Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

369

Dirk Taubner

Finite Representations of CCS and TCSP Programs by Automata and Petri Nets

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo Hong Kong

Author

Dirk A. Taubner
Institut für Informatik, Technische Universität München
Arcisstraße 21, D–8000 München 2, Federal Republic of Germany

CR Subject Classification (1987): D.1.3, D.3.1, F.1.1–2, F.3.2–3, F.4.3

ISBN 3-540-51525-9 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-51525-9 Springer-Verlag New York Berlin Heidelberg

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

To my parents

Foreword

There are two main approaches to a theory of concurrent distributed computations: the theory of Petri nets and the Milner/Hoare theory of CCS/CSP. They are based on different philosophies and emerged from two different classical notions of computability. The Petri net approach developed (in the early 60s) from the ideas around Turing machines and automata; it has concurrency and causality as its basic concepts. CCS/CSP grew (in the middle of the 70s) out of ideas around the λ-calculus and concepts in programming; it has communication and composition as its basic notions.

Petri nets are equipped with a natural notion of partial order semantics (the processes introduced by Petri in 1976, which model concurrency explicitly), while originally CCS/CSP has an interleaving semantics (which models concurrency by nondeterminism).

In recent years both approaches began to influence each other and to converge. In particular Petri nets are being developed such that they can be used for a variety of purposes: for system description, as a specification and programming language, and as a formal semantics for languages like CCS and CSP. We are now in the phase where constructions allowing compositionality and modularity are built into Petri nets, where we look for hierarchical net constructions and refinement techniques, and for methods of formal reasoning (about or by using nets) — see for example the ESPRIT Basic Research Action 3148 "Design Methods Based on Nets" (DEMON). The deep and broad theory developed around CCS/CSP and related concepts has a great impact on this development.

On the other hand, ideas and techniques from the field of Petri nets influence more and more the CCS/CSP domain. And, at least in my opinion, the power and the problems inherent in the application of the CCS/CSP operators as well as in the implementation of CCS/CSP-based languages, can be particularly well understood and studied by translating these operators into constructors for nets and for automata. This thesis is an especially good proof for this opinion.

Munich, June 1989 Wilfried Brauer

Preface

This work relates different approaches for the modelling of parallel processes.

On the one hand there are the so-called 'process algebras' or 'abstract programming languages' with Milner's Calculus of Communicating Systems (CCS) and the theoretical version of Hoare's Communicating Sequential Processes (CSP) as main representatives.

On the other hand there are machine models, viz. the classical finite state automata (transition systems), for which however more discriminating notions of equivalence than equality of languages are used; and secondly there are differently powerful types of Petri nets, namely safe, respectively general (place/transition) nets, and predicate/transition nets.

Within a uniform framework the syntax and the operational semantics of CCS and TCSP are explained. We consider both, Milner's well-known interleaving semantics which is based on infinite transition systems, as well as the new distributed semantics introduced by Degano, De Nicola, Montanari, and Olderog which is based on infinite safe nets.

The main part of this work contains three syntax-driven constructions of transition systems, safe nets, and predicate/transition nets respectively. Each of them is accompanied with a proof of consistency.

Due to intrinsic limits, which are also investigated here, neither for transition systems and safe nets, nor for general place/transition nets does a finite consistent representation of all CCS and TCSP programs exist. However sublanguages which allow finite representations are discerned. On the other hand the construction of finite predicate/transition nets is possible for all CCS programs in which every choice and every recursive body starts sequentially.

The work is a revised version of my PhD thesis. I am particularly grateful to my supervisor Prof. W. Brauer for his kind guidance, for many helpful discussions, and for his continuous very efficient support. Additionally I thank Prof. Mogens Nielsen for his advice and for having acted as the second referee.

My special thanks go to Walter Vogler. In numerous discussions and with many suggestions for improvements he has been a great help. I have also benefitted from valuable comments of Volker Diekert and Ursula Goltz.

I thank the Siemens AG and in particular Prof. H. Gumin for supporting my work by an Ernst von Siemens Scholarship. Especially I appreciate the additional time which made it possible for me to do this revision.

Munich, June 1989 Dirk Taubner

Contents