Typer une collection par la présentation d'un groupe

Olivier Michel et Jean-Louis Giavitto

LRI u.r.a. 410 du CNRS, Bâtiment 490, Université de Paris-Sud,
F-91405 Orsay Cedex, France.
email: {michel|giavitto}@lri.fr

Résumé: Nous rapportons ici les premiers travaux concernant l'extension de la notion de *collection* en 81/2.

81/2 est un langage déclaratif permettant la définition de *tissus* par des équations. Un tissu est soit une séquence temporelle de collections soit une collection de séquences. Une collection est un aggrégat de données manipulé comme un tout. Par exemple les vecteur en APL sont des collections mais pas les vecteurs en C car les opérations sur les vecteurs passent par l'accès à la valeur d'un élément. Un tissu est une structure de données particulièrement adaptée à la représentation de la trajectoire d'un système dynamique. La simulation des systèmes dynamiques constitue une grande part des applications des supercalculateurs et les collections sont un support naturel pour l'expression puis l'exploitation du parallélisme de données.

Du point de vue de ces applications de simulations, les collections sont essentiellement utilisée pour modéliser la discrétisation d'un morceau d'espace : par exemple on utilise un vecteur pour enregistrer la variation de température aux points de discrétisation d'une barre dans la résolution numérique d'un problème de diffusion de chaleur. Les collections, manipulées comme des tout, sont particulièrement adaptées à la simulation de phénomènes physiques car les mêmes lois s'appliquent en chaque point de l'espace, ce qui se traduit par des calculs qui peuvent s'exprimer globalement sur les collections. On trouve dans les langages de programmation toutes sortes de structures de collections : l'ensemble (SETL), le multi-ensemble (GAMMA), le vecteur (*LISP), l'imbrication de vecteurs (NESL) et bien sur le tableau multidimensionnel (APL, HPF, MOA, multidimensional LUCID, 81/2...).

La structure de tableau est sans doute une des plus expressive pour représenter la discrétisation d'un morceau d'espace car elle correspond naturellement à une discrétisation en grille euclidienne. Mais elle présente plusieurs gros inconvénients : par exemple une opération naturelle dans un espace, qui est d'aller d'un point à un des points voisins, doit être codée en terme de manipulations d'index ; la topologie de l'espace discrétisé n'est pas explicite et la représentation de topologies plus générales que la grille euclidienne est loin d'être évidente ; un tableau correspond à une région connexe de l'espace dont les bornes doivent être fixées à l'avance ; etc.

C'est pourquoi nous essayons d'étendre la notion de tableau vers une notion plus générale de *champs de données* sur un espace discret dont la géométrie est spécifiée par une structure de groupe. La géométrie est décrite par la présentation finie d'un groupe (pour l'instant nous nous restreignons aux groupes abéliens pour lesquels il existe de nombreux outils effectifs). Les principales opérations data-parallèles s'étendent à ce nouveau type de collections. Par exemple, le groupe agit sur la collection comme un ensemble de permutations particulières qui s'interprètent comme des déplacements privilégiés (i.e. des communications privilégiées dans le cas d'une implantation data-parallèle), les générateurs du groupe constituant les déplacements élémentaires. La réduction nécessite la donnée d'un axe de réduction, qui généralise l'opérateur d'axe

Journées du GDR Programmation.
22, 23 et 24 novembre 1995.
Grenoble.

existant en APL, et qui correspond à un sous-groupe (un sous-groupe normal dans le cas des géométries non-abéliennes) plutôt qu'à une dimension. Un théorème de structure sur les groupes abéliens permet d'implanter ces groupes comme des tableaux modulo un changement de générateurs.

Une des difficultés à laquelle se heurte le développement d'une notion de *collection orientée groupe* est le problème de la spécification d'un sous-domaine fini dans un groupe non-fini. Par ailleurs, même en se restreignant à une structure classique de tableau, la simulation d'un processus de croissance (comme par exemple la croissance d'une plante) se trouve (trop) limitée par les contraintes imposées par des frontières spécifiées par des bornes homogènes suivant chaque dimension. Une solution évidente est de développer une structure de collections paresseuses, à la manière des listes paresseuses qu'on trouve par exemple dans HASKELL. Les collections deviennent alors des collections potentiellement infinies dont les valeurs définies sont toutes dans un support borné dans un espace spécifié par la présentation d'un groupe.