Studies in Computational Intelligence, Volume 24

Editor-in-chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 8. Srikanta Patnaik, Lakhmi C. Jain, Spyros G. Tzafestas, Germano Resconi, Amit Konar (Eds.)
Innovations in Robot Mobility and Control, 2005
ISBN 3-540-26892-8

Vol. 9. Tsau Young Lin, Setsuo Ohsuga, Churn-Jung Liau, Xiaohua Hu (Eds.)
Foundations and Novel Approaches in Data Mining, 2005
ISBN 3-540-28315-3

Vol. 10. Andrzej P. Wierzbicki, Yoshiteru Nakamori
Creative Space, 2005
ISBN 3-540-28458-3

Vol. 11. Antoni Ligęza
Logical Foundations for Rule-Based Systems, 2006
ISBN 3-540-29117-2

Vol. 12. Jonathan Lawry
Modelling and Reasoning with Vague Concepts, 2006
ISBN 0-387-29056-7

Vol. 13. Nadia Nedjah, Ajith Abraham, Luiza de Macedo Mourelle (Eds.)
Genetic Systems Programming, 2006
ISBN 3-540-29849-5

Vol. 14. Spiros Sirmakessis (Ed.)
Adaptive and Personalized Semantic Web, 2006
ISBN 3-540-30605-6

Vol. 15. Lei Zhi Chen, Sing Kiong Nguang, Xiao Dong Chen
Modelling and Optimization of Biotechnological Processes, 2006
ISBN 3-540-30634-X

Vol. 16. Yaochu Jin (Ed.)
Multi-Objective Machine Learning, 2006
ISBN 3-540-30676-5

Vol. 17. Te-Ming Huang, Vojislav Kecman, Ivica Kopriva
Kernel Based Algorithms for Mining Huge Data Sets, 2006
ISBN 3-540-31681-7

Vol. 18. Chang Wook Ahn
Advances in Evolutionary Algorithms, 2006
ISBN 3-540-31758-9

Vol. 19. Ajita Ichalkaranje, Nikhil Ichalkaranje, Lakhmi C. Jain (Eds.)
Intelligent Paradigms for Assistive and Preventive Healthcare, 2006
ISBN 3-540-31762-7

Vol. 20. Wojciech Penczek, Agata Półrola
Advances in Verification of Time Petri Nets and Timed Automata, 2006
ISBN 3-540-32869-6

Vol. 21. Cândida Ferreira
Gene Expression on Programming: Mathematical Modeling by an Artificial Intelligence, 2006
ISBN 3-540-32796-7

Vol. 22. N. Nedjah, E. Alba, L. de Macedo Mourelle (Eds.)
Parallel Evolutionary Computations, 2006
ISBN 3-540-32837-8

Vol. 23. M. Last, Z. Volkovich, A. Kandel (Eds.)
Algorithmic Techniques for Data Mining, 2006
ISBN 3-540-33880-2

Vol. 24. Alakananda Bhattacharya, Amit Konar, Ajit K. Mandal
Parallel and Distributed Logic Programming, 2006
ISBN 3-540-33458-0

Alakananda Bhattacharya
Amit Konar
Ajit K. Mandal

Parallel and Distributed Logic Programming

Towards the Design of a Framework for the Next Generation Database Machines

With 121 Figures and 10 Tables

Springer

Dr. Alakananda Bhattacharya
Artificial Intelligence Laboratory
ETCE Department
Jadavpur University
Calcutta 700032
India
E-mail: b_alaka2@hotmail.com

Prof. Dr. Ajit K. Mandal
Artificial Intelligence Laboratory
ETCE Department
Jadavpur University
Calcutta 700032
India
E-mail: ajit.k.mandal@vsnl.com
ajit.k.mandal@ieee.org

Prof. Dr. Amit Konar
Visiting Professor
Department of Math. and Computer Science
University of Missouri, St. Louis
8001 Natural Bridge Road, St. Louis
Missouri 63121-4499
USA

Permanently working as
Professor
Department of Electronics and Tele-communication Engineering
Jadavpur University
Calcutta 700032
India
E-mail: konaramit@yahoo.co.in

Library of Congress Control Number: 2006925432

ISSN print edition: 1860-949X
ISSN electronic edition: 1860-9503
ISBN-10 3-540-33458-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-33458-3 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

Printed in The Netherlands

Cover design: deblik, Berlin
Typesetting by the authors and SPI Publisher Services
Printed on acid-free paper SPIN: 11588498 89/SPI 5 4 3 2 1 0

Preface

Foundation of logic historically dates back to the times of Aristotle, who pioneered the concept of truth/falsehood paradigm in reasoning. Mathematical logic of propositions and predicates, which are based on the classical models of Aristotle, underwent a dramatic evolution during the last 50 years for its increasing applications in automated reasoning on digital computers.

The subject of Logic Programming is concerned with automated reasoning with facts and knowledge to answer a user's query following the syntax and semantics of the logic of propositions/predicates. The credit of automated reasoning by logic programs goes to Professor Robinson for his well-known resolution theorem that provides a general scheme to select two program clauses for deriving an inference. Until now Robinson's theorem is being used in PROLOG/DATALOG compilers to automatically build a Select Linear Definite (SLD) clause based resolution tree for answering a user's query.

The SLD-tree based scheme for reasoning undoubtedly opened a new era in logic programming for its simplicity in implementation in the compilers. In fact, SLD-tree construction suffices the need for users with a limited set of program clauses. But with increase in the number of program clauses, the execution time of the program also increases linearly by the SLD-tree based approach. An inspection of a large number of logic programs, however, reveals that more than one pair of program clauses can be resolved simultaneously without violating the syntax and the semantics of logic programming. This book employs this principle to speed up the execution time of logic programs.

One question that naturally arises: how does one select the clauses for concurrent resolution? Another question that crops up in this context: should one select more than two clauses together or pairs of clauses as groups for concurrent resolution? This book answers these questions in sufficient details. In fact, in this book we minimize the execution time of a logic program by grouping sets of clauses that are concurrently resolvable. So, instead of pairs, groups of clauses with more than two members in a group are resolved at the same time. This may give rise to further questions: how can we ensure that the selected groups only are concurrently resolvable, and members in each group too are maximal? This in fact is a vital question as it ensures the optimal time efficiency (minimum execution time) of a logic program. The optimal time efficiency in our proposed system is attained by mapping the program clauses onto a specialized structure that allows

each group of resolvable clauses to be mapped in close proximity, so as to participate in the resolution process. Thus n-groups of concurrently resolvable clauses form n clusters in the network. Classical models of Petri nets have been extended to support the aforementioned requirements.

Like classical Petri nets, the topology of network used in the present context is a bipartite graph having two types of nodes, called places and transitions, and directed arcs connected from places to transitions and transitions to places respectively. Clauses describing IF-THEN rules (knowledge) are mapped at the transitions, with predicates in IF and THEN parts being mapped at the input and the output places of the transitions. Facts described by atomic predicates are mapped at the places that too share predicates of the IF or the THEN parts of a rule. As an example, let us consider a rule: (Fly(X) ¬Bird(X).) and a fact: (Bird(parrot)¬.). The above rule in our terminology is represented by a transition with one input and one output place. The input and the output places correspond to the predicates: Bird(X) and Fly(X) respectively. The fact: Bird(parrot) is also mapped at the input place of the transition. Thus, a resolution of the rule and the fact is possible because of their physical proximity on the Petri net architecture. It can be proved by method of induction easily that all members in a group of resolvable clauses are always mapped on the Petri net around a transition. Thus a number of groups of resolvable clauses are mapped on different transitions and the input-output places around them. Consequently, a properly designed firing rule can ensure concurrent resolution of the groups of clauses and generation and storage of the inferences at appropriate places. The book aimed at realizing the above principle by determining appropriate control signals for transition firing and resulting token saving at desired places.

It is indeed important to note that the proposed scheme of reasoning covers the notion of AND-, OR-, Stream- and Unification-parallelisms. It is noteworthy that there are plenty of research papers with hundreds of scientific jargons to prohibit the unwanted bindings in AND-parallelisms, but very few of them are realistic. Implementation of the Stream-parallelism too is difficult, as it demands design of complex control strategies. Fortunately, because of the structural benefits of Petri nets, AND- and Stream-parallelisms could have been realized by our proposed scheme of concurrent resolution automatically. The most interesting point to note is that these parallelisms are realized as a byproduct of the adopted concurrent resolution policy, and no additional computation is needed to implement the former.

The most important aspect of this book, probably, is the complete realization of the proposed scheme for concurrent resolution on a massively parallel architecture. We verified the architectural design with VHDL and the implementations were found promising. The VHDL source code is not included in the book for its sheer length that might have enhanced its volume three times its current size. Finally, the book concludes on the possible application of the proposed parallel and distributed logic programming for the next generation database machines.

The book comprises of six chapters. Chapter 1 provides an introduction to logic programming. It begins with a historical review on the last 50 years evolution of symbolic paradigms in Artificial Intelligence. The chapter then outlines the logic of propositions and predicates, the resolution principles and its application in automated theorem proving. Gradually, the chapter progresses through a series of reviews on logic programs, its realization with stacks, the PROLOG language, and stability of interpretations in a logic program. The chapter also reviews four typical parallel architectures used for conventional programs. It also includes discussions on possible types of parallelisms in logic programs.

Chapter 2 extensively reviews the existing models of parallelisms in logic programs, such as the RAP-WAM architecture, Parallel AND-OR logic programming language, Kale's AND-OR tree model, CAM based architecture for a PROLOG machine. A performance analysis of PROLOG programs on different machine architectures is also introduced in this chapter. It then highlights the need of Petri nets in logic programming and ends with a discussion on the scope of the book.

Chapter 3 provides formal definitions to Petri nets and related terminologies. Main emphasis is given on concurrency in resolution. The chapter introduces an extended Petri net model for logic programming and explains resolution of program/data clauses with forward and backward firing of transitions in the Petri net model. An algorithm for automated reasoning is then proposed and explained with a typical Petri net. The chapter includes a performance analysis of the proposed algorithm with special references to speed up and resource utilization rate for both the cases of limited and unlimited resources.

Chapter 4 is devoted to the design of a massively parallel architecture that automates the reasoning algorithm presented in chapter 3. It begins with an introduction to the overall architecture in a nutshell.

The chapter then gradually explores the architectural details of the modules—namely Transition History File, Place Token Variable Value Mapper, Matcher, Transition Status File, First Pre-Condition Synthesizer and Firing Criteria Testing Logic. The chapter then analyzes the performance of the hardwired engine by computing a timing analysis with respect to the system clock.

Prior to mapping the user's logic program to the architecture proposed in Chapter 4, a pre-processing software is needed for parsing the user's source codes and mapping the program components on to the architecture. Chapter 5 provides a discussion on the design aspects of a pre-processor. The chapter outlines the design of a Parser to be used for our application. It then introduces the principles of mapping program components, such as clauses, predicates, arc function variables and tokens onto the appropriate modules of the architecture.

Chapter 6 indicates the possible direction of the book in the next generation database machines. It begins with an introduction to Datalog language, highlighting all its specific features in connection with logic program based data

models. The LDL system architecture is presented, emphasizing its characteristics in negation by failure, stratification and bottom-up query evaluation. Principles of designing database machines with Petri nets are also narrated in the chapter. The scope of Petri net based models in data mining is also examined at the end of the chapter.

January 1, 2006

Artificial Intelligence Lab.
ETCE Department
Jadavpur University

Alakananda Bhattacharya,
Amit Konar,
and Ajit K. Mandal.

Acknowledgements

The authors would like to thank many of their friends, colleagues and co-workers for help, cooperation and support, without which the book could not be completed in the present form.

First and foremost the authors wish to thank Professor A. N. Basu, Vice Chancellor, Jadavpur University for providing them the necessary support to write the book. They are equally indebted to Professor M. K. Mitra, Dean, Faculty of Engineering and Technology, Jadavpur University for encouraging them to write the book. During the preparation of the manuscript, Professor C. K. Sarkar, the present HOD and Professor A. K. Bandyopadhyay and Professor H. Saha, the past two HODs helped the authors in various ways to successfully complete the book.

The authors would like to thank Saibal Mukhopadhyay and Rajarshi Mukherjee for simulating and verifying the proposed architecture with VHDL. They are also indebted to a number of undergraduate students of ETCE department, Jadavpur University for helping them in drawing some of the figures of the book. They are equally indebted to Saswati Saha, an M. Tech. student of ETCE department for providing support in editing a part of the book.

The first author is indebted to her parents Mrs. Indu Bhattacharya and Mr. Nirmal Ranjan Bhattacharya for providing her all sorts of help in building her academic career and their moral and mental support to complete the book in the present form. She is equally grateful to her in-laws Mrs. Kabita Roy and Mr. Sunil Roy for all forms of supports they extended to household affairs and their patience and care for the author's beloved child Antariksha. The first author would also like to thank her elder brother Mr. Anjan K. Bhattacharya, brother-in-law Mr. Debajit Roy and her sister-in-law Mrs. Mahua Roy for their encouragement in writing this book. She would like to pay her vote of thanks to her uncle Late N. K. Gogoi, who always encouraged her to devote her life for a better world rather than living a routine life only. She also thanks her cousin brother Gunturu (Sudeet Hazra) and her friend Madhumita Bhattacharya who continued insisting for successful completion of the book. Lastly, the author thanks her husband Abhijit for his understanding to spend many weekends lonely. The acknowledgement will remain incomplete if the author fails to record the help and support she received from her onetime classmate and friend Sukhen (Dr. Sukhen Das). Lastly, the author would like to express her joy and happiness to her dearest son Antariksha and her nephew Anjishnu whose presence helped her wade through the turbulence of home, office and research during the tenure of her work.

The second and the third authors would also like to thank their family members for extending their support to write this book.

The authors gratefully acknowledge the academic support they received from UGC sponsored projects on i) AI and Expert Systems Applied to Image Processing and Robotics and ii) University with Potential for Excellence Program in Cognitive science.

Artificial Intelligence Lab.
ETCE Department,
Jadavpur University.

Alakananda Bhattacharya,
Amit Konar,
and Ajit K. Mandal.

Contents