Dr. DRAGOŞ VAIDA

Le langage algorithmique ALGOL pour les ordinateurs petits et moyens

Dans cette note, l'on présente les idées principales de la monographie [16] sous presse, le langage algorithmique ALGAMS pour les ordinateurs moyens [15], [17] et un exemple effectif de programmation automatique [12].

Au cours des dernières dix années, le développement de la programmation des ensembles électroniques a subi une transformation considérable, grâce à l'affirmation d'un nouveau principe dans la logique de l'utilisation de ces ensembles — *le principe de la programmation à l'aide des langages algorithmiques.*

Un langage algorithmique est un langage de programmation *indépendant* de la machine que l'on utilise. Pour que la machine soit capable d'exécuter le programme obtenu, il faut qu'il soit traduit en langage machine. Cette conversion est réalisée automatiquement par le calculateur lui-même, grâce à un programme de traduction appelé compilateur. Un programme écrit en langage algorithmique peut être utilisé à différentes machines à calculer, partout où un tel programme compilateur existe.

Par rapport à la programmation en code machine, l'usage des langages algorithmiques rend plus facile à pratiquer les multiples réajustements de la programmation initiale, indispensables pour toute expérimentation ; de cette manière, la programmation n'est plus une longue besogne, fastidieuse et coûteuse.

A présent, le modèle de tout langage algorithmique est le langage ALGOL [1] — [4]. A ce sujet, les lecteurs de la littérature scientifique française trouveront une traduction du rapport originel [5], des monographies [6] — [8] et des manuels préparés par différentes compagnies [9] — [11].

Cette note n'est pas conçue pour exposer l'ALGOL aux programmeurs, mais elle est destinée à ceux ayant une certaine expérience dans ce domaine et voulant écrire un compilateur algébrique. Notre but est celui de donner une orientation dans le domaine de certaines recherches concernant la description des processus de calcul numérique pour les ordinateurs petits et

moyens [12] — [16]. L'on présente surtout les idées principales de la monographie [16], sous presse, le langage algorithmique ALGAMS pour les ordinateurs moyens [15], [17] et un exemple effectif de programmation automatique [12].

1. Dans cette monographie, par rapport au langage ALGOL, on propose certains principes de simplification, aboutissant à un *langage algorithmique simplifié*, primitif au sens de [18]. Ce langage est un sous-langage du langage ALGOL, au sens que tout programme écrit dans le langage simplifié est un programme en ALGOL et a le même sens dans les deux langages. Dans le cas des petits ordinateurs, l'adoption de ces principes nous semble nécessaire.

Les principes de simplification proposés sont les dix suivants :

I. LES CONCEPTS ET LES UNITÉS SYNTAXIQUES DU LANGAGE ALGOL

Le commentaire, la chaîne, l'opérateur logique et l'instruction de procédure sont exclus.

II. L'IDENTIFICATEUR

On utilise l'alphabet majuscule pour former les identificateurs. Une lettre minuscule représente le même symbole que la lettre majuscule correspondante.

III. LA VARIABLE INDICÉE

Une variable indicée doit avoir au plus deux indices ; de cette manière, les tableaux se réduisent aux vecteurs et aux matrices.

IV. L'INDICATEUR FONCTIONNEL

Les indicateurs de fonction se réduisent aux indicateurs fonctionnels $<$identificateurs$>$ ($<$liste des arguments$>$), dont les arguments sont des expressions arithmétiques.

V. L'EXPRESSION ARITHMÉTIQUE

L'opération de division spéciale dite entière $\div$ est exclue. En plus, l'on admet seulement les expressions arithmétiques simples, en excluant le cas dans lequel, parmi les diverses expressions arithmétiques simples, une seule est choisie en fonction des valeurs actuelles de certaines expressions booléennes.

VI. L'INSTRUCTION D'AFFECTATION

L'instruction d'affectation est de la forme $\overline{V} := \overline{E}$, où $\overline{V}$ est une variable et $\overline{E}$ une expression arithmétique.

VII. L'EXPRESSION DE DÉSIGNATION

Une étiquette est composée d'un identificateur. Les instructions sont marquées par au plus une étiquette. L'indicateur d'aiguillage doit avoir un

seul indice. Enfin, les seules expressions de désignation admises sont les expressions de désignation simples.

VIII. L'INSTRUCTION CONDITIONNELLE

L'instruction conditionnelle doit avoir la forme *SI* <relation > *ALORS ALLER A* < expression de désignation >. Dans une telle instruction, les opérateurs de relation admis sont les trois suivants : $<$, $=$, $>$. D'autre part, les expressions booléennes doivent être de la forme d'une relation dont le second membre est 0 (zéro).

IX. L'INSTRUCTION POUR

Dans l'élément d'une liste de *POUR*, la variable sous contrôle doit être une variable simple. D'autre part, les éléments avec *TANT QUE* sont exclus.

X. LA DÉCLARATION

On ne peut avoir ni variables simples rémanentes, ni tableaux rémanents. Une liste d'aiguillage est une suite d'étiquettes séparées par des virgules.

2. Dans la monographie [16], d'autre part, l'on présente un nouveau langage informationnel — *le langage algorithmique réduit* L. Ce langage, adapté aux nécessités des calculs numériques sur de petits ordinateurs, constitue une réalisation particulière des principes de simplification en question. L'adoption de cette variante est *optionnelle.*

En dehors des principes de simplification mentionnés, l'on impose les quatre restrictions suivantes.

1 (I). Les concepts et les unités syntaxiques du langage ALGOL qui ont été exclus sont les suivants : le facteur de cadrage ; l'indicateur de fonction ; l'instruction *POUR* ; l'instruction composée ; le déclarateur ; le bloc.

2 (II). Dans le langage algorithmique réduit L, l'identificateur peut être composé par au plus deux caractères. Le premier caractère qui est une lettre ne peut pas être choisi d'une manière arbitraire parce qu'il intervient comme une déclaration. En effet, cette lettre dénote le type de la variable ou la catégorie de la quantité désignée par l'identificateur (variable simple, variable du type entier, variable indicée, étiquette, identificateur d'aiguillage).

3 (III). Dans le langage algorithmique L, l'on admet un seul indice. L'expression en indice doit être une variable du type entier (simple) ou bien une variable du type entier (simple) $\pm$ < entier sans signe >. Les variables indicées sont du type réel. Dans la déclaration d'un tableau, la borne inférieure doit être 1.

4 (VII). L'on a un seul indicateur d'aiguillage O [*G*].

A propos du langage algorithmique réduit L, il est à remarquer que dans le travail [13], les organigrammes sont introduites de telle manière qu'elles soient des représentations graphiques des programmes écrits dans le langage L. Les éléments constitutifs des organigrammes correspondent aux instructions du langage, partout où une telle correspondence a un sens. En établissant une telle correspondence, on a assuré pour les organigrammes le caractère algorithmique.

A titre d'exemple, citons le problème suivant : *Etant donnée la suite des valeurs* $x_1, x_2, \ldots, x_{100}$, *écrire la description algorithmique dans le langage* L *et faire l'organigramme pour le calcul de la moyenne*

$$A = \frac{1}{100} \sum_{h}^{1,100} x_h$$

et de la variance

$$B = \frac{1}{100} \sum_{h}^{1,100} x_h^2 - A^2.$$

La description algorithmique est la suivante :

DÉBUT A, B, H ; MX [1 : 100] ;
A : = O ;
B : = O ;
H : = 1 ;
U : *LIRE* (MX [H]) ;
A : = A + MX [H] ;
B : = B + MX [H]↑2 ;
SI H — 100 = 0 *ALORS ALLER À* T ;
H : = H + 1 ;
ALLER A U ;
T : A : = 0.01 × A ;
B : = 0.01 × B — A↑2 ;
ÉCRIRE (A) ;
ÉCRIRE (B)
FIN

A cette description algorithmique correspond l'organigramme de la fig. 1

Il est aussi à remarquer que le langage algorithmique réduit L est convenable pour la programmation automatique sur les ordinateurs construits en Roumanie CIFA 102, DACICC 1, MECIPT 2 et CET 500.

3. En passant aux ordinateurs moyens nous nous arrêtons aux travaux du groupe GAMS [1] dans le domaine de la programmation automatique. Con-

[1] GAMS — Grouppa Avtomatizatsii Programmirovaniya dlya Machine Srednevo Tipa (en russe).

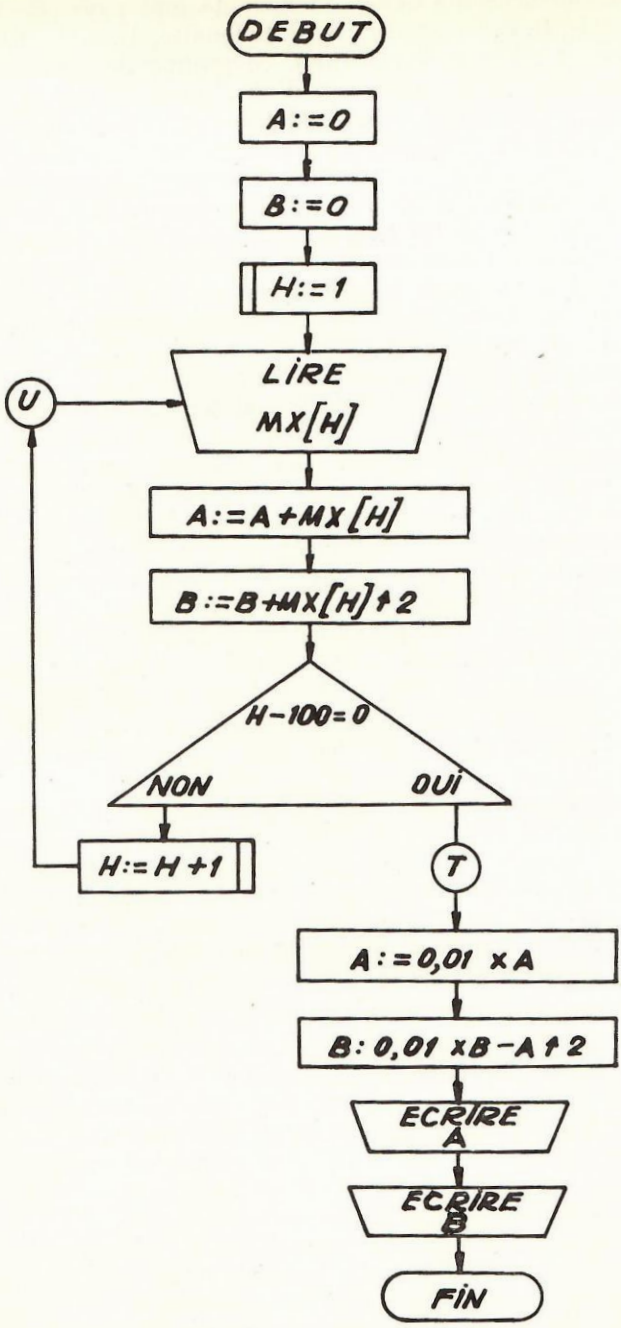
DEBUT
A:=0
B:=0
H:=1
LIRE
MX[H]
U
A:=A+MX[H]
B:=B+MX[H]↑2
H-100=0
NON
OUI
H:=H+1
T
A:=0,01 xA
B:0,01 xB-A↑2
ECRIRE
A
ECRIRE
B
FIN

Fig. 1

stitué par des représentants des Académies de sept pays (R. D. Allemande, R. P. de Bulgarie, R. P. Hongroise, R. P. Polonaise, R. S. de Roumanie, R. S. Tchécoslovaque et l'Union Soviétique), ce groupe de travail s'est proposé comme premier objectif la réalisation d'un standard pour la description algorithmique des processus de calcul, adapté aux calculs numériques sur des ordinateurs moyens. Le langage algorithmique adopté est appelé ALGAMS. Ce langage est le résultat de la discussion d'un projet préparé par un collectif, sous la direction scientifique de M. A. Mazurkiewicz [17].

Quant à la définition du langage ALGAMS par rapport au langage SUBSET ALGOL [19], nous renvoyons le lecteur au rapport originel [17] et à l'étude [15]. Le langage ALGAMS coïncide avec la restriction établie par le groupe de travail WG 2.1 IFIP, à l'exception des quatre points suivants:

A. On exclut les restrictions 3.3.4.3 (l'élévation à la puissance) et 4.7.3.2 (l'appel par nom) de [19].

B. On introduit les notions d'identificateur extérieur et d'identificateur de la mémoire et l'on modifie, par conséquent, la définition de la notion de tableau. Les concepts introduits permettent la description formelle de l'échange d'information entre la mémoire rapide et la mémoire lente.

C. On introduit les notions de part et d'identificateur d'une part et l'on modifie, par conséquent, la définition de la notion de bloc. Les nouvelles notions facilitent le traitement des segments d'un programme.

D. On introduit 11 fonctions et procédures standard, afin de faciliter la programmation et la description formelle de l'emploi de la bibliothèque des programmes standard. Les fonctions standard sont les huit suivantes: DIV(E1, E2), RES(E1, E2), TANG(E), ARCSIN(E), ARCCOS(E), ARC(E1, E2), MAX(E1, E2, ... , EN), MIN(E1, E2, ... , EN).

Les procédures standard sont les trois suivantes: STOP, LIBRARY, COPY.

Ayant en vue ces caractéristiques, on déduit que le langage ALGAMS est une extension du langage SUBSET ALGOL.

En plus, le langage ALGAMS offre des moyens spéciaux qui servent à préciser la structure des informations, tant sur le support externe que dans la mémoire et, par suite, les conversions à appliquer. L'introduction des données se fait au moyen d'une procédure de fonction, l'extraction des résultats au moyen des procédures d'édition et d'impression [20].

4. En conclusion, signalons une recherche appliquée concernant la programmation automatique effective à l'aide d'un calculateur électronique de l'Institut de Physique Atomique de Bucarest [12]. Le travail cité se rattache à la réalisation d'une *routine interprétative* pour les calculs effectués dans l'analyse et la simplification des réseaux électriques.

A l'aide du calculateur donné C_1, l'on simule un autre calculateur C_2, admettant la mise en série et la mise en paralèlle comme opérations fondamentales (v. fig. 2).

Soit R le registre calculateur simulé du C_2. Grâce au programme de simulation, le calculateur C_2 admet comme opérations fondamentales les six suivantes:

LIRE 01-A	Lecture et reproduction en R des contenus des mémoires A, A + 1, A + 2 et A + 3.
SERIE 02-A	Calcul de l'élément équivalent à deux éléments mis en série, le premier étant l'élément contenu dans le

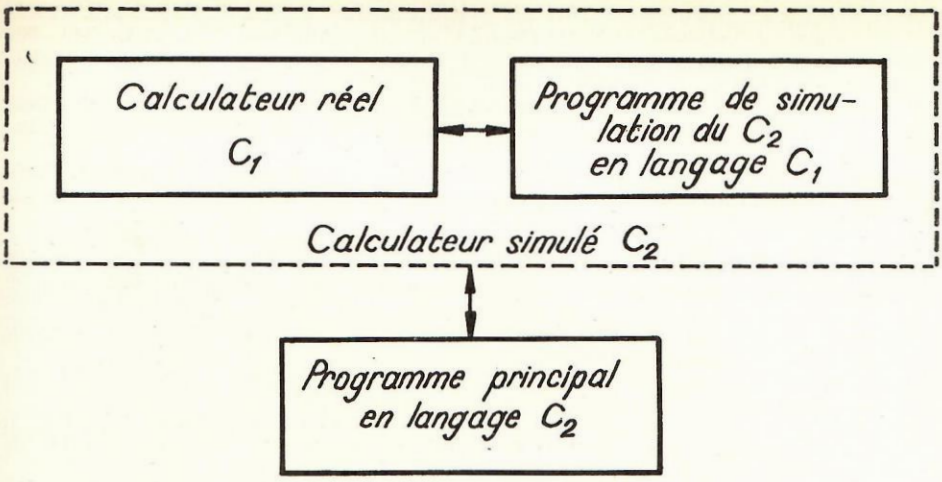

Fig. 2

registre R et le second celui défini par les contenus des mémoires A, A + 1, A + 2 et A + 3.

PARALLÈLE 04-A	Calcul analogue pour la mise en parallèle
ÉCRIRE 11-A	Enrégistrement du R dans les mémoires A, A + 1, A + 2 et A + 3.
IMPRIMER 16-000	Impression en décimal du contenu de R.
FIN 00-000	Arrêt du calculateur.

L'emploi de la routine interprétative réalisée permet l'analyse et la simplification des réseaux énergétiques à l'aide d'un programme que l'on écrit immédiatement, en mettant 02 là où il y a une mise en série et 04 là où il y a une mise en parallèle. De cette manière, le programme n'est qu'une transposition mécanique du réseau considéré.

BIBLIOGRAPHIE

1. Perlis A. J., Samelson K., *Preliminary Report — International Algebraic Language ALGOL.* Comm. ACM 1,12 (1958), 8—22; Num. Math. 1, 1 (1959), 41—60.
2. Backus J. W. *The Syntax and Semantics of the proposed International Algebraic Language of the Zürich ACM-GAMM Conference. Information Processing,* Proceedings of ICIP Paris (15—20 June, 1959), Munich. R. Oldenbourg Butterworths, 1960, 125—132.
3. Naur P. (ed.). *Report on the Algorithmic Language ALGOL 60.* Comm. ACM 3, 5 (1960), 299—314; Num. Math. 2 (1960), 106—136.

4. Naur P. (ed.). *Revised Report on the Algorithmic Language ALGOL 60.* Num. Math. 4(1963) 420—453 ; Comm. ACM 6, 1 (1963), 1—17 ; Computer Jour. 5, 4 (1963), 349—367
5. Genuys F., Poyen J., Vauquois B. *Rapport sur le langage algorithmique ALGOL 60.* Chiffres 3, 1 (1960), 1—44.
6. Bolliet L., Gastinel N., Laurent P. J., *Un nouveau langage scientifique ALGOL.* Manuel pratique. Paris, Hermann, 1964 (Actualités scientifiques et industrielles 1310).
7. Arsac J., Lentin A., Nivat M., Nolin L. *ALGOL-Théorie et pratique.* Paris, Gauthier-Villars, 1965 (Publications de l'Institut de programmation de la Faculté des sciences de Paris).
8. Broise P., *Le langage ALGOL — Application à des problèmes de recherche opérationnelle.* Paris, Dunod, 1965 (Monographies de recherche opérationnelle, IV).
9. Compagnie Européenne d'Automatisme Electronique *Manuel de programmation ALGOL.* Paris, 1965 (NT 1152) (Système de trait ment d'informations CAF 510).
10. Compagnie Européenne d'Automatisme Electronique. *ALGOL Manuel d'uti.isation.* Paris, 1965 (C 65—30) (Série 90).
11. Société Européenne pour le Traitament de l'Information. *Manuel de référence ALGOL.* Paris (No. 10.519).
12. Cavadia I., Meiltz G., Moldovan N., Vaida D., Zamfirescu I. *Metode de programare automată — Rutine interpretative.* Bucarest, Institut de Physique Atomique, 1964 (MC 6).
13. Vaida D. *Operatori și scheme logice pentru descrierea proceselor de calcul.* St. Cerc. Fiz. XVIII, 5 (1966), 525—584.
14. Vaida D. *Un limbaj informațional nou pentru descrierea algoritmică a proceselor de calcul.* St. Cerc. Calcul Econ. Cibernetică Econ. 1 (1966), 67—76.
15. Vaida D. *Limbajul algoritmic elaborat de grupul GAMS.* Aut. Electron. 10, 3 (1966), 133—136.
16. Vaida D. *Descrierea proceselor de calcul pentru calculatoare electronice mici.* Bucarest Ed. Technique, 1967 (sous presse).
17. Определение языка ALGAMS по отношению к SUBSET ALGOL. Varsovie, Institut de Machines Mathématiques de l'Académie Polonaise de Sciences, 1963.
10. Kiyono T., Nagao M. *Comments on the ALGOL System for the small and medium size Computers. Symbolic Languages in Data Processing* (Proc. Symp. ICC Rome, March 26—31, 1962). London, Gordon and Breach Sc. Publ., 1962, 253—262.
19. Report on SUBSET ALGOL 60 (IFIP). Comm. ACM 7, 10 (1964), 626—628.
20. Knuth D. E. (Chairman). *A Proposal for input-output Conventions in ALGOL 60.* Comm ACM 7, 5(1964), 273—283.

THE ALGORITHMIC LANGUAGE ALGOL FOR SMALL AND MEDIUM-SIZE DIGITAL COMPUTERS

The introduction of this paper is a short comment concerning the definition of the algorithmic language ALGOL as a mean to describe numerical processes in a machine-independent and easily readable manner which allows automatic translation into machine program. Assuming that the reader is already familiar with ALGOL, the author defines the purpose of this paper: to summarize five works on the algorithmic languages for small and medium-size digital computers [12] — [16]. This enables the programmer to deal with efficient logical principles when building such a language. In the first part of this paper, the author formulates ten principles of simplification, in the case when ALGOL is used for a small electronic computer [16]. A particular realization of these principles leads to a new informational language — the reduced algorithmic language [16]. The second part of the paper is devoted to this language. In the next part, the author deals with the language ALGAMS for medium-size computers. Four characteristics define ALGAMS with respect to SUBSET ALGOL as an extension [15], [17]. The fourth part contains an example of effective automatic programming [12].

АЛГОРИФМИЧЕСКИЙ ЯЗЫК ALGOL ДЛЯ СРЕДНИХ И МАЛЫХ ЭЛЕКТРОННЫХ СЧЕТНЫХ МАШИН

Введение этой работы представляет относительно краткий комментарий определения алгорифмического языка ALGOL как средства для описания числовых методов, независимо от потребляемой счетной машины. Этот алгоритмический язык, представляющий легко расшифровываемую форму, позволяет автоматический ввод данных в программу машин. Предположив, что читатель уже знаком с ALGOL-ом, автор определяет цель этой работы: выводы по пяти работам об алгорифмическом языке для средних и малых электронных машин [12] — [16].

Так, програмирующему даются эффективные логические принципы для построения определенного языка. В первой части этой работы автор формулирует десять принципов упрощения, используемые при расчетах, в которых ALGOL употребляется как малая электронная машина [16].

Особенностью этих принципов является то, что получаем новый информационный язык — сокращенный алгорифмический язык [16]. Вторая часть работы посвящена этому языку. В следующей части автор занимается языком ALGAMS для средних счетных электронных машин. Четыре свойства определяют ALGAMS как развертывание SUBSET ALGOL-а [15], [17]. Четвёртая часть содержит пример эффективного автоматического программирования [12].