

UNIVERSITE SCIENTIFIQUE ET MEDICALE
et INSTITUT NATIONAL POLYTECHNIQUE
de GRENOBLE

MATHÉMATIQUES APPLIQUÉES ET INFORMATIQUE

Laboratoire associé au CNRS n°7

B.P. 53 - 38041 GRENOBLE cedex France

TRANSFORMATION DE PROGRAMMES RECURSIFS

G. VEILLON

RR. n° 22 - Novembre 1975

IST
128

RAPPORT DE RECHERCHE

Résumé

Ce rapport présente une technique pratique de transformations de programmes récursifs, permettant notamment de réaliser ces programmes dans un langage non récursif. Le principe de transformation est de préserver les états du programme en utilisant l'invariance de certaines pré et post conditions.

Abstract

This report presents a practical method to transform recursive programs, especially for a realization in a non recursive language. The principle of the transformation is to preserve states of the program by the use of some pre and post conditions.

INTRODUCTION

L'utilisation de la récursivité a souvent été considérée comme une coutume universitaire peu réaliste. L'élégance et la simplicité d'écriture ne compensaient pas, aux yeux des programmeurs, le coût d'une exécution complexe, et l'utilisation d'un langage peu courant dans les applications. En pratique, l'application d'un algorithme récursif dans un langage n'ayant pas cette propriété pose souvent des problèmes résolus empiriquement.

Or il est clair que l'écriture récursive est particulièrement satisfaisante dans une construction "descendante" de programmes. Elle permet d'accéder facilement à un programme sûr, et évite de traiter au même niveau les problèmes de gestion de pile et la description générale de l'algorithme.

Nous proposons ici une technique de transformation d'algorithmes permettant de construire systématiquement une forme non récursive, et de rechercher une réduction de la complexité. Pour cela nous rappellerons tout d'abord l'axiomatique de Floyd et Hoare, que nous utiliserons dans l'étude de la validité de certains programmes récursifs.

Partant de trois exemples très classiques, nous étudierons les transformations de certains programmes récursifs en programmes itératifs, l'élimination des paramètres et l'élimination des appels récursifs.