Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

348

P. Deransart B. Lorho
J. Małuszyński (Eds.)

Programming Languages Implementation and Logic Programming

International Workshop PLILP '88
Orléans, France, May 16–18, 1988
Proceedings

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo

Editors

Pierre Deransart
INRIA-Rocquencourt, Domaine de Voluceau
B.P. 105, F-78153 Le Chesnay Cedex, France

Bernard Lorho
Université d'Orléans, Faculté des Sciences
Laboratoire d'Informatique Fondamentale (LIFO)
B.P. 6759, F-45067 Orléans Cedex 2, France

Jan Maluszyński
Department of Computer and Information Science
Linköping University
S-58183 Linköping, Sweden

CR Subject Classification (1987): F.4.1–2, D.3.1, D.3.4, F.3.3, I.2.3

ISBN 3-540-50820-1 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-50820-1 Springer-Verlag New York Berlin Heidelberg

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

PREFACE

PLILP '88, the first international Workshop on Programming Languages Implementation and Logic Programming, was held from May 16 to May 18, 1988 in Orléans. PLILP '88 has been organized by the Laboratoire d'Informatique Fondamentale d'Orléans (LIFO-Université d'Orléans) and Institut National d'Informatique et d'Automatique (INRIA-Rocquencourt).

The aim of the workshop was to discuss whether research on the implementation of programming languages and research on logic programming can mutually benefit from each other's results.The intention was to bring together researchers from both fields, especially those working in the area of their intersection.

Problems such as formal specification of compilers and syntax-based editors, program analysis and program optimization have been traditionally studied by implementors of algorithmic languages and have resulted in a number of well-established notions, formalisms and techniques. At the same time, an increasing number of people use logic programming as a way of specifying compilers or other programming environment tools, taking advantage of the relatively high level of logic programming and the growing efficiency of Prolog implementations.

On the other hand, research on logic programming raises the questions of analysis of logic programs and their optimization. These are motivated primarily by compiler construction for logic programs, by studies on the methodology of logic programming and by the attempts to amalgamate logic programming and functional programming.

Research in the field of logic programming, including its applications to the implementation of other programming languages, may or may not refer to the well-known results of the other field. In the first case the field of logic programming may benefit from these results. For example application of LR parsing techniques may contribute to a more efficient implementation of definite clause grammars. On the other hand, techniques of logic programming may contribute to the development of the other field. As an example, one may consider the use of logic programs for compiler specification.

The purpose of the workshop was to review the techniques developed in one (or both) of the fields which could also be of some help in the other one and to facilitate the transfer of expertise. It seems important to compare notions used in both fields : pointing out similarities between them may prevent rediscovering results already known, while studying the differences may contribute to the transfer of technology.

The workshop consisted of a series of invited talks and a panel discussion. This book presents some of the most significant talks.

We gratefully acknowledge the financial support provided by the following institutions :
- INRIA,
- Université d'Orléans,
- GRECO de Programmation et Outils pour l'Intelligence Artificielle du CNRS.

Le Chesnay, Orléans, Linköping
December 1988

Pierre Deransart
Bernard Lorho
Jan Maluszynski

Table of Contents

Attribute Grammars in Logic Programming

Logic Programming for Programming Environments