Parallel & Constraint Logic Programming

PARALLEL AND CONSTRAINT LOGIC PROGRAMMING:

An Introduction to Logic, Parallelism and Constraints

by

Ioannis Vlahavas
Aristotle University of Thessaloniki, Greece

Panagiotis Tsarchopoulos
European Commission, Belgium

Ilias Sakellariou
Aristotle University of Thessaloniki, Greece

KLUWER ACADEMIC PUBLISHERS
Boston / Dordrecht / London

Distributors for North, Central and South America:
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA
Telephone (781) 871-6600
Fax (781) 871-6528
E-Mail <kluwer@wkap.com>

Distributors for all other countries:
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS
Telephone 31 78 6392 392
Fax 31 78 6546 474
E-Mail <orderdept@wkap.nl>


Electronic Services <http://www.wkap.nl>

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

Printed on acid-free paper.

Printed in the United States of America

PARALLEL AND CONSTRAINT LOGIC PROGRAMMING:

An Introduction to Logic, Parallelism and Constraints

THE KLUWER INTERNATIONAL SERIES
IN ENGINEERING AND COMPUTER SCIENCE

to Chrysoula
Ioannis

to Mercedes
Panagiotis

to Katerina
Ilias

The most popular representative of logic programming languages nowadays is Prolog, interest in which is increasing in the recent years, since it offers significant advantages for the development of applications involving symbolic computation and reasoning. Unfortunately, Prolog's approach to logic programming seems to have a major disadvantage concerning its application to "real world" problems: inefficiency. In order to overcome this problem two different routes have been followed by the research community. The first involves the execution of Prolog programs in parallel. The second is Constraint Logic Programming, a significant extension, or better a new logic programming paradigm based on Prolog-like language i.e. first-order Horn Clauses. While the former aims to increase Prolog's efficiency in general, the latter focuses on the area of combinatorial search problems, an area where Prolog seems to be highly inefficient.

Exploiting the parallelism that is naturally embedded in the declarative semantics of logic programs seems to be the most promising approach of implementing parallel computer systems. This approach offers a number of significant advantages, as for example, parallelization without explicit annotations by the user, which leads to an easy to program parallel computational model. Techniques for executing Prolog in parallel are still an active area of research worldwide and have led to many promising results and to a number of successful implementations that have opened the way for the use of Prolog in real life applications.

Constraint Logic Programming (CLP), an area of extreme research interest in the recent years, extends the semantics of Prolog in such a way that the combinatorial explosion, a characteristic of most problems in the field of Artificial Intelligence, can be tackled efficiently. By employing dedicated to each domain solvers, instead of the unification algorithm, CLP drastically reduces the search space of the problem and thus leads to increased efficiency in the execution of logic programs. CLP offers the possibility of solving complex combinatorial problems in an efficient way, and at the same time maintains the advantages offered by the declarativeness of logic programming.

The aim of this book is to present parallel and constraint logic programming, offering a basic understanding of the two fields to the newly introduced to the area reader. Before going into the discussion of these two extensions, the reader has to be familiar with the fundamentals of conventional logic programming. The first part of the book gives this introduction to the fundamental aspects of conventional logic programming, necessary for the understanding of the parts that follow. The second part includes an introduction to parallel logic programming, architectures and implementations proposed in the area. Finally, the third part presents the principles of the constraint logic programming. The last two parts also include descriptions of the supporting facilities for the two paradigms in two popular systems: ECLiPSe and SICStus. These platforms have been selected mostly because they offer both parallel and constraint features. Commented and explained examples are also included in the relevant parts, offering a valuable guide and a first practical experience to the reader. Finally, applications of the covered paradigms are presented.

The authors felt that a book of this kind should provide some theoretical background, necessary for the understanding of the covered logic programming paradigms, and a quick start to the reader interested in writing parallel and constraint logic programming programs. However it is out of the scope of this book to provide a deep theoretical background of the two areas. In that sense, this book is addressed to a public interested in obtaining a knowledge of the domain, without going through the time and effort consuming phase of understanding the extensive theoretical work done in the field, namely postgraduate and advanced undergraduate students in the area of logic programming.

We believe that the book fills a gap in the current bibliography since, to our knowledge, there does not exist a comprehensive book of this level that covers the areas of conventional, parallel, and constraint logic programming.

Acknowledgments

We would like to thank Professor Ahmed K. Elmagarmid who encouraged us to write this book. We wish also to thank Dr. Petros Kefalas, Mr. Christos Varelas and Mr. Ioannis Refanides for their constructive comments on the final draft of this book.

Contents

1 INTRODUCTION

The meaning of programs, expressed in conventional languages, is defined in terms of the behavior they invoke within the computer. The meaning of programs expressed in logic, on the other hand, can be defined in machine independent, human oriented terms.

The semantics of a statement in a procedural language is quite complex, because the meaning is both dependent on a context and indicates a modification to the context. In a procedural language the effect of a statement depends on the statements executed before it, and at the same time it changes the context for all statements that follow. Declarative programming, on the other hand, allows parts of a program to be examined and understood in isolation; statements are context independent.

A critical property of a programming language is its level of abstraction. Separating the meaning of a program from any particular computational model, gives the freedom to pick alternative implementations of a program. Stating what is to be computed without how it is to be computed, encourages the programmer to think about the intend of a program and the description of relationships in it, without having to worry about changes in the storage of the computer.

Separation of the logic of a program from the control is a basic property of the logic based approach to programming. Its declarative power and reduced development time, for most application areas, are also very appealing properties; unfortunately increased execution time compared with that of imperative

programming languages, have drastically reduced the scope of application of languages like Prolog.

One solution to Prolog's inefficiency problem is provided by executing logic programs in parallel. Research on parallel execution models has been contacted since the beginning of the eighties and has reported significant results. A large number of models has been proposed in the literature, that aim to exploit mainly two types of parallelism, AND- and OR-parallelism, either by exploiting each type separately or both simultaneously.

Parallelism approached through logic programming offers a number of advantages compared to that of imperative languages. One of the most important is that parallel execution can be achieved without the use of explicit annotations from the programmer. This fact allows programs developed originally for sequential systems to execute with no (or minor) modifications in parallel.

On the other hand, the generalization of unification, one of Prolog's most basic operations, gave rise to a new powerful paradigm: constraint logic programming (CLP). It dramatically improves Prolog's performance in combinatorial problems and allows the application of logic programming to numerous real world applications. The combination of the declarative style of programming with high performance, made CLP one of the most promising fields of logic programming today, as indicated by the extensive research effort put to it by the logic programming community.

Without doubt the above areas of logic programming present great scientific and practical interest. This book is an effort to present in a simple manner, both parallel and constraint logic programming, through descriptions of the research results in each area. In order to illustrate the practical applications of the above, case studies of two successful implementations are presented: SICStus and ECLiPSe. We have selected these two from a large number of available systems for two reasons: first, both these systems support parallelism and constraints; second, these are well known, robust and efficient logic programming systems, and are the outcome of years of research in the academic institutes that developed them.

The book is organized in three parts (chapters). The first part begins with a brief presentation of fundamental issues of mathematical logic; both propositional and first order predicate logic are presented. The part continues by describing basic notions of logic programming, and, naturally, moves to the description of Prolog, the major representative of logic programming today. It ends by providing some simple examples of Prolog programs, to illustrate both its use and power.

The second part is dedicated to the presentation of the parallel execution of logic programs. It contains descriptions of the implementation problems and models introduced to solve them, concerning the two major forms of parallelism in logic programs: OR-parallelism and AND-parallelism. The efforts for the successful implementation of systems aiming to exploit AND and OR parallelism simultaneously are presented. Finally, the parallel features of ECLiPSe and SICStus are discussed together with some examples illustrating their use.

Constraint logic programming is discussed in the last part. The fundamental idea of this logic programming paradigm is presented through the use of an example in the first section of the part. The presentation continues by introducing some basic ideas in CLP, such as declarative and operational semantics, constraint solving algorithms and implementation issues. As in the previous part, the CLP features of ECLiPSe and SICStus are presented with some examples that demonstrate their use. Finally, applications of CLP to various fields are described and the part ends by recording future trends in the area.