Lecture Notes in Computer Science 908
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Advisory Board: W. Brauer D. Gries J. Stoer

Josyula Ramachandra Rao

Extensions of the UNITY Methodology

Compositionality, Fairness and Probability in Parallelism

Springer

Series Editors

Gerhard Goos
Universität Karlsruhe
Vincenz-Priessnitz-Straße 3, D-76128 Karlsruhe, Germany

Juris Hartmanis
Department of Computer Science, Cornell University
4130 Upson Hall, Ithaca, NY 14853, USA

Jan van Leeuwen
Department of Computer Science, Utrecht University
Padualaan 14, 3584 CH Utrecht, The Netherlands

Author

Josyula Ramachandra Rao
IBM Thomas J. Watson Research Center
P. O. Box 704, Yorktown Heights, NY 10598, USA

CR Subject Classification (1991): D.1.3, D.2.2, D.3, F.3, G.3

ISBN 3-540-59173-7 Springer-Verlag Berlin Heidelberg New York

CIP data applied for

Typesetting: Camera-ready by author
SPIN: 10485587 06/3142-543210 - Printed on acid-free paper

Preface

The UNITY methodology marks an important milestone in research on program verification. The methodology shows how a simple programming notation and a small set of carefully engineered operators can be used to reason effectively about a wide variety of parallel programs. The goal of this treatise is to push these ideas further in order to explore and understand the limitations of this approach. We attempt to do this in three ways. First, we apply UNITY to formulate and tackle problems in parallelism such as compositionality. Second, we extend and generalize the notation and logic of UNITY in an attempt to increase its range of applicability. Finally, we develop paradigms and abstractions useful for algorithm design. We summarize our contributions below.

In designing a system of processes, it is desirable to have a guarantee that the progress made by each individual process is inherited by the system as a whole. Such a guarantee would aid in developing parallel programs in a compositional way. We use UNITY logic to formulate such a guarantee and use formal methods to derive sufficient (and yet practical) conditions for the guarantee to hold. These conditions require process interactions to obey certain commutativity conditions. Our restrictions permit compositional reasoning about progress properties of parallel programs and provide a rigorous justification for including certain syntactic features in parallel programming languages.

The nature of fairness assumed in executing a UNITY program determines the progress properties that can be proven of the program. Our second contribution is a uniform framework for the *systematic* design of proof rules for proving progress under a spectrum of fairness assumptions ranging from pure nondeterminism to strong fairness. Proofs of soundness and relative completeness of the synthesized rules follow by checking a set of simple conditions. Unlike existing work in this area, our proofs do not use ordinals.

One special notion of fairness that is being increasingly used by algorithm designers is that associated with tossing a coin. Of late, programmers have started using probabilistic transitions in designing simple and efficient algorithms for problems that may not have a deterministic solution. We generalize UNITY program to permit probabilistic transitions and develop a UNITY–like theory to design and prove the correctness of probabilistic parallel programs. We illustrate our theory with examples from random walks and mutual exclusion.

Finally, we propose a new paradigm for the design of probabilistic parallel programs called *eventual determinism*. The paradigm provides a means of com-

bining probabilistic and deterministic algorithms to take advantage of both. The proofs of such algorithms use the probabilistic generalization of UNITY. We illustrate the paradigm with examples from conflict–resolution and self–stabilization.

Our investigations and results reaffirm the promise of UNITY: we conclude that it provides a versatile medium for posing and solving many of the diverse problems of parallelism.

Acknowledgements: This book is based on my doctoral dissertation which was completed at the University of Texas at Austin in August 1992. The work reported here has been deeply influenced by discussions with several people and I would like to take this opportunity to thank some of them.

I owe an enormous debt of gratitude to my supervisor, Professor J. Misra and to Professor Edsger W. Dijkstra. I have been greatly influenced by their taste in research topics and their clarity of thought. In writing this book, I have tried to achieve their conception of simplicity and elegance while aspiring for their high standards of rigor and excellence. I have also had the privilege of improving my work through discussions with Professor C. A. R. Hoare. I will remain indebted to him for his valuable criticisms and timely words of encouragement.

Chapters 4 and 5 of this thesis represent collaborative work with two of my colleagues, Ernie Cohen and Charanjit Jutla respectively: it has been a pleasure to work with them. I would also like to thank my colleagues in the UNITY group and the Distributed Systems Discussion Group at Austin especially Mike Barnett, Ken Calvert, Ted Herman, and Dave Naumann.

I gratefully acknowledge the financial support that I received from the Office of Naval Research, the Texas Advanced Research Program, the National Science Foundation, and the University of Texas at Austin.

I have been extremely fortunate to have enjoyed the company of good friends at all stages of my life. In particular, I would like to thank Asoke Chattopadhyaya, Vipin Chaudhary, Leena and Manoj Dharwadkar, T. Krishnaprasad, Lyn and David Loewi, Linda Mohusky, Vijaya and K. Muthukumar and Bikash Sabata. Finally, I will remain indebted to my parents, my sisters, Surya and Sundari, and my wife, Sailaja, for their love and emotional support.

Yorktown Heights, New York
December 1994

Josyula R. Rao

Table of Contents