RAIRO. INFORMATIQUE THÉORIQUE

MAURICE TCHUENTE

Sur l'élimination itérative des situations de blocage dans un système

RAIRO. Informatique théorique, tome 14, nº 1 (1980), p. 57-66
<http://www.numdam.org/item?id=ITA_1980__14_1_57_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 14, n° 1, 1980, p. 57 à 66)

SUR L'ÉLIMINATION ITÉRATIVE DES SITUATIONS DE BLOCAGE DANS UN SYSTÈME (*)

par Maurice TCHUENTE [1]

Communiqué par M. SINTZOFF

Résumé. — *On s'intéresse à une méthode itérative proposée pour éliminer, dans un système, les évolutions qui mènent à des situations de blocage : on présente deux méthodes d'analyse permettant, dans certains cas simples, de démontrer le caractère fini de cette itération.*

Summary. — *We consider an iterative scheme proposed for the elimination of deadends in a system: we present two methods which permit, in some simple cases, to demonstrate that the iteration is finite.*

INTRODUCTION

Un système non déterministe $S=[S_1 \mid S_2 \mid \ldots \mid S_N]$ est défini par la donnée de N instructions conditionnelles

$$S_i: \quad si\ A_i\ alors\ x:=\alpha_i(x),$$

où X est l'ensemble des états possibles du système; A_i est un prédicat sur X; on dira que A_i garde S_i; α_i est une application de X dans X.

A un instant t, si $x \in X$ est l'état du système, et si l'instruction S_j est sélectionnée pour exécution, alors l'état du système à l'instant $t+1$ est $\alpha_j(x)$ si $A_j(x)=vrai$, et x si $A_j(x)=faux$; le non-déterminisme se traduit par le fait que n'importe quelle instruction peut être sélectionnée pour exécution à l'instant t; ainsi, toute suite à éléments dans $\{S_1, S_2, \ldots, S_N\}$ correspond à une évolution possible du système à partir d'un état initial.

Soit par ailleurs B un prédicat sur X appelé but; dans la suite un prédicat P sur X sera confondu avec $\{x \in X : P(x)=vrai\}$.

(*) Reçu mars 1979, révisé juillet 1979.
([1]) Laboratoire I.M.A.G., Grenoble.

Lorsque le système se trouve dans un état

$$x \notin A_1 \vee A_2 \vee \ldots \vee A_N \vee B,$$

aucun des prédicats A_i n'est satisfait, et le système ne peut plus évoluer vers le but B; x sera alors appelé un état de blocage.

Ce modèle est particulièrement adapté à la formalisation de la coopération entre plusieurs processeurs; B représente alors les états de bon fonctionnement du système; c'est par exemple le cas des algorithmes d'exclusion mutuelle avec autostabilisation dans un réseau d'automates [2, 6].

Partant d'un système S, on s'intéresse à la détermination d'un nouveau système $T=[T_1 \mid T_2 \mid \ldots \mid T_N]$ qui est une version améliorée de S en ce sens que :

(1) toutes les évolutions de T sont possibles dans S;

(2) tout état initial d'où il est possible d'atteindre B par S est un état initial d'où il est possible d'atteindre B par T;

(3) les évolutions de S qui mènent à une situation de blocage (se terminent sans atteindre B) sont éliminées dans T;

Il faut noter que T, bien que ne se bloquant pas, peut cycler indéfiniment sans jamais atteindre le but B.

L'une des méthodes itératives proposées dans [4, 7] permet d'obtenir les prédicats B_i qui gardent les T_i, comme bornes inférieures de suites décroissantes de prédicats :

$$B_i^{n+1} \leqq B_i^n \leqq A_i; \qquad B_i = \bigwedge_{n \geqq 0} B_i^n.$$

Il est clair que cette technique est intéressante surtout lorsque ces bornes sont atteintes en un nombre fini de pas; c'est évidemment le cas lorsque les A_i sont des parties finies de X. On expose ici deux méthodes d'analyse qui, dans certains cas simples où les A_i sont de cardinalité infinie, permettent de démontrer que l'itération en question est finie.

I. NOTATIONS ET DÉFINITIONS

Les symboles $\vee$, $\wedge$, $\neg$, désignent respectivement la disjonction, la conjonction et la négation. On note Z (resp. Q) l'ensemble des nombres entiers (resp. rationnels).

Les instructions d'un système $S=[S_1 \mid S_2 \mid \ldots \mid S_N]$ sont notées :

$$si\ A_i\ alors\ s_i \qquad (i=1, 2, \ldots, N),$$

où :

– X est l'ensemble des états possibles du système;

– A_i est un prédicat sur X; on dira que A_i *garde* S_i;

– s_i est l'instruction d'affectation $x := \alpha_i(x)$.

Au système S est associé un prédicat B appelé *but*, tel que

$$B \Rightarrow \neg\left(\bigvee_{j=1}^{N} A_j\right).$$

Les états de *blocage* du système sont caractérisés par

$$\neg\left(\bigvee_{j=1}^{N} A_j \vee B\right).$$

DÉFINITION : Transformateurs de prédicats pre (., .).

Soit R un prédicat sur X :

– pre (s_i, R) caractérise l'ensemble des états tels que, après exécution de l'affectation s_i, R est satisfait;

– pre (S_i, R) caractérise l'ensemble des états tels que le prédicat qui garde S_i est satisfait et, après exécution de S_i, R est satisfait :

$$\text{pre}(S_i, R) = A_i \wedge \text{pre}(s_i, R);$$

– pre (S, R) caractérise l'ensemble des états tels *qu'il existe* une instruction S_i dont l'exécution mène à la satisfaction de R :

$$\text{pre}(S, R) = \bigvee_{i=1}^{N} \text{pre}(S_i, R).$$

On vérifie aisément que les transformateurs de prédicats pre $(s_i, .)$, pre $(S_i, .)$ et pre $(S, .)$ sont monotones :

$$k_1 \Rightarrow k_2 \text{ entraîne } \text{pre}(I, k_1) \Rightarrow \text{pre}(I, k_2) \qquad (I = s_i, S_i, S).$$

II. FORMALISATION ET SOLUTION DU PROBLÈME [4, 7]

Le problème consiste, partant de

$$S_i: \quad \textit{si } A_i \textit{ alors } x := \alpha_i(x) \qquad (i=1, \ldots, N),$$

à construire

$$T_i: \quad \textit{si } B_i \textit{ alors } x := \alpha_i(x) \qquad (i=1, \ldots, N),$$

où les B_i sont les *solutions maximales* du système suivant :

$$\left.\begin{array}{lcl} (a) & B_i \;\Rightarrow\; A_i & (i=1, \ldots, N), \\ (b) & B_i \;\Rightarrow\; \operatorname{pre}(s_i, BB) & (i=1, \ldots, N), \\ & \text{où}\quad BB=B_1 \vee B_2 \vee \ldots \vee B_N \vee B, & \end{array}\right\} \qquad (1)$$

ce système peut encore s'écrire :

$$B_i \;\Rightarrow\; \operatorname{pre}(S_i, BB) \qquad (i=1, \ldots, N). \qquad (1)$$

La validité d'un système équivalent à (1) est formellement analysée dans [7] pour le cas où $B \equiv$ *faux*, c'est-à-dire où T doit cycler indéfiniment.

Interprétation

(a) exprime que toute évolution de T est possible dans S;

(b) exprime le fait que, partant d'un état x qui satisfait B_i, on atteint après exécution de T_i soit le but B, soit un état satisfaisant un prédicat B_j, ce qui exclut le blocage.

Le fait que nous recherchions la solution maximale garantit que toute évolution de S menant au but B sera préservée dans T.

On est donc ramené à la recherche de la solution maximale d'une inéquation associée à une fonction monotone dans le treillis complet $(\mathscr{P}(X))^n$.

THÉORÈME [5] : *Soit* $(E, \leqq)$ *un treillis complet, et* f *une application monotone de* E *dans* E.

1. *L'inéquation* $x \leqq f(x)$ *et l'équation de point fixe* $x = f(x)$ *admettent la même solution maximale* $\bar{x}$.

2. *Si de plus, pour toute suite décroissante de* E, $\{x_n;\ n \geqq 0\}$ *on a*

$$f\Big(\bigwedge_{n\geqq 0} x_n\Big) = \bigwedge_{n\geqq 0} f(x_n)$$

et si 1 *désigne l'élément maximal du treillis, alors*

$$\bar{x} = \underset{n\geqq 0}{\quad} f^n(1).$$

Solution itérative du problème [4, 7]

vrai désigne le prédicat toujours satisfait; on part d'une initialisation

$$B_i^0 \equiv vrai \qquad (i=1, \ldots, N)$$

pour $n \geqq 0$,

$$B_i^{n+1} = \operatorname{pre}(S_i, BB^n) \qquad (i=1, \ldots, N),$$

$$BB^n = \bigvee_{j=1}^{N} B_j^n \vee B,$$

on a alors

$$B_i = \bigwedge_{n \geqq 0} B_i^n \qquad (i=1, \ldots, N).$$

DÉFINITION : On dit que l'itération est *finie*, ou *stationnaire*, si :
Il existe $n \geqq 0$:

$$B_i^n = B_i^{n+1} \qquad (i=1, \ldots, N).$$

Ceci correspond aux cas où la méthode itérative permet d'obtenir la solution en un nombre fini de pas.

Commentaire

Dans la plupart des problèmes tels que la synchronisation, l'ensemble des états du système, X, est fini; en conséquence, l'ensemble des parties de X, noté $\mathscr{P}(X)$, est aussi fini et les suites générées par le processus itératif sont décroissantes dans un ensemble fini et sont donc stationnaires; le seul problème qui subsiste alors est celui du volume de calculs qui peut devenir prohibitif.

Lorsque X est infini, la suite $\{ B_i^n;\ n \geqq 0 \}$ peut soit avoir une infinité de termes distincts, soit être stationnaire. Ces deux comportements étant fondamentalement différents, il est intéressant de pouvoir dire à quelle catégorie un système appartient. Nous présentons ici deux méthodes d'analyse permettant de répondre à cette question dans certains cas simples.

Signalons qu'il existe dans la littérature des tentatives pour remplacer, dans le cas non stationnaire, la suite exacte infinie par une suite approchée finie [4].

III. MÉTHODE I

Rappelons que s_i désigne l'instruction d'affectation

$$x := \alpha_i(x).$$

La méthode I suppose connue l'expression explicite des transformateurs de prédicats $\operatorname{pre}(s_i, .)$; l'idée est alors très simple car, connaissant une expression formelle des éléments des suites $\{ B_i^n;\ n \geqq 0 \}$, on donne une condition suffisante pour que ces suites appartiennent à une partie finie de $\mathscr{P}(X)$, ce qui garantit leur caractère stationnaire.

Supposons que l'on ait $\{1, 2, \ldots, N\} = I \cup J$ avec :

– pour $i \in I$, α_i est une bijection de X dans X; l'application réciproque est notée φ_i;

– pour $j \in J$, α_j est constante : $\alpha_j(x) = b_j$, $\forall x \in X$.

NOTATIONS : $(F[X_1\ X]_1\ o)$ désigne le monoïde des applications de X dans lui-même.

$$C = \{A_1, \ldots, A_N, B\}; \qquad \Phi = \{\varphi_i, i \in I\};$$

Φ^* est le sous-monoïde de $(F[X_1\ X]_1\ o)$ généré par Φ

$$\Phi^*(C) = \{\varphi(A) : \varphi \in \Phi^*, A \in C\}$$

$\overline{\Phi^*(C)}$ est la fermeture de $\Phi^*(C)$ par rapport aux opérations $\wedge$, $\vee$.

PROPRIÉTÉ 1 : $\forall n \geqq 1$: on a $B_i^n \in A_i \wedge \overline{\Phi^*(C)}$ et $B_j^n \in \{A_j, \emptyset\}$, pour $i \in I$ et $j \in J$.

Démonstration : Nous raisonnons par récurrence sur n.

$n = 1$: $B_i^1 = A_i$ $(i = 1, \ldots, N)$ et la propriété est vraie.

Supposons le résultat acquis pour $n - 1$ et montrons qu'il est encore vrai pour n; posons

$$BB^{n-1} = B_1^{n-1} \vee \ldots \vee B_N^{n-1} \vee B;$$

pour $j \in J$,

$$B_j^n = A_j \wedge \operatorname{pre}(s_j, BB^{n-1}) = \begin{cases} A_j & \text{si } b_j \in BB^{n-1}, \\ & \emptyset \text{ sinon;} \end{cases}$$

pour $i \in I$,

$$B_i^n = A_i \wedge \operatorname{pre}(s_i, BB^{n-1})$$

$$= A_i \wedge \varphi_i(B_1^{n-1} \vee \ldots \vee B_N^{n-1} \vee B) = A_i \wedge \left(\bigvee_{j=1}^{N} \varphi_i(B_j^{n-1}) \vee \varphi_i(B)\right);$$

en utilisant l'hypothèse de récurrence on déduit que $B_i^n \in A_i \wedge \overline{\Phi^*(C)}$. ■

PROPRIÉTÉ 2 : Si, pour tout $i \in I$, pour tout $A \in \{A_1, \ldots, A_N, B\}$, $A_i \wedge \Phi^*(A)$ est fini dans $\mathscr{P}(X)$, alors l'itération est stationnaire.

Démonstration : On montre que $A_i \wedge \overline{\Phi^*(C)}$ est fini; la suite $\{B_i^n; n \geqq 0\}$, décroissante dans un ensemble fini, est donc stationnaire. ■

Exemple 1 : *Énoncé* (nous adoptons une présentation imagée afin de favoriser l'intuition) : Soit une région plane $[a, b] \times [c, d]$ dont une partie est recouverte d'eau. Un oiseau qui s'y promène, vole d'un point à l'autre avec les contraintes suivantes : lorsqu'il s'envole d'un point (x_1, x_2), il ne peut se poser qu'en l'un des points $(0, 0)$, (x_1+1, x_2-1), (x_1+2, x_2+1). Peut-il ainsi se promener indéfiniment sans jamais se mouiller ?

Solution : soit $L \subset [a, b] \times [c, d]$ l'ensemble des régions non recouvertes d'eau. On a alors un système $S = [S_1 \mid S_2 \mid S_3]$.

$$S_1: \quad si\ L\ alors\ (x_1, x_2) := (0, 0),$$
$$S_2: \quad si\ L\ alors\ (x_1, x_2) := (x_1+1, x_2-1),$$
$$S_3: \quad si\ L\ alors\ (x_1, x_2) := (x_1+2, x_2+1).$$

En posant $B = \emptyset$, le problème revient à construire un nouveau système T déduit de S par élimination des évolutions qui mènent à des situations de blocage. La propriété qui suit montre qu'on est dans un cas où l'itération est stationnaire.

PROPRIÉTÉ 3 : Soit $X = \mathbb{R}^m (m \geqq 1)$, $S = [S_1 \mid \ldots \mid S_N]$ où les fonctions α_i sont de deux formes :

$$i \in I: \quad (x_1, \ldots, x_m) \rightarrow (a_1 x_{\sigma_1}, \ldots, a_m x_{\sigma_m}) + v_i,$$
$$j \in J: \quad (x_1, \ldots, x_m) \rightarrow v_j,$$

où $a_i \in \{-1, 1\}$, $v_j \in Q^m$; σ permutation de $\{1, 2, \ldots, m\}$.

Si $A_1, \ldots, A_N$, B sont bornés dans R^m, alors l'itération est stationnaire.

Démonstration : Adoptons les notations introduites plus haut; après réduction des v_i au même dénominateur, on peut montrer que les fonctions réciproques $\varphi \in \Phi^*$ sont de la forme :

$$(x_1, \ldots, x_m) \rightarrow (b_1 x_{\sigma_1}, \ldots, b_m x_{\sigma_m}) + (e_1 v, \ldots, e_m v)$$

où $b_i \in \{-1, 1\}$, $e_i \in Z (i=1, \ldots, m)$; σ permutation de $\{1, 2, \ldots, m\}$, v fixe dans Q.

Montrons que pour $i \in I$, $A \in \{A_1, \ldots, A_N, B\}$, $A_i \wedge \Phi^*(A)$ est fini.

En effet $x \in A_i$, $y \in A$, $\varphi \in \Phi^*$, $x = \varphi(y)$ entraîne

$$(x_1, \ldots, x_m) = (b_1 y_{\sigma_1}, \ldots, b_m y_{\sigma_m}) + (e_1 v, \ldots, e_m v) \quad (\star).$$

On déduit :

$$\sum_{i=1}^{m} |e_i| \leqq \frac{2}{|v|} \sup_{\xi_i \in AA} \sum_{i=1}^{m} |\xi_i| \qquad \text{où} \quad AA = \bigvee_{j=1}^{N} A_j \vee B.$$

Les paramètres qui interviennent dans l'égalité (★), à savoir b_i, e_j, σ, sont en nombre fini, majoré par une constante indépendante de x et y. Ceci entraîne que $A_i \wedge \Phi^*(A)$ est fini; d'après la propriété 2, l'itération est stationnaire. ■

IV. MÉTHODE II

A tout système $S=[S_1 | \ldots | S_N]$ ayant X pour ensemble d'états, on peut associer le graphe orienté G défini comme suit :

X est l'ensemble des sommets; pour tout $x \in B$, on a une boucle en x; pour tout $x \in A_i (i=1, \ldots, N)$, on a un arc allant de x à $\alpha_i(x)$.

Il est clair que le graphe G représente le comportement dynamique de S; les états que le système peut avoir en évoluant à partir d'un état x sont représentés par les sommets qu'on peut atteindre à partir de cet état, en suivant les arcs de G.

DÉFINITION : Une impasse de G de longueur p est un chemin $[x_0, x_1, \ldots, x_p]$ tel que $x_p \notin A_1 \vee A_2 \vee \ldots \vee A_N \vee B$ et tous les chemins de G contenant l'arc (x_0, x_1) sont de longueur $\leqq p$. Les évolutions de S qui mènent à une situation de blocage sont donc des impasses de G.

PROPRIÉTÉ 4 : $\forall i \in \{1, 2, \ldots, N\}, \forall n \geqq 1, \forall x \in A_i : x \in B_i^n \Leftrightarrow x$ est origine d'un chemin de longueur n commençant par l'arc $(x, \alpha_i(x))$.

Démonstration : Si $n=1$, $B_i^1 = A_i$ et le résultat est acquis; pour $n \geqq 2$, la démonstration se fait par récurrence sur n. ■

Notons

$$S^n=[S_1^n | \ldots | S_N^n] \quad (n \geqq 1); \qquad S_i^1 = S_i;$$

$$S_i^n: \quad si\ B_i^n\ alors\ x := \alpha_i(x).$$

G^n le graphe orienté associé à S^n.

PROPRIÉTÉ 5 : 1. G^n se déduit de G par suppression des impasses de longueur $\leqq n-1$.

2. L'itération est stationnaire si et seulement si les impasses de G ont une longueur majorée par une constante.

Démonstration : 1. G^n se déduit de G^{n-1} par suppression des impasses de longueur 1; on raisonne ensuite par récurrence sur n.

2. Évident. ■

PROPRIÉTÉ 6 : Posons $\psi = \{\alpha_1, \ldots, \alpha_N\}$; ψ^* le sous-monoïde de $(F[X_1 X]_1 o)$ engendré par ψ. Pour $x \in X$, posons $\psi^*(x) = \{\alpha(x), \alpha \in \psi^*\}$. S'il existe $l \in N$ tel que

$$\forall x \in AA, \quad \text{Card}(\psi^*(x) \cap AA) \leqq l, \qquad \text{où} \quad AA = A_1 \vee \ldots \vee A_N,$$

Alors l'itération est stationnaire.

Démonstration : Si $[x_0, x_1, \ldots, x_n]$ est une impasse de G de longueur n, alors $x_i (i=0, \ldots, n-1)$ sont des sommets distincts de $\psi^*(x_0) \cap AA$, donc $n \leqq l$.

Les impasses de G étant de longueur majorée par l, l'itération est stationnaire. ■

Nous sommes maintenant en mesure de traiter des cas qui échappent à la méthode I, d'abord parce que les α_i ne sont supposées ni bijectives, ni constantes, et parce qu'on ne fait aucune hypothèse sur le but B.

PROPRIÉTÉ 7 : Soit $X = \mathbb{R}^m$, $S = [S_1 | \ldots | S_N]$ un système où les S_i sont de la forme

$$si\ A_i\ alors\ (x_1, \ldots, x_m) := (a_1 x_{\sigma_1}, \ldots, a_m x_{\sigma_m}) + v_i,$$

où $a_i \in \{-1, 0, 1\}$, $v_j \in Q^m$, σ est une permutation de $\{1, 2, \ldots, m\}$; si les A_i sont bornés dans $\mathbb{R}^m$, alors l'itération est stationnaire.

Démonstration : On montre que les fonctions $\alpha \in \{\alpha_1, \ldots, \alpha_N\}^*$ sont de la forme

$$(x_1, \ldots, x_m) \rightarrow (b_1 x_{\sigma_1}, \ldots, b_m x_{\sigma_m}) + (e_1 v, \ldots, e_m v),$$

où $b_i \in \{-1, 0, 1\}$, $e_i \in Z$, σ une permutation de $\{1, 2, \ldots, m\}$, v fixe dans Q. Par une technique strictement analogue à celle de la propriété 3, on montre que les conditions de la propriété 6 sont vérifiées; l'itération est donc stationnaire. ■

DISCUSSION

Nous avons analysé l'une des méthodes itératives proposées dans [4, 7] pour l'élimination des situations de blocage dans un système, en recherchant des conditions suffisantes pour que l'itération soit stationnaire.

Dans le cas où la suite générée a une infinité de termes distincts, il suffit d'examiner les trois exemples

$$E_1 : \left\{ \left[0, \frac{1}{n} \right]; n \geqq 1 \right\} \qquad E_2 : \left\{ \left[\frac{-1}{n}, 1 + \frac{1}{n} \right]; n \geqq 1 \right\}$$

$$E_3 : \{ [n, +\infty[; n \geqq 1 \}$$

pour se rendre compte que les notions de convergence et d'approximation doivent être manipulées avec précaution. Il s'agit en effet de savoir dans quelle mesure et en quel sens une technique itérative générant une suite décroissante de prédicats sur un ensemble X :

$$\{ B_i^n; n \geqq 0 \} \qquad \text{avec} \quad B_i = \bigwedge_{n \geqq 0} B_i^n$$

peut être considérée comme une méthode effective de calcul de B_i. Ce problème n'a pas été abordé ici et est analogue à ce que A. Arnold et M. Nivat appellent dans [1] notion de calcul infini d'un programme récursif. Comme dans [1], en munissant l'ensemble X d'une structure d'espace métrique, on peut apporter des éléments de réponse.

Plus précisément, si d est une distance sur X, on peut définir pour toute partie A de X les notions suivantes :

diamètre de A,

$$\operatorname{diam}(A)=\sup\{d(a, b);\ a, b\in A\};$$

voisinage de A de rayon r,

$$V(A, r)=\{x\in X;\ \exists\, a\in A,\ d(x, a)\leqq r\}.$$

On peut alors définir au moins deux niveaux d'efficacité :

niveau 1 : diam $(B_i^n - B_i)$ tend vers 0 quand n tend vers l'infini;

niveau 2 : pour tout r positif, il existe n' tel que pour $n\geqq n'$:

$$B_i^n\subset V(B_i, r).$$

Les exemples E_1 et E_2 correspondent respectivement aux niveaux 1 et 2.

Par contre l'exemple E_3 échappe aux deux critères et dans ce cas on peut dire que la méthode itérative n'est pas efficace.

REFERENCES

1. A. Arnold et M. Nivat, *Non Deterministic Computation Spaces*, Second International workshop on semantics of programming languages, Bad-Honnef, mars 1979.
2. E. W. Dijkstra, *Self-Stabilizing Systems in Spite of Distributed Control*, Comm. A.C.M., vol. 17, 1974, p. 643-644.
3. E. W. Dijkstra, *A Discipline of Programming*, Prentice-Hall, New Jersey, 1976.
4. M. Sintzoff, *Inventing Program Construction Rules*, Dans *Constructing Quality Software*, P. Hibbard et S. Schuman, éd., North. Holland, Amsterdam, 1978, p. 471-501.
5. A. Tarski, *A Lattice Theoretical Fixpoint Theorem and its Applications*, Pac. J. Math., vol. 5, 1955, p. 285-310.
6. M. Tchuente, *Sur l'auto-stabilisation dans un réseau d'automates*, R.R., n° 111, 1978, Maths Appliquées Grenoble.
7. A. van Lamsweerde et M. Sintzoff, *Formal Derivation of Strongly Correct Concurrent Programs*, Acta Informatica, vol. 12, 1979, p. 1-31.