



Revue de mathématiques spéciales, Volume 108, Numéros 6 à 10

- Author : B. Niewengłowski
- Publisher : Librairie Vuibert, 1998
- pages : 753 pages
- N° Class : 621/897

Le langage de manipulations symboliques Scheme fait partie de la famille Lisp. Il a été choisi depuis 1980 par les enseignants du MIT et le nombre de ses adeptes n'a fait qu'augmenter. Il fait maintenant son entrée dans les premiers et deuxièmes cycles des universités françaises. Ce simple fait en dit long sur les qualités de Scheme comme support à l'enseignement de la programmation et de ses fondements. Cet ouvrage de référence est conçu pour être utilisable avec profit aussi bien par un lecteur débutant que par un lecteur plus expérimenté. La richesse de son contenu permet de le recommander tout au long d'un cursus en informatique, du DEUG au DEA. C'est non seulement un livre sur la programmation fonctionnelle avec Scheme, mais plus généralement un excellent ouvrage d'accompagnement pour des cours sur l'intelligence artificielle, la logique, la sémantique, la compilation,... Les exemples utilisés ne sont pas de simples exemples d'école. Ils sont pertinents, significatifs et parfois même amusants. Le livre est structuré en deux parties : La première partie introduit progressivement le lecteur aux joies de la programmation avec Scheme (ou Lisp). Y sont expliqués les différents styles de programmation : fonctionnel, symbolique, par continuations, par objets, impératif, par macros, par flots,... Les méthodes récursives y jouent un rôle central, aussi l'auteur s'est efforcé d'en dissiper les mystères en les présentant sous une grande variété de points de vue. Chaque chapitre comporte des compléments sur des sujets variés : fractals, jeu donjon et dragon, contraintes et CAO, extension objet de Scheme, simulation et coroutines, visualisation en 3D, calcul formel, vie artificielle.... Ils ont aussi pour but d'exercer le lecteur à l'écriture de programmes de taille plus importante. De nombreux exemples et exercices permettent au lecteur de progresser à son rythme. La seconde partie explique la théorie et l'implantation des concepts de base en programmation : machines et automates finis, analyse lexicale et syntaxique, formatage de texte ou de formules, systèmes experts et Prolog, systèmes déductifs, lambda calcul, sémantique, interprétation, compilation.... La possibilité de programmer puis d'expérimenter une théorie est l'une des meilleures façons de l'assimiler, d'où le slogan de l'auteur : " Implanter pour mieux comprendre. "