NOUVELLE ARITHMÉTIQUE D'ACCÈS MÉMOIRE

C. Jousselin

Que cela soit les langages de programmation par type abstrait, c'est-à-dire la forme la plus moderne de la programmation structurée, ou les langages dits d'intelligence artificielle tels que Lisp, Prolog ou Smalltalk, les maîtres-mots de ces nouvelles formes de conception de logiciels sont : structure de données, structure de programmes, programmation par les données. Ces techniques sont poussées à l'extrême par les langages d'intelligence artificielle où il n'existe plus aucune différence de forme entre données et programmes. Le concept de type abstrait de données utilise la notion de type générique, qui permet de définir de nouvelles structures et d'accéder aux informations qu'elles renferment à l'aide de sélecteurs (fonction de projection).

Cette tendance générale à la structuration est due à un besoin de méthode de conception et d'augmentation de la lisibilité des logiciels pour en faciliter leur maintenance ; cela se traduit par une augmentation du temps d'accès aux informations stockées en mémoire.

Les instructions et les données ont pour dénominateur commun leur stockage en mémoire. Au lieu d'essayer d'accélérer les procédures existantes d'accès aux informations à structure complexe, avec la même structure mémoire, on s'est intéressé à la modification de la structure d'adressage de la mémoire elle-même, afin qu'elle soit plus proche des structures d'informations qui y seront stockées [1]. Il s'ensuivra naturellement une simplification des mécanismes d'accès dont on peut espérer un gain de temps significatif.

L'étude menée à LEP porte sur la modélisation mathématique des mécanismes d'accès à des structures d'informations, en particulier l'accès à la mémoire centrale, à l'aide d'une nouvelle arithmétique d'accès, basée sur la théorie des groupes. Quelle est la structure mathématique cachée de la mémoire centrale ? Comment est-elle vue par l'organe de traitement de l'information qu'est l'unité centrale ? Ce sont les questions fondamentales à se poser face à ce besoin crucial de stockage d'informations structurées.

La mémoire centrale est un ensemble fini d'éléments mémoire. Sa structure est une structure de groupe, au sens mathématique de la théorie des groupes [2-5], qui est imposée par une loi de translation des adresses. Les modes d'adressage, servant à la navigation dans la mémoire, utilisent cette loi (fig. 1). La mémoire ne contient donc pas en elle sa structure, l'adresse associée à chaque élément mémoire n'est

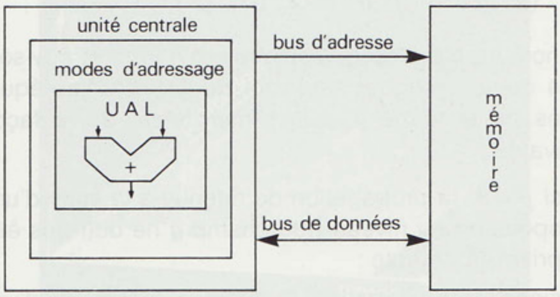

Fig. 1. — Loi de translation dans son environnement.

qu'un identificateur : il n'existe aucun lien (aucune loi), a priori, entre les adresses notées 00, 01 ou 10. Actuellement, la loi de translation d'adresse, implantée généralement dans une unité arithmétique et logique (UAL), n'est autre que la loi d'addition des entiers, ce qui confère à la mémoire une structure de groupe cyclique (donc commutatif), dont le graphe associé est un cercle (mémoire bien improprement appelée plan mémoire, on devrait plutôt parler de cercle mémoire). La configuration de la mémoire centrale détermine les structures de données et de programmes simples et rapides d'accès. Avec la configuration en cercle de la mémoire actuelle, ces structures sont des tableaux à une dimension (arcs de cercle).

Le cercle mémoire peut se représenter à l'aide de la notation des groupes par générateurs et relations :

$$\{t \mid t^n = t^0\} ,$$

où t correspond au champ d'adressage et n est le nombre d'éléments de la mémoire. Un élément mémoire d'adresse i se notera t^i. Une translation d'adresse de valeur j à partir de l'élément d'adresse i doit être considérée comme la composition des éléments t^i et t^j par la loi de translation, notée par un point (.) :

$$t^i \,.\, t^j = t^{i+j} ,$$

où + est la loi d'addition des entiers.

Le concepteur de logiciel utilisant, par exemple, une structure d'arbre binaire aimerait parcourir ses arbres à l'aide des sélecteurs droit et gauche (respectivement d et g). Or ces sélecteurs ne commutent pas entre eux : l'adressage droit puis gauche n'est pas équivalent à l'adressage gauche puis droit. Pour réaliser un tel mécanisme d'accès non commutatif, il faut au minimum utiliser une structure de groupe à deux générateurs correspondant aux deux types de déplacements d et g. Un élément mémoire d'adresse $a = (i, j)$ se notera $(g^i \,.\, d^j)$, où g et d sont deux parties du champ d'adressage.

Par mesure de simplification, on ne présentera qu'un type de loi générale de translation d'un groupe à deux générateurs :

$$\{g, d \mid d^n = d^0, g^m = d^p, (d^1 \,.\, g^1) = (g^1 \,.\, d^y)\} ;$$

le nombre d'éléments mémoire est $n \times m$, et p, y sont des caractéristiques de la loi du groupe. Les équations de la forme $g^i = d^j$ s'interprètent de la façon suivante :

- si $j = 0$, la propagation de retenue à la suite d'une opération au niveau i du champ g ne doit pas être prise en compte ;
- sinon, la propagation de retenue à la suite d'une opération au niveau i du champ g doit se faire au niveau j du champ d.

La loi générale de translation est :

$$(g^i \,.\, d^j) \,.\, (g^k \,.\, d^l) = (g^{i+k} \,.\, d^{j \times y^k + l}),$$

$+$ et $\times$ étant respectivement les lois d'addition et de multiplication des entiers.

Les opérations sur les champs de bits à effectuer par l'UAL sont : pour le champ g : $i + k$, pour le champ d : $j \times y^k + l$.

Pour illustrer l'utilisation d'une telle structure mémoire, la figure 2 montre que l'on peut stocker deux arbres de sept éléments ; la figure 3 montre la simplicité d'un tel dispositif d'adressage.

Cette nouvelle arithmétique d'adressage trouve d'autres applications que l'adressage des structures de données complexes. Elle peut être utilisée pour adresser les nœuds d'un réseau d'automates, si le graphe du réseau est un graphe de groupe.

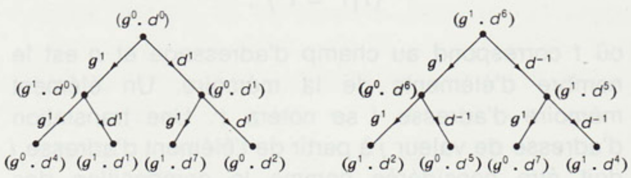

Fig. 2. — Deux arbres de sept éléments dans le groupe

$$\{g, d \mid d^8 = d^0, g^2 = d^4, (d \,.\, g) = (g \,.\, d^{-1})\}$$

où $d^{-1} = d^7$.

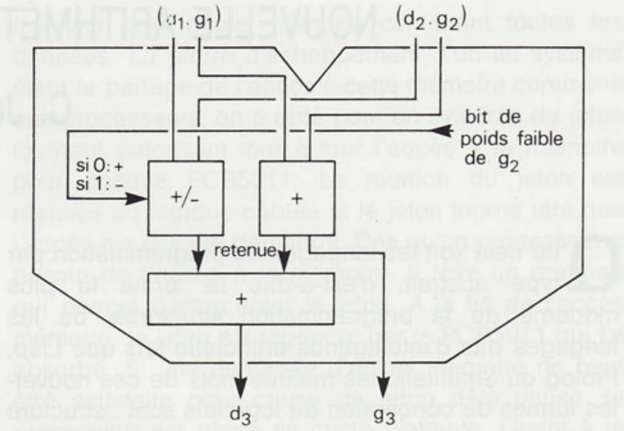

Fig. 3. — Dispositif d'adressage associé à un groupe de la forme :

$$\{d, g \mid d^N = d^0, g^M = d^P, (d \,.\, g) = (g \,.\, d^{-1})\} .$$

Cette modélisation des mécanismes d'adressage est une généralisation des mécanismes existants (adressage direct et paginé), adaptable à l'accès aux structures d'informations complexes.

L'auteur remercie J.M. Nicolas et J.P. Moskowitz pour leur contribution à la mise au point de cette théorie.

1 FENWICK (P.M.). *Addressing operations for automatic data structure accessing.* Computer Architecture News, 12, 1, (1984), 44-57.

2 MAGNUS (W.), KARRAS (D.). *Combinatorial group theory.* Solitar, Interscience ed., New-York, (1966).

3 MUTAFIAN (C.). *Le défi algébrique,* tome 1. Vuibert, Paris, (1975).

4 BABAI (L.), KANTOR (W.M.), LUKS (E.M.). *Computational complexity and the classification of finite simple groups.* IEEE 1983 Foundations of Computer Science, IEEE Cat. 83CH1938-0, (1984), 162-171.

5 BABAI (L.), SZEMERÉDI (E.). *On the complexity of matrix group problems.* IEEE 1984 Foundations of Computer Science, IEEE Cat. 84CH2085-9, (1985), 229-250.