MÉMOIRES DE LA S. M. F.

EVELYNE TOURNIER

Présentation générale de REDUCE-2

Mémoires de la S. M. F., tome 49-50 (1977), p. 195-199

<http://www.numdam.org/item?id=MSMF_1977__49-50__195_0>

Calculateurs en Math. (1975 - LIMOGES)
Bull. Soc. math. France
Mémoire 49-50, 1977, p. 195-199.

PRESENTATION GENERALE DE REDUCE-2

par Evelyne TOURNIER

I - INTRODUCTION -

Pendant les dix dernières années des progrès considérables ont été faits dans le domaine de la résolution des problèmes algébriques par ordinateurs, et de nombreux systèmes ont été produits.

Parmi ces principaux systèmes, nous trouvons :

- FØRMAC [1]
- SYMBAL [2]
- MATHLAB [3]
- REDUCE-2 [4]
- SCRATCHPAD [5]
- SAC-1 [6]
- MACSYMA [10]

On pourra trouver une étude comparative de ces systèmes et leur développement futur dans [7].

REDUCE-2 est un système interactif approprié à la manipulation algébrique. Ce système a été choisi pour la richesse des possibilités qu'il offre, et parce qu'il a une syntaxe bien connue, de type ALGØL, ce qui rend son apprentissage rapide et son utilisation relativement simple.

Les principales possibilités qu'il offre à l'utilisateur sont :

- manipulation des polynômes et fractions rationnelles
- différentiation
- substitution d'expressions et sous-expressions
- simplification des expressions automatique et contrôlée par l'utilisateur
- possibilité d'extension du système (par exemple définition de nouveaux objets).

REDUCE est primitivement conçu pour un travail en mode interactif dans un environnement en temps partagé.

Un programme est donc constitué par une série de commandes dont chacune est interprétée et évaluée par le système, par ordre séquentiel, avant de passer à l'étape suivante.

REDUCE peut évidement être utilisé dans un environnement " batch processing ". Il y a cependant une différence fondamentale entre le mode interactif et le mode " batch ".

- En mode interactif, toutes les fois que le système trouve une ambiguïté en un point quelconque du calcul (par exemple l'omission d'un type dans une affectation), il demande à l'utilisateur, à ce point, quelle est l'interprétation correcte.

- En mode " batch ", ce n'est pas pratique de terminer des calculs dans de tels cas, après corrections, cela nécessite de resoumettre l'ensemble du programme.

REDUCE est écrit dans son propre langage source, et il est basé sur un sous-ensemble de lISP 1.5 [8], ceci rendant son implémentation plus facile.

II - PRINCIPALES POSSIBILITES DU LANGAGE -

Pour une description complète du langage, on pourra se référer à [4]. Présentons simplement les propriétés les plus intéressantes pour le calcul algébrique.

1) Les variables -

Il existe les types suivants :

- INTEGER entier de précision infinie, ce qui permet le travail en rationnels
- REAL
- SCALAR C'est le type de base de la variable symbolique.

2) Les expressions -

a) Simplification :

La simplification est un des problèmes importants des systèmes de calcul formel, et la façon de la traiter est assez variée; on trouvera une classification de ces différentes méthodes dans le papier de J. Moses [9].

Dans REDUCE, on obtient les résultats suivants :

- La simplification automatique : elle a lieu dans toutes les expressions et sous-expressions qui ne sont pas des fractions.

exemple : X(4 * X + Y) -X * * 2 donne X(3 * X + Y)

mais (X + 1)(X + 2)/(X * * 2 + 2 * X + 1) ne donne pas automatiquement (X + 2)/(X + 1), il faut le demander de la façon suivante :

- La simplification contrôlée : Donc, par exemple, dans le cas d'une fraction on utilisera l'option GCD qui applique le PGCD sur le numérateur et dénominateur des expressions.

Cette option GCD peut avoir deux états :

ON GCD; c'est-à-dire en fonction

ØFF GCD; c'est-à-dire hors de fonction

b) Substitution :

La puissance de la substitution est aussi très variable d'un système à l'autre elle est importante dans REDUCE et peut se faire à deux niveaux :

- Substitution locale :

exemple : SUB(X = X + 1, Y = 1, X * * 2 + Y * * 2)

donne $X^2 + 2X + 2$

Cette substitution se fait de la façon suivante :

1. en simplifiant l'expression si nécessaire
2. en remplaçant les variables de la liste par leur substitution
3. en resimplifiant l'expression obtenue si nécessaire.

- Substitution globale :

exemple : LET X = Y * * 2 + 1, H(X, Y) = X -Y;

X sera substitué par Y * * 2 + 1

H(X, Y) sera substitué par X -Y

Remarque -

Soit LET H(X, Y) = X -Y ; cette commande remplace H(X, Y) par X -Y, mais non H(X, Z) ou toute autre fonction de H.

Cependant, ceci est possible de la façon suivante :

exemple : FØR ALL X, Y LET H(X, Y) = X -Y; aura pour effet de remplacer H(X, Y) par X -Y ∀ X, Y dans toutes les expressions où figure H(X, Y).

3) Les opérateurs -

REDUCE utilise deux types d'opérateurs :

- Les opérateurs infixés : qui apparaissent entre leurs arguments. Ce sont tous les opérateurs logiques, relationnels, arithmétiques et symboliques.

(Ex : ØR | AND | MEMBER ... | + | * | CØNS)

- Les opérateurs préfixés : qui apparaissent en tête de leurs arguments.

Exemples : - DF opérateur de dérivation

DF(X * * 2 + LØG(X), X, 2)

- les fonctions trigonométriques

CØS(X + Y)

Remarque -

L'utilisateur peut aussi définir de nouveaux opérateurs, ce qui lui offre un champ d'applications très important.

```
Exemple 1 :   ØPERATØR F;
              FØR ALL U LET F(U) = U * * 3 + 2;
Exemple 2 :   ØPERATØR G;
              FØR ALL Q LET DF (G(Q), Q) = (Q + 1) * * 2
Exemple 3 :   ØPERATØR N, P;
              NØNCØM N;
              P(X) * P(Y) * P(X); donne  P(X)**2 * P(Y)
              N(X) * N(Y) * N(X); donne  N(X) * N(Y) * N(X)
Exemple 4 :   INFIX &
              FØR ALL  U, V LET & (U, V) = U * V + U * * 2;
              U & V; donne  V * U + U * * 2
```

4) Calcul matriciel :

MATRIX est un type de variable, donc nous trouvons :

- des expressions matricielles :

Exemple : MATRIX A, B, C, D;

C : = 1/B; est interprété comme B * * (-1), c'est-à-dire l'inverse de B

D : = A * * 2 + MAT((1, X), (X, Y))/2;

l'opérateur MAT permet de générer une matrice par ligne

- des opérateurs avec des arguments matriciels :

DET A donne le déterminant de la matrice A

TRACE A donne la trace de la matrice A

TP A donne la transposée de la matrice A.

5) Le contrôle de l'exécution :

a) Les options :

Un certain nombre d'options permettent de contrôler à tout instant les entrées, les sorties et l'exécution.

Ces options peuvent être mise à ØN (c'est-à-dire en fonction) ou à ØFF (hors fonction). Elles sont toutes mises à un certain état (par défaut) par le système lui-même et on peut les changer en tout point du programme.

Exemple : ØN INT; signifie que le travail sera exécuté en mode interactif.

ØN FORT; permet de sortir des formules directement exploitables en FØRTRAN.

b) Espace de travail et sauvegarde :

- Au niveau interne, REDUCE utilise la notion d'espace de travail qui est un environnement dans lequel sont sauvegardées les affectations et évaluations.

A tout instant, on peut demander quel est le contenu de cet espace de travail qui est en fait la dernière expression calculée.

- Au niveau externe, on peut, par exemple, sauvegarder une expression sur un fichier pour une utilisation ultérieure.

Ce ne sont là qu'un aperçu des propriétés importantes de ce langage qui sont dues :

- à la possibilité d'extension de la syntaxe (c'est-à-dire que des commandes de REDUCE permettent d'en modifier la grammaire).
- à la possibilité d'extension de la sémantique (par exemple de définition de nouveaux objets).

et ceci est dû au fait que REDUCE est écrit dans son propre langage source.

BIBLIOGRAPHIE

[1] R.G. TOBEY.- FØRMAC-PL/1 interpreter, Users Reference Manual, IBM contributed program library 360 D 03.3004, Hawthrone, New-York.- October 1967.

[2] M. ENGELI.- User's manual for the formular Manipulation language SYMBAL.- The University of Texas at Austin computer center.- Austin, Texas (1968)

[3] C. ENGELMAN.- MATHLAB 68 In A. J. Morell (Ed.), Information Precessing 68.- North-Holland Amsterdam (1969) p. 469-467.

[4] A.C. HEARN.- REDUCE 2 USER'S manual University of UTAH, Salt Lake City, March 1973.

[5] F.W. BLAIR, J.H. GRIESMER and R.D. JENKS.- An interative facility for symbolic mathematics, IBM Research Report. No. RC2766 January 1970.

[6] G.F. COLLINS.- The SAC-1 system : An introduction and syrvey Proc. 2nd symposium on symbolic and algebraic manipulation.- (Assoc. Comp. mach, N.Y1971 144-152).

[7] Y. SUNBLAD.- Symbolic mathematical systems Now and in the Future.- (SIGSAM bulletin Proceeding of Eurosam 1974 p. 1 - 8).

[8] Mc CARTY J., ABRAHAMS P.W., EDWARDS D.J., HART T.P., LEVIN M. I.- LISP 1.5 programmer's manual MIT, Press CAMBRIDGE, MASS.- (Aug 1962).

[9] J. MOSES.- Algebraic simplification, A guide to the Perplexed.- Proceeding of the 2nd symposium on Symbolic and Algebraic Manipulation §March (March 23-25 1971).

[10] J. MOSES.- MACSYMA primer - Project MAC - M.I.T. (1973).

Evelyne TOURNIER
U.E.R. de Math appliquée et Informatique
Université Scientifique et Médicale
de GRENOBLE
B.P. 53 38041 GRENOBLE CEDEX