur du déterminant sont des permutations qui sont stables sur $[1, s_1]$ et sur $[s_1 + 1, s_2]$, soit en gros deux permutations π_1 et π_2 de $[1, s_1]$ et $[1, s_2]$, à support disjoints donc on a bien $sgn(\pi) = sgn(\pi_1)sgn(\pi_2)$. D'où :

$$\begin{array}{rcl} DET^T(M) & = & \sum_{\pi_1,\pi_2} sgn(\pi_1)sgn(\pi_2) \prod^{T_1}(M_{1,\pi_1(1)} \dots M_{s_1,\pi_1(s_1)}) \prod^{T_2}(M_{s_1+1,\pi_2(s_1+1)} \dots M_{s_2,\pi_2(s_2)} \\ & = & DET^{T_1}(M_1) \times DET^{T_2}(M_2) \end{array}$$

ce que nous voulions.

Cas $\Phi = \Phi_1 + \Phi_2$ On pose désormais

$$M = \begin{bmatrix} 1 & v & 0 & 0 \\ 0 & M_1 & v_1 & 0 \\ M2[1] & 0 & v_2 & M_2[2^+] \end{bmatrix}$$

où v est le vecteur ligne avec un -1 à gauche et des 0 ailleurs, v_1, v_2 des vecteurs colonnes avec un -1 en bas et des 0 ailleurs et $M_2[1]$ la première colonne de M_2 et $M_2[2^+]$ le reste de M_2. Cette matrice est de taille $s = s_1 + s_2 + 1$ et a la forme voulue. On définit T pour qu'il ait les propriétés suivantes :

$$\prod^{T}(-1, f_1, \dots, f_{s_1}, -1, \dots - 1) = (-1)^{s-s_1} \prod^{T_1}(f_1, \dots, f_{s_1})$$

$$\prod^{T}(f_1, -1 \dots - 1, f_2, \dots f_{s_2}, -1, \dots - 1) = (-1)^{s-s_2} \prod^{T_2}(f_1, \dots, f_{s_2})$$

Soit π une permutation. En regardant la matrice M on se rend compte que si $\pi(1) > 2$, alors $M_{i,\pi(i)} = 0$ et donc ce genre de permutation ne participent pas à la valeur du déterminant. Pour les mêmes raisons que dans le cas précédent, on a nécessairement $\pi(s_1 + 1) = s_1 + 2$ ou $\pi(s) = s_1 + 2$ Étudions les autres cas :

(i) Si $\pi(1) = 1$ et $\pi(s_1 + 1) = s_1 + 2$, on a forcément un i tel que $M_{i,\pi(i)} = 0$: ces permutations ne participent pas au déterminant

(ii) Si $\pi(1) = 1$ et $\pi(s) = s_1 + 2$, et $M_{i,\pi(i)} \neq 0$ pour tout i, alors $\pi(i) \in \{2, s_1 + 1\}$ pour $i \in \{2, s_1 + 1\}$ et $M_{i,\pi(i)} = -1$ ailleurs.

(iii) Si $\pi(1) = 2$ et $\pi(s_1 + 1) = s_1 + 2$, et $M_{i,\pi(i)} \neq 0$ pour tout i, alors $\pi(i) \in \{1, s_1 + 3 \dots s\}$ pour $i \in \{s_1 + 2 \dots s\}$ et $M_{i,\pi(i)} = -1$ ailleurs.

(iv) Si $\pi(1) = 2$ et $\pi(s) = s_1 + 2$, on a forcément un i tel que $M_{i,\pi(i)} = 0$: ces permutations ne participent pas au déterminant

Seules les permutations de type (ii) et (iii) participent au calcul du déterminant, et on remarque qu'elles agissent localement sur M_1 et M_2. On aura donc :

$$DET^T(M) = \sum_{\pi \in (ii)} (-1)^{sgn(\pi)} \prod^{T}(M_{1,\pi(1)} \dots M_{s,\pi(s)}) + \sum_{\pi \in (iii)} (-1)^{sgn(\pi)} \prod^{T}(M_{1,\pi(1)} \dots M_{s,\pi(s)})$$

$$= \sum_{\pi \in (ii)} (-1)^{sgn(\pi)} \prod^{T} (-1, M_{2,\pi(2)} \dots M_{s_1+1,\pi(s_1+1)}, -1, \dots -1) +$$

$$\sum_{\pi \in (iii)} (-1)^{sgn(\pi)} \prod^{T} (M_{1,\pi(1)}, -1 \dots, -1, M_{s_1+2,\pi(s_1+2)}, \dots M_{s,\pi(s)})$$

$$= \sum_{\pi_1} (-1)^{sgn(\pi_1)} \prod^{T_1} (M_{1;1,\pi(1)} \dots M_{1;s_1,\pi(s_1)} + \sum_{\pi_2} (-1)^{sgn(\pi_2)} \prod^{T_2} (M_{2;1,\pi(1)}, M_{2;s_2,\pi(s_2)})$$

$$= DET^{T_1}(M_1) + DET^{T_2}(M_2)$$

ce que nous voulions. Le passage de la signature de π à celle de π_1 fonctionne puisque l'arbre T que nous utilisons met $(-1)^{s-s_1}$ en facteur. Donc si π a p orbites, π_1 à $p-(s-s_1) = p-(s_2+1)$ orbites, puisqu'on enlève celles induites par π_2.

2.3 Le permanent simule des sommes booléennes

Ce point-là est sûrement le plus corsé de la preuve originale, puisqu'il fait intervenir des gadgets sur les branching programs et des constructions loin d'être évidentes. Cependant, les auteurs ici réutilisent astucieusement ces constructions du cas associatif et commutatif pour montrer qu'elles sont toujours valables dans le cas non-associatif. Ils observent que dans le cas associatif et commutatif et en caractéristique différente de 2, pour une $m \times m$ matrice M donnée, on construit une matrice

$$M' = \begin{bmatrix} M & M_1 \\ M_2 & M_3 \end{bmatrix}$$

de taille au plus $5s_e + m$ où s_e est le nombre d'occurence de e dans M, avec M_1, M_2, M_3 des matrices contenant uniquement des éléments du corps telle que :

$$PERM(M') = PERM(M)|_{e=0} + PERM(M)|_{e=1}$$

On peut en fait utiliser la même construction pour le cas non-associatif en explicitant les arbres utilisés pour la multiplication à gauche de l'égalité. Donc si M est une matrice, on a :

$$PERM^{(T,P)}(M') = PERM^{T}(M)|_{e=0} + PERM^{T}(M)|_{e=1}$$

où P est l'arbre binaire "peigne" de taille $5s_e$ qui fait le produit en associant de gauche à droite. On peut observer cela en associant tout monôme non-associatif et non-commutatif au monôme associatif et commutatif correspondant et en comparant leurs coefficients.

En itérant l'opération sur un nombre polynomial de variables $e_1, \dots, e_s$, on peut faire une somme exponentielle booléenne en augmentant la taille de la matrice d'au plus $5(s_{e_1} + \dots + s_{e_s})$ puisque quand on construit la somme sur e_i, on ne change pas le nombre d'occurence de e_j pour $i < j$.

Ce raisonnement ne fonctionne hélas qu'en caractéristique 2 puisqu'il utilise implicitement les gadgets de la preuve originale. Cependant il semble s'adapter sans mal au cas de l'hamiltonien, en définissant l'hamiltonien non-associatif similairement au permanent :

$$HC_n(M) = \sum_{\sigma cycles} \prod^{\mathcal{T}_n} (M_{1,\sigma(1)} \dots M_{m,\sigma(m)})$$

ce qui permettrait de généraliser le résultat de l'article en enlevant la condition sur la caractéristique. Cependant, il faut encore vérifier qu'on peut construire explicitement une projection des formules sur l'hamiltonien comme dans la partie précédente ce que nous n'avons toujours pas réussi à faire.

3 Séparation

Finalement, les auteurs proposent un polynôme de $\mathsf{VNP}_{\bar{A},C}$ qui n'est pas calculable dans $\mathsf{VP}_{\bar{A},C}$. Cela prouve la séparation des deux classes en quelque caractéristique que ce soit. Soit S_n une suite d'arbre binaire à n feuilles, constructible en temps polynomial en n. On pose :

$$V_n(z_0, z_1) = \sum_{s<s'} \prod^{S_n} z_{s(0)} \dots z_{s(n)} \prod^{S_n} z_{s'(0)} \dots z_{s'(n)}$$

On commence par remarquer que ce candidat n'en est pas du tout un dans le cas associatif et commutatif puisqu'il a seulement 2 variables et donc un nombre polynomial en n de monômes commutatifs et associatifs : il est donc explicitement constructible dans VP associatif et commutatif. La difficulté pour calculer ce polynôme dans le cas non-associatif provient vraiment du fait qu'on a un nombre exponentiel de monômes dû à la faible structure.

Ce polynôme est dans $\mathsf{VNP}_{\bar{A},C}$. Il suffit de remarquer que :

$$z_{s(i)} = s(i)z_1 + (1 - s(i))z_0$$

et qu'on peut tester $s < s'$ et construire S_n en temps polynomial en n.

Pour montrer que ce polynôme n'est pas dans $\mathsf{VP}_{\bar{A},C}$, les auteurs exhibent une borne inférieure sur le nombre de porte de multiplication utilisée par un circuit calculant V_n. Le lemme clé utilisé est le suivant, que nous admettons ici :

Lemme 3.1 *Si*

$$\sum_{i,j\in[1,k];i<j} x_i x_j = \sum_{i\in[1,m]} f_i g_i$$

avec f_i et g_i homogènes de degré 1, alors $m \geq (k-1)/2$.

Supposons qu'on ait un circuit Φ qui calcule V_n et qui soit homogène (on ne perd qu'un facteur polynomial à faire cette hypothèse). Soient $v_1 \dots v_m$ les portes de multiplication de la forme : $v_i = u_i \times w_i$ avec $deg(v_i) = 2n$. On va prouver que m est exponentiel en n en utilisant la borne donnée par le lemme. Il est facile de voir que $\Phi = \sum_{i=1}^{m} a_i \Phi_{u_i} \Phi_{w_i}$ avec a_i un élément du corps.

Si on remplace désormais chaque monôme de la forme $\prod^{S_n} z_{s(0)} \dots z_{s(n)}$ par une variable x_s (on a donc un x_s par élément $s \in \{0,1\}^n$), on peut montrer que :

$$\sum_{s<s'} \prod^{S_n} z_{s(0)} \dots z_{s(n)} \prod^{S_n} z_{s'(0)} \dots z_{s'(n)} = \sum_{i\in[1,m]} f_i g_i$$

où les f_i et g_i sont sensiblement les polynômes Φ_{u_i} et Φ_{w_i} où l'on a remplacé les produits selon S_n par des variables du type x_i (les auteurs montrent ici que ces polynômes ne contiennent bien que des produits de cette forme-là).

On obtient donc en utilisant le lemme que $m \geq \frac{(2^n-1)}{2}$ ce qui termine la preuve et donne :

$$\mathsf{VP}_{\bar{A},C} \neq \mathsf{VP}_{\bar{A},C}$$

et donc aussi

$$\mathsf{VP}_{\bar{A},\bar{C}} \neq \mathsf{VP}_{\bar{A},\bar{C}}$$

Conclusion

Le présent article a été très agréable à lire. Les auteurs explicitent régulièrement leur plan et procèdent en plusieurs petites étapes simples. Nous avons essayé de rendre compte de la structure très organisée des preuves de complétudes, tout en essayant de trouver les parallèles qu'il y a entre la preuve de complétude du permanent dans le cas courant et celle-ci, notamment comment les auteurs substituent la notion de branching programs à une construction matricielle explicite, ce qui ne favorise pas l'intuition mais qui permet d'obtenir une preuve plus générale. L'article étant long, nous avons choisis de présenter toutes les idées en n'explicitant pas toutes les preuves ou en passant certains détails techniques sous silence. Néanmoins l'article original est très précis. Cependant nous espérons avoir donné suffisamment d'intuition pour faciliter la compréhension des preuves. Il semble possible d'étendre les travaux des auteurs à l'hamiltonien ce qui permettrait d'obtenir un problème complet même en caractéristique 2. Cependant, la construction des matrices serait plus compliquée (dans le cas associatif et commutatif, on projette sur des hamiltoniens de taille double) et le gain théorique assez négligeable puisque les auteurs séparent les classes non-associatives pour toute caractéristique.

Références

[1] L. Blum, M. Shub, and S. Smale. On a theory of computation over the real numbers ; np completeness, recursive functions and universal machines. *Foundations of Computer Science, Annual IEEE Symposium on*, 0 :387–397, 1988.

[2] P. Bürgisser. *Completeness and Reduction in algebraic theory*. Springer, 2000.

[3] N. Portier G. Malod. Characterizing valiant's algebraic complexity classes. *J.Complexity 24*, 2008.

[4] Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, STOC '91, pages 410–418, New York, NY, USA, 1991. ACM.

[5] Amir Yehudayoff Pavel Hrube, Avi Wigderson. Relationless completeness and separations. *IEEE Computational Complexity Conference*, 2010.

[6] L. G. Valiant. Completeness classes in algebra. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, STOC '79, pages 249–261, New York, NY, USA, 1979. ACM.