Lecture Notes in Computer Science 1862, pp. 61-70

UNE TENTATIVE MALHEUREUSE DE CONSTRUIRE UNE STRUCTURE ELIMINANT RAPIDEMENT LES QUANTEURS

Bruno POIZAT
poizat@desargues.univ-lyon1.fr

Introduction.

Je cherche à construire une structure M éliminant rapidement les quanteurs : à toute formule existentielle $(\exists \underline{y})\ \varphi(\underline{x},\underline{y})$, où φ est sans quanteurs, doit être associée une formule $\psi(\underline{x})$, également libre de quanteurs, qui lui soit équivalente dans M , la longueur de ψ étant polynomialement bornée en fonction de celle de φ ; autrement dit, pour une certaine constante c , $|\psi| \leq |\varphi|^c$. Tant qu'à faire, j'aimerais aussi que ψ fût calculée par un algorithme polynomial à partir de φ .

Pourquoi je veux faire ça ? Parce que si c'était vrai de la structure à deux éléments $M = \{0,1\}$, dans le langage réduit à l'égalité et aux deux constantes 0 et 1 (ou même à la seule relation $x = 0$!), eh bien on répondrait positivement à deux questions ouvertes en complexité, $NC^1 = P = NP$, version uniforme ou non suivant que l'élimination serait algorithmique ou pas (que ça implique $P = NC^1$ vient de ce que j'écris les formules sous la forme usuelle, et non pas comme des circuits booléens). La même chose se produirait si la structure était finie, hormis le cas trivial d'une structure à un élément que nous excluons, ou bien si son langage fini ne comportait que des relations et pas de fonctions (voir [POIZAT 1995]).

Donc, ce que je veux construire, c'est une structure ayant une propriété qui, si elle était possédée par la structure à deux éléments, provoquerait un cataclysme en Théorie de la Complexité ! Comme vous le voyez, je m'exerce au passe-temps favori des complexionistes, qui consiste à ne démontrer des théorèmes qu'après les avoir transportés dans des contextes où ils n'ont plus aucune signification. Il est en effet très improbable qu'une structure construite spécialement pour répondre à cette question ait un quelconque enjeu algorithmique, si bien que l'intérêt du problème ne réside que dans la plus ou moins grande difficulté de sa solution.

Si on autorise un langage infini, il devient complètement trivial : prenons un modèle M de la théorie des ensembles, par exemple celui des

ensembles héréditairement finis, formé des entiers naturels munis de l'appartenance d'Ackerman, dite aussi relation des bits, s'il est permis de s'exprimer ainsi ($x \in y$ s'il y a un 1 à la $x°$ place du développement de y en base 2) ; la fonction qui à (x,y) associe le couple $\langle x,y\rangle = \{\{x\},\{x,y\}\}$ permet de représenter le uple $(x_1, \ldots x_n)$ comme la "liste" $\langle x_1 , \langle x_2 , \langle \ldots\ldots \langle x_n , 0\rangle \ldots \rangle\rangle\rangle$; une fois identifiés à certains éléments de M les symboles servant à l'écriture des formules $\varphi(\underline{x})$, toute formule $\varphi(\underline{a})$ à paramètres $\underline{a}$ dans M se représente ainsi par un élément "$\varphi(\underline{a})$" de M , qu'en maths modernes, où tout est ensemble, on considère comme $\varphi(\underline{a})$ elle-même plutôt que comme un codage qui en respecte la taille. Nous considérons alors la structure dont l'univers est M , dans le langage comprenant (outre l'égalité) la constante 0 , la fonction binaire $\langle x,y\rangle$, et une suite infinie de prédicat unaires V_1 , ... V_n , V_{n+1} , ... construite par récurrence, V_{n+1} étant l'ensemble des formules $\varphi(\underline{a})$ vérifiées par M qui ne font intervenir que les n premiers prédicats V_1 , ... V_n : la formule $\varphi(\underline{x})$, quel que soit son rang de quantification, est équivalente dans M à V_{n+1}("$\varphi(\underline{x})$") , laquelle est sans quanteurs, et pratiquement de même longueur que φ ! Il y a aussi une manière encore plus débile d'éliminer, la "morleysation" d'une quelconque structure, qui associe à chaque formule $\varphi(\underline{x})$ un nouveau symbole relationnel $R_\varphi(\underline{x})$ (voir [POIZAT 1985] p. 89).

Si on veut que le problème présente quelque difficulté, et par conséquent quelqu'intérêt, il est indispensable de se limiter à un langage fini. Il faut alors se garder des structures M trop riches en pouvoir d'auto-expression, comme l'appartenance entre ensembles, ou même l'arithmétique, qui permettent de représenter les formules par des éléments de M et de les manipuler (ce que ne permet pas la fonction de couple à elle seule ; mais rien ne nous empêchait d'ajouter $\in$ à la structure ci-dessus), car un célèbre résultat de Tarski nous assure qu'elles ne peuvent éliminer les quanteurs (l'ensemble des codes d'énoncés Π_1 vrais n'est pas Σ_1 ; voir [POIZAT 1985] ch. 7). D'autre part, au moins dans un premier temps, on essaye seulement d'éliminer un bloc de quanteurs existentiels, et pas un bloc de quanteurs alternant arbitrairement.

Nous conservons cet idée de prédicat de vérité, et nous cherchons à construire M , structure de langage fini comportant un prédicat unaire $V(x)$, ayant la propriété qu'à toute formule existentielle $(\exists\, \underline{y})\ \varphi(\underline{x},\underline{y})$ soit associé un terme $\tau_\varphi(\underline{x})$, de longueur modérée par rapport à celle de φ , de sorte que M satisfasse :

$$(\forall\, \underline{x}) \qquad (\exists\, \underline{y})\ \varphi(\underline{x},\underline{y}) \ \leftrightarrow\ V(\tau_\varphi(\underline{x})) \qquad .$$

La difficulté vient bien sûr de ce que la formule (libre) φ mentionne le prédicat V . On voit bien là tout l'artifice de la construction : il ne s'agit pas d'élaborer un algorithme pour vérifier quelque chose dans le monde

réel ; on se donne au contraire l'algorithme a priori, et on construit une vérité qui s'y adapte ! Ça serait bien sûr beaucoup plus intéressant de montrer que certaines structures "naturelles" éliminent rapidement, ou bien lentement ; mais il faut s'attendre à ce que les questions de ce genre soient plus ou moins équivalentes à des problèmes de complexité standards ([KOIRAN xxxx], [FOURNIER & KOIRAN xxxx]), ayant la réputation d'être insolubles.

Et même cette réponse artificielle, je suis incapable de la donner entièrement. Je vais seulement construire ici M satisfaisant :

$$(\forall\, x) \qquad (\exists\, \underline{y})\ \varphi(x,\underline{y}) \ \leftrightarrow\ V(\tau_\varphi(x))$$

pour les formules existentielles où il ne reste *qu'une seule* variable libre x . Cela vient de ce que j'ai besoin d'un contexte modèle théorique facilement maîtrisable, et que je n'ai pas trouvé mieux que de me limiter à des fonctions unaires, qui ne permettent pas la formation de termes en plusieurs variables. Et c'est déjà assez compliqué comme ça !

Nous examinerons en conclusion quelques propriétés algorithmiques de la structure construite, et nous explorerons une voie qui pourrait mener à la découverte d'une autre structure plus satisfaisante. Pour l'instant, place à la construction !

Section 1. Deux successeurs.

Nous adoptons un langage L comprenant trois symboles de fonctions unaires s_0 , s_1 , p , et un prédicat unaire V . Les deux premières fonctions sont appelées successeurs, et la troisième prédécesseur. Fixons-en dès à présent l'interprétation.

Le modèle de base S est l'algèbre libre, dans le langage de deux fonctions unaires, engendrée par un élément r qu'on appellera sa racine ; les éléments de S s'écrivent sous la forme $\theta(r)$, où θ est un terme en s_0 , s_1 , $\theta = s_{\varepsilon 1} \circ s_{\varepsilon 2} \circ \ldots \circ s_{\varepsilon m}$. La longueur m du terme θ sera aussi aussi appelée niveau de l'élément $\theta(r)$, la racine r étant de niveau nul. Si $\theta(r) = \theta'(r)$, c'est que les deux termes θ et θ' sont identiques. S a la forme d'un arbre binaire infini, composé de sa racine r , de ses deux successeurs $s_0(r)$ et $s_1(r)$, puis des quatre successeurs d'iceux $s_0(s_0(r))$, $s_0(s_1(r))$, $s_1(s_0(r))$ et $s_1(s_0(r))$, et ainsi de suite.

Le prédécesseur est l'inverse des successeurs, c'est-à-dire que $p(r) = r$, $p(s_0(x)) = p(s_1(x)) = x$.

Nous appelons bloc B une copie de S muni d'un quelconque prédicat unaire V_B. Nous considérons la classe C des structures M, de langage L, formées de la juxtaposition de blocs qui se répètent chacun une infinité de fois : si B figure dans M, il s'y trouve aussi une infinité de blocs isomorphes à B.

Les structures de C éliminent les quanteurs : c'est très facile à voir par des méthodes modèle-théoriques impliquant la compacité. Mais nous, étant dans l'obligation de contrôler la taille des formules éliminantes, n'avons d'autre choix que de procéder péniblement à cette élimination.

Si n est un entier, et x un élément d'un bloc, nous appelons triangle de hauteur n et de racine x la formule caractérisant à l'isomorphie près le morceau du bloc composé des éléments qui sont situés au plus n niveaux au-dessus de x ; c'est la conjonction des formules satisfaites par x, de la forme $p(x) = x$ ou $p(x) \neq x$, $V(x)$ ou $\neg V(x)$, et $V(y)$ ou $\neg V(y)$ pour chaque $y = s_{\varepsilon 1} \circ s_{\varepsilon 2} \circ \ldots \circ s_{\varepsilon m}(x)$, $m \leq n$. Le n-voisinage de x, c'est le triangle de racine $p^n(x)$ et de hauteur $2n$.

Lemme d'élimination. *Considérons* M *dans* C*, une formule existentielle* $(\exists\, \underline{y})\, \varphi(x,\underline{y})$*, et l'entier* n *valant quatre fois le nombre total d'occurrences dans* φ *des symboles* s_0, s_1 *et* p*. Alors, pour déterminer si un élément* x *de* M *satisfait cette formule, il suffit de connaître le* n*-voisinage de* x*, et de savoir quels sont les* n*-triangles réalisés dans* M.

Nous appelerons *portée* de la formule φ ce nombre n figurant dans l'énoncé du lemme.

Démonstration. Nous commençons par mettre φ sous forme disjonctive, ce qui fait exploser sa taille, mais pas celles des constituants : φ s'écrit comme une disjonction exponentielle de conjonctions où figurent exactement une fois, sous forme positive ou négative, chacune des formules atomiques de φ : dans chaque conjonction figure le même nombre ω de symboles de fonction que dans φ. Comme le quanteur existentiel saute au-dessus de la disjonction, il suffit de traiter chacune de ces conjonctions $\varphi_i(x,\underline{y})$.

Les constituants de $\varphi_i(x,\underline{y})$ sont de la forme $V(\theta(u))$, $\neg V(\theta(v))$, $\theta'(u) = \theta''(v)$, $\theta'(u) \neq \theta''(v)$, où u et v désignent des variables, distinctes ou confondues, prises dans le uple $x^\frown\underline{y}$, et où θ, θ', θ'' désignent des termes du langage L, en s_0, s_1 et p.

Nous dirons qu'une variable v prise dans $\underline{y}$ est à distance 1 de x si il y a dans φ_i une équation $\theta'(v) = \theta''(x)$; qu'elle est à distance 2 de x si elle n'en est pas à distance 1, mais figure dans une équation $\theta'(v) =$

$\theta''(u)$, où u est à distance 1 de x ; et plus généralement, on dit qu'une variable v est voisine de x si elle est reliée à cette dernière par une chaîne d'équations figurant dans φ_i , sa distance à x étant la longueur minimale d'une telle chaîne.

Pour chaque voisine de x , nous faisons le choix d'une équation de φ_i qui la relie à x , ou bien à une variable strictement plus proche de x . Nous notons E la conjonction des équations choisies.

Pour éliminer, on procède ainsi : on écrit chaque équation $\theta'(v) = \theta''(u)$ de E , dont v est la variable la plus lointaine, sous la forme $v = \theta(u)$; cela peut obliger à distinguer un nombre exponentiel de cas, s'exprimant par des conditions sur u , ce qui a pour influence d'enfler encore la disjonction, et d'ajouter quelques conditions à chacune des conjonctions . Par exemple, si l'équation est $p(v) = u$, il faut distinguer les trois cas $u = p(u) = v$, $v = s_0(u)$, $v = s_1(u)$. Ce qui importe, c'est que la taille de E n'est pas affectée au cours du procès, et que dans les conditions ajoutées les termes n'ont pas une longueur supérieure à ω .

Quand on a fini, le système E permet d'exprimer chaque voisine v de x comme un terme en une variable plus proche ; comme le nombre total de symboles de fonction de E est majoré par ω , v se met par composition sous la forme $v = \eta(x)$ où le terme η est de longueur au plus ω . On substitue alors partout les variables voisines de x par leur valeur, ce qui ne produit que des termes de longueur majorée par 2ω .

Quand on a ainsi éliminé les voisines de x , chaque système $\psi(x,\underline{y})$ obtenu est formé de trois parties :

- une conjonction $\psi_0(x)$ de conditions portant sur des termes en x de longueur inférieure à 2ω ;

- une conjonction $\psi_1(x,\underline{y})$ d'inéquations $\theta'(x) \neq \theta''(y)$;

- une conjonction $\psi_2(\underline{y})$ de conditions portant sur des variables qui n'ont pas été touchées par l'élimination, et qui étaient toutes présentes à l'origine : dans ψ_2 , il n'y a pas plus de ω occurrences de fonctions.

De plus, l'hypothèse de répétition des blocs nous permet de jeter ψ_1 ; en effet, $(\exists\, \underline{y})\ \psi(x,\underline{y})$ équivaut à $\psi_0(x) \wedge (\exists\, \underline{y})\ \psi_2(\underline{y})$, puisque s'il existe des $\underline{y}$ dans M satisfaisant ψ_2 , il en existe dont aucun élément ne soit dans le bloc de x .

On choisit alors une variable dans $\underline{y}$ par rapport à laquelle on applique le même procédé. A la fin on obtient une décomposition de la formule originelle en une monstrueuse disjonction de formules du type

$\varphi_0(x) \wedge (\exists\, y_1)\, \varphi_1(y_1) \wedge .... \wedge (\exists\, y_m)\, \varphi_m(y_m)$; comme tous les termes qui y figurent ont leur longueur majorée par 2ω , leur satisfaction ne dépend que de ce qui est indiqué dans l'énoncé du lemme. **Fin**

Remarques. 1. Ce n'est pas ici qu'il est essentiel que x soit un élément et non un uple ; l'élimination fonctionne aussi bien pour un uple.
2. Ehud Hrushovski m'a suggéré d'employer des fonctions-successeurs dans ce contexte, et Farzad Didehvar a contribué à l'élimination.

Section 2. Le prédicat de vérité.

Nous choisissons maintenant le terme τ_φ . Les habitants de la ville de Sour, au sud du Liban, ont inventé il y a quelques 35 siècles un système très pratique de représentation graphique de la parole comportant seulement 22 signes, qu'ils ont appelé alphabet, et qu'un citoyen américain a réduit bien plus tard à deux symboles. Cela nous permet d'écrire une formule φ comme un mot binaire $(\varepsilon_1, ... \varepsilon_m)$, $\varepsilon_i = 0$ ou 1 . Nous demandons que la longueur de ce mot vaille au moins 72 , et soit aussi supérieure ou égale à $24n$, où $n = 4\omega$ est l'entier qui a été appelé portée de φ , et qui figure dans le lemme d'élimination. Vu la présence des parenthèses, la traduction en Morse et la corrélation entre la portée et le nombre de symboles de la formule, ça ne demande pas beaucoup de bourrage !

Par ailleurs, pour nous faciliter la vie, nous demandons aussi qu'il y ait un signal de début de formule : toute formule commence par 00 , et on n'y trouve ensuite plus jamais deux 0 consécutifs. J'aurais pu simplifier légèrement la présentation en introduisant un troisième successeur, spécialisé dans le rôle de caractère de début de formule : comme ce n'était pas décisif, j'en ai fait l'économie.

Cela étant convenu, à $\varphi = (\varepsilon_1, ... \varepsilon_m)$ nous associons tout simplement le terme $\tau_\varphi = s_{\varepsilon_m} \circ ... \circ s_{\varepsilon_1}$.

Nous dirons qu'un élément y d'une structure dans C est contraint s'il est de la forme $y = \tau_\varphi(x)$, où φ est une formule, et qu'il est libre sinon. Si y est contraint, il ne l'est que d'une seule manière : si $\tau_\varphi(x) = \tau_\psi(x')$, alors $\varphi = \psi$, et $x = x'$, puisque, pour trouver la formule qui contraint éventuellement y , on en remonte les prédécesseurs jusqu'à ce qu'on trouve deux 0 consécutifs. Ce n'est d'ailleurs pas là que le signal de début s'impose, puisque les parenthèses, garantes de la non ambiguïté de lecture, interdisent à une formule d'être segment final d'une autre. Son utilité véritable, c'est de forcer les formules qui passent par y , à

l'exception de celles qui concernent son prédécesseur (dans le cas où $y = s_0(p(y))$), de concerner toutes un même élément.

On voit la nécessité d'au moins deux successeurs, non seulement pour avoir une longueur raisonnable pour le terme τ_φ , mais aussi pour la non ambiguïté de la lecture des formules : il est clair que des confusions du genre $\tau_\varphi(x) = \tau_\psi(x')$ handicaperaient lourdement la construction du prédicat de vérité.

Ce prédicat V est construit niveau par niveau à partir du bas. Pour ce qui est des points libres du niveau en construction, on met toutes les possibilités d'appartenance et de non appartenance à V , ce qui conduit à démultiplier ce qui a déjà été construit en plusieurs blocs ; c'est ainsi qu'au début, jusqu'au niveau 71 , il n'y a que des points libres, si bien qu'on trouve au bas des divers blocs des copies de tous les triangles de hauteur 71 possibles.

Par contre, quand il s'agit de placer dans V ou dans $\neg V$ un point de la forme $\tau_\varphi(x)$, où x est au moins $24n$ niveaux en dessous, nous devons faire un pari sur l'avenir, c'est-à-dire anticiper de la satisfaction, ou de la non-satisfaction, de la formule $(\exists\, \underline{y})\ \varphi(x,\underline{y})$ par l'élément x dans la structure en cours de construction. Nous connaissons le n-voisinage de x , obtenu aux niveaux antérieurs, mais l'information qui nous manque, c'est de savoir quel sont les n-triangles qu'on trouvera dans la structure quand la construction en sera achevée. *Nous faisons comme si nous les avions tous déjà obtenus,* plus précisément, nous faisons l'hypothèse qu'il n'y aura pas d'autres n-triangles présents dans cette structure que ceux qui y ont été obtenus en dessous du niveau $24n$ (pour $n \geq 1$; les formules de portée nulle ne posent pas de problèmes, car les quatre triangles possibles de hauteur 0 sont tous obtenus aux niveaux 0 ou 1). D'après le lemme d'élimination, cette hypothèse nous permet de définir une stratégie pour déterminer la satisfaction de $(\exists\, \underline{y})\ \varphi(x,\underline{y})$: si c'est oui, on met $\tau_\varphi(x)$ dans V , et sinon en dehors. Quand on a fini, on démultiplie tous les blocs de la structure obtenue.

Section 3. L'art de faire des hypothèses.

C'est bien beau de faire des hypothèses, à condition qu'elles se vérifient. Pour que la vérité devinée corresponde à la vérité vraie, nous devons montrer que le modèle M obtenu satisfait à l'hypothèse qui a été faite lors de sa construction, à savoir que tout triangle de hauteur $n \geq 1$ qui s'y trouve y a une copie isomorphe située toute entière en-dessous du niveau $24n$. C'est une conséquence du lemme suivant, puisque $6n + n < 24n < 24(n+1)$.

Lemme de descente. *Soit* x *dans* M *de niveau strictement supérieur à* 6n *; alors ils existe* x' *dans* M *, de niveau au plus* 6n *, tel que :*

(i) le triangle de racine x' *, et montant jusqu'au niveau* 24(n+1) - 1 *, est isomorphe à celui de même hauteur et de racine* x

(ii) le chemin obtenu en prenant les prédécesseurs de x' *passe par* $s_1(r)$ *, et non pas par* $s_0(r)$ *, où* r *est la racine du bloc de* x' *; il est semblable au chemin de longueur correspondante partant de* x *(c'est-à-dire que si* $s_1(r) = p^s(x')$ *, c'est le même terme qui exprime* x' *en fonction de* $p^s(x')$ *et* x *en fonction de* $p^s(x)$ *).*

Démonstration. Par induction sur n . Pour $n \leq 2$, on peut remplir comme on veut le triangle au-dessus de $x' = s_1(r)$ tant qu'on reste en dessous du niveau 72 , puisque tous les points y sont libres.

Soit donc $n \geq 3$, et soit m la partie entière de n/3 : $3m \leq n < 3(m+1)$. Comme $1 \leq m < n$, l'hypothèse d'induction s'applique à m .

Soit donc x dans M , de niveau supérieur à 6n . Nous voulons choisir x' , de niveau au plus 6n , tel que le triangle au-dessus de x' se soit rempli lors de la construction de M de la même façon que le triangle au-dessus de x , cela jusqu'à ce qu'on atteigne le niveau 24(n+1) - 1 . Les points libres au-dessus de x' ne nous gêneront pas, puisque nous avons eu la possibilité de les remplir comme on l'a voulu. Quant au points contraints, comme nous restons en dessous du niveau 24(n+1) , ils correspondent à des formules φ de portée au plus n : par la construction même de M , si deux points u et v de M ont des n-voisinages isomorphes, et si le premier satisfait $V(\tau_\varphi(u))$, l'autre aussi, et réciproquement.

Nous devons tenir compte des points contraints au-dessus de x ; certains concernent le prédécesseur de x , ou bien x lui-même, ou encore des éléments au-dessus de x . Si on ne rencontre que ceux-là, il suffira que le n-voisinage de p(x') soit choisi isomorphe à celui de p(x) , celui de x' isomorphe à celui de x , et celui d'un point v' , situé moins de n niveaux au-dessus de x' , isomorphe à celui du point v correspondant situé au-dessus de x ; en effet, comme les points situés plus haut dans les triangles ont leur n-voisinage entièrement situé au-dessus de x et x' , le triangle au-dessus de x' se remplira ensuite de manière isomorphe à celui au-dessus de x .

Mais il est aussi possible que x soit sur le trajet de formules beaucoup plus longues, correspondant toutes à un unique élément u , le premier que l'on rencontre après un double 0 quand on chemine le long des

prédécesseurs de x . Dans ce cas, nous dirons que x a une queue - le chemin qui joint x à u - et il faudra aussi nous soucier de reproduire le n-voisinage de u .

Nous distinguons deux cas :

(i) x n'a pas de queue, ou bien a une queue de longueur strictement supérieure à $n + 6m$. Dans ce cas on choisit y' de niveau au plus $6m$ associé par la $m°$ étape du lemme à $y = p^{n+1}(x)$. Soit x' l'élément correspondant à x . Comme $6m + n + 1 + 2n < 24(m+1)$, $p(x)$ et $p(x')$ ont des n-voisinages isomorphes, ainsi que x et x' , et aussi les points v et v' situés moins de n niveaux au-dessus ; par ailleurs x' n'a pas de queue (on la lui a coupée), de sorte que les points éventuellement contraints par la longue queue de x sont maintenant libres, et qu'on peut les mettre dans V ou dans $\neg V$ ad libitum, si bien qu'il y a une façon de faire qui donne le même triangle au-dessus de x' qu'au-dessus de x , tant qu'on reste en-dessous du niveau $24(n+1)$.

(ii) x a une queue de longueur inférieure ou égale à $n + 6m$. Dans ce cas-là, nous devons reproduire les n-voisinages de x , de $p(x)$, des points v , mais aussi du point u qui est au bout de la queue, et pour cela il nous suffit de reproduire grâce à l'hypothèse de récurrence le triangle de racine $p^n(u)$ et de hauteur $n + 6m + n + 2n$ en-dessous du niveau $24(m+1)$: ça marche parce que $12m + 4n < 24(m+1)$.

Nous avons recopié x en x' à un niveau majoré par $6m + n + 1$ dans le premier cas, $12m + 2n$ dans le second. Il ne reste plus qu'à observer que $2n + 12m \leq 6n$. **Fin**

Remarque. Quand le lapin blanc est sorti du chapeau, on aime bien connaître le truc. Cette démonstration dépend de trois paramètres A , B et C : on demande que la longueur des formules soit au moins An ; on cherche à reproduire les racines des triangles de hauteur n en dessous du niveau Bn ; pour le pas de récurrence, on divise n par C , ce qui demande AC comme longueur minimum de formule. On pose $m = [n/C]$, et on coupe les queues à $n + Bm$; le premier cas demande que $Bm + n + 1 + 2n < A(m+1)$, le second que $2Bm + 4n < A(m+1)$, et la conclusion que $2Bm + 2n \leq Bn$. Cela donne $2B + 4C \leq A$, $2B/C + 2 \leq B$, soit encore, pour $C \geq 2$, $B \geq 2C/C-2$, $A \geq 4C(C-1)/C-2$; le minimum pour A est atteint en $C = 2 + \sqrt{2}$, et vaut $12 + 8\sqrt{2}$. J'ai choisi les paramètres $C = 3$, $B = 6$, $A = 24$, valeur très proche du minimum ; d'autres choix raisonnables sont $C = 4$, $B = 4$, $A = 24$, ou $C = 6$, $B = 3$, $A = 30$. Les grandes valeurs de C - et de A ! - font tendre au contraire B vers sa borne inférieure.

Conclusion.

Si on ajoute au langage de M une constante r nommant une racine ($p(r) = r$), on obtient une structure qui a un algorithme de décision rapide pour ses énoncés existentiels, puisque $(\exists \underline{y})\ \varphi(r,\underline{y})$ est vrai si et seulement si $V(\tau_\varphi(r))$ l'est : pour voir si l'énoncé est vrai, on calcule $\tau_\varphi(r)$ et on teste s'il satisfait V ! C'est peut-être pas bien convaincant comme exemple, puiqu'il s'agit d'un "algorithme" au sens de la structure M , où, suivant les conventions de [BCSS 1998] ou [POIZAT 1995], ou plus généralement de la Complexité Algébrique, on considère par définition comme algorithmique l'évaluation d'une formule (et même d'un circuit) libre de quanteurs. Je ne crois pas beaucoup améliorer mon cas en faisant observer que le problème analogue pour la structure $\{0,1\}$ est NP-complet.

Cependant, on pourra poser $0 = s_0(r)$ et $1 = s_1(r)$ pour respecter les conventions de [POIZAT 1995] ; comme un énoncé n'est rien d'autre qu'un objet syntaxique, la propriété décrite au paragraphe ci-dessus s'interprète ainsi : tout problème booléen qui est NP au sens de M est P au sens de M ! Il est très facile de construire des structures ou n'importe quel problème booléen est P grâce à l'intervention d'un paramètre approprié (voir les "dictionnaires" de [GOODE 1994]), mais dans le cas présent, il n'est nul besoin d'ajouter des paramètres pour transformer un algorithme NP en un algorithme P ayant même effet sur des données booléennes : je ne connais pas de méthode de construction plus simple pour obtenir cette propriété.

On observe aussi que la formule $V(\tau_\varphi(x))$ est horriblement séquentielle, alors que j'ai annoncé dans l'introduction un contrepoint à $NC^1 = P$! Cela vient de ce que, puisqu'il y a des fonctions, il n'est plus possible de paralléliser les formules, si bien que la connection entre "formules écrites sous forme traditionnelle" et "circuits de profondeur logarithmique" est perdue !

Je conjecture que notre modèle M n'élimine plus rapidement dès qu'il reste deux variables libres : la candidate à une élimination difficile, c'est la formule existentielle en les deux variables libres x et x' qui exprime le fait qu'il existe ε_1 , ... ε_n tels que soient vérifiées $V(s_{\varepsilon_1}. \ldots s_{\varepsilon_n}(x))$ et $\neg V(s_{\varepsilon_1}. \ldots s_{\varepsilon_n}(x'))$; elle s'écrit :

$(\exists\ y_1) \ldots (\exists\ y_n)\ (\exists\ z_1) \ldots (\exists\ z_n)$
$((y_1 = s_0(x) \wedge z_1 = s_0(x')) \vee (y_1 = s_1(x) \wedge z_1 = s_1(x'))) \ \wedge\ \ldots\ \wedge$
$((y_{i+1} = s_0(y_i) \wedge z_{i+1} = s_0(z_i)) \vee (y_{i+1} = s_1(y_i) \wedge z_{i+1} = s_1(z_i))) \ \wedge\ \ldots\ \wedge$
$V(y_n) \ \wedge\ \neg V(z_n)$.

Il est probable qu'on ne puisse décider de la véracité de cette formule sans faire dans les pires cas beaucoup de tests, c'est-à-dire, selon les mots de [KOIRAN-FOURNIER xxxx], qu'il n'y ait pas d'arbres de décision de profondeur polynomiale pour les problèmes NP au sens de M . Je pense donc que M satisfait $P \neq NP$, et même $P \neq NBP$. Pour en avoir le coeur net, il faudrait examiner le problème suivant : soit B_0 un bloc de M dont tous les points libres sont hors de V ; si je perturbe sa construction en mettant un certain point libre dans V , quels sont les points contraints qui vont être touchés ? combien y en aura-t'il à un niveau donné ?

Comme nous l'avons dit, la faiblesse de la méthode vient de ce qu'elle n'utilise que des fonctions unaires. Pour éliminer rapidement les quanteurs sans restriction sur $\underline{x}$, la tentation est de coller un prédicat de vérité sur la fonction-couple de la Théorie des Ensembles, qui n'est autre qu'une algèbre libre, à une infinité de générateurs, dans le langage d'une fonction binaire (c'est un exemple d'algèbre de Malcev). C'est cette méthode qui a donné facilement dans [POIZAT 1995] p. 188 un exemple de structure où P = NBP ; mais ça risque d'être beaucoup plus dur quand on est en présence de quanteurs véritables, car si la théorie de l'algèbre de Malcev elle-même est facilement contrôlable ([BOUSCAREN-POIZAT 1988]), elle devient sauvage quand on lui ajoute un prédicat unaire arbitraire, puisque l'arité unaire n'est pas plus simple qu'une autre en présence d'une bijection entre l'univers et son carré cartésien !

Ce dont nous avons profité, c'est que la théorie des modèles pour deux successeurs est non seulement absolument triviale, mais qu'elle le reste si on introduit des prédicats unaires à peu près arbitraires. C'est ça qui nous a permis de prévoir le comportement final de l'objet que nous étions en train de construire. Dans cette nouvelle situation infiniment plus complexe, il nous faudrait une classe de prédicats qui restent sous contrôle quand on les plaque sur la fonction-couple, tout en laissant assez de liberté de choix pour pouvoir représenter la vérité. Ça n'a pas l'air si simple, et je n'ai pas de candidat ; d'un autre côté, je ne vois aucune raison métaphysique qui interdise l'existence d'une telle structure, et je suis étonné de n'avoir jamais rencontré dans mes lectures de tentative d'en fabriquer une : vu la célébrité des résultats de Gödel et de Tarski, affirmant l'impossibilité de définir les énoncés prouvables ou les énoncés vrais, il est surprenant de constater que personne n'ait voulu, par la construction de contre-exemples, en délimiter précisément le cadre de validité.

L'idéal, ça serait d'exhiber $\tau_\varphi(\underline{x})$ de profondeur logarithmique, et de construire V tel que $(\exists\, \underline{y})\ \varphi(\underline{x},\underline{y})$ équivaille à $V(\tau_\varphi(\underline{x}))$.

On pourrait aussi chercher à éliminer en suivant de tout autres principes, par exemple en déterminant un $\underline{y}$, s'il en existe, parmi ceux qui satisfont $\varphi(\underline{x},\underline{y})$ (à ce propos, la question 1 au bas de la p. 181 de [POIZAT 1995] est complètement stupide, puisque dans un corps fini toute fonction est polynôme ; comme le montre le lemme d'élimination, on a également des fonctions de Skolem décrites par des termes dans le langage de un ou plusieurs successeurs, augmenté du sélecteur et de la fonction caractéristique de l'égalité ; mais le problème est celui de la complexité du uple de termes associé à la formule φ), ou bien en décrivant une bonne raison expliquant si oui ou non il y a un $\underline{y}$ qui satisfait $\varphi(\underline{x},\underline{y})$. Ce serait quelque chose qui ressemblerait d'avantage à l'idée qu'on se fait d'un algorithme d'élimination que la simple consultation d'un oracle miraculeux. Il est aussi possible que ce que j'ai fait ici soit naïvement compliqué, et que certaines structures très simples et bien connues éliminent de toute évidence et à une vitesse foudroyante ; mais j'en doute.

Références.

[BCSS 1998] Lenore BLUM, Felipe CUCKER, Michael SHUB & Steve SMALE, **Complexity and real computations,** Springer.

[BOUSCAREN-POIZAT 1988] Elisabeth BOUSCAREN & Bruno POIZAT, Des belles paires aux beaux uples, **the Journal of Symbolic Logic,** vol. 53, p. 434-442.

[FOURNIER & KOIRAN xxxx] Hervé FOURNIER & Pascal KOIRAN, Lower bounds are not easier for the Reals, prépublications LIP-ENSLyon, n° 1999-21.

[GOODE 1994] John B. GOODE, Accessible telephone directories, **the Journal of Symbolic Logic,** vol. 59, p. 92-105.

[KOIRAN xxxx] Pascal KOIRAN, Transfer theorems via sign conditions, prépublications LIP-ENSLyon, n° 2000-13.

[POIZAT 1985] Bruno POIZAT, **Cours de Théorie des Modèles,** Nur al-Mantiq wal-Ma'rifah ; version anglaise par Moses KLEIN, **A Course in Model Theory,** Springer, 2000.

[POIZAT 1995] Bruno POIZAT, **Les petits cailloux,** Nur al-Mantiq wal-Ma'rifah ; Aléas Editeur, 15 quai Lassagne, 69001 Lyon, France.

Bruno POIZAT

Institut Girard Desargues,
Mathématiques, bâtiment 101,
Université Claude Bernard (Lyon-1),
43, boulevard du 11 novembre 1918,
69622 Villeurbanne cedex, France

poizat@desargues.univ-lyon1.fr