

New Books

Kybernetika, Vol. 24 (1988), No. 3, 227--237

Persistent URL: <http://dml.cz/dmlcz/125356>

Terms of use:

© Institute of Information Theory and Automation AS CR, 1988

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library*
<http://project.dml.cz>

Knihy došlé do redakce / (Books received)

Micha Hofri: Probabilistic Analysis of Algorithms: On Computing Methodologies for Computer Algorithms Performance Evaluation. (Texts and Monographs in Computer Science.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. XV + 240 pages; 14 figs.; DM 88,—.

System Development and Ada — CRAI Workshop on Software Factories and Ada, Capri, Italy, May 26—30, 1986, Proceedings (*A. N. Habermann, U. Montanari, eds.*). (Lecture Notes in Computer Science 275.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. V + 305 pages; DM 45,—.

ECOOP '87 — European Conference on Object-Oriented Programming, Paris, France, June 15—17, 1987, Proceedings (*J. Bézuvin, J. - M. Hullot, P. Cointe, H. Lieberman, eds.*). (Lecture Notes in Computer Science 276.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. VI + 273 pages; DM 40,50.

Graph Reduction. Proceedings of a Workshop, Santa Fé, New Mexico, USA, September 29 — October 1, 1986 (*Joseph H. Fasel, Robert M. Keller, eds.*). (Lecture Notes in Computer Science 279.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. XVI + 450 pages; DM 60,50.

Mathematical Models for the Semantics of Parallelism — Advanced School, Rome, Italy, September 24—October 1, 1986 (*Joseph H. Fasel, Robert M. Keller, eds.*). (Lecture Notes in Computer Science 280.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. V + 231 pages; DM 36,—.

Visualization in Programming — 5th Interdisciplinary Workshop in Informatics and Psychology, Schärding, Austria, May 20—23, 1986, Selected Contributions (*P. Gorny, M. J. Tauber, eds.*). Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. VII + 210 pages; DM 36,—.

Category Theory and Computer Science — Edinburgh, U. K., September 7—9, 1987, Proceedings (*D. H. Pitt, A. Poigné, D. E. Rydeheard, eds.*). (Lecture Note in Computer Science 283.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. V + 300 pages; DM 45,—.

D. A. Carlson, A. Hawrie: Infinite Horizon Optimal Control — Theory and Applications. (Lecture Notes in Economics 290.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. XI + 254 pages; DM 52,—.

Stochastic Differential Systems — Proceedings of the IFIP-WG 7/1 Working Conference, Eisenach, GDR, April 6—13, 1986 (*H. J. Engelbert, W. Schmidt, eds.*). (Lecture Notes in Control and Information Sciences 96.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1986. XII + 381 pages; DM 96,—.

Control Problems for Systems Described by Partial Differential Equations and Applications — Proceedings of the IFIP-WG 7.2 Working Conference, Gainesville, Florida, February 3—6, 1986 (*I. Lasiecka, R. Triggiani, eds.*). (Lecture Notes in Control and Information Sciences 97.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. VIII + 400 pages; DM 108,—.

A. Alonftis: Stochastic Adaptive Control: Results and Simulations. (Lecture Notes in Control

and Information Sciences 98.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. XII + 120 pages; DM 36,—.

Pattern Recognition Theory and Application (*Pierre A. Devijver, Josef Kittler, eds.*) (NATO ASI SERIES — Series F: Computer and Systems Sciences, Vol. 30.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. XI + 543 pages; DM 178,—.

Modelling, Robustness and Sensitivity Reduction in Control Systems (*Ruth F. Curtain, ed.*) (NATO ASI Series — Series F: Computer and Systems Science, Vol. 34.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. IX + 492 pages; DM 148,—.

Natural Language Parsing Systems (*Leonard Bolc, ed.*) (Symbolic Computation — Artificial Intelligence.) Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987. XVIII + 367 pages; 151 figs; DM 98,—.

Hans Loeper, Hans-Jörg Jäkel, Wolfgang Otter: Compiler und Interpreter für höhere Programmiersprachen. (Informatik — Kybernetik — Rechentechnik 17.) Akademie-Verlag, Berlin 1987. 390 Seiten; M 50,—.

The Knowledge Frontier — Essays in the Representation of Knowledge (*Nick Cercone, Gordon McCalla, eds.*) (Symbolic Computation — Artificial Intelligence.) Springer-Verlag, New York—Berlin—Heidelberg—London—Paris—Tokyo 1987. 550 pages; 93 figs.; DM 78,—.

WILLIAM F. CLOCKSIN, CHRISTOPHER S. MELLISH

Programming in PROLOG

Third, Revised and Extended Edition.

Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987.

XIV + 281 pages; DM 52,—.

When reading a three, five, or more years old review of a book dealing with PROLOG (including the review of the first edition of a title in question, published in this journal in 1983), we can see that at least one half of the review is devoted to a detail explanation, what the PROLOG programming language may be and how its philosophy and methodology differ from those on which the “classical” programming languages are based. We follow here this pattern only with non-negligible doubts and hesitations, as it may be taken as the same and very rough under-estimation of the reader’s (natural) intelligence as when explaining, in a review of an undergraduate textbook on algebra or arithmetics, what algebra and arithmetics are. This shift can be seen as one of the most evident demonstrations of the great success and popularity achieved by PROLOG within the last few years.

Hence, as it is almost notorically known, in our days, PROLOG is a language working rather over facts and dependencies among facts, expressed in an appropriate first-order predicate language, than over numerical values of computable functions, like the “classical” programming languages do. A PROLOG program does not consists in a sequence of instructions, what to do with some input numerical values, but rather in a collection of facts and dependencies of implicative form (database) together with some more formula(s), i.e. hypotheses, the deducibility of which from the database is to be either simply verified, if there are no variables in these hypotheses, or satisfied by choosing and listing all the substitutions of names of considered objects for variables which satisfy the tested formulas (given a database dealing relations among the members of a family, we may either simply test, whether John and Paul are brothers, or to list all John’s brothers — in the extend in which the given database allows to do so, of course!).

Almost every day brings new and new examples how many and how various problems can be converted into and solved by the means of these basic tasks. What is of great importance, and should be carefully taken into consideration by anybody beginning to work with PROLOG, is the fact that the way in which the given goal is being transformed into sub-goals and satisfied

is given by the PROLOG language itself (as a matter of fact, it is given by the well-known backtracking exhaustive principle), not by the user himself, as in the other programming languages. The user may influence the work of his PROLOG programs essentially in two ways. First, by the so called "cut operation", which enables to generate just one solution to a goal or sub-goal in case the exhaustive search for all solutions is useless because of the nature of the main goal. Second, the order of items in databases and the order of goals posed by the program may be of crucial importance as far as the computational complexity or even the solvability are concerned, and a clever user may exploit this fact as a powerful tool for improving the qualities of his programs. It is why the detailed knowledge of how the program works is more important in the case of PROLOG than in the "classical" languages, when the knowledge of the language on a "manual" level is, from the user's viewpoint, quite satisfactory. It is not the case of PROLOG and one of the reasons for which the reviewed book should be highly appreciated are the pedagogical qualities with which the nature of PROLOG programs, their implementation, behaviour and work in actual cases are explained.

Let us briefly survey the contents of the reviewed book, using the characteristics given by the authors themselves. First chapter gives feelings for what it is like to program in PROLOG and introduces objects, relationships, fact, rules and variables. A more detailed presentation of PROLOG syntax and data structures can be found in Chapter 2. The next chapter deals with representing objects and relationships by trees and list and develops several standard PROLOG programming techniques. Chapter 4 describes the actual run of PROLOG programs and introduces the cut operation to modify the control sequence of running PROLOG programs. The fifth chapter develops facilities available for the input and output of characters and structures, and the next one deals with the core built-in predicates. Chapter 7 offers a number of example programs, covering a wide range of interests. Chapter 8 deals with the flow of the control model, hints about common bugs and techniques of debugging. The use of PROLOG for some aspects of analysing natural language is briefly discussed in the next chapter. Chapter 10 describes the close relations of PROLOG to the predicate calculus, clausal forms, resolution theorem proving and logic programming. The last, eleventh chapter presents a selection of suggested exercises, projects and problems. Appendix contains answers to select exercises, a note on clausal form program listing and some remarks concerning different versions of PROLOG, including Edinburgh PROLOG and micro-PROLOG.

When compared with the two earlier editions, the reviewed volume proves higher qualities in the way of explanation which is more lucid and easy to understand. The contents of almost all chapters have been revised, completed and enriched, and more illustrative examples have been added. The study of this textbook needs not any preliminary knowledge and, due to the specific nature of PROLOG language, the students not yet familiar with any "classical" programming language are not handicapped, we are in a strong temptation to say, that they are even favoured. Because of reasons briefly mentioned above, a detailed textbook on PROLOG is an inevitable complementary tool for anybody who knows only the manual-like presentation of a particular version of PROLOG and the reviewed book can be sincerely recommended for these purposes. From the formal point, of view, the volume keeps the traditional high level of Springer-Verlag publishing house.

Ivan Kramosil

JOHANN CH. FREYTAG

Translating Relational Queries into Iterative Programs

Lecture Notes in Computer Science 261.

Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987.

X + 250 pages; DM 27,—.

The book is the published version of author's Ph. D. thesis from 1985. The objective of this

work is to investigate the problem of translating set-oriented query specifications into iterative programs. The translation is based on functional programming and program transformation techniques.

The book consists of 7 chapters covering three groups of problems. The first two chapters are introductory. Chapter 1 surveys some important concepts of relational DBMSs such as a query optimizer and a query evaluation plan (QEP) conceived as subcomponents of the logical database processor. The problem of generating a program from QEP is called the translation problem for relational queries here. The solution described in this thesis gives two algorithms that translate QEPs into programs directly executable on the physical database processor. The second class of concepts concerns functional programming and program transformation.

Chapter 2 describes the data model supporting QEPs. A set of operators on relations and access tables is defined. Access tables (indices) support fast access to individual tuples of relations. The operators give the possibility to handle queries which are called select-project-join queries in database area. On the physical level five operators are defined to operate on sequences of tuples (so called streams). Together with the mechanism of function defining, these operators are sufficient to specify programs for each action of a QEP. Generally, to express iterative programs the language is extended by an assignment statement and a loop statement (the target language).

Chapter 3 presents a term rewriting system which simplifies expressions in the language of terms constituting the target language. The Church-Rosser property of the system is proved. In the rest of the chapter transformation steps are developed, and a complete transformation algorithm is presented. The algorithm replaces recursion by iteration in the final step.

An inefficiency of the algorithm led to the design of the second one, which generates more quickly the final iterative program. Chapter 4 covers this problem. The theory used in the algorithm is motivated by LISP operator map. It is interesting that final products of both algorithms are syntactically identical.

The second part of the book (Chapter 5) is devoted to the problem of generating efficient programs for the evaluation of aggregate functions. Author's solution gives transformations independent of any particular aggregate function performed in QBE.

Finally, the third part informs the reader about two implementations of DBMS based on the principles developed in the book. The LISP dialect T developed at Yale University and C language were used as implementation languages.

The major contribution of this book is in the integrating role of functional programming and program transformation applied on query processing.

Unfortunately, the book does not mention related works involving optimization techniques of select-project-join queries which were deeply studied in database area.

The book is interesting and might be useful reading for relational DBMS implementation experts.

Jaroslav Pokorný

EGON BÖRGER, Ed.

Computation Theory and Logic

Lecture Notes in Computer Science 270.

Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987.

IX + 422 pages; DM 60,50.

This collection contains 37 interesting contributions dealing with various aspects of computation theory (computer science) and mathematical logic and has been published in the memory of Prof. Dieter Rödding (1937—1984), on the occasion of the 50th anniversary of his birth and three years after his premature death. The choice of contributions seems to be motivated rather

by the fact that their authors were friends and fellows of Prof. Rödding than by some thematical limitations or restrictions. Hence, it is very difficult to systemize the presented contributions into some thematical groups, so we shall not try to do so and we shall prefer to mention sequentially all the contributions respecting the alphabetical order of their authors according to which they are presented.

K. Ambros-Spies investigates pairs A, B of recursive sets such that neither A nor B can be computed in polynomial time but every set which reduces to both A and B is polynomial time computable. G. Asser deals with primitive recursive functions of one variable which take the space of words over a fixed alphabet into itself. Existential fixed-point logic, proposed and investigated by A. Blass and Y. Gurevich, represents an interesting fragment of second-order logic, the expressive power of which is great enough to describe and handle, through the fixed-point operator, some important problems of the theory of computation. Some unsolvable decision problems for PROLOG programs are presented by E. Börger. In his rather philosophical essay, H.-J. Brämik argues in favour of the idea that the only way how to understand a mathematical theorem is to prove it. One of Prof. Rödding's main fields of interest was the theory of modular decomposition of finite automata, some new results in this direction are introduced by A. Brüggemann-Klein and R. Klein. H.-G. Carstens deals with algorithms for graph coloring and proves an effective version of the five-color-extension-theorem. Polynomials in infinitely many variables and their relations to the so called Buchberger's algorithm are studied by D. E. Cohen. E. Cohors-Fresenborg introduces the notion of microworld and argues in favour of its use at the methodological and pedagogical level, when solving the question, how can pupils at early secondary level get an insight into fundamental concepts of computer programming. Skolem normal forms of formulas containing the least fix-point operator are investigated by E. Dahlhaus.

M. Deutsch devoted his contribution to spectral representation of recursively enumerable and co-enumerable predicates. Very interesting problem of parallel inductive expertise on partial recursive functions, when several machines work simultaneously, in a co-operative way, in order to estimate a partially recursive function given a few of its values, is posed and solved by A. Drosdol and B. Schinzel. H.-D. Ebbinghaus presents a formalization of domino game, which can serve as an interesting and powerful nontraditional tool for investigation of computational complexity. The contribution by E. Engeler deals with cooperating expert systems. G. Germano and S. Mazzanti propose a new model of generalized computability with the aim to connect the traditional Gödel-Church conception of computability with the more abstract modern approaches. First-order spectra of unary first-order sentences with general quantifiers are investigated by E. Grandjean. G. Hasenjaeger presents a short and informal survey on the early history of register machines. The only contribution dealing with probabilistic algorithms is that by M. Karpinski and R. Verbeek, who investigate the possibilities to separate some time and space complexity classes for probabilistic algorithms (in author's terms: Monte-Carlo space and Monte-Carlo time). H. Kleine-Büning and T. Lettmann solve the problem of representation and updating of databases expressed by Horn formulas. The algorithmization of the construction of mutually orthogonal Latin squares, is proposed by H. Kull and E. Specker. Certain classes of finite functions defined on initial segments of the set \mathbb{N} of non-negative integers are investigated by M. Kummer under the title "Negative results about the length problems". T. Lickteig and H. Volger present some results on the complexity of powers. D. Mundici proves some results concerning the Turing complexity of approximately finite-dimensional C^* -algebras with lattice-ordered Grothendieck group. Some remarks about the SASL (St. Andrew's Static Language) system and its use for verification of functional programming languages are introduced by K. Nökel, R. Reibold and M. M. Richter. T. Ottmann, G. Thiemt and Ch. Ullrich present an interesting survey paper on numerical stability of simple geometric algorithms in the plane. Communication oriented semantics for concurrent systems are studied by L. Priese.

The contribution by B. Scarpellini presents a class of exp-time machines which can be simulated

by polytape machines. I. Schwank investigates a special kind of automata realizing preferences. Number-theoretic systems of Gentzen type with λ -rule are studied by H. Schwichtenberg. Some comments on the book of late Prof. J. R. Büchi are introduced by D. Siefkes under the title "Grammars for terms and automata" (the book is to be published by Springer Verlag). A very interesting paper of philosophical and methodological nature is that by W. Sieg, dealing with relative consistency of formalized theories (unfortunately written in German). W. A. Slaby investigates translating systems which learn on the base of a few examples of the translation process in question. First steps towards a theory of complexity over more general data structures are presented by V. Spersneider. The contribution by D. Spreen and H. Stahl deals with the power of single-valued nondeterministic polynomial time computations. A variant of the well-known Ehrenfeucht-Fraïssé game which is appropriate for an analysis of the expressive power of star-free regular expressions is proposed by W. Thomas. K. W. Wagner deals with "almost context-free" languages, finally, the last contribution in the collection is that by I. Wegener dealing with complexities of symmetric boolean functions.

All the contributions (up to two items) are written in English and their extents range from four to twenty-one pages. The shorter ones are conceived rather as extended abstracts, a few of them are presented on informal and philosophical level, but the greatest part of the presented contributions meet the demands of very high theoretical and mathematical level including precise mathematical models and structures, formalized assertions and their proofs. Hence, the volume meets also a traditional very high level of LNCS series and can be recommended to everybody interested in the topics of contemporary theory of recursive functions and computer science.

Ivan Kramosil

PHILIP TRELEAVEN, MARCO VANNESCHI, Eds.

Future Parallel Computers

Proceedings of an Advanced Course held in Pisa, Italy, June 9—20, 1987

Lecture Notes in Computer Science 272.

Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987

IV + 492 pages; DM 60,50.

In this review we shall often take profit of the preface introducing the volume in question, as this preface offers an excellent insight into the domain of interest as well as far as the contents of the volume is concerned. Interest in parallel computers has shown a dramatic increase in recent years. Since Japan launched its national Fifth Generation Project to develop parallel computers for use in the 1990's most other major industrial countries have started comparable national research programs, e.g. a significant portion of the European Community ESPRIT Program is devoted for these sakes. The competition between the national research programs has been a catalyst for parallel computer development.

The growing importance of parallel computers was reflected in the Advanced Course entitled "Future Parallel Computers" and held in Pisa, on June 9—20, 1986. The course was sponsored by ESPRIT Project 415: "Parallel Architectures and Languages for Advanced Information Processing — a VLSI — direct approach" and additional sponsorship was given by the six companies constituting the Project 415 consortium.

The aim of the Advanced Course was: firstly to present tutorials on the main classes of parallel computers, secondly to illustrate these classes of computers by examining important parallel systems being developed, and finally to study, in depth, important topics that influence all classes of parallel computers. This Course structure is followed in the 15 Chapters of this Proceedings.

Chapters 1 to 5 present the major classes of future parallel computers. In Chapter 1 Baiardi and Vanneschi cover parallelism issues in multi-style computers; this chapter may also serve

as a good introduction to the subject of parallel computers. In Chapter 2 May, Shepherd and Keane examine communicating process architectures using as a basis the designs of the Transputer system and the Occam system. The next chapter, written by Gurd and his colleagues, deals with dataflow computers. Reduction languages and systems are presented by Kluge and Schmittgen in Chapter 4. In Chapter 5 Bibel and his colleagues describe parallel inference (i.e. logic) machines.

Chapters 6 to 9 are of more technical nature and examine key research projects, each investigating a specific class of parallel computer systems. These chapters draw on the influential work of ESPRIT 415. In Chapter 6 Bronnenberg, Janssens, Odijk and van Twist describe the architecture of the DOOM parallel object-oriented computer being developed at Philips. In Chapter 7 Karia describes investigations at GEC towards a parallel architecture for functional languages. The BULL DDC Delta data driven computer, a parallel machine for symbolic processing, is presented by Gonzales-Rubio, Bradier and Rohmer in Chapter 8. In Chapter 9 Goto and Uchida deal with the influence of Japanese fifth generation project on high performance parallel inference machines.

Chapters 10 to 15 cover important research topics that are relevant to all classes of parallel computers. In Chapter 10 Giloi examines interconnection networks for massively parallel computing systems. Sami and Scarabottolo deal, in Chapter 11, with fault tolerance in parallel architectures. Chapter 12, written by Mehring and Aposporidis, is devoted to a multi-level simulator for VLSI, which is devoted at AEG as a part of Project 415. An introduction to systolic architectures is presented by Quinton in Chapter 13. Chapter 14, by Attardi, deals with concurrency in a knowledge base. Finally, in Chapter 15 Treleven, Refenes, Lees and McCabe present a detailed tutorial and survey on computer architectures for artificial intelligence.

When compared with the other volumes of the Lecture Notes in Computer Science series, the papers in the reviewed volume seem to be written at a more technical and informal level; their contributions consists in describing or suggesting hardware and software tools for parallel computing and in discussing their properties, either actual (for already existing systems) or hypothetical (when discussing the future development). Because of its tutorial character, the volume can be sincerely recommended to computer specialists beginning their work in the domain of parallel computing systems.

Ivan Kramosil

MARISA VENTURINI ZILLI, Ed.

Mathematical Models for the Semantics of Parallelism

Proceedings of the Advanced School held in Rome, Italy, September 24 to October 1, 1986

Lecture Notes in Computer Science 280.

Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1987.

IV + 230 pages; DM 36,—.

The contemporary need for a comprehensive and clear presentation of the several semantical approaches to parallelism motivated the stress on mathematical models, by means of which comparisons among different approaches can also be performed in a perspicuous way. The volume contains eight invited tutorial lectures presented at the Advanced School on Mathematical Models for the Semantics of Parallelism, held in Rome at the "Istituto per le Applicazioni del Calcolo Mauro Ricone", from September 24 to October 1, 1986. The list of participants of the Advanced School contains 32 names, almost all of them (with four exceptions) being computer science specialists from various Italian universities, technical Schools and other institutes.

The first lecture by L. Aceto, R. de Nicola and A. Fantechi deals with equivalences testing for event structures. They propose a flexible abstraction mechanism for models of concurrency which allows systems “looking the same” to be considered equivalent. Three different semantic models for concurrent systems are obtained, which can be used as the basis for defining interleaving, multisets or partial ordering semantics of concurrent systems. In the next, rather long and detailed (sixty pages), lecture by P. America and J. de Bakker, operational and denotational semantic models are designed for languages with process creation, and the relationships between the two semantics are investigated. The presentation is organized in four sections dealing with a uniform and static, a uniform and dynamic, a nonuniform and static, and a nonuniform and dynamic language, respectively. Here uniform/nonuniform refers to a language with uninterpreted/interpreted elementary actions, and static/dynamic to the distinction between languages with a fixed/growing number of parallel processes.

E. Astesiano and G. Reggio offer, in the next lecture, an outline of the so called SMoLCS methodology for the specification of concurrent systems and languages. Its main novelties lie in a high level of modularity and parametrization and in the fact that, within the same homogeneous framework, functions, data types and concurrency can be handled together — a concurrent system is algebraically specified as an abstract data type obtained by instantiating a parametrized schema (a parametrized abstract data type) for defining and composing process specifications. The purpose of the next lecture by M. Broy and T. Streicher is to outline shortly distinct views and ways for the description of distributed systems, to analyse and clarify the relations between various approaches to such systems, distributed programs, their specifications, and the definition of their meaning. Sometimes these distinct approaches are hard to compare and it seems rather confusing that although they look rather different nevertheless are supposed to treat the same subject.

The fifth lecture by P. Degano, R. de Nicola and U. Montanari defines a new class of Petri nets, called augmented conditions/event systems. The so called Sigma-CCS system from this class is used to give a new operational semantics to Milner’s calculus of communicating systems. CCS is provided with a semantics which is able to describe concurrency and causal dependencies, moreover, an adequate linguistic level is guaranteed for the particular class of Petri nets which can be defined through CCS systems. This lecture is followed by that by J.-Y. Girard, who discusses the relevance of a new logic, called linear logic and proposed by Girard himself, to computer science and to parallel computations in particular. This lecture is written on a rather informal and philosophical level with a more detailed explanation presented in a forthcoming study by Girard and Mascari. The next lecture deals with universal models in categories for process synchronization and is presented by A. Labella and A. Pettorossi. They consider a generic category of processes with morphisms which are labelled by strings of actions belonging to a monoid. The category of trees is defined and is proved to be optimal for most synchronizations described in the literature. Finally, the last paper of the volume by G. Mirkowska and A. Salwicki deals with an axiomatic definition of MAX-model of concurrency. For each concurrent program P there exists a set $Ax(P)$ of modal formulas such that the structure of admissible parallel executions of the program P is a Kripke model of the set $Ax(P)$ and any Kripke model of $Ax(P)$ is an extension of the structure of all admissible parallel executions of the program P .

The papers in the volume are written on a very high level of mathematical formalization and abstraction and rather extended preliminary knowledge in mathematical logic and the theory of algorithms seems to be inevitable for their more detailed study. Supposing the reader meets these demands he will appreciate the volume in question very sincerely, as it yields a top-level survey of the new, but very important and promising area of applied mathematical logic and model theory.

Ivan Kramosil

G. REINELT

The Linear Ordering Problem: Algorithms and Applications

Research and Exposition in Mathematics 8.
Heldermann Verlag Berlin 1985.
158 pages; DM 38,—.

The Linear Ordering Problem belongs to the class of hard combinatorial problems — so-called NP-hard problems — and its importance stems not only from theoretical point of view, but also from its wide occurrence in economics, scheduling, social sciences and even sport. The problem can be formulated as follows: Given a complete directed graph $D = (V, A)$ and arc weights $c(a)$ for each arc $a \in A$, the aim is to find a spanning acyclic tournament in D such that the sum of the weights of its arcs is as large as possible.

The results in the monograph under review originate from the work on the Acyclic subgraph problem, which was presented in the treatise of M. Jünger, and it should be stated without doubts that this book preserves the high scientific standard of the foregoing title. The whole treatise is divided into seven chapters.

The first chapter is intended to give a sufficient introduction to the graph theory, polyhedral theory and computational complexity theory. With regard to further exposition, maybe more detailed results from polyhedral theory would be appreciated. In the second chapter the linear ordering polytope is defined and various classes of facet defining inequalities for this polytope are derived. The solution of a particular case of the linear ordering problem is shown to be equivalent to the problem of minimizing a linear objective function over this polytope. The focus of the treatise lies in the third chapter, where the cutting plane algorithm for the solution of the linear ordering problem is described. Chapter 4 reports about the performance of algorithm when applied to the so-called triangulation problem for input-output tables, which is one of the frequently solved problems in economics. In Chapter 5 several aspects of the triangulation problem are discussed from the economic point of view.

A review of previous algorithms and approaches to the solution of the linear ordering problem is given in Chapter 6, which presents results that previously were dispersed in the literature. The volume is supplied with an extensive bibliography which includes also some unrelated titles.

The author succeeded in balancing theory and practicality, and thus the book can be recommended to wide distribution in the research environment. There is only one minor theoretical drawback: the author doesn't consider the other possible approaches to the investigation of the linear ordering problem, such as the characterization of the set of feasible solutions by its associated independence system, the effectivity of some simple heuristic algorithms. Let us hope that the results covering these open questions will appear soon.

Pavel Trska

YU. I. ARSHAVSKY, I. M. GELFAND, G. N. ORLOVSKY

Cerebellum and Rhythmical Movements

Studies in Brain Function 13. Překlad ruského originálu *Mozečok i upravljenie ritmičeskimi dviženijami*, Nauka, Moskva 1984.

Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo 1986.
Stran VIII + 166; 86 obrázků; cena 158,— DM.

Publikace je závažným a originálním příspěvkem k řešení problému funkční úlohy mozečku a organizace některých motorických subsystémů centrální nervové soustavy pro řízení lokomoce a jiných pohybů těla. Autoři zde prezentují výsledky své soustavné experimentální a koncepční

práce, jejíž začátky jsou zhruba ve druhé polovině šedesátých let. Vůbec však nejde o prostý souhrn nashromážděných dat, ale o vývojem poznávání vykrystalizovanou a bohatě experimentálními údaji dokumentovanou hypotézu.

Organizace mozkových subsystémů patří nepochybně k typickým tématům kybernetiky (resp. neurokybernetiky), o čemž svědčí i celá řada modelů — často velmi důmyslných —, které byly v uplynulých desetiletích publikovány. Mezi nimi je i několik ve své době podnětných modelů, vztahujících se k problematice funkce mozečku. Je — žel — třeba souhlasit s autory, že — ač je vztah mozečku k motorice evidentní — problém faktické jeho úlohy v řízení motoriky vyřešen není. Je třeba také souhlasit se stanoviskem — ke kterému se autoři dopracovali na základě dřívějších vlastních zkušeností —, že totiž konstrukce spekulativních modelů může být podnětná, ale v určité fázi vývoje poznání jsou nezbytné dostatečně verifikovatelné koncepce. Toto stanovisko charakterizuje pojetí v publikaci uvedených výsledků.

Obsah je rozdělen do sedmi kapitol zaměřených postupně na míšni mechanizmy chůze, na vstupní signály mozečku, na signály přenášené sestupnými drahami, na vztahy mezi vstupními a generovanými signály mozečku, na aktivitu mozečkových neuronů, na externí vstupy okruhu mícha—mozeček a na prezentaci výsledné koncepce úlohy mozečku v řízení lokomoce. Tato koncepce, jejíž empirické a experimentální základy jsou bohatě dokumentovány, je zhruba vyjádřena následujícími tézami: 1. Motorická aktivita živočichů spočívá na omezeném počtu motorických synergizmů (programů); každý z nich může být uveden v činnost relativně jednoduchým povelům. 2. Do mozečku přichází detailní informace o aktivitě motorických synergizmů. 3. Informace přicházející do mozečku jsou transformovány tak, aby reprezentovaly podstatné aspekty aktivity motorických synergizmů. 4. Mozeček sám nerealizuje motorické synergizmy a také se nepodílí na jejich spouštění. 5. Mozeček reguluje přenos signálů z různých motorických center mozku do páteřní míchy. 6. Mozeček koordinuje různé motorické synergizmy a přizpůsobuje je prostředí.

Problematika funkcí mozečku je v knize pojednávána především z neurofyzilogického hlediska; nieméně tato monografie obsahuje materiál, který je užitečný pro každého, kdo se zabývá modelováním mozkových funkcí, zejména pak systému nervového řízení lokomoce živočichů.

Zdeněk Wunsch

ROMAN HUŠEK, JOSEF LAUBER

Simulační modely

SNTL — Nakladatelství technické literatury, Praha 1986.

Stran 351; 46 obr., 30 tab.; cena 24,— Kčs.

V době, kdy rozmach mikropočítačů umožňuje rozsáhlé nasazení výpočetní techniky do výzkumu, řízení, vývoje a projektování složitých systémů i na pracovištích, která nejsou vybavena právě špičkovou výpočetní technikou, byla recentovaná publikace vydána a doporučena jako příručka pro studenty vysokých ekonomických škol. Je to šťastná událost, neboť simulace je hlavní technikou k výše uvedeným manipulacím týkajícím se složitých systémů a nadto technikou univerzální, platnou prakticky bez omezení na charakter vyšetřovaných systémů a na obor, do něhož bychom takové systémy jednotlivě kvalifikovali.

Hned na počátku je třeba upozornit, že kniha má název, který zní dosti obecně a který je jen částečně reflektován obsahem knihy: ta je zaměřena především na simulační modely systémů, které se vyskytují v ekonomických oborech, tedy především systémů, v nichž existují fronty a náhodné vlivy. To však není ve sporu s výše uvedeným tvrzením o univerzálnosti simulace: ti, kdo se setkají se simulací mimo ekonomickou sféru, tedy se simulací biologických či technických systémů, a kdo se tedy nezaměřil přímo na vznik front a na studium náhodných vlivů, by dnes už přece jen měli něco vědět jak o frontách, tak o náhodných vlivech ve složitých dynamic-

kých systémech, takže i pro ně je kniha vhodným zdrojem nových informací a stimulů k tvůrčí práci.

Členění knihy odpovídá výše uvedenému zaměření na simulaci v ekonomii. První kapitola obsahuje obecné informace o simulaci — její definici, kontext a základní třídění — a úvodní, stručné a lakonické informace o jejím použití. Druhá kapitola obsahuje podrobný výklad toho, co je třeba znát o stochastických prvcích v simulačních modelech. Třetí kapitola je zaměřena na metody Monte Carlo. Následuje dvojice kapitol zaměřených na realizaci simulačních modelů na počítačích: čtvrtá kapitola se zabývá převodem paralelních dějů simulovaných systémů na sériově implementované algoritmy číslicových počítačů, tj. především programovými prostředky pro modelování událostí, procesů a front, zatímco pátá kapitola ukazuje, jak ze záplavy programovacích složitostí vysvětlených v kapitole čtvrté dávají východisko simulační jazyky, především jazyky pro diskrétní simulaci. Některým z nich, konkrétně GPSS, CSL, jazykům SIMSCRIPT, GASP IV, SIMON a Q-GERT jsou věnovány důkladnější partie, jiným jen poznámky. Důkladnější popis je věnován i jazyku pro objektově orientované programování SIMULA, aniž by ovšem byla uváděna souvislost simulace se znalostními systémy a objektově orientovaným programováním.

Šestá kapitola je zaměřena na dnes aktuální, ale ne definitivně vyřešené problémy manipulace se simulačními modely. Sedmá kapitola obsahuje třídění a příklady použití simulace v různých oblastech operačního výzkumu a osmá kapitola je poněkud obdobně zaměřena na aplikace v ekonometrii. Závěrečná devátá kapitola je obecně orientovaná na využití simulace.

Kdo sleduje současný trend ve vývoji výpočetní techniky, kdy knihy zastarávají už během doby od sazby přes korektury po distribuci do knihkupectví, pochopí, že zaměření knihy na dávkové zpracování úloh na počítačích není anachronismem, nýbrž logickým důsledkem reality výpočetní techniky. To, že v knize není pojednáváno o interaktivní práci, o počítačové grafice v simulaci a o možnostech použití umělé inteligence v řízení simulačních studií, je pochopitelné, uvědomíme-li si, že v době přípravy rukopisu nikdo nemohl tušit, že i u nás bude v době jeho vydání poměrně velký počet profesionálních osobních počítačů vybavených přijatelnými grafickými perifériemi. Chceme však upozornit ještě na jeden aspekt této jakési „zastaralosti“ knihy: interaktivní grafická práce v simulaci může vést k velmi povrchnímu manipulování s ne právě věrnými, ale jinak efektními simulačními modely, a právě zde může mít kniha velmi důležitý výchovný vliv; jestliže se dnešní studenti na jejím základě seznámí s vnítní, tedy matematickou a implementační povahou simulačních modelů, mohou při svém setkání s interaktivními prostředky moderní mikroprocesorové simulace zapracovat na velmi vysoké profesionální úrovni a spojit klasické vidění simulačních modelů s nejprogresivnějšími technickými i programovými prostředky pro styk člověka s počítačem.

Z tohoto hlediska lze recenzovanou knihu doporučit nejen studentům ekonomie, ale i studentům jiných oborů — např. elektrotechniky, strojírenství, matematiky a chemické technologie, stejně jako pracovníkům výzkumu, vývoje, řízení a projektování. Výklad o statistických partiích diskrétní simulace, o programovacích a implementačních problémech a technikách a o použití simulace je nesmírně hodnotný, poučný a cenný.

Eržen Kindler