REVUE FRANÇAISE D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE. SÉRIE ROUGE

G. WERNER

Quelques remarques sur la complexité des algorithmes

Revue française d'informatique et de recherche opérationnelle. Série rouge, tome 4, nº R2 (1970), p. 33-50

<http://www.numdam.org/item?id=M2AN_1970__4_2_33_0>

R.I.R.O.
(4e année, R-2, 1970, p. 33-50)

QUELQUES REMARQUES SUR LA COMPLEXITE DES ALGORITHMES

par G. WERNER (1)

Résumé. — *Le problème de l'estimation de la complexité pour un algorithme d'une certaine classe est défini indépendamment d'un type particulier de machines. On montre l'existence de certaines familles de fonctions récursives qui croissent trop vite pour être calculables par un algorithme dont la complexité est inférieure à une certaine borne. Ces fonctions sont utilisées pour la construction d'une infinité de classes récursivement énumérables de fonctions récursives. Celles-ci peuvent être définies par une classe d'algorithmes de complexité bornée pour lesquels le problème de l'estimation est soluble.*

INTRODUCTION

La notion de complexité ou de coût des calculs, c'est-à-dire des suites de configurations internes d'une certaine classe d'automates, peut servir à diverses restrictions de la calculabilité correspondante et par là à des classifications d'algorithmes et de fonctions [1-4].

M. O. Rabin [5] a montré que les « degrés de complexité » de systèmes canoniques de Post forment un ordre partiel; cette classification peut être étendue aux fonctions calculables par de tels systèmes. Si l'ensemble des semi-fonctions récursives $\{ \varphi_i \}$ est donné par une gödelisation acceptable (2), alors, la définition de la complexité d'une classe de fonctions calculables peut être rendue indépendante d'une classe particulière de machines par une définition axiomatique de l'ensemble des « fonctions de complexité » $\{ \Phi_i \}$ correspondantes [6].

M. Blum montre l'existence de fonctions dont tout algorithme peut être remplacé par un autre algorithme pour la même fonction et moins complexe que le précédent. D'autre part, il montre qu'il existe une fonction h et une autre fonction τ telles que pour toute classe récursivement énumérable de

(1) Institut de Mathématiques Appliquées de l'Université de Grenoble.
(2) C'est-à-dire par une énumération effective qui satisfait au théorème de l'itération ou théorème S^m_n.

fonctions de complexité $\{ \Phi_i \}, \Phi_i < \Phi_{\tau(i)} < h[x, \Phi_i(x)]$ pour tout i et pour presque tout x.

Nous sommes ici concernés par une sorte de problème inverse, c'est-à-dire estimer, pour un algorithme donné, sa complexité.

Pour certaines classes de fonctions de complexité $\{ \Phi_i \}$ on montre qu'il existe des fonctions qui « croissent trop vite », pour être calculables par un algorithme de complexité inférieure à $h[x, \Phi_i(x)]$; sous certaines conditions h est une telle fonction.

Or, la question de décider si un algorithme donné calcule une fonction bornée par celle calculée par un autre algorithme est indécidable pour la classe des semi-fonctions $\{ \varphi_i \}$; ainsi la fonction caractéristique de l'ensemble $B = \{ \langle x, y \rangle \mid \varphi_x \leqslant \varphi_y \Leftrightarrow (\forall u)(\exists z)[u < z$ et $[[\varphi_x(z)$ et $\varphi_y(z)$ définis et $\varphi_x(z) \leqslant \varphi_y(z)]$ ou $[\varphi_x(z) = \varphi_y(z) =$ indéfini$]]] \}$ n'est pas récursive. En effet, ce n'est qu'un cas particulier d'un théorème de Rice qui dit que toute propriété qui définit une dichotomie non-triviale de la classe des semi-fonctions récursives est indécidable.

Il est par contre possible de trouver des classes restreintes d'algorithmes pour lesquelles le problème suivant a une solution : soit A et B des classes d'algorithmes, existe-t-il une fonction récursive Ψ telle que

$$(*) \quad (\forall x) [[x \in A \cap B \rightarrow \Psi(x) = 1] \,\&\, [x \in A \cap \bar{B} \rightarrow \Psi(x) = 0]]?$$

Soit par exemple $\{f_k(x)\}$ la classe de fonctions définies par $f_0(x) = x$, $f_{k+1}(x) = 2^{f_k(x)}$; un indice peut être trouvé uniformément pour tout f_k ; soit A la classe de ces indices et soit $B = \{ \langle x, y \rangle \mid f_x < f_y \}$, le problème B est décidable relatif à la classe A comme on prouve facilement par induction sur la construction de A.

Le problème d'estimer la complexité d'un algorithme d'une certaine classe C revient à trouver une classe A et un procédé effectif K qui fait correspondre à tout élément x de C un élément $K(x)$ de A tel que :

$$(**) \begin{cases} \text{(i)} \ (\forall x \in \mathsf{C})(\exists y \in A)(\forall z \geqslant z_0)\Phi_x(z) \leqslant f_y(z) \\ \text{(ii)} \ (\forall x \in \mathsf{C})(\forall z \geqslant z_0)\Phi_x(z) \leqslant f_{K(x)}(z) \\ \text{(iii)} \ B = \{ \langle x, y \rangle \mid f_x < f_y \} \text{ est décidable relatif à } A. \end{cases}$$

On montre dans le paragraphe 1 que le problème (*) est indécidable pour un certain nombre de classes d'algorithmes ; le problème de l'estimation par contre s'avère soluble pour certaines de ces classes. Ceci pose les problèmes suivants :

(1) Quelles sont les classes A, C d'algorithmes telles que :

$A \subset \mathsf{C}$, $(\forall x \in \mathsf{C})(\exists y \in A)\varphi_x \leqslant f_y$ et (*) est décidable pour A?

(2) Quelles sont les classes C pour lesquelles il existe un K récursif tel que :

$$(\forall x) x \in \mathsf{C} \text{ et } K(x) \in A \Rightarrow \Phi_x \leqslant f_{K(x)}?$$

Soit $\lambda yxf_y(x) = \lambda xyh(x, y)$ une fonction récursive. Si le problème de l'estimation s'avère insoluble pour la classe de tous les algorithmes de complexité bornée $\{ i \mid (\exists y)(\forall x \geqslant x_0)\Phi_i(x) \leqslant f_y(x) \}$, la classe de fonctions récursives totales correspondantes

$$F_h = \{f \mid (\exists i)(\exists y)(\forall x \geqslant x_0)[f(x) = \varphi_i(x) \,\&\, \Phi_i(x) \leqslant h(x, y)] \}$$

est-elle une classe récursivement énumérable, c'est-à-dire existe-t-il une fonction qui énumère au moins un indice pour toute fonction dans F_h ?

On montre que si $\lambda yxh(x, y)$ a certaines propriétés alors il existe une fonction universelle $\lambda ix\varphi_{i_0}(< i, x >)$ pour la séquence de fonctions $\lambda xf_i(x)$, $i = 0, 1, 2, \ldots$ telle que $\lambda ixU((i)_0, x, \varphi_{i_0}(< (i)_1, x >))$ est une fonction d'énumération pour F_h qui appartient à une hiérarchie de classes restreintes d'algorithmes pour F_h pour lesquelles le problème de l'estimation (**) a une solution.

(3) Quelle est la complexité de $\lambda ixU(i, x)$?

M. Blum [7] a montré que la complexité de la fonction

$$g(x) = \begin{cases} 1 \text{ si } (\exists y < h(x))[T(x, x, y)] \\ 0 \text{ sinon} \end{cases}$$

est plus grande que $h(x)$; de combien?

Dans ce qui suit on essaie de donner une réponse partielle à ces questions. On verra que les problèmes (1) et (2) ont une solution pour une classe importante d'algorithmes. Le problème (3) ou problème de complexité d'une fonction universelle pourrait être appelé problème de complexité d'un compilateur (interprétatif) pour une classe d'algorithmes de complexité bornée ; on verra qu'elle peut être estimée et que dans un certain sens on n'ajoute que très peu à la complexité de $\varphi_i(x)$ si $\varphi_i(x)$ est suffisamment complexe.

1. L'INDECIDABILITE DU PROBLEME (*) POUR CERTAINES CLASSES D'ALGORITHMES

On prouve d'abord que le problème (*) est indécidable relatif à la classe des algorithmes des fonctions récursives totales $A = \{ x \mid \varphi_x \text{ total} \}$, puis on étend ce résultat à une certaine classe d'algorithmes de fonctions récursives primitives et récursives élémentaires. Un résultat similaire sera prouvé pour la classe des algorithmes obtenus par des schémas de définitions récursifs primitifs ou élémentaires par une méthode différente qui donne un corollaire intéressant.

Définition 1.1. Soit M_x la machine numéro x dans une liste de toutes les machines (ou programmes) d'un certain type M; soit $\varphi_x^{(k)}$ la semi-fonction récursive calculée par M_x : $M_x(x_1, \ldots, x_k)$ s'arrête avec $\varphi_x^{(k)}$ dans un registre désigné ou $M_x(x_1, ..., x_k)$ ne s'arrête pas et $\varphi_x^{(k)}$ est indéfini. On appelle x (le code pour) un algorithme ou indice de φ_x.

On a la proposition suivante :

P.1.2. Le problème (*) est indécidable relatif à $A = \{ x \mid \varphi_x \text{ total} \}$. Soit

$$\Psi_z^{(1)} = \varphi_{f(z)}(y) = \text{def} \begin{cases} 1 \text{ si } \varphi_z(z) \text{ converge en } \leqslant y \text{ pas de } M_z \\ 0 \text{ si } \varphi_z(z) \text{ ne converge pas en } \leqslant y \text{ pas de } M_z. \end{cases}$$

Pour tout z, $\varphi_{f(z)}(y)$ est une fonction totale associée à z qui peut être effectivement trouvée donné z;

$$\text{on a } \varphi_{f(z)}(y) = \begin{cases} 1 \text{ si } y \geqslant y_0 \text{ et } \varphi_z(z) \text{ converge en } y_0 \text{ pas} \\ 0 \text{ partout si } \varphi_z(z) \text{ ne converge pas.} \end{cases}$$

Soit g une fonction caractéristique récursive pour

$$\{ \langle x, y \rangle \mid \varphi_x < \varphi_y; x, y \in A \}$$

et soit x_0 un indice pour la fonction constante λz [1]; alors si g était récursive :

$g(f(x), x_0) = 1$ si $\varphi_{f(x)}(z) = 0$ pour tout z et $\varphi_{x_0} = 1 \Leftrightarrow \varphi_x(x)$ ne converge pas
$g(f(x), x_0) = 0$ si $\varphi_{f(x)}(z) = 1$ pour tout $z \geqslant z_0$ et $\varphi_{x_0} = 1 \Leftrightarrow \varphi_x(x)$ converge,
g serait fonction caractéristique récursive pour le problème de l'arrêt, ce qui est impossible.

On considère maintenant des classes d'algorithmes « de complexité bornée »:

P.1.3. Le problème (*) demeure indécidable relatif à $A = \{ x \mid$ il existe une fonction élémentaire h telle que $(\forall y)\varphi_x(y)$ converge en $\leqslant h(y)$ pas $\}$.

Démonstration comme ci-dessus :

$$\Psi_z = \varphi_{f(z)}(y) = \begin{cases} 1 \text{ si } (\exists u \leqslant y)[T(z, z, u)] \\ 0 \text{ si } (\forall u \leqslant y)[\neg T(z, z, u)] \end{cases}$$

$(\exists u \leqslant y)[T(z, z, u)]$ est le prédicat : « $\varphi_z(z)$ converge en $\leqslant y$ pas ».

On suppose [4], [8] qu'il existe une gödelisation de M telle que $T(x, y, z)$ est un prédicat récursif élémentaire et cette classe de prédicats est close par rapport à la quantification bornée, donc Ψ_z est une fonction élémentaire. Par conséquent f peut être choisi de façon que $M_{f(z)}$ s'arrête en $h(y)$ pas au plus, h élémentaire, car A contient un indice pour toute fonction récursive élémentaire.

Définition 1.4. On dira que M_x est récursif primitif ou $x \in P \Leftrightarrow P$ est un ensemble récursif d'indices tel que $\pi : P \rightarrow F_{PR}$ est une numération de Gödel des fonctions récursives primitives équivalente à la numération standard de F_{PR}.

M_x récursif élémentaire ou $x \in E$ est défini de la même façon.

Définition 1.5. Soit $f_0(x), f_1(x), \ldots, f_k(x), \ldots, \max\limits_{k \leqslant x} \{ f_k(x) \}$ une suite de fonctions telle que pour tout $k \geqslant 0$:

(1) $(\forall\, n)(\forall\, x \geqslant x_0) f_k^n(x) < f_{k+1}(x)$; récursif primitif (élémentaire pour $k = 0, 1$);

(2) $f_k(x)$ est monotone croissant et non-borné : $f_k(x+1) > f(x) \geqslant x$;

(3) toute fonction récursive primitive est majorée par un $f_k^n(x)$.

P.1.6. L'ensemble d'indices $A_k = \{ x \mid (\exists\, n)(\forall\, z)\varphi_x(z) \leqslant f_k^n(z) \;\&\; x \in P(E) \}$ n'est pas récursif.

On utilise le

Lemme 1.7. Soit

$$B = \{ x \mid x \in P \;\&\; \varphi_x \text{ non-décroissant et } \varphi_x(y+1) \leqslant \varphi_x(y) + 1 \}$$

alors les ensembles

$$C = \{ x \mid x \in B \;\&\; \varphi_x(N) \text{ est fini} \}$$

et

$$D = \{ x \mid x \in B \;\&\; \varphi_x(N) \text{ est infini} \}$$

ne sont pas récursivement énumérables.

Soit $\{ W_i \}$ une énumération standard de tous les ensembles récursivement énumérables, on réduit

$A_1 = \{ x \mid W_x \text{ fini} \}$ à C et $A_2 = \{ x \mid W_x \text{ infini} \}$ à D par une certaine fonction récursive f;

donc : $$x \in A_1 \Leftrightarrow f(x) \in C \text{ et } x \in A_2 \Leftrightarrow f(x) \in D.$$

Construction de f : soit

$$W_x = \operatorname{dom} \varphi_x = \{ y \mid \varphi_x(y) \text{ converge} \Leftrightarrow (\exists\, z) T(x, y, z) \};$$

on pose

$\varphi_{f(x)}(0) = 0$

$$\varphi_{f(x)}(z+1) = \begin{cases} \varphi_{f(x)}(z) & \text{si} \quad (\forall\, y \leqslant z)[T(x, y, z+1) \Rightarrow (\exists\, n \leqslant z) T(x, y, n)] \\ \varphi_{f(x)}(z) + 1 & \text{si} \quad (\exists\, y \leqslant z)[T(x, y, z+1) \;\&\; (\forall\, n \leqslant z) \neg T(x, y, n)]. \end{cases}$$

Avec une gödelisation récursive élémentaire de M $\varphi_{f(x)}$ est élémentaire, donc

$$f(x) \in C \Leftrightarrow (\exists\, v)(\forall\, y)[v < y \Rightarrow (\forall\, u) \neg T(x, y, u)]$$
$$\Leftrightarrow (\exists\, w)(\forall\, z)[w < z \Rightarrow \varphi_{f(x)}(z) = \varphi_{f(x)}(w)] \Leftrightarrow x \in A_1$$

$$f(x) \in D \Leftrightarrow (\forall\, v)(\exists\, y)[v < y \;\&\; (\exists\, u) T(x, y, u)]$$
$$\Leftrightarrow (\forall\, w)(\exists\, z)[w < z \;\&\; \varphi_{f(x)}(z) > \varphi_{f(x)}(w)] \Leftrightarrow x \in A_2$$

Or, A_1 est Σ_2-complet et A_2 est Π_2-complet (1) [10], nous avons donc prouvé le lemme en déterminant le type d'isomorphisme récursif de C et de D.

Démonstration du théorème P.1.6. :

Soit $h_k(x, 0) = x$, $h_k(x, y+1) = f_k(h_k(x, y))$; $h_k(x, y) = f_k^y(x)$, et soit

$$A_1' = \{ x \mid x \in P \ \& \ \varphi_x(N) \text{ fini} \} \text{ et } A_2' = \{ x \mid x \in P \ \& \ \varphi_x(N) \text{ infini} \}$$

des ensembles d'indices de fonctions récursives primitives. A_1 et A_2 sont réductibles à A_1' et A_2' respectivement; il suffit de poser $W_x = \varphi_{g(x)}(N)$ où $g(x)$ est l'indice d'une fonction récursive primitive qui énumère W_x. Par conséquent, A_1' est réductible à C et A_2' à D par un $h(z)$.

Soit, pour tout $z \in P$, $n(z) \in P$ un indice de $\lambda x h_k(x, \varphi_{h(z)}(x))$, alors par définition 1.5.(3)

$$(\forall x)\varphi_{n(z)}(x) \leqslant \lambda x f_k^y(x) \text{ pour un certain } y \Leftrightarrow \varphi_{h(z)} \text{ borné}$$

$$(\forall x \geqslant x_0)\varphi_{n(z)}(x) > \lambda x f_k^y(x) \text{ pour tout } y \Leftrightarrow \varphi_{h(z)} \text{ non-borné.}$$

Corollaire 1.8. L'ensemble d'indices A_k est du type Σ_2($\exists\ \forall$-définissable). La signification profonde de la proposition 1.6. est donnée par le

Corollaire 1.9. Il n'y a pas de fonction récursive $K(z)$ telle que

$$(\forall x)(\forall z \in P)[\varphi_z(x) \leqslant f_k^n(x) \Leftrightarrow f_k^n(x) = \varphi_{K(z)}(x)].$$

Supposons le contraire : soit B_k un ensemble récursif d'indices pour $\{ f_k^n(x), n = 0, 1, \dots \}$: alors A_k serait réductible à B_k par K, donc récursif. Mais A_k n'est pas récursif.

2. UN CAS SOLUBLE D'UN PROBLEME D'ESTIMATION

Dans le paragraphe précédent on a trouvé que le problème (*), c'est-à-dire de savoir si φ_x est borné par φ_y, $x, y \in \mathsf{C}$ n'a pas de solution pour un certain nombre de classes importantes d'algorithmes C. Soit maintenant C une classe d'algorithmes et $\lambda xy\varphi_{i_0}(\langle x, y \rangle)$ une séquence récursivement énumérable de fonctions $f_0(x), f_1(x), \dots$ telles que

(1) $\lambda y\varphi_{i_0}(\langle x, y \rangle) = \lambda y f_x(y)$ pour tout x,

(2) il existe une fonction récursive α qui énumère un ensemble d'indices A, $A \subset \mathsf{C}$ pour $\{f_x(y)\}$ tel que le problème (*) est décidable relatif à A;

(3) $(\forall i)[i \in \mathsf{C} \Rightarrow (\exists k)(\forall x \geqslant x_0)[\varphi_i(x) \leqslant f_k(x)]]$.

(1) Un ensemble E est Σ_2-complet s'il est $\exists\ \forall$-définissable et si tout ensemble $\exists\ \forall$-définissable est réductible à E.

Une telle classe existe; on essaie donc de construire la classe C telle que le problème suivant ait une solution :

Définition 2.1. Le problème de l'estimation est soluble pour la classe d'algorithmes $\mathsf{C} \Rightarrow$ il existe $\lambda xy\varphi_{i0}(\langle x, y \rangle)$ avec les propriétés (1) à (3) et une fonction récursive $h : \mathsf{C} \to A$ telle que

$$(\forall f)[f = \varphi_i(x) \ \& \ i \in \mathsf{C} \Rightarrow (\forall x \geqslant x_{0f})\varphi_i(x) \leqslant \varphi_{h(i)}(x)].$$

On montre d'abord que E, la classe de définitions de fonctions élémentaires qui est close par rapport aux schémas de la substitution simultanée, de la somme finie et du produit fini, est une classe pour laquelle ce problème de l'estimation a une solution :

P.2.2. Le problème de l'estimation de la définition 2.1 est soluble pour la classe E de définitions (équationnelles) de fonctions.

Lemme 2.3. (Propriétés de clôture de la classe E) :

E est close par rapport à :

(i) la minimisation bornée,

(ii) la récursion primitive bornée,

(iii) $\mathrm{Rel}(E) = \{ R \mid (\exists f \in E)R(\bar{x}) \Leftrightarrow f(\bar{x}) = 0 \}$ est clos par rapport à la quantification bornée et par rapport aux opérations de la logique propositionnelle.

(i) On utilise les indices récursifs élémentaires, similaires aux indices de Kleene [9], pour la caractérisation de la classe E des dérivations de fonctions récursives élémentaires. Soit donc E la plus petite classe de dérivations qui contient

(1)
$$
\begin{array}{lll}
U_i^n = \lambda x_1 \ldots x_n[x_i] & (1 \leqslant i \leqslant n) \text{ codifié} & \langle 1, i, n \rangle \\
C_i^n = \lambda x_1 \ldots x_n[i],\ i, n \geqslant 0 & & \langle 2, i, n \rangle \\
f_1 = \lambda xy[x + y] & & \langle 3, 2 \rangle \\
f_2 = \lambda xy[x \dot{-} y] & & \langle 4, 2 \rangle
\end{array}
$$

(2) et qui est close par rapport à la substitution simultanée

$$f^{(n)}(\bar{x}) = g^{(m)}(g_1^{(n)}(\bar{x}), \ldots, g_m^{(n)}(\bar{x})) \qquad \langle 5, n, \underline{g}, \underline{g_1}, \ldots, \underline{g_m} \rangle$$

(3) et par rapport à la somme finie et au produit fini

$$f^{(n+1)}(\bar{x}, y) = \sum_{i<y} g(\bar{x}, i) \qquad \langle 6, n+1, \underline{g} \rangle$$

$$f^{(n+1)}(\bar{x}, y) = \prod_{i<y} g(\bar{x}, i) \qquad \langle 7, n+1, \underline{g} \rangle$$

où $\underline{g}, \underline{g_1}, \ldots, \underline{g_m}$ sont les indices de fonctions $g, g_1, \ldots, g_m$ introduites par un des schémas de (1), (2) ou (3).

On dira que $g^{(n+1)}(\bar{x}, y) \in E$ si $g \in E$.

Soit

$g^{(n+1)}(\bar{x}, y) \in E$ alors $f(\bar{x}, y)$

$$= \left\{ \begin{array}{l} \mu z < [g(\bar{x}, z) = 0] \text{ si } (\exists z < y)[g(\bar{x}, z) = 0] \\ 0 \text{ sinon} \end{array} \right\} \in E$$

car
$$f(\bar{x}, y) = \sum_{t<y} \prod_{z \leqslant t} sg(g(\bar{x}, z))$$

(ii) Soit $f^{(n+1)}(\bar{x}, y)$ défini par

$$f(\bar{x},0) = g(\bar{x}) \qquad g^{(n)}, h^{(n+2)} \text{ et } j^{(n+1)} \in E$$

$$f(\bar{x}, y+1) = h(\bar{x}, y, f(\bar{x}, y))$$

$$f(\bar{x}, y) \leqslant j(\bar{x}, y)$$

avec $\quad k(\bar{x}, y) = \langle j(\bar{x}, 0), \ldots, j(\bar{x}, y) \rangle$

$$f(\bar{x}, y) = (\mu z \leqslant k(\bar{x}, y)[(z)_0 = g(\bar{x}) \wedge (\forall i < y)[(z)_{i+1} = h(\bar{x}, i, (z)_i]])_y$$

où $\lambda x_0 \ldots x_n \langle x_0, \ldots, x_n \rangle = z$ et $\lambda zy(z)_y$, $\langle (z)_0, \ldots, (z)_n \rangle = z$, sont des fonctions de codifications des n-uplets qui peuvent être choisies dans E [8].

(III) $\qquad \neg [f(\bar{x}) = 0] \Leftrightarrow [[1 \dot{-} f(\bar{x})] = 0]$

$$[(f(\bar{x}) = 0) \wedge (g(\bar{x}) = 0)] \Leftrightarrow [[f(\bar{x}) + g(\bar{x})] = 0]$$

$$(\forall y < z)[f(\bar{x}, y) = 0] \Leftrightarrow \sum_{y<z} f(\bar{x}, y) = 0$$

Si l'on pose $\lambda xy[x.\, y] = \sum_{i<y} x$ et $\lambda xy[x^y] = \prod_{i<y} x$ on obtient une autre caractérisation de E :

(i) $\qquad U_i^n, C_i^n, \lambda x[x+1], \lambda xy[x^y] = f_3(x, y)$

(ii) close par rapport à la substitution simultanée et par rapport à la récursion primitive bornée.

Lemme 2.4. La suite de fonctions $\lambda y g_k^x(y)$, $x = 0, 1, 2, \ldots$ satisfait au propriétés (1) à (3) de la définition 2.1.

Soit

$$h(0, y) = y$$
$$h(x+1, y) = g(h(x, y));$$
$$h(x.\, y) = g^x(y) \text{ et avec } g = f_3(k, y) = g_k(y) = k^y$$

on a $h_k(x, y) = g_k^x(y) = k^{k^y}$, les itérées successives de k^y.

(1) $\lambda y \varphi(\langle x, y \rangle) = \lambda y h_k(x, y) = g_k^x(y) \in E$ pour tout x, mais

$$\lambda z \varphi(z) \in E \Leftrightarrow \varphi(\langle x, y \rangle) + 1 \in E \Leftrightarrow \varphi(\langle x, y \rangle) + 1 \leqslant \varphi(\langle y_0, y \rangle)$$

pour un certain y_0, donc $\varphi(\langle y_0, y_0 \rangle) + 1 \leqslant \varphi(\langle y_0, y_0 \rangle)$, une contradiction et $\varphi(z) \notin E$.

(2) Construction de la fonction récursive α_k qui énumère un ensemble d'indices pour $\lambda y g_k^x(y)$ pour lequel le problème (*) est décidable :
soit $j = \langle 7, 1, \langle 2, k, 1 \rangle \rangle$ l'indice de $g_k(y) = k^y$, la fonction

$$\alpha(0) = \langle 1, 1, 1 \rangle$$
$$\alpha(n+1) = \langle 5, 1, j, \alpha(n) \rangle$$

définit un ensemble d'indices élémentaires pour $g_k^n(y)$; on vérifie aisément que $g_k^{x+1}(y) > g_k^x(y) > y$ pour tout $x > 0$ et tout y.

(3) $$(\forall\, i)[i \in \mathsf{C} \Rightarrow (\exists n)(\forall\, x \geqslant x_0)[\varphi_i(x) \leqslant g_k^n(x)]].$$

Cette propriété est facile à prouver par induction sur la construction de la classe C des indices élémentaires;

(i) i indice d'une fonction initiale : immédiat

(ii) substitution : si $f(x) \leqslant g_k^n(x)$ et $f_i(x) \leqslant g_k^{n_i}(x)$, $1 \leqslant i \leqslant m$

$$(\forall\, x \geqslant x_0) f(f_1(x), \ldots, f_m(x)) \leqslant g_k^n(\max(f_1, \ldots, f_m)) \leqslant g_k^{\,n+\max\limits_{1\leqslant i \leqslant m}(n_i)}(x)$$

somme finie et produit fini : si $f(x) \leqslant g_k^n(x)$

$$(\forall\, x \geqslant x_0) \sum_{i<x} f(i) \leqslant \sum_{i<x} g_k^n(i) \leqslant x \cdot g_k^n(x) \leqslant g_k^{n+1}(x)$$

$$(\forall\, x \geqslant x_0) \prod_{i<x} f(i) \leqslant \prod_{i<x} g_k^n(i) \leqslant [g_k^n(x)]^y \leqslant k^{\,x \cdot g_k^n(x)} \leqslant g_k^{n+2}(x)$$

Pour la démonstration de la proposition 2.2, il suffit maintenant de poser :

$h'(\underline{f}) = 0$ si $(\underline{f})_0 = 1, 2, 4$

$h'(\underline{f}) = 1$ si $(\underline{f})_0 = 3$

$h'(\underline{f}) = h'(\underline{g}) + \max(h'(\underline{g}_1), \ldots, h'(\underline{g}_m)$ si $\underline{f} = \langle 5, n, \underline{g}, \underline{g}_1, \ldots, \underline{g}_m \rangle$

$h'(\underline{f}) = h'(\underline{g}) + 1$ si $\underline{f} = \langle 6, n+1, \underline{g} \rangle$

$h'(\underline{f}) = h'(\underline{g}) + 2$ si $\underline{f} = \langle 7, n+1, \underline{g} \rangle$

$h'(f) =$ indéfini sinon

on pose $h'' = \alpha h'(\underline{f})$.

On peut aussi déterminer la fonction h' pour les autres schémas de clôture de E, puis avec l'énumération standard des indices récursifs élémentaires $\pi : N \rightarrow E$, on obtient une fonction récursive totale $h = \pi^{-1}h''\pi(x)$.

3. LA SOLUTION DU PROBLEME (**) POUR LA CLASSE E

On considère maintenant une certaine classe de machines de Minsky et une mesure de complexité fixée :

Définition 3.1. $\{ M_i \}$ est la classe de machines de Minsky $\Leftrightarrow \{ M_i \}$ est une des opérations de base sur le contenu X du registre $x(x = 0, 1, 2, ...)$:

(i) $(\forall x)(\forall y)Sx : X := X + 1; Nx : X := 0; Txy : X := Y \in \{ M_i \}$;

ou $\{ M_i \}$ est une opération portant sur une suite d'instructions

(ii) $(\forall x)M_p, M_q \in \{ M_i \} \Rightarrow \left\{ \begin{array}{l} RxM_pF : \text{itérer } M_p \text{ tant que } X \neq 0 \\ M_pM_q : \text{effectuer } M_p \text{ puis } M_q \end{array} \right\} \in \{ M_i \}$

Définition 3.2. M_i calcule φ_i avec complexité $\Phi_i \Leftrightarrow (\forall x)[M_i(x)$ s'arrête après exactement $\Phi_i(x)$ pas avec $\bar{\varphi}_i(x) = \langle x, \varphi_i(x) \rangle$ dans les registres ou M_i ne s'arrête jamais et $\varphi_i(x) = \Phi_i(x) =$ indéfini].

P.3.3. La classe des algorithmes (M-indices) de complexité bornée $\bar{E} = \{ i \mid M_i$ calcule φ_i avec complexité Φ_i & $(\exists n)(\forall x)\Phi_i(x) \leqslant g_k^n(x) \}$ n'est pas récursivement énumérable.

$$\text{Soit } \varphi_{f(z)}(x) = \begin{cases} g_k^x(x) & \text{si } (\exists\, n \leqslant x)T(z, z, n) \\ 1 & \text{si } (\forall\, n \leqslant x) \neg T(z, z, n) \end{cases}$$

donc $(\exists\, x_0)(\forall\, x > x_0)\varphi_{f(z)}(x) = g_k^x(x) \Leftrightarrow (\forall\, n)(\exists\, x_0)(\forall\, x > x_0)\Phi_{f(z)}(x) > g_k^n(x)$ et $(\forall\, x)\varphi_{f(z)}(x) = 1 \Leftrightarrow (\exists\, n)(\forall\, x)\Phi_{f(z)}(x) \leqslant g_k^n(x)$

$$\text{Soit } g(z, n) = \begin{cases} 1 & \text{si } (\forall\, x)\Phi_z(x) \leqslant g_k^n(x) \Leftrightarrow \langle z, n \rangle \in \alpha \\ 0 & \text{si } (\exists\, x_0)(\forall\, x > x_0)\Phi_z(x) > g_k^n(x) \Leftrightarrow \langle z, n \rangle \notin \alpha \end{cases}$$

$$(\exists\, n)g(f(z), n) = 1 \Leftrightarrow (\exists\, n)(\forall\, x)\varphi_{f(z)}(x) = 1 \Leftrightarrow (\exists\, n)(\forall\, x)\Phi_{f(z)}(x) \leqslant g_k^n(x) \Leftrightarrow \varphi_z(z) \text{ diverge}$$

donc $\bar{E} = \{ z \mid (\exists\, n)[\langle z, n \rangle \in \alpha] \}$ n'est pas récursivement énumérable.

La classe $\bar{E}$ et la classe correspondante de fonctions $F_{\bar{E}} = \{ f \mid (\exists\, i \in \bar{E})\varphi_i = f \}$ est close par rapport à un certain nombre de schémas de définitions $S : S(\bar{E}) \subset \bar{E}$. En particulier, $\bar{E}$ est close par rapport aux schémas de clôture de E, S_E. Ces propriétés de clôture permettent de construire, pour tout sous-ensemble récursivement énumérable $\bar{F}$ de $\bar{E}$, une fonction récursive k, telle que $(\forall\, i \in S(\bar{F}))\,[\Phi_i = \varphi_{k(i)}\ \&_{k(i)} \in S_E(\bar{F})]$. Si l'on pose $\bar{F} = E$, la classe des algorithmes des fonctions récursives élémentaires, on voit que E et $\bar{E}$ contiennent un indice pour toutes leurs fonctions de complexité.

Comme M peut être arithmétisé dans E, on en déduit que pour toute fonction dans $F_{\bar{E}}$ il existe un indice récursif élémentaire pour la même fonction et tel que le problème de l'estimation de la complexité (**) ait une solution.

P.3.4. Le problème (**) est soluble pour $E \Leftrightarrow (\exists\, k$ fonction récursive totale$)\,(\forall\, i \in E)[k(i) \in E\ \&\ \Phi_i = \varphi_{k(i)} \leqslant \varphi_{n(i)}]$.

Pour la démonstration, il suffit de montrer l'existence d'une telle fonction récursive totale k, puis on pose $n(i) = h_1 k(i)$, où h_1 correspond à la fonction récursive totale de la définition 2.1.

On remarque d'abord (et on prouve aisément par induction sur la définition des indices récursifs élémentaires) qu'il existe un procédé effectif (en fait, un « compiler ») qui, à tout indice récursif élémentaire donné, fait correspondre un M-indice équivalent (voir aussi la définition 1.4).

On détermine maintenant l'influence de certains schémas de définitions sur les fonctions de complexité de la classe $\bar{E}$.

Lemme 3.5. La classe $\bar{E}$ est close par rapport au schéma de

(i) la composition des fonctions,
(ii) l'itération bornée,
(iii) la somme finie et du produit fini,
(iv) la minimisation bornée.

Il suffit de se borner aux fonctions singulières sans restriction de la généralité car si $\lambda z f^{(1)}(z) = \lambda z g^{(n+1)}((z)_0, \ldots, (z)_n)$ on peut associer

$$\lambda x_0 \ldots x_n g^{(n+1)}(x_0, \ldots, x_n) \text{ avec } \lambda x f^{(1)}(x) = f^{(1)}(\langle x_0, \ldots, x_n \rangle)$$

et $\bar{E}$ contient une fonction de Cantor (des couples).

On adopte une méthode de [8] pour prouver que si f et g sont calculables avec fonctions $\Phi_{\underline{f}}$, $\bar{\varphi}_{\underline{f}}$ et $\Phi_{\underline{g}}$, $\bar{\varphi}_{\underline{g}}$ $\underline{f}$ et $\underline{g} \in \bar{E}$, alors

(i) $h(x) = fg(x)$ est calculable avec

$$\bar{\varphi}_{\underline{h}}(x) = \bar{\varphi}_{\underline{j}}(\bar{\varphi}_{\underline{f}}(\bar{\varphi}_{\underline{g}}(x))) = \bar{\varphi}_j(\bar{\varphi}_{\underline{f}}(\langle x, g(x) \rangle))$$
$$= \bar{\varphi}_{\underline{j}}(\langle x, g(x), f(g(x)) \rangle) = \langle x, fg(x) \rangle$$

$$\Phi_{\underline{h}}(x) = \Phi_{\underline{g}}(x) + \Phi_{\underline{f}}(g(x)) + \Phi_{\underline{j}}(f(g(x))) \qquad (\Phi_{\underline{j}}(x) = 1)$$

donc $\Phi_{\underline{h}}(x) = \varphi_{k(\underline{h})}(x)$ est récursif élémentaire dans f, g, Φ_f et $\Phi_{\underline{g}}$ et avec le lemme 2.4(3) il suit que $k(\underline{h}) \in \bar{E}$,

(ii) $h(x) = f^{(x)_1}(g((x)_0) \leqslant j(x)$ est calculable avec

$$\bar{\varphi}^i_{\underline{f}}(\langle (x)_0, (x)_1, 0, g((x)_0) \rangle) = \langle (x)_0, (x)_1 \dot{-} i, i, h(\langle (x)_0, i \rangle) \rangle$$

$$\Phi_{\underline{h}}(x) = \Phi_{\underline{g}}(x) + \Phi_{\underline{f}}^{(x)_1}(\bar{\varphi}_{\underline{g}}(x))$$

$$= \Phi_{\underline{g}}(x) + 1 + \sum_{i \leqslant (x)_1} (\Phi_{\underline{f}}(\langle (x)_0, (x)_1 \dot{-} i, i; h(\langle (x)_0, i \rangle) \rangle + 2)$$

$\Phi_{\underline{h}}(x) = \varphi_{k(\underline{h})}(x)$ est élémentaire en $g, f, \Phi_{\underline{g}}$ et $\Phi_{\underline{f}}$;

$k(\underline{h}) \in \bar{E} \Leftrightarrow (\forall\, x)[h(x) \leqslant j(x)]$, ce schéma est non-effectif car

$$(\forall\, x)[h(x) \leqslant j(x)]$$

n'est pas en général décidable pour $\bar{E}$ par P.1.3,

(iii) $f = \sum_{i<x} g(i)$ est calculable avec $\Phi_{\underline{f}}(x) = \sum_{i<x} \Phi_{\underline{g}}(i) + \sum_{1 \leqslant i \leqslant x} (3g(i) + 1) + 1$ et similairement pour $f(x) = \prod_{i<x} g(i)$,

(iv) ce schéma est équivalent à certains schémas précédents.

Définition 3.6. $\mathcal{G}(C)$ est la classe de fonctions de complexité de la classe d'algorithmes C.

Corollaire 3.7. $\mathcal{G}(E) \subset E$.

Immédiat par le lemme 3.5.

La définition explicite de la fonction récursive totale $k(x)$ de la proposition P.3.4, peut également être fondée sur la démonstration du lemme.

Le problème de l'estimation n'a évidemment pas de solution pour la classe $\bar{E}$, car celle-ci n'est pas récursivement énumérable par la proposition P.3.3 et par conséquent n'est pas le domaine de définition d'une fonction récursive.

La proposition P.3.3, permet également de montrer qu'il existe des fonctions qui « croissent trop vite » pour être calculables par un algorithme de complexité inférieure à une certaine borne.

Corollaire 3.8. Il n'y a pas d'indice dans $\bar{E}$ pour la fonction $f(x) = g_k^{\alpha(x)}(x)$, $\alpha(x) \in F_{\bar{E}}$ et non-borné.

Supposons le contraire; on obtient une contradiction :

Soit $g_k^x(x) = h_k(x, x)$, $h_k(x, y) = g_k^y(x)$ et $g_k^{\alpha(x)}(x) = h_k(x, \alpha(x))$.

Soit $\underline{f} \in \bar{E}$ un indice pour $h(x, \alpha(x))$; h_k est obtenu de g_k par itération et composition, d'après le lemme 3.5 :

$\Phi_{\underline{f}}(x) \leqslant \varphi_{h_1 k(\underline{f})}(x) = g_k^{K(\underline{f})}(x)$ pour une certaine fonction $K(x)$ récursive élémentaire et par 3.5.(ii) :

$$\Phi_{\underline{f}}(x) > \sum_{i \leqslant \alpha(x)} \Phi_{\underline{g_k}}(x, \alpha(x) \dot{-} i, i, h_k(x, i)) > h_k(x, \alpha(x)) > g_k^{K(\underline{f})}(x).$$

4. CLASSES RECURSIVEMENT ENUMERABLES DE FONCTIONS RECURSIVES ET HIERARCHIES

Il résulte des sections (2) et (3) que pour la classe E des algorithmes (M-indices) récursifs élémentaires le problème de l'estimation de la complexité (**) est soluble. Le même problème n'a pas de solution pour la classe de toutes les machines $\bar{E}$ dont la fonction de complexité est bornée par une des fonctions récursives élémentaires g_k^n.

On utilise maintenant une méthode de gödelisation de M [8] pour la construction de fonctions d'énumération de classes de fonctions récursives totales, $F_h = \{f \mid f = \varphi_z \;\&\; (\exists i)(\forall\, x)\Phi_z(x) \leqslant \varphi_{i0}(\langle i, x \rangle)\}$ où $\varphi_{i0}(\langle i, x \rangle)$ énumère une suite de « fonctions d'estimations » $\lambda x\varphi_{i_0}(\langle i, x \rangle) = \lambda xh(x, y)$. Lorsque l'on choisit $\lambda x\varphi_{i_0}(\langle i, x \rangle) = \lambda xg_k^i(x)$ on obtient la classe des fonctions $F_{\bar{E}}$. La fonction d'énumération correspondante détermine un certain sous-ensemble de E, celui-ci contient donc (au moins) un indice pour toute fonction dans $F_{\bar{E}}$. Il en résulte que les classes de fonctions

$$F_E = \{f \mid (\exists\, x \in E)\varphi_x = f\} \text{ et } F_{\bar{E}} = \{f \mid (\exists x \in \bar{E})\varphi_x = f\}$$

coïncident. Ce résultat peut être généralisé; on montre que l'on peut effectivement construire une infinité de classes récursivement énumérables $F_{h_0} \subset F_{h_1} \subset F_{h_2} \subset \ldots$ qui coïncident avec certaines hiérarchies de fonctions récursives totales. En outre, si $\{f_i(x)\}$ est une suite de fonctions d'estimation d'une certaine classe d'algorithmes, la classe de fonctions correspondantes F_h peut être caractérisée à l'aide de toute fonction $\varphi_{i_0}(\langle i, x \rangle)$ qui énumère une infinité d'éléments de $\{f_i(x)\}$.

Soit maintenant $\lambda x\varphi_{i_0}(\langle y, x \rangle) = \lambda xf_0^y(x), y = 0, 1, 2, \ldots$ une suite de « fonctions d'estimation » récursives totales pour la complexité des algorithmes d'une certaine classe C. On sait (section 3) que $\lambda yxh(x, y) = \lambda yxf_0^y(x)$ et plus généralement $\lambda yxh(x, \alpha(y))$, α non-borné, n'est pas C-calculable.

Définition 4.1. Soit $\lambda xf_0^y(x), y = 0, 1, 2, \ldots$ une suite de fonctions telles que

(1) $(\forall\, y)f_0^y(x + 1) > \lambda xf_0^y(x) \geqslant x$ et $\lambda xf_0^{y+1}(x) \geqslant \lambda xf_0^y(x)$; $f_0(x) \geqslant 2^x$;

(2) $(\forall\, i \in \mathsf{C})(\exists k)(\forall\, x)\Phi_i(x) \leqslant f_0^k(x)$;

(3) en particulier : $(\exists i \in \mathsf{C})(\exists k)(\forall\, x)\varphi_i = f_0 \Rightarrow \Phi_i(x) \leqslant f_0^k(x)$ c'est-à-dire qu'il existe un algorithme pour f_0 dans C.

P.4.2. Soit $\lambda xh(x, \alpha(y))$, α élémentaire et non-borné, une suite de fonctions satisfaisant à la définition 4.1 alors,

(i) la classe de fonctions $F_h = \{g \mid g = \varphi_z \Rightarrow (\exists\, i)(\forall\, x)\Phi_z(x) \leqslant \lambda xh(x, \alpha(i))\}$ est une classe récursivement énumérable.

(ii) Il existe une classe d'algorithmes qui contient au moins un indice pour tout $g \in F_h$ et pour laquelle le problème de l'estimation de la complexité est soluble.

(i) Il suffit de construire une fonction universelle pour cette classe de fonctions, c'est-à-dire d'indiquer un procédé qui énumère au moins un indice pour toute fonction de cette classe. On procède d'abord à la construction d'une certaine « forme normale » pour les fonctions de F_h; l'énumérabilité récursive en est un corollaire.

(ii) Soit $\underline{h}$ et $\underline{f_o}$ des indices pour h et f_0 respectivement; un algorithme d'une fonction universelle pour F_h peut être construit dans la classe $S_E(\underline{h})$ $(= E(\underline{h}))$ des définitions par des schémas récursifs élémentaires en h et tel que pour tout i fixé, si $\varphi_i = g \in F_h$, il fournit un indice pour g dans $E(\underline{f}_0)$. Le problème de l'estimation (**) est soluble pour cette classe.

Lemme 4.3. Il existe une arithmétisation de M telle que la fonction

$\lambda ixt\ U(i, x, t) =$ code de la configuration des registres de M_i si $t \geqslant \Phi_i(x)$

$\lambda ixt\ U(i, x, t) = 0$ si $t < \Phi_i(x)$

est récursive élémentaire.

Rödding montre dans [8] pour une machine M différente, mais équivalente, qu'il existe une codification $\{p_i\}$ des machines $\{M_i\}$ telle que $U(i, x, t)$ est même sub-élémentaire (la classe des fonctions sub-élémentaires contient les fonctions initiales de E et est close par rapport à la substitution simultanée et par rapport à la somme finie; toute fonction sub-élémentaire est bornée par un polynôme).

Étant donné un M-indice x on définit une x-configuration C_x par $\langle x, i, x_0, ..., x_n \rangle$ où i est le code d'une instruction de base de $M_x, (p_x)_1 = n$ et $x_0, ..., x_n$ sont les contenus des registres.

On prouve d'abord sans difficulté ([4], [8]) que les prédicats $M(p_x) \Leftrightarrow p_x$ est le code de $M_x \in \{M_i\}$ et $\text{Conf}\,(p_x, y) \Leftrightarrow y$ est x-configuration sont sub-élémentaires et que les codes p_x pour les machines M_x peuvent être énumérés par une fonction récursive élémentaire. Par conséquent, la fonction successeur $\lambda y S(x, y)$ pour les x-configurations y :

$S(x, y) =$ configuration successeur de y, si une telle configuration existe

$S(x, y) = y$ si y est finale

est sub-élémentaire, et la fonction de configuration :

$$C(i, x, t) = \text{« } i\text{-configuration } C_i \text{ après } t \text{ pas de } M_i \text{ »}$$

peut être définie par une récursion bornée :

$C(i, x, 0) = \langle i, F(i), x \rangle = g(i, x), F(i)$ la première instruction de base de M_i

$C(i, x, t + 1) = S(i, (C(i, x, t)) = h(i, C(i, x, t))$

$C(i, x, t) \leqslant j(i, x, t)$

avec $\lambda x g(i, x)$ et $\lambda x h(i, x)$ sub-élémentaires, donc $\lambda ixt\, C(i, x, t)$ élémentaire (si l'on borne $(p_i)_1 = n \leqslant n_0$ on peut trouver un $\lambda xtj(i, x, t)$ sub-élémentaire).

Par définition de $S(x, y)$, $C(i, x, t+1) = C(i, x, t)$ lorsque $C(i, x, t)$ est une configuration finale $\langle i, L(i), \varphi_i(x) \rangle$:

$$\Phi_i(x) = \mu t[C(i, x, t) = C(i, x, t+1)] \text{ et}$$
$$\varphi_i(x) = (C(i, x, \mu t[C(i, x, t) = C(i, x, t+1)]))_2 = D(C(i, x, \Phi_i(x))).$$

Finalement, on pose

$$\lambda ixyU(i, x, y) = \begin{cases} D(C(i, x, y)) \text{ si } y \geqslant \Phi_i(x) \\ 0 \qquad\qquad\quad \text{si } y < \Phi_i(x). \end{cases}$$

Démonstration de P.4.2. :

(i) On construit une fonction $\lambda ikxg(i\,;\, k, x)$ telle que

$$\lambda xg(i, k, x) = f(x) \text{ si } \mu t[C(i, x, t) = C(i, x, t+1)] \leqslant f_0^{\beta(k)}(x)$$
$$\lambda xg(i, k, x) = 0 \quad \text{ si } \mu t[C(i, x, t) = C(i, x, t+1)] > f_0^{\beta(k)}(x)$$

où $\beta(x) = \max\limits_{i \leqslant x} \{ \alpha(i) \}$ est non-décroissant et non-borné ;

donc $\varphi_i(x) = f(x) \in F_h \Rightarrow (\exists k) f(x) = \lambda xg(i, k, x)$

$\varphi_i(x) = f(x) \notin F_h \Rightarrow (\forall\, k)\lambda xg(i, k, x) = j(x) \neq f(x)\ \&\ j(x) \in F_h$

$\lambda ikxg(i, k, x) = \lambda ikxU(i, x, f_0^{\beta(k)}(x))$ est cette fonction.

Par le théorème S_n^m[9], il existe une fonction récursive $S(e, i, k)$ telle que $\lambda xU(i, x, f_0^{\beta(k)}(x)) = \lambda x\varphi_{S(e,i,k)}(x)$ où e est un indice pour U. Donc, si

$$i \in \mathrm{C} \Rightarrow (\exists\, k)\lambda xg(i, k, x) = f = \varphi_i(x) : F_h \subseteq \{ \varphi_{S(e,i,k)} \mid i, k \in N \}$$

$$\text{et si } j \in S(e, i, k) \Rightarrow \left\{ \begin{array}{l} (a)\ \varphi_{S(e,i,k)} = \varphi_i\ \&\ \Phi_i \leqslant f_0^{\beta(k)} \quad \text{ou} \\ (b)\ \varphi_{S(e,i,k)} = 0\ \ \&\ \Phi_i = f_0^{\beta(k)} \end{array} \right\}$$

$$i \in \mathrm{C} : \{ \varphi_{S(e,i,k)} \mid i, k \in N \} \subseteq F_h$$

(ii) D'après le lemme, un indice e pour $\lambda ikxU(i, x, f_0^{\beta(k)}(x))$ peut être trouvé dans $E(\underline{h})$, $\underline{h}$ un indice pour $\lambda yxf_0{}^{(y)}(x)$. D'autre part, $S(e, i, k) \subset E(\underline{f}_0)$, donc $E(\underline{f}_0)$ contient au moins un indice pour toute fonction dans F_h et le problème (**) est soluble pour $E(\underline{f}_0)$.

Corollaire 4.4. — (i) $\lambda inxU(i, x, g_k^{\alpha(n)}(x))$ énumère la classe $F_{\overline{E}}$;

(ii) $$F_E = F_{\overline{E}}.$$

(i), (ii) immédiat par P. 4.2. (i), (ii).

Corollaire 4.5.

(i) $f(x) = \begin{cases} 1 \text{ si } (\exists\, u \leqslant h(x) T(x, x, n) \\ 0 \text{ sinon} \end{cases}$, $h(x)$ récursif, est calculable avec complexité $\Phi_{\underline{f}}(x) = g_0(x, h(x))$, g_0 récursif élémentaire.

(ii) Pour tout α non-borné, $h(x, \alpha(y))$ détermine la même classe F_h.

(i) Soit $\lambda ix D_1[C(i, x, h(x))) = \begin{cases} 1 \text{ si } \Phi_i(x) \leqslant h(x) \\ 0 \text{ si } \Phi_i(x) > h(x) \end{cases}$;

$\lambda x D_1(C(x, x, h(x))) = \varphi_e(x, h(x))$, $e \in E$ par la définition de C ; donc $\Phi_e(x, h(x)) = \varphi_{k(e)}(x, h(x)) = g_0(x, h(x))$; $k(e) \in E$.

(ii) Soit $\alpha(x)$ tel que $(\forall\, u)(\exists\, x)(\exists\, i)[u < x \;\&\; x = \alpha(i)]$;

$$\varphi_e(i, x, y) = \lambda iyx U(i, x, h(x, \alpha(y))) = \lambda ikx U(i, x, f_0^k(x)) = \varphi_f(i, k, x)$$

pour une infinité de k, donc $\bigcup_{y=0}^{\infty} \{ g_{S(e,i,y)} \} = \bigcup_{k=0}^{\infty} \{ g_{S(f,i,k)} \}$.

On généralise maintenant la construction de certaines hiérarchies de fonctions récursives à l'aide d'un théorème de Rabin [5] et Blum [6] :

P.4.6. — Soit $\lambda x h_0(x, \alpha(y)) = \lambda x f_0^{\alpha(y)}(x)$, α non-borné, une suite de fonctions récursives, $f_0(x)$ satisfaisant à la définition 4.1. On peut construire une hiérarchie de classes récursivement énumérables de fonctions $F_{h_0} \subset F_{h_1} \subset F_{h_2} \subset \ldots$ et d'algorithmes correspondants pour lesquels le problème (**) a une solution.

Soit $E(\underline{f}_0)$ la classe de définitions de fonctions élémentaires en f_0 ; $F_{h_0} = F_{E(f_0)}$ par la proposition P.4.2. Donné f_0, il suffit de construire $f_1(x) \geqslant \max_{i \leqslant x} [f_0^{\alpha(i)}(x)] = f_0^{\beta(x)}(x)$, $\beta(x)$ non-borné et non décroissant, de telle façon que $F_{h_1} = F_{E(\underline{f}_1)}$.

On prouve d'abord que le théorème de Rabin-Blum s'applique à la classe $\{ g_i(x) \}$ des fonctions de complexité de $E(\underline{f}_0)$: soit $\lambda x E(\langle i, x \rangle) = f_i(x)$ une fonction d'énumération pour les fonctions récursives élémentaires en f_0 ; la proposition P.3.4 permet de trouver un indice $k(i)$ pour la fonction de complexité correspondante $g_i(x)$ et le théorème S^m_n donne $g_i(x) = f_{k(i)}(x) = \varphi_{tk(i)}$.

Le prédicat $\quad \lambda ixy[g_i(x) = y] \Leftrightarrow \lambda ixy[M(i, x, y) = 1]$
$\qquad\qquad\quad \lambda ixy[g_i(x) \neq y] \Leftrightarrow \lambda ixy[M(i, x, y) = 0]$

est récursif, car g_i est total et borné.

Théorème 4.7. (Rabin-Blum) :

Pour tout g_i il existe une fonction $f : N \longrightarrow \{0, 1\}$ telle que :

(i) si j est un indice pour f, $g_i(x) < \Phi_j(x)$ pour presque tout x ;

(ii) il existe un indice k pour f tel que $\varphi_k = f = \varphi_{\tau(i)}$ avec une certaine fonction récursive τ ;

(iii) il existe une fonction récursive h telle que $\Phi_{\tau(i)}(x) = \Phi_k(x) < h(x, g_i(x))$ pour presque tout x.

Pour le calcul de $f(x)$ on considère l'ensemble de fonctions $\{\varphi_0(x), ..., \varphi_x(x)\}$; on cherche le premier indice $i \leqslant x$ qui n'a pas été utilisé pour le calcul de $f(j)$, $j < x$ et qui est tel que $\Phi_i(x) \leqslant g(x)$, puis on pose $f(x) = 1 \dot{-} \varphi_i(x)$ si un tel i existe

et $f(x) = 0$ sinon.

On montre sans difficulté que l'algorithme $s(i)$ pour la fonction $f(x)$ du théorème peut être effectué par une machine de Minsky en moins de $f_0^{n+k}(x)$ k constant, si $g_i(x) \leqslant f_0^n(x)$. Donc, pour tout $g_i(x)$ donné, $f(x)$ possède un indice dans $\underline{E}(f_0)$

si $\qquad g_i = f_{k(i)}$, alors $f(x) = f_{sk(i)} = f_{\tau(i)}$ et $\tau(i) \in \underline{E}(f_0)$.

La fonction de complexité de $f(x)$ est $f_{k\tau(i)}$, elle peut être majorée par une itérée de $f_0(x)$:

$$f_{k\tau(i)}(x) \leqslant f_{h_1 k\tau(i)}(x) = f_0^{\alpha(i)}(x) = h_o(x, \alpha(i))$$

où la fonction non-bornée $\alpha(i)$, qui dépend de l'énumération $E(\langle i, x \rangle)$, peut être effectivement déterminée.

La fonction $h(x) = \max\limits_{i \leqslant x} [f_0^{\alpha(i)}(x)] = f_0^{\beta(x)}(x)$ est une majorante pour toutes les fonctions $f_{k\tau(i)}$.

Soit $f_1 \geqslant f_0^{\beta(x)}(x)$ une autre fonction qui satisfait à la définition 4.1. Le procédé peut être itéré et donne une hiérarchie d'algorithmes $\underline{E}(f_0) \subset \underline{E}(f_1) \subset ...$ et de fonctions $F_{h_0} \subset F_{h_1} \subset ...$ correspondantes. L'inclusion propre des classes suit si l'on prouve que $\underline{E}(f_1)$ contient une fonction d'énumération pour F_{h_0} : $\lambda ikx U(i, x, f_0^{\alpha(k)}(x))$. Il suffit de montrer que $f_0^{\alpha(y)} \in \underline{E}(f_1)$ mais $f_0^{\alpha(y)}(x) = \mu z \leqslant f_1(\max(x, y)) [f_0^{\alpha(y)}(x) = z]$ et cette fonction est élémentaire en f_1, ce qui prouve P.4.6.

BIBLIOGRAPHIE

[1] J. HARTMANIS, R. E. STEARNS. « On the Computational Complexity of Algorithms » *Trans. Amer. Math. Soc.*, 117, pp. 285-306 (1965).

[2] P. M. LEWIS II, R. E. STEARNS et HARTMANIS J. « Memory bounds for the recognition of context-free and context-sensitive languages », 1965, *IEEE Conference Reord on Switching Circuit Theory and Logical Design*, pp. 191-202, *IEEE*, New-York (1965).

[3] J. HARTMANIS, « Tape-Reversal Bounded Turing Machine Computations », *Journal of Computer and System Sciences* 2, Nr 2, pp. 117-135 (1968).

[4] R. W. RITCHIE, « Classes of Predictably Computable Functions », *Trans. Amer. Math. Soc.*, 106, pp. 139-173 (1963).

[5] M. O. RABIN, « Degrees of Difficulty of Computing a Function and a Partial Ordering of Recursive Sets », *Technical Report* Nr 2, Hebrew University, Jerusalem, (1960).

[6] Manuel BLUM, « A Machine-Independent Theory of the Complexity of Recursive Functions », *Journal A.C.M.*, avril 1967, vol. 14, Nr 2, pp. 322-336.

[7] Manuel BLUM, « Recursive Function Theory and Speed of Computation », *Canadian Math. Bull.* 9 (1966), pp. 745-750.

[8] Dieter RÖDDING, « Klassen rekursiver Funktionen », *Lecture Notes in Math.* 70, 1968, Springer-Verlag, Berlin, pp. 159-222.

[9] S. C. KLEENE, « Extension of an Effectively Generated Class of Functions by Enumeration », *Colloquium Math.* VI (1958), pp. 67-78.

[10] Hartley ROGERS, *Theory of Recursive Functions and Effective Computability*, Mac Graw-Hill Book Company, New York, 1967.