

MÉMOIRES DE LA S. M. F.

JEAN BERSTEL

ARCO pour la programmation arithmétique et combinatoire

Mémoires de la S. M. F., tome 49-50 (1977), p. 11-13

<http://www.numdam.org/item?id=MSMF_1977__49-50__11_0>

© Mémoires de la S. M. F., 1977, tous droits réservés.

L'accès aux archives de la revue « Mémoires de la S. M. F. » (<http://smf.emath.fr/Publications/Memoires/Presentation.html>) implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

ARCO
POUR LA PROGRAMMATION ARITHMETIQUE
ET COMBINATOIRE

par Jean BERSTEL

On présente une première version du langage de programmation ARCO qui est une extension du FORTRAN et qui est conçu pour faciliter la programmation d'algorithmes arithmétiques et combinatoires.

I - Introduction -

L'arithmétique et la combinatoire font appel aux ordinateurs pour effectuer des calculs depuis la naissance de l'informatique. Les programmes écrits pour mener à bien ces calculs rivalisent d'ingéniosité; les algorithmes sont souvent difficiles et leur programmation compliquée. Il apparaît que la complexité de la mise en œuvre pratique des algorithmes réside principalement dans deux faits : d'une part, le volume de certains calculs est tel que, malgré la rapidité des ordinateurs, l'on ne peut arriver à les achever sans se servir de propriétés extrêmement fines des objets considérés et de particularités très spécifiques que l'ordinateur a utilisé; d'autre part, les langages de programmation évolués courants sont très différents des langages mathématiques de description des phénomènes arithmétiques et combinatoires (cf. (1)).

Nous nous proposons de décrire un essai de résolution du deuxième problème par la construction d'un langage de programmation mieux adapté à l'arithmétique et à la combinatoire. Notons tout de suite que les deux faits énoncés ci-dessus ne peuvent recevoir un remède commun.

En effet, plus un langage permet une programmation aisée d'un grand nombre de problèmes, moins il sera apte à tirer avantage des spécificités d'un problème particulier. De même, plus un langage est "évolué" et moins il pourra profiter des facilités de telle ou telle machine. Or ceci constitue un handicap très important dans la mesure où les temps d'exécution sont toujours encore l'obstacle principal de l'achèvement de calculs arithmétiques ou combinatoires. Un langage spécialisé de portée relativement large ne peut donc être conçu que comme un outil de première approche, de "dégrossissage" qui permet de mieux cerner les paramètres importants, ou d'avoir un premier aperçu du problème en question.

2 - Le langage ARCO -

Le langage ARCO est une extension de FORTRAN V. Cette version de Fortran a la particularité de permettre l'accès, en Fortran, à des parties de mots et au bit, répondant ainsi à l'une des exigences formulées en (1).

Outre les propriétés du Fortran, ARCO possède le type ENTLONG qui permet la déclaration de variables ou de tableaux d'entiers de taille arbitraire. L'arithmé-

tique usuelle est étendue à ce type ce qui permet donc l'écriture habituelle d'expressions arithmétiques, de sous-programmes, de commons etc.. utilisant des variables de ce type.

D'autre part, le langage ARCO possède un certain nombre de primitives (fonctions bibliothèque) qui lui sont propres. Y figurent des fonctions d'entrée-sortie adaptées au type ENTLONG, des fonctions booléennes comme DIVISE, PAIR, IMPAIR, PREM (vraie si l'argument est un nombre premier), un quotient avec reste (le quotient entier et le reste sont accessibles par une seule instruction), une décomposition en facteurs premiers, une fonction PGCD, une fonction ALEA qui donne un nombre "au hasard".

Enfin, il a été ajouté, en vue de faciliter l'écriture de programmes, une instruction de somme implicite (et une de produit implicite) qui calcule la somme d'une suite de termes entre les bornes précisées et satisfaisant des conditions. Par exemple, la formule

$$a = \sum_{\substack{i=1 \\ i|n^2}}^n \frac{i^2}{n+i-1}$$

se programme

$$A = \text{SOMME}(I = 1, N \ \& \ \text{DIVISE}(I, N * N) \ \& \ I * I / (N + I - 1))$$

Pour plus de précisions, voir ⁽³⁾.

3 - L'implantation de ARCO -

Le compilateur ARCO est un préprocesseur au compilateur Fortran V Univac qui possède, lui, un optimiseur efficace. Le préprocesseur est écrit en Fortran. Il détecte les instructions spécifiques au langage ARCO et les traduit en une succession d'appels de sous-programmes correspondant à l'exécution de ces instructions. Les instructions purement Fortran sont directement transmises au compilateur Fortran

Citons la représentation des entiers longs; il y en a deux : d'une part, les entiers sont représentés en tableaux sur un nombre fixe de mots ce qui permet une description très rapide du nombre; dans la version actuelle, les nombres ayant moins de 160 chiffres décimaux sont représentés ainsi.

D'autre part, il existe une représentation sur un nombre arbitraire de mots chaînés en liste. Le compilateur choisit a priori la représentation par tableau; en cas de dépassement de capacité, le passage à la représentation par liste se fait automatiquement. Des informations sur le compilateur sont contenues dans ⁽³⁾.

4 - Extensions -

Il y a plusieurs directions dans lesquelles des développements peuvent être envisagés; le désir de souplesse d'emploi ne doit toutefois pas masquer la nécessité d'une exécution rapide, condition sine qua non de l'utilité du langage.

Tout d'abord, le nombre de primitives peut être considérablement élargi, en ajoutant les fonctions arithmétiques usuelles (Euler, Möbius, etc.) et les fonctions

combinatoires standard (binomiaux, Stirling). Cette addition peut se faire de suite sans remaniement du compilateur (voir (3)).

Un soin particulier doit être apporté à la conception des sous-programmes calculant les fonctions standard, en vue d'obtenir une rapidité maximale. En ce qui concerne les opérations arithmétiques, les algorithmes efficaces sont bien connus (4). Pour d'autres fonctions, les évaluations de la "complexité des algorithmes" sont souvent asymptotiques ou pourvues de constantes dont seule la pratique peut évaluer l'incidence.

Il serait utile de disposer, en plus de l'arithmétique en multiprécision, d'une arithmétique des rationnels et d'une arithmétique des entiers modulo. L'approche de (2) d'une arithmétique générale qui permet à l'utilisateur de spécifier la structure arithmétique dans laquelle il désire se placer (polynômes, rationnels, corps quadratiques ...) est très intéressante pour des développements dans cette direction.

REFERENCES

=====

- (1) P. BARRUCAND.- Languages, in A. Birch(ed), Computers in Numbers Theory, Academic Press, (1971), p.423-427.
- (2) M. BORSU.- Adaption du langage FORTRAN au calcul dans une structure algébrique quelconque.- Thèse 3^e cycle.- Orsay (1972).
- (3) D. KANJA, R. RIPP.- Le langage de programmation ARCO.- Rapport de D.E.A.- Paris (1975).
- (4) D.E. KNUTH.- The art of computer programming.- vol. 2.- Seminumerical algorithms Addison Wesley (1969)

J. BERSTEL
Institut de Programmation
Université Pierre et Marie Curie
4, Place Jussieu

75230 - PARIS Cedex 05
