Preface

Software design and program development is widely recognised as a highly creative task. The reason for this understanding is that industrial-strength software has some inherent conceptual complexity which can, as in the case of concurrent, reactive systems, easily exceed the human intellectual capacity. Finding ways to master this complexity is therefore one of the major challenges in computer science today.

So far, a common approach in software engineering has been to apply during the design phase a variety of structured techniques like top-down design, decomposition, and abstraction in order to cope with the complexity of large software systems. Only after the design is completed does intensive testing in the implementation phase ensure reliability, usually understood as absence of program errors. However, this approach neglects the fact that central aspects of software design and program development have a strong formal character that in principle admits tool support for the construction of reliable and correct computer systems. A crucial precondition for the success of such a computer-aided effort is, in fact, the availability of methods and techniques which perform the required formal reasoning.

This monograph aims to provide the theoretical foundations needed for the verification of reactive, sequential infinite-state systems. In particular, we will develop two new algorithms that allow us to automatically verify important aspects, like safety or liveness properties, of the given infinite-state system. As we deal with infinite-state spaces, many theoretical topics are involved, including process algebras, fixpoint theory, modal logics, and model checking. To stress the importance of a sound foundation for the developed verification methods, we put particular emphasis on the presentation of the formal framework which we hope will be of use also for future extensions.

This monograph is a revised version of my doctoral dissertation which was submitted to the Faculty of Mathematics and Natural Sciences of the Rheinisch-Westfälische Technische Hochschule Aachen and accepted in July 1995.

I would like to thank my supervisor Bernhard Steffen who introduced me to the subject and provided many inspiring discussions that greatly influenced the contents of my thesis. I also want to thank him and his wife Tiziana for their kind hospitality during my several visits to Passau.

I also thank Klaus Indermark for his constant support as well as his enthusiasm during my employment at the Department of Computer Science at Aachen.

Thanks are also due to Didier Caucal, who after a discussion in Passau initiated my work on the bisimulation equivalence problem by pointing me to some of his previous research. I also want to thank him for his many useful comments that influenced much of the topic of Chapter 5, as well as for his kind support during my stay at IRISA in Rennes where we continued the work on the bisimulation problem for context-free processes.

I am also indebted to Colin Stirling who influenced implicitly, and after we met personally in Edinburgh also explicitly, the presentation of the model checking algorithm and its associated theory.

Finally, I am also grateful both to an anonymous referee and to Markus Schweighofer, who gave a number of hints on a draft version that helped to improve the final presentation.

Last but not least, warm thanks to my wife Simone for taking good care of me by providing constant emotional and practical support.

Dortmund, October 1997 *Olaf Burkart*

Foreword

After almost twenty years of concurrency theory, we face a wide spectrum of formalisms – process algebras, models for concurrency, and application specific languages – which all come with their specific expressive power and conceptual complexity. There is currently no sign that this tendency to diversity will end. In response to this state of the art, various tools have been developed, each addressing very specific scenarios on the basis of tailored methodologies. They provide keys to the practical use of all the described theory, as they have the potential to constitute an interface between theory and practice on a purely phenomenological level.

The typical development of such tools, which I am convinced provide a new momentum to concurrency theory and even to formal methods in general, goes through three sometimes overlapping phases: a *conceptual phase*, where the underlying decidability issues are studied, usually by pure mathematical reasoning, a *complexity-oriented realization phase*, where appropriate data structures and algorithms are designed and implemented, and first case studies are performed, and a *'civilisation' phase*, where people look at application scenarios and profiles, at appropriate interfacing to industrial environments and user communities, and where the investigation of the practical behaviour of algorithms on concrete applications gains importance over the usual worst case reasoning.

Currently, although the first 'civilised' formal-method based tools have become reality, the main effort is still invested in the first two phases, which still provide a huge potential for investigation.

The contributions of this monograph belong to the first and second phase. New concepts and algorithms are provided, for both model checking and equivalence checking, which drastically extend the scope of automatic verification, and which, nevertheless, have the potential to give directions for efficient implementation. In fact, the results of the underlying dissertation include the first effective model checking algorithm for infinite state systems, namely the class of context-free and pushdown processes, which can be regarded as procedural extensions of finite automata. The underlying second-order semantics is rather elegant and surprisingly efficient: context-free processes can be model checked essentially in time proportional to the size of the argument process. Only the size of the property to be checked is critical.

Moreover, as it has turned out in the meantime, second-order semantics allow comparatively simple extensions to formulae of higher alternation depth and to further generalised process calculi.

In addition, the first bisimulation checking algorithm of elementary complexity is presented, which covers all context-free processes. This algorithm, although extremely intricate and computationally expensive, paves the way towards efficient implementation: for the first time it allows an estimation of the worst case complexity of this class of processes. One may well expect that, as already experienced in the case of normed context-free processes, which over the years have been shown to admit polynomial bisimulation checking, the complexity result provided in this thesis will be drastically improved in the near future, making context-free bisimulation checking a tool of practical relevance.

All these contributions are based on an impressive collection of new algebraic theorems, which are interesting already on their own: they provide in fact a strong intuition about infinite state systems, properties of parallel composition, and differences between normed and unnormed processes, as well as between context-free and pushdown processes in a branching-sensitive scenario. Thus this monograph also provides a comprehensive overview of the foundations of infinite state verification for the considered classes of processes, and reveals essential differences between the new branching-sensitive theory and 'classical' automata and formal language theory.

Summarising, the reader will find elegant and deep theory as well as comprehensible algorithms, which I am convinced will be the key for a better understanding of process theory. In fact, I believe that corresponding implementations will enhance this understanding, and tuned versions will enter modern environments for concurrent system design in the near future in the general course of formal methods integration. This general trend has recently been observed for finite state model checking, which, like the now omnipresent type checking, quickly conquered the industrial hardware design arena. I am convinced that this trend will continue for other fully automatic verification techniques like the ones presented in this thesis. Thus this monograph provides a wealth of valuable information, both for pure theoreticians interested in algebraic theories and for tool builders who are open to conquering new ground in their desire to construct practically relevant tools.

Dortmund, November 1997 *Bernhard Steffen*

http://www.springer.com/978-3-540-63982-4

Automatic Verification of Sequential Infinite-State Processes

Burkart, O.

1997, X, 166 p. 1 illus., Softcover
ISBN: 978-3-540-63982-4