Lecture Notes in Computer Science 1032
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

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

Patrice Godefroid

Partial-Order Methods for the Verification of Concurrent Systems

An Approach to the State-Explosion Problem

Springer

Author

Patrice Godefroid
AT&T Bell Laboratories
1000 E. Warrenville Road, Naperville, IL 60566-7013, USA

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Godefroid, Patrice:
Partial order methods for the verification of concurrent systems : an approach to the state explosion problem / Patrice Godefroid. - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1996
(Lecture notes in computer science ; 1032)
ISBN 3-540-60761-7
NE: GT

CR Subject Classification (1991): F.3.1, D.2.4, C.2.2, F.1.2, D.2.4

ISBN 3-540-60761-7 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10512431 06/3142 - 5 4 3 2 1 0 Printed on acid-free paper

Foreword

For many years, research on verification dealt almost exclusively with semantic and logical issues. A typical thesis on the subject would consider some delicate programming feature and show how a proof system could be extended to handle this feature. A completeness proof added some theoretical meat, and a laboriously worked out toy example showed that it was, at least in principle, possible to use the new proof system. With few exceptions, little thought was given to implementing the verification approach and making it painlessly usable by the typical practicing software developer.

Although this thesis deals with verification, it is far from following the pattern outlined above. Indeed, rather than being about a logical system for verification, it is about *algorithms* for verification. Its starting point is the simple technique of state-space exploration, which as such, or elaborated into model checking, is attracting growing attention for the verification of concurrent systems. Patrice Godefroid addresses a main limiting factor of this approach: the explosion of the number of states due to the modeling of concurrency by interleaving. Noticing that, as indicated by partial-order semantics for concurrency, this limiting factor is not inherent, he proceeds to develop a family of algorithms that make it possible to avoid it. Furthermore, these algorithms have been implemented, and experiments show that they can work very well in practice.

The general pattern of this thesis is thus to turn logical and semantic ideas into exploitable algorithms. It is part of the trend that views verification as a computer-aided (and as algorithmic as possible) activity, not as a paper and pencil one. After all, if software engineers are going to check their designs, they need, as all other engineers, computer support to do so.

There is much valuable information in this thesis. The verification systems builder will find a guide to implementing partial-order state space exploration techniques. The user of verification systems will gain a knowledge of how these methods work, and of why they are entirely reliable. The computer scientist will enjoy the beauty of clear algorithms crystallizing from the simple semantic concept of independent transitions.

Pierre Wolper

Preface

State-space exploration techniques are increasingly being used for debugging and proving correct finite-state concurrent reactive systems. The reason for this success is mainly the simplicity of these techniques. Indeed, they are easy to understand, easy to implement, and, last but not least, easy to use: they are fully automatic. Moreover, the range of properties that they can verify has been substantially broadened thanks to the development of model-checking methods for various temporal logics.

The main limit of state-space exploration verification techniques is the often excessive size of the state space, due, among other causes, to the modeling of concurrency by interleaving. However, exploring *all* interleavings of concurrent events is not a priori necessary for verification: interleavings corresponding to the same concurrent execution contain related information. One can thus hope to be able to verify properties of a concurrent system without exploring all interleavings of its concurrent executions. This work presents a collection of methods, called *partial-order methods*, that make this possible.

The intuition behind partial-order methods is that concurrent executions are really partial orders and that concurrent events should be left unordered since the order of their occurrence is irrelevant. However, rather than choosing to work with direct representations of partial orders, the methods we develop keep to an interleaving representation of partial orders, but attempt to limit the expansion of each partial-order computation to just *one* of its interleavings, instead of all of them. More precisely, given a property, partial-order methods explore only a reduced part of the global state-space that is sufficient for checking the given property. In this work, three types of properties are considered: absence of deadlocks, safety properties, and properties expressed by linear-time temporal-logic formulas.

The techniques and algorithms we describe have been implemented in an add-on package for the protocol verification system SPIN. This *Partial-Order Package* has been tested on numerous examples, including several industrial-size communication protocols. When the coupling between the processes is very tight, partial-order methods yield no reduction, and the partial-order search becomes equivalent to a classical exhaustive search. When the coupling between the processes is very loose, the reduction is very impressive: in some cases, the number of states that need to be visited for verification can be reduced from exponential to polynomial in the size of the system description (code). For most realistic examples, partial-order methods provide a

significant reduction of the memory and time requirements needed to verify protocols.

This monograph is a revised version of my PhD thesis, submitted in 1994 to the University of Liège. This work would not have been possible without the technical and moral support of my thesis advisor, Pierre Wolper. He introduced me to the field of verification, and opened doors for me in the research community. His enthusiastic supervision has been a continuous source of encouragement to me. I consider myself fortunate that I had access to his valuable guidance.

I am grateful to the other members of my reading committee, Professors Raymond Devillers, Pascal Gribomont, Amir Pnueli, Daniel Ribbens, Joseph Sifakis, and Antti Valmari, for their very careful review of this work.

It has been a great pleasure for me to work closely with Didier Pirottin during these last three years. I am thankful to Didier for numerous insightful discussions, and for his help in implementing algorithms presented in this work.

I wish to thank Gerard Holzmann for freely sharing his considerable experience in validating communication protocols. I learned how to build verification tools mainly from his work and from discussions with him. He provided me with many challenging examples of communication protocols, which have been (and still are) a very good source of inspiration to me. He also made possible an exciting visit to AT&T Bell Laboratories during the summer of 1992.

I have had the opportunity to discuss my research with many scientists at various conferences and seminars. I thank all of them for being helpful and encouraging. I am particularly grateful to Mark Drummond, Pascal Gribomont, Froduald Kabanza, Doron Peled, and Antti Valmari for very fruitful discussions. Special thanks also go to Bernard Boigelot, Philippe Lejoly, and Luc Moreau for reading and commenting on an early version of this text, and to Gerald Luettgen and Bernhard Steffen for detailed comments and suggestions.

This work was financially supported by the European Community ESPRIT projects SPEC (3096) and REACT (6021), and by the Belgian Incentive Program "Information Technology – Computer Science of the Future", initiated by the Belgian State – Prime Minister's Service – Science Policy Office, which I gratefully acknowledge. I thank AT&T Bell Laboratories for the additional time which made it possible for me to complete this revision of my thesis.

Last but not least, I wish to thank my parents for their constant moral support, and Anne-Christine for her love, for sharing ups and downs, and for reminding me, when necessary, that computer science is not the most important thing in life.

Contents

List of Figures