# Construction d'une structure où $\mathcal{P} = \mathcal{NP}$

Martin Bodin

1er Décembre 2011

**Résumé**

Nous exposons ici les résultats des articles [Pru06] et [Poi00] qui cherchent tout deux à construire des structures finies où l'élimination des quantificateurs est faisable en temps polynomial.

L'article de Bruno Poizat ne résout que le cas où une seule variable est libre dans la formule à éliminer. Celui de Mihai Prunecsu s'inspire fortement de ce dernier, mais améliore la construction en permettant l'élimination sur des formules avec autant de variables libre que voulu.

## Choix et objectifs

Nous cherchons à construire une structure $M$ dans laquelle l'élimination des quantificateurs peut se faire en temps polynomial. C'est à dire qu'il existe un algorithme polynomial qui à toute formule existentielle $\exists \vec{y}, \phi(\vec{x}, \vec{y})$ associe une formule $\psi(\vec{x})$ équivalente (où $|\psi|$ est polynomialement bornée par $|\phi|$).

Dans une telle structure, il serait donc possible d'éliminer toute suite de quantificateurs en temps polynomial en la longueur de la formule passée en entrée et la hiérarchie polynomiale s'effondrerait dès le premier rang : dans $M$, on aurait $\mathcal{P} = \mathcal{NP}$. Bien que la construction d'une telle structure $M$ sera assez artificielle, l'idée serait ensuite d'essayer de transposer ses propriétés sur la structure booléenne $\{0, 1\}$, ce qui améliorerait peut-être la compréhension de $\mathcal{P} = \mathcal{NP}$ dans les structures habituelles.

On impose de plus à cette structure d'être finie. Dans le cas contraire, il est assez facile de la construire, mais la structure est très loin de la structure booléenne objectif. On peut en effet considérer $M = \mathbb{N}$, ce qui permet de définir une relation d'appartenance [1] (dite *appartenance d'Ackerman*) $x \in y$ par le fait que le $x^e$ bit de $y$ dans son écriture en base 2 est 1. On peut alors encoder des couples $(x, y)$ par l'ensemble $\{\{x\}, \{x, y\}\}$, puis des uples $(x_1, \dots, x_n)$ par $(x_1, (x_2, \dots (x_{n-1}, x_n) \dots))$. On peut ainsi utiliser un codage des formules dans $M$ de tel sorte que la longueur du codage de $\phi(\vec{x})$ soit proportionnel à la taille de $\phi(\vec{x})$. On considère alors la structure sur $M$ contenant l'égalité, la constante 0, la fonction binaire de construction de couples et d'une suite infinie de prédicats unaires $V_1$, …, $V_n$, … construite inductivement : $V_{n+1}(x)$ est vraie si $x$ est le codage d'une formule $\phi$ vraie qui ne fait intervenir que les $n$ premiers prédicats $V_1, \dots V_n$. Pour toute formule $\phi(\vec{x})$ n'utilisant que les $n$ premiers prédicats $(V_i)$, $\phi(\vec{x})$ est équivalente à $V_{n+1}(C)$ où $C$ est le codage dans $M$ de $\phi(\vec{x})$.

1. On aurait pu tout aussi bien prendre n'importe quel modèle de la théorie des ensembles.

Plus généralement, on impose à la structure construite de ne pas être trop expressive (en particulier, elle ne contiendra pas l'arithmétique). Dans le cas contraire, le théorème de Tarski, qui énonce que l'ensemble des codes de formules vraies sur un langage ne sont pas exprimables par une formule de ce même langage, nous empêcherait d'éliminer les quantificateurs.

La notion de codage est toutefois gardée et on suppose que $M$ contient un prédicat unaire $V$ tel que toute formule de la forme $\exists \vec{y}, \phi(\vec{x}, \vec{y})$ admette un codage $\tau_\phi(x)$ tel que $V(\tau_\phi(\vec{x}))$ soit équivalente à $\exists \vec{y}, \phi(\vec{x}, \vec{y})$. Ce prédicat symbolise ainsi la vérité d'une formule et l'on cherche à le définir de manière ad hoc afin que l'équivalence soit vraie (en sachant que $V$ peut apparaître dans $\phi$).

# 1 Premières définitions

On commence par définir un langage $B$ (appelé *bloc*) constitué de deux fonctions successeurs $s_0$ et $s_1$ et d'un symbole de constante $r$, qui génèrent $B$ librement :

$$B := \{s_{\epsilon_1} \circ \ldots \circ s_{\epsilon_n}(r) | n \in \mathbb{N}, \forall i, \epsilon_i \in \{0, 1\}\}$$

**Définition 1.** *L'entier $n$ donné dans la définition des éléments de $B$ est la* taille*. Les éléments d'une taille donnée forme un* niveau*.*

On ajoute ensuite dans la définition un prédécesseur $p$ défini par

$$\forall x \in B, (p(x) = x \Leftrightarrow x = r) \wedge p \circ s_0(x) = x \wedge p \circ s_1(x) = x$$

**Remarque.** On notera qu'aucune équation de la forme $s_0(x) = s_1(y)$ n'est satisfiable et que toute équation de la forme $s_\epsilon(x) = s_\epsilon(y)$ est équivalente à $x = y$ (ce qui n'est pas le cas pour le prédécesseur).

On ajoute enfin un prédicat de vérité $V$ sur $B$.

**Définition 2.** *On dira que les éléments $x$ de $B$ sont* noirs *ou* blancs *en fonction de si $V(x)$ est vrai ou non, $B$ constituant maintenant un* bloc coloré*.*

**Définition 3.** *Un* triangle de hauteur $n$ *est une conjonction $T(x)$ de $2^{n+1} - 1$ termes de la forme $V(t(x))$ ou $\neg V(t(x))$ pour tous les termes $t(x)$ de longueur inférieure à $n$ en $x$.*

Il y a $2^{2^{n+1}-1}$ triangles de hauteur $n$.

**Définition 4.** *Le $n$-*voisinage *de $\vec{x}$ est une conjonction des tous les triangles de racine $p^n(x_i)$ et de hauteur $2n$, ainsi que de toutes les formules réalisées par $\vec{x}$ de la forme $p(y) = y$, $p(y) \neq y$, $y = y'$ et $y \neq y'$ où $y$ et $y'$ parcourent tous les termes qui apparaissent dans ces triangles.*

*Si $\vec{x}$ est un singleton, on parlera de* voisinage individuel*.*

Intuitivement, le $n$-voisinage d'un tuple exprime avec une « précision » $n$ à quoi ressemble ce tuple, à isomorphisme près. En particulier, si tous les éléments de $\vec{x}$ ont une taille inférieure à $n$, le $n$-voisinage de $\vec{x}$ décrit très précisément à isomorphisme près $\vec{x}$.

**Exemple.** Si $\vec{x} = (r, s_0(r))$ et que l'on a $V(y)$ pour tout $y$, son 1-voisinage est

$$V(r) \wedge V(s_0(r)) \wedge V(s_1(r)) \wedge V(s_0(r)) \wedge V(s_0 \circ s_0(r)) \wedge V(s_1 \circ s_0(r)) \wedge$$

$$r = p(r) \wedge s_0(r) \neq p \circ s_0(r) \wedge s_1(r) \neq p \circ s_1(r) \wedge s_0 \circ s_0(r) \neq p \circ s_0 \circ s_0(r) \wedge$$

$$s_1 \circ s_0(r) \neq p \circ s_1 \circ s_0(r) \wedge s_0(r) = s_0(r) \wedge \ldots$$

Où la suite de la formule nie l'égalité entre tous les termes différents de l'exemple.

Le $n$-voisinage d'un tuple est donc une formule de taille exponentielle en celui-ci, mais avec quelques redondance.

On identifiera la formule donnée par un $n$-voisinage et l'ensemble des éléments qui obéissent à cette formule. Une autre manière de présenter le $n$-voisinage de $\vec{x}$ et de considérer l'ensemble des tuples ne pouvant pas être distingués de $\vec{x}$ en moins de $n$ questions (sur des égalités ou inégalités structurelles ou sur le prédicat $V$).

**Définition 5.** *Deux éléments $x$ et $y$ seront dits $n$-dépendants si leurs $n$-voisinages respectifs ne sont pas disjoints. La $n$-dépendance est réflexive et symétrique, mais pas transitive (si $x$ et $y$ sont $n$-dépendants et que $y$ et $z$ le sont aussi, alors $x$ et $z$ sont $2n$-dépendants, mais pas nécessairement $n$-dépendants).*

L'idée va ici être de choisir le prédicat $V$ de telle sorte que les $n$-voisinages soient exprimables de manière polynomiale en $n$.

**Définition 6.** *Un prédicat $V$ sera dit* générique *si pour tout voisinage individuel $\mathcal{N}(x)$ réalisé par $(M, V)$, $\mathcal{N}(x)$ est réalisé un nombre infini de fois.*

Une structure composée d'une union infinie et disjointe de blocs de couleur identique a toujours un prédicat générique : tous ces blocs sont en effets isomorphes (à un changement de racine et de fonctions successeurs près) et tout résultat qui s'applique à l'un s'applique aux autres.

Dans la suite, on omettra les symboles de fonctions ainsi que la constante $r$ pour des soucis de lisibilité : on écrira donc $(B, V)$ pour la structure $(B, s_0, s_1, r, p, V)$.

La suite de la construction est la suivante :

- Un lemme d'élimination qui permet de réduire le problème de savoir si $\vec{x}$ satisfait la formule $\exists \vec{y}, \phi(\vec{x}, \vec{y}$ à la connaissance du $n$-voisinage de $x$ et des $n$-triangles de $x$ réalisés dans $M$, où $n$ est relié à la taille de $\phi$.
- Grâce à ce lemme, on a rejeté l'étude de $\exists \vec{y}, \phi(\vec{x}, \vec{y})$ sur des propriétés locales à $\vec{x}$. L'étape suivante consiste à encoder cette information dans une formule sans quantificateur, ce qui passera par imposer une autre propriété sur $V$.
- Un certain type de formules (quantifiées) sont ensuite encodées dans $V$.

# 2 Lemme d'élimination

On considère une structure $(M, V)$ union infinie disjointe de blocs avec $V$ générique.

**Lemme 1.** *Soit $\phi(\vec{x})$ une formule de taille $n$ équivalente à une formule existentielle sur $\vec{x}$. Il existe une formule sans quantificateurs $\lambda(\vec{x})$ équivalente dans $M$ tout $\vec{x}$ à $\phi(\vec{x})$ telle que tous les termes de $\lambda(\vec{x})$ on une taille au plus $2n$.*

Ainsi, pour savoir si un tuple $\vec{x}$ de $M$ satisfait $\phi(\vec{x})$, il suffit de connaître le $2n$-voisinage de $(\vec{x}, r)$, ainsi que les $2n$-voisinages individuels réalisés pas $M$.

*Démonstration.* On commence par éliminer l'existentielle en mettant $\phi$ sous forme normale disjonctive. Ceci augmente exponentiellement la taille de $\phi$, mais chacune des sous-formules de $\phi$ contient le même nombre de symboles de fonction (le même que dans $\phi$), puisque dans chacune de ses sous-formules apparaissent exactement une fois chacune des formules atomiques de $\phi$ (ou leur négation).

L'existentielle se propageant sur chacune de ces sous-formules $\phi_i(\vec{x}, \vec{y})$, il suffit de traiter ces dernières. Chacune des formules atomiques constituant $\phi_i(\vec{x}, \vec{y})$ est de la forme $V(t(u))$, $\neg V(t(u))$, $t_1(u) = t_2(v)$ et $t_1(u) \neq t_2(v)$ où $u$ et $v$ désignent des éléments de $\vec{x}$ et $\vec{y}$ et $t$, $t_1$ et $t_2$ des éléments de $M$ (des suites de compositions de $s_0$, $s_1$ et $p$).

Notons que les égalités permettent de construire beaucoup d'information sur les termes, puisque $\phi_i(\vec{x}, \vec{y})$ est une conjonction. On dira que deux variables sont *voisines* si elles sont reliées par une suite d'égalité de la forme $t_1(u) = t_2(v)$. On choisit ensuite un chemin entre deux formules voisines de telle sorte que le graphe de relation entre les différentes variables soit un arbre étiqueté en des éléments de $\vec{x}$. On réduit ensuite cet arbre d'équation en une forme canonique $u = t(v)$, $u$ étant à . Ceci augmente encore la disjonction de façon exponentielle puisqu'une équation de la forme $p(u) = v$ se décompose en trois cas : $u = p(u) = v$, $u = s_0(v)$ et $u = s_1(v)$. Cependant, chacune des conjonctions reste de taille identique, puisque l'on a remplacé l'égalité $p(u) = v$ que par une seule de ces trois possibilités ($u$ et $v$ étant fusionnées dans le premier cas).

Comme $\phi$ était de taille $n$, chacun des $\phi_i(\vec{x}, \vec{y})$ est lui aussi de taille $n$. Il est maintenant possible de réécrire chaque voisin $v$ d'un élément $x_i$ de $\vec{x}$ sous la forme $v = t(x_i)$, et donc de les remplacer dans la formule $\phi_i(\vec{x}, \vec{y})$. Chaque terme qui apparaît dans cette nouvelle formule est cependant de taille inférieure à $2n$.

On peut donc décomposer cette nouvelle sous-formule en trois types de conjonctions :

- Des conditions $\phi_0(\vec{x})$ sur des termes de $\vec{x}$ de longueur inférieur à $2n$.
- Des inéquations $\phi_1(\vec{x}, \vec{y})$ de la forme $t_1(x_i) \neq t_2(y_j)$.
- Des conditions $\phi_2(\vec{y})$ sur des variables voisines d'aucun élément de $\vec{x}$. Chacun de ces termes est de longueur inférieur à $n$.

Le fait que $M$ soit une union infinie de blocs permet de choisir des éléments $\vec{y}$ qui ne sont pas dans les blocs de $\vec{x}$ : ils satisfont alors toutes les propriétés de $\phi_1(\vec{x}, \vec{y})$ et comme tous les blocs sont isomorphes, ils satisfont $\phi_2(\vec{y})$ si et seulement si leurs images dans les blocs de $\vec{x}$ le satisfaisaient déjà. On peut ainsi éliminer la conjonction $\phi_1(\vec{x}, \vec{y})$.

Pour éliminer $\phi_2(\vec{y})$, il suffit de réappliquer le procédé sur $\vec{y}$. On augmente encore le nombre de disjonctions, mais chacun des termes qui y figurent ont une longueur inférieure à $2n$. □

# 3 Un $V$ éparse

Afin de réduire encore cette disjonction, une idée est de se limiter à un prédicat $V$ telle qui n'accepte qu'un très petit nombre de point dans le $n$-voisinage d'un point.

**Définition 7.** *Un prédicat $V$ est dit* éparse *si tout $x$ qui le vérifie est de la forme $s_1^n s_0 s_{\epsilon_1} \dots s_{\epsilon_n}(r)$ avec $r$ une racine de $M$ ($p(r) = r$).*

Un prédicat éparse va donc colorier en blanc une très grande partie des éléments de $M$. À partir de maintenant, on considérera toujours $(M, V)$ comme étant une union infinie de blocs identiques [2] avec $V$ éparse.

On a d'ailleurs le résultat suivant :

**Lemme 2.** *Si $x$ est de taille strictement supérieure à $3m$, son $m$-voisinage individuel contient au plus un point noir, qui est de la forme $s_1^n p^m(x)$ avec $n \in [\![0, 2m]\!]$.*

*Démonstration.* Soit $h$ la taille de $x$. Supposons par l'absurde qu'il existe deux tels points noirs. Comme $V$ est éparse, ils sont de taille de la forme $2l + 1$. On a $h - m \leqslant l$ (car ils sont dans le $m$-voisinage de $x$) et $h + m \geqslant 2l$ (car $V$ est éparse). On obtient alors successivement $l \leqslant 2m$ puis $h \leqslant 3m$, qui est une contradiction. □

**Remarque.** Cette nouvelle hypothèse sur la structure permet de régler le problème posé par Bruno Poizat qui l'a fait conjecturer que son raisonnement ne fonctionnait pas lorsque l'on devait éliminer plus d'une formule. Son contre-exemple exploitait justement le fait que l'on ne sache rien sur le prédicat $V$ : il est par exemple possible d'exprimer à l'aide de formules logiques le fait qu'il existe pour $x$ et $x'$ donnés des $\epsilon_1$, …, $\epsilon_n$ tels que $V(s_{\epsilon_1} \circ \dots s_{\epsilon_n}(x))$ et $\neg V(s_{\epsilon_1} \circ \dots s_{\epsilon_n}(x))$.

Avec l'hypothèse que $V$ est éparse, on sait tout de suite la forme que ces $\epsilon_i$ doivent respecter pour vérifier cette propriété et la recherche est beaucoup moins longue (et retombe dans le polynomial).

Il est maintenant possible de construire une formule (sans quantificateurs) qui exprime quel est le $m$-voisinage individuel d'un $x$ en une complexité très raisonnable :

**Lemme 3.** *Il est possible de construire en un temps linéaire en $m$ une formule $\beta(x)$ déterminant le $m$-voisinage individuel de $x$ donné.*

*Démonstration.* On commence par extraire (toujours à isomorphisme près) la structure de $x$ jusqu'à profondeur $3m$ en calculant $x$, $p(x)$, …, $p^{3m}(x)$ et en vérifiant à chaque étape ce que l'on a enlevé ($s_0$, $s_1$ ou que l'on est à une racine).

Si l'on tombe sur une racine à un moment donné, disons à l'étape $k$, il suffit d'exprimer que $x$ est de taille $k$, c'est à dire que $p^k(x) = p^{k+1}(x) \wedge p^{k-1}(x) \neq p^k(x)$ et que $x$ est égal à une certaine suite $t$ de $s_0$ et de $s_1$, ce qui s'exprime très bien par $x = t(p^k(x))$.

Sinon, on commence par exprimer le fait que $x$ commence par une certaine suite de $s_0$ et de $s_1$, sans arriver à une racine. De plus, il existe alors d'après le lemme précédent au plus un point noir dans le $m$-voisinage de $x$, et il est de la forme $s_1^n p^m(x)$, avec $n \in [\![0, 2m]\!]$. S'il en existe un — disons $s_1^n p^m(x)$ —, on ajoute $V(s_1^n p^m(x))$ à la formule. S'il n'en existe pas, il est possible de tricher légèrement et d'ajouter un symbole $\Sigma$ signifiant qu'il n'existe pas de points

2. C'est à dire à peu de choses près que $V$ se comporte de la même manière dans tous ces blocs.

noirs dans le $m$-voisinage, la valeur de $m$ se déduisant facilement du reste de la formule.

On obtient ainsi bien une formule exprimant le $m$-voisinage de $x$ en un temps linéaire. □

On peut montrer de manière relativement similaire le lemme généralisé suivant :

**Lemme 4.** *Si $\vec{x}$ est un tuple de longueur $k$, il est possible de calculer en temps $\mathcal{O}(mk^2)$ une formule $\beta(\vec{x})$ qui expriment le $m$-voisinage de $\vec{x}$.*

# 4 Description de l'encodage

Nous avons maintenant tout l'outillage nécessaire pour encoder les formules sur les voisinages. Le point clé maintenant est la présence dans la structure $(M, V)$ de deux « caractères » différents $s_0$ et $s_1$ qui permettront d'encoder directement dans la structure toutes les formules dont nous avons besoin.

Dans le papier [Pru06], les variables sont nommées par un encodage assez lourd de la forme $x''^{\cdots\prime}$ (c'est à dire par un encodage unaire). Cela ne modifie pas la complexité (et même va améliorer légèrement la taille de $\beta(x)$ !) si on les note par un codage plus efficace logarithmique en le nombre de variables.

Pour des raisons techniques, toutes les formules encodées seront supposées être de la forme

$$\forall \vec{x}, \beta(\vec{x}) \Rightarrow \exists \vec{y}, \phi(\vec{x}, \vec{y})$$

Il est maintenant temps de définir un $V$ éparse.

Comme on l'a déjà supposé, $M$ est composé d'une infinité de blocs identiques par isomorphisme. Il suffit donc de définir $V$ sur un seul de ces blocs.

Pour les éléments qui ne sont pas des codes (i.e. ne représentent pas une formule, ou pas une formule de la bonne forme), on considérera que seuls les $s_1^n s_0^{n+1}(r)$ seront noirs. Ceci, avec quelques conditions sur les couleurs des codes, permet que $V$ soit éparse.

On colorie temporairement tous les codes en blanc, puis on applique récursivement le coloriage suivant (en prenant l'ordre lexicographique sur les codes) : si un code $x$ représentant une formule $\beta(\vec{x}) \Rightarrow \exists \vec{y}, \phi(\vec{x}, \vec{y})$ est vraie au moment où on la considère (car elle peut très bien effectuer des tests de la forme $V(y)$ où $y$ est un code qui n'a pas encore été considéré), on la colorie en noir, sinon on la laisse en blanc.

L'idée est bien entendue que si $\theta$ est une formule de la bonne forme et $[\![\theta]\!]$ son code, on ait équivalence entre $\theta$ et $V([\![\theta]\!])$.

**Lemme 5.** *On considère $\theta = \forall \vec{x}, \beta(\vec{x}) \Rightarrow \exists \vec{y}, \phi(\vec{x}, \vec{y})$, où $\phi(\vec{x}, \vec{y})$ est de taille $n$. La véracité de $\theta$ dans $M$ est vérifiable en ne connaissant que la couleur des termes $t(r)$ de longueur inférieure strictement à $2n^2$ et une liste des $4n^2$-voisinages (à isomorphisme près) vérifiés par $M$.*

Il est donc maintenant possible de connaître la valeur de vérité de $\theta(r)$ en connaissant tous les termes de longueur inférieure à $2n^2$ et les $8n^2$-voisinages réalisés dans $M$.

*Démonstration.* Par le lemme 1, la formule $\exists \vec{y}, \phi(\vec{x}, \vec{y})$ est équivalente à une formule sans quantificateurs $\lambda(\vec{x})$ ne contenant que des termes en $\vec{x}$ ou $r$ de taille inférieure strictement à $2n$. Il suffit donc d'éliminer la quantification universelle $\forall \vec{x}, \beta(\vec{x}) \Rightarrow \lambda(\vec{x})$, c'est à dire éliminer une existentielle $\exists \vec{x}, \beta(\vec{x}) \wedge \neg\lambda(\vec{x})$.

Il suffit maintenant d'observer comment $\beta(\vec{x})$ a été construit : c'est une conjonction qui contient pour chaque variable de $\vec{x}$ et $r$ au pire $n$ relations de la forme $V(t_k(z_k)) \wedge z_k = t_{k-1}(z_{k-1}) \wedge \ldots \wedge z_1 = t_0(z_0)$ avec $z_0$ étant un élément de $\vec{x}$ ou égal $r$ et chacun des $t_i$ de taille inférieur à $2n$. On obtient ainsi une borne en $4n^2$. □

Par un argument de comptage, il est maintenant possible de montrer que tous les codages d'une formule vraie sont maintenant coloriés en noir dans cette structure.

On a donc ainsi construit une structure $(M, V)$ sous la forme d'une union infinie de blocs identiques avec $V$ générique et éparse tel que toute formule qui possède un encodage [3] ai la même valeur de vérité dans $M$ que l'application du prédicat $V$ à son codage.

# 5 $\mathcal{P} = \mathcal{NP}$ (dans cette structure)

Maintenant que la structure a été construite, montrons comment l'utiliser pour résoudre SAT en temps polynomial dans $M$ : on a en hypothèse une formule de la forme $\exists \vec{y}, \phi(\vec{x}, \vec{y})$ et de taille $n$ et on cherche s'il existe de telles variables $\vec{y}$ pour une instantiation des variables $\vec{x}$ données en entrée également sous la forme d'un tuple. L'algorithme prend ben entendu cette formule en binaire sous la forme d'un élément de $M$.

Le nombre $k$ de variables libres dans la formule $\exists \vec{y}, \phi(\vec{x}, \vec{y})$ est tel que [4] $k < 2n$.

Par l'algorithme donné par le lemme 4, il est donc possible de calculer une description courte $\beta(\vec{x})$ du $2n$-voisinage de $\vec{u}$ en temps $\mathcal{O}(n^3)$ [5].

Voici maintenant l'algorithme pour décider de SAT en temps $\mathcal{O}(n^3)$ :

- On commence par calculer le $\beta(\vec{x})$ évoqué précédemment en $\mathcal{O}(n^3)$. Cette formule exprime à un isomorphisme près le $2n$-voisinage du tuple $(\vec{u}, r)$.
- On construit ensuite la formule $\theta = \forall \vec{x}, \beta(\vec{x}) \Rightarrow \exists \vec{y}, \phi(\vec{x}, \vec{y})$, donc on calcule le code $[\![\theta]\!]$.
- On vérifie si $V([\![\theta]\!])$ est validé dans $M$.

On a construit $M$ pour cela : le résultat de ce dernier test est exactement la vérification de la formule $\theta$ équivalente à la formule initiale.

On a donc $\mathcal{P} = \mathcal{NP}$ dans $M$.

# Conclusion

Ce rapport résume les résultats et preuves des deux articles [Poi00] et [Pru06]. Comme les définitions différent légèrement d'un article à l'autre, des choix on dû être faits ici. J'ai en particulier modifié l'alphabet utilisé par Mihai Prunecsu

3. C'est à dire de la forme $\forall \vec{x}, \beta(\vec{x}) \Rightarrow \exists \vec{y}, \phi(\vec{x}, \vec{y})$.
4. On obtenait un meilleur rendu dans l'article [Pru06] à cause du codage en unaire des variables de $\phi$ dans $M$.
5. $\mathcal{O}(n^2)$ dans l'article [Pru06] pour la même raison.

de façon à ce que les variables soient écrites sous forme plus compacte. Cela ne change pas le résultat (même s'il dégrade légèrement son efficacité), mais je trouve que la version où la $n^e$ variable prenne une place $\mathcal{O}(n)$ dans $M$ est légèrement moins propre.

J'espère ne pas avoir été trop flou sur les définitions, ayant moi-même mis beaucoup de temps pour comprendre le fait que l'on travaille à isomorphisme près dans une structure contenant une infinité de fois le même bloc.

Cela est peut-être dû au fait que ces articles soient relativement récents, mais il y a eu relativement peu de suite pour l'instant : citons cependant [Gaß06] qui généralise cette construction en construisant à partir d'un langage à signature finie donné une nouvelle structure dans laquelle $\mathcal{P} = \mathcal{NP}$ (ou $\mathcal{P} \neq \mathcal{NP}$).

Pour en revenir à la remarque initiale, on remarque qu'à cause du prédicat $V$ calculé récursivement en fonction de tous les arguments inférieurs et donc de manière exponentielle en son argument, on ne peut pas plonger ce modèle dans la structure booléenne de manière polynomial (en tout cas pas tel quel). Ceci ne montre donc pas que $\mathcal{P} = \mathcal{NP}$ dans le sens usuel, dommage.

# Références


[Gaß06] C. Gaßner. From structures with $\mathcal{P} \neq \mathcal{NP}$ to structures with $\mathcal{P} = \mathcal{NP}$ and reverse. 2006.

[Poi00] B. Poizat. Une tentative malheureuse de construire une structure éliminant rapidement les quanteurs. In *Computer Science Logic*, pages 61–70. Springer, 2000.

[Pru06] M. Prunescu. Structure with fast elimination of quantifiers. *Journal of Symbolic Logic*, pages 321–328, 2006.