

BOOK REPORTS

The Book Reports section is a regular feature of *Computers & Mathematics with Applications*. It is an unconventional section. The Editors decided to break with the longstanding custom of publishing either lengthy and discursive reviews of a few books, or just a brief listing of titles. Instead, we decided to publish every important material detail concerning those books submitted to us by publishers, which we judge to be of potential interest to our readers. Hence, breaking with custom, we also publish a complete table of contents for each such book, but no review of it as such. We welcome our readers' comments concerning this enterprise. Publishers should submit books intended for review to the Editor-in-Chief,

Professor Ervin Y. Rodin
Campus Box 1040
Washington University in St. Louis
One Brookings Drive
St Louis, MO 63130, U.S.A.

Logic Programming and Non-monotonic Reasoning. Edited by Luís Moniz Pereira and Anil Nerode. The MIT Press. Cambridge, MA. (1993). 494 pages. \$35.00.

Contents:

I. Implementation. 1. Autonomous control of hybrid systems with declarative controllers. (W. Kohn and A. Nerode). 2. Implementing stable semantics by linear programming. (C. Bell *et al.*). 3. Implementing semantics of disjunctive logic programs using fringes and abstract properties. (M. Müller and J. Dix). 4. Connectionist approach to finding stable models and other structures in non-monotonic reasoning. (R. Vingrálek). 5. SLS-resolution without floundering. (W. Drabent). II. MBNF and related topics. 6. Extended logic programs as autoepistemic logic and logic programming. (V. Lifschitz and G. Schwarz). 7. Reflexive autoepistemic logic and logic programming. (V.W. Marek and M. Truszczyński). 8. Minimal knowledge + negation as failure = only knowing (sometimes). (J. Chen). 9. Autoepistemic logic programming. (P.A. Bonatti). III. Stability and related topics. 10. An assumption-based framework for non-monotonic reasoning. (A. Bondarenko *et al.*). 11. Contributions to the stable model semantics of logic programs with negation. (S. Costantini). 12. A characterization of stable models using a non-monotonic operator. (F. Teusink). 13. Negation as failure to support. (A. Torres). 14. Negation as partial failure. (B. Mobasher *et al.*). IV. Disjunctive LP, inconsistency handling. 15. Recent complexity results in logic programming and non-monotonic reasoning, and why they matter. (G. Gottlob). 16. Relating disjunctive logic programs to default theories. (C. Sakama and K. Inoue). 17. Rational default logic and disjunctive logic programming. (A. Mikitiuk and M. Truszczyński). 18. Reasoning with inconsistency in extended deductive databases. (G. Wagner). 19. Diagnosis and debugging as contradiction removal. (L.M. Pereira *et al.*). V. Nonstandard semantics. 20. Tools for deductive databases. (Y. Sagiv). 21. Scenario semantics of extended logic programs. (J.J. Algeres *et al.*). 22. An abductive framework for generalized logic programs. (G. Brewka). 23. Justification semantics: A unifying framework for the semantics of logic programs. (M. Denecker and D. De Schreye). 24. A non-monotonic reasoning formalism using implicit specificity information. (P. Geerts and D. Vermeir). 25. Reasoning in open domains. (M. Gelfond and H. Przymusińska). VI. Constructive logic. 26. An intuitionistic interpretation of finite and infinite failure. (L.T. McCarty and R. van der Meyden). 27. Canonical Kripke models and the intuitionistic semantics of logic programs. (F. Dong and L.V.S. Lakshmanan). 28. Answer sets and constructive logic, II: Extended logic programs and related non-monotonic formalisms. (D. Pearce). 29. A sequent axiomatization of three-valued logic with two negations. (D.R. Busch).

Object-Oriented Programming: The CLOS Perspective. Edited by Andreas Paepcke. The MIT Press. Cambridge, MA. (1993). 352 pages. \$39.95.

Contents:

I. Introduction. 1. An introduction to CLOS. (Linda G. DeMichiel). 2. CLOS in context: The shape of the design space. (Daniel G. Bobrow *et al.*). II. The metaobject protocol. 3. User-level language crafting: Introducing the CLOS metaobject protocol. (Andreas Paepcke). 4. Metaobject protocols: Why we want them and what else they can do. (Gregor Kiczales *et al.*). 5. Metaobject programming in CLOS. (Giuseppe Attardi). 6. The silica window system: The metalevel approach applied more broadly. (Ramana Rao). III. Comparisons with other languages. 7. CLOS and C++. (Linda G. DeMichiel). 8. CLOS, Eiffel, and Sather. (Heinz W. Schmidt and Stephen M. Omohundro). 9. CLOS and smalltalk. (Pierre Cointe). IV. CLOS uses and methodology. 10. Documenting protocols in CLOS: Keeping the promise of Reuse. (John Collins). 11. CLOS & LispView: Users' experiences distilled. (Rick Dinitz *et al.*). 12. Using CLOS to implement a hybrid knowledge. (Jiri Dvorak and Horst Bunke). V. Implementation. 13. TICLOS: An implementation of CLOS for the explorer family. (Patrick H. Dussud). 14. Efficient method dispatch in PCL. (Gregor Kiczales and Luis H. Rodriguez Jr.).

Mathematical Modelling of Complex Mechanical Systems. Volume 1: Discrete Models. By K. Arczewski and J. Pietrucha. Ellis Horwood Limited. West Sussex, England. (1993). 293 pages. \$81.00.

Contents:

1. Basic notions of modelling. 2. The framework for modelling. 3. Modelling by means of balance laws. 4. Modelling using variational principles. 5. Modelling by means of graphs.

The Computer from Pascal to von Neumann. By Herman H. Goldstine. Princeton University Press. Princeton, New Jersey. (1993). 378 pages. \$17.95.

Contents:

Part 1. The historical background up to World War II. 1. Beginnings. 2. Charles Babbage and his analytical engine. 3. The astronomical ephemeris. 4. The universities: Maxwell and Boole. 5. Integrators and planimeters. 6. Michelson, Fourier coefficients, and the Gibbs phenomenon. 7. Boolean algebra: $x^2 = xx = x$. 8. Billings, hollerith, and the census. 9. Ballistics and the rise of the great mathematicians. 10. Bush's differential analyzer and other analog devices. 11. Adaptation to scientific needs. 12. Renaissance and triumph of digital means of computation. Part 2. Wartime developments: ENIAC and EDVAC. 1. Electronic efforts prior to the ENIAC. 2. The ballistic research laboratory. 3. Differences between analog and digital machines. 4. Beginnings of the ENIAC. 5. The ENIAC as a mathematical instrument. 6. John von Neumann and the computer. 7. Beyond the ENIAC. 8. The structure of the EDVAC. 9. The spread of ideas. 10. First calculations on the ENIAC. Part 3. Post-World War II: The von Neumann machine and The Institute for Advanced Study. 1. Post-EDVAC days. 2. The Institute for Advanced Study Computer. 3. Automata theory and logic machines. 4. Numerical mathematics.

5. Numerical meteorology. 6. Engineering activities and achievement. 7. The computer and UNESCO. 8. The early industrial scene. 9. Programming languages. 10. Conclusions. Appendix: world-wide developments.

Term Graph Rewriting. Theory and Practice. By M. R. Sleep, M. J. Plasmeijer and M. C. J. D. van Eekelen. John Wiley & Sons, Inc. New York, NY. (1993). 385 pages. \$29.95.

Contents:

1. An introduction to term graph rewriting. 2. Partial type assignment in left linear applicative term rewriting systems. 3. How to get confluence for explicit substitutions. 4. An infinitary church-rosser property for non-collapsing orthogonal term rewriting systems. 5. The functional strategy and transitive term rewriting systems. 6. Graph rewriting systems for efficient compilation. 7. A fibration semantics for extended term graph rewriting. 8. A new term graph rewriting formalism: Hyperedge replacement jungle rewriting. 9. Abstract reduction: Towards a theory via abstract interpretation. 10. A lattice for the abstract interpretation of term graph rewriting systems. 11. Event structures and orthogonal term graph rewriting. 12. The adequacy of term graph rewriting for simulating term rewriting. 13. Translations into the graph grammar machine. 14. An algebraic framework for the transformation of attributed graphs. 15. Hypergraph rewriting: Critical pairs and undecidability of confluence. 16. A quick look at tree transductions. 17. Paragon specifications and their implementation. 18. MONSTR: Term graph rewriting for parallel machines. 19. A graph rewriting model enhanced with sharing for OR-parallel execution of logic programs. 20. A new process model for functions. 21. Parallel execution of concurrent clean on ZAPP. 22. Graph-based operational semantics of a lazy functional language. 23. Graph rewriting using the annotated functional strategy. 24. Implementing logical variables and disjunctions in graph rewrite systems. 25. Process annotations and process types. 26. JALPA: A functional modular programming language based on extended graphical term rewriting.

Practical UNIX Security. By Simson Garfinkel and Gene Spafford. O'Reilly & Associates, Inc. Sebastopol, CA. (1991). 482 pages. \$29.95.

Contents:

Part I. UNIX and UNIX security basics. 1. Introduction. 2. Users and passwords. 3. Users, groups, and the superuser. 4. The UNIX filesystem. Part II. Enforcing security on your system. 5. Defending your accounts. 6. Securing your data. 7. The UNIX log files. 8. Protecting against programmed threats. Part III. Communications and security. 9. Modems. 10. UUCP. 11. Networks and security. 12. Sun's NFS. 13. Kerberos and secure RPC. 14. Firewall machines. Part IV. Handling security incidents. 15. Discovering a break-in. 16. Denial of service attacks and solutions. 17. Computer security and U.S. law. Part V. Other security topics. 18. Encryption. 19. Physical security. Part VI. Appendices. A. UNIX security checklist. B. Important files. C. UNIX processes. D. How Kerberos works. E. Other sources.

Dealing With Complexity. An Introduction to the Theory and Application of Systems Science. Second Edition. By Robert L. Flood and Ewart R. Carson. Plenum Publishing Corporation. New York, NY. (1993). 280 pages. \$295.00.

Contents:

1. Systems: Origin and evolution, terms and concepts. 2. Systems and complexity. 3. Systems and measurement. 4. Systems and modeling: Diagrams and system identification. 5. Systems view of management and organizations. 6. Systems approach to "problem solving". 7. Systems theory in international relations. 8. Building models of dynamic processes. 9. Quantitative cybernetics. 10. System and model decomposition. 11. Systems science: Making sense of the philosophical issues.

Parallel Algorithms for Digital Image Processing, Computer Vision and Neural Networks. Edited by Ionnis Pitas. John Wiley & Sons. New York, NY. (1993). 395 pages. \$34.95.

Contents:

1. Introduction to parallel digital image processing. 2. Low level parallel image processing. 3. Parallel FFT-like transform algorithms on transputers. 4. Parallel edge detection and related algorithms. 5. Parallel segmentation algorithms. 6. MIMD and SIMD parallel range data segmentation. 7. Parallel stereo and motion estimation. 8. Parallel implementations of the backpropagation learning algorithm based on network topology. 9. Parallel neural computation based on algebraic partitioning. 10. Parallel neural computing based on network duplicating. 11. Parallel eikona: A parallel digital image processing package. 12. Parallel architectures and algorithms for real time computer vision.

The Network Nation: Human Communication via Computer. By Starr Roxanne Hiltz and Murray Turoff. The MIT Press. Cambridge, MA. (1993). 557 pages. \$24.95.

Contents:

Part I. The Nature of computerized conferencing 1. 1. An overview: Computerized conferencing and related technologies. 2. Development and diversification of computerized conferencing. 3. Social and psychological processes in computerized conferencing. Part II. Potential applications and impacts of computerized conferencing. 4. Potential impacts of computerized conferencing on managerial and staff functions. 5. Computer-mediated

communications and the disadvantaged. 6. Public use. 7. Science and technology. 8. Research imperatives and opportunities. Part III. Projecting the future: The technology and its regulation. 9. Structured communications. 10. The human-machine interface: Design, dilemmas, and opportunities. 11. Technology, economics, and utility. 12. Policy and regulation. 13. Societal impacts of computerized conferencing. 14. Superconnectivity: Computers, communication, and social organization.

Learning in Embedded Systems. By Leslie Pack Kaelbling. The MIT Press. Cambridge, MA. (1993). 176 pages. \$29.95.

Contents:

1. Introduction. 2. Foundations. 3. Previous approaches. 4. Interval estimation method. 5. Divide and conquer. 6. Learning Boolean functions in k -DNF. 7. A generate-and-test algorithm. 8. Learning action maps with state. 9. Delayed reinforcement. 10. Experiments in complex domains. 11. Conclusions. Appendix A. Statistics in GTRL. Appendix B. Simplifying Boolean expressions in GTRL.

UNIX Power Tools. By Jerry Peek, Tim O'Reilly and Mike Loukides. Bantam Books. New York, NY. (1993). 1119 pages. \$59.95.

Contents:

1. Introduction. 2. Logging in. 3. Logging out. 4. Passwords. 5. Organizing your home directory. 6. Setting up your terminal. 7. Shell and environment variables. 8. Setting your shell prompt. 9. How the shell interprets what you type. 10. Saving time on the command line. 11. Aliases. 12. The lessons of history. 13. Job control. 14. Redirecting input and output. 15. Moving around in a hurry. 16. Wildcards. 17. Where did I put that? 18. Finding files with find. 19. Linking, renaming, and copying files. 20. Creating and reading archives. 21. Backing up files. 22. More about managing files. 23. File security, ownership, and sharing. 24. Removing files. 25. Other ways to get disk space. 26. Showing what's in a file. 27. Regular expressions. 28. Searching through files. 29. Comparing files. 30. Spell checking, word counting, and textual analysis. 31. vi tips and tricks. 32. Creating custom commands in vi. 33. GNU Emacs. 34. Batch editing. 35. The sed stream editor. 36. You can't quite call this editing. 37. Sorting. 38. The awk programming language. 39. Perl, a pathologically eclectic rubbish lister. 40. Starting, stopping, and killing processes. 41. Time and performance. 42. Delayed execution. 43. Terminal and serial line settings. 44. Problems with terminals. 45. Printing. 46. Shell programming for the uninitiated. 47. Shell programming for the initiated. 48. Shell script debugging and gotchas. 49. C shell programming ... NOT. 50. Office automation. 51. Working with numbers. 52. Help—Online documentation, etc. 53. Miscellaneous useful programs and curiosities. 54. What's on the disk.

MH & xmh: E-mail for Users and Programmers. By Jerry D. Peek. O'Reilly & Associates, Inc. Sebastopol, CA. (1992). 687 pages. \$29.95.

Contents:

Part I. Introduction to MH and xmh. 1. MH and xmh. 2. MH and the UNIX filesystem. 3. Setting up for MH and xmh. Part II. Using MH. 4. Tour through MH. 5. Reading your mail with MH. 6. Sending mail with MH. 7. Finding and organizing mail with MH. Part III. Customizing MH. 8. Making MH and xmh work your way. 9. New versions of MH commands. 10. MH formatting. 11. Processing new mail automatically. 12. Introduction to shell programming for MH. 13. MH shell programs. Part IV. Using and customizing xmh. 14. Tour through xmh. 15. Using xmh. 16. Customizing xmh. Part V. Appendices. A. Where can you go from here? B. Early history of MH. C. Reference list. D. Converting messages to MH. E. Copies of files over the network. F. The execut programs. G. Customizing xmh: Configuration files. H. Glossary. I. Manual pages.

High Performance Computing. By Kevin Dowd. O'Reilly & Associates, Inc. Sebastopol, CA. (1993). 371 pages. \$25.95.

Contents:

I. Modern computer architectures. 1. What is high performance computing? 2. RISC computers. 3. Memory. II. Porting and tuning software. 4. What an optimizing computer does. 5. Clarity. 6. Finding porting problems. 7. Timing and profiling. 8. Understanding parallelism. 9. Eliminating clutter. 10. Loop optimizations. 11. Memory reference optimizations. 12. Language support for performance. III. Evaluating performance. 13. Industry benchmarks. 14. Running your own benchmarks. IV. Parallel computing. 15. Large scale parallel computing. 16. Shared-memory multiprocessors.

The Whole Internet: User's Guide and Catalog. By Ed Krol. O'Reilly & Associates, Inc. Sebastopol, CA. (1992). 376 pages. \$24.95.

Contents:

1. What is this book about? 2. What is the Internet? 3. How the Internet works. 4. What's allowed on the Internet. 5. Remote login. 6. Moving files: FTP. 7. Electronic mail. 8. Network news. 9. Finding software. 10. Finding someone. 11. Tunneling through the Internet: Gopher. 12. Searching indexed databases: WAIS. 13. Hypertext spanning the Internet: WWW. 14. Other applications. 15. Dealing with problems.

Power Programming With RPC. By John Bloomer. O'Reilly & Associates, Inc. Sebastopol, CA. (1992). 486 pages. \$29.95.

Contents:

1. Introduction to remote procedure calling. 2. Network computing today. 3. Developing high-level RPC applications. 4. Protocol compiling and lower-level RPC programming. 5. UNIX networking and interprocess communication. 6. Application development: Networked parallel image processing. 7. Distributing existing applications. 8. Managing RPC servers. 9. Multiple clients and servers. 10. RPC under windowing systems. 11. ONC transport-independent RPC. 12. Advanced programming issues.

Essential System Administration. By Eileen Frisch. O'Reilly & Associates, Inc. Sebastopol, CA. (1991). 440 pages. \$29.95.

Contents:

1. Introduction to system administration. 2. The UNIX way. 3. Startup and shutdown. 4. User accounts. 5. Security. 6. Automating routine tasks. 7. Managing system resources. 8. Filesystems and disks. 9. Backup and restore. 10. Terminals and modems. 11. Printers and the spooling subsystem. 12. TCP/IP network management. 13. Accounting.

UNIX In A Nutshell. By Daniel Gilly and the staff of O'Reilly & Associates, Inc. O'Reilly & Associates, Inc. Sebastopol, CA. (1992). 272 pages. \$9.95.

Contents:

Part I. Commands and shells. 1. Introduction. 2. UNIX commands. 3. The UNIX shell: An overview. 4. The Bourne shell and Korn shell. 5. The C shell. Part II. Text editing. 6. Pattern matching. 7. The Emacs editor. 8. The Vi editor. 9. The Ex editor. 10. The Sed editor. 11. The Awk scripting language. Part III. Text formatting. 12. Nroff and Troff. 13. mm Macros. 14. ms Macros. 15. me Macros. 16. Preprocessors. Part IV. Software development. 17. The SCCS utility. 18. The RCS utility. 19. The make utility. 20. Program debugging. Part V. Loose ends. 21. ASCII character set. 22. Command index.

TCP/IP Network Administration. By Craig Hunt. O'Reilly & Associates, Inc. Sebastopol, CA. (1993). 472 pages. \$29.95.

Contents:

1. Overview of TCP/IP. 2. Delivering the data. 3. Name service concepts. 4. Getting started. 5. Basic configuration. 6. Configuring the interface. 7. Configuring routing. 8. Configuring DNS name service. 9. Network applications. 10. sendmail. 11. Troubleshooting TCP/IP. 12. Network security. 13. Internet information resources. A. Network contacts. B. Forms, forms, forms. C. A gated reference. D. named reference. E. Sample sendmail.cf. F. Selected TCP/IP headers. G. Reference for passwd+. H. Software sources.

DNS and BIND in a Nutshell. By Paul Albitz and Cricket Liu. O'Reilly & Associates, Inc. Sebastopol, CA. (1993). 381 pages. \$29.95.

Contents:

1. Background. 2. How does DNS work. 3. Where do I start? 4. Setting up BIND. 5. DNS and electronic mail. 6. Configuring hosts. 7. Maintaining BIND. 8. Growing your domain. 9. Parenting. 10. nslookup. 11. Reading BIND debugging output. 12. Troubleshooting DNS and BIND. 13. Programming with the resolver library routines. 14. Miscellaneous. A. DNS message format and resource records. B. Compiling and installing BIND on a Sun. C. Top-level domains. D. Domain registration form. E. IN-ADDR.ARPA registration.

Practical C Programming. By Steve Oualline. O'Reilly & Associates, Inc. Sebastopol, CA. (1993). 396 pages. \$24.95.

Contents:

1. The basics of program writing. 2. Style. 3. Basic declarations and expressions. 4. Arrays, qualifiers, and reading numbers. 5. Decision and control statements. 6. The programming process. 7. More control statements. 8. Variable scope and functions. 9. The C preprocessor. 10. Bit operations. 11. Advanced types. 12. Simple pointers. 13. File input/output. 14. Debugging and optimization. 15. Floating point. 16. Advanced pointers. 17. Modular programming. 18. Portability problems. 19. C's dustier corners. 20. Putting it all together. 21. Programming adages.

Lex & Yacc. By John R. Levine, Tony Mason and Doug Brown. O'Reilly & Associates, Inc. Sebastopol, CA. (1993). 366 pages. \$29.95.

Contents:

1. Lex and Yacc. 2. Using Lex. 3. Using Yacc. 4. A menu generation language. 5. Parsing SQL. 6. A reference for Lex specifications. 7. A reference for Yacc grammars. 8. Yacc ambiguities and conflicts. 9. Error reporting and recovery. A. AT&T Lex. B. AT&T Yacc. C. Berkeley Yacc. D. GNU bison. E. Flex. F. MKS Lex and Yacc. G. Abraxas Lex and Yacc. H. POSIX Lex and Yacc. I. MGL compiler code. J. SQL parser code.

System Performance Tuning. By Mike Loukides. O'Reilly & Associates, Inc. Sebastopol, CA. (1991). 312 pages. \$24.95.

Contents:

1. Introduction to system performance. 2. Monitoring system activity. 3. Managing the workload. 4. Memory performance. 5. Disk performance issues. 6. Network performance. 7. Terminal performance. 8. Kernel configuration. A. Real-time processes in system V4. B. A performance tuning strategy.

Computer Security Basics. By Deborah Russell and G.T. Gangemi Sr. O'Reilly & Associates, Inc. Sebastopol, CA. (1991). 448 pages. \$29.95.

Contents:

Part I. Overview. 1. Introduction. 2. Some security history. Part II. Computer security. 3. Computer system security and access controls. 4. Viruses and other wildlife. 5. Secure system planning and administration. 6. Inside the orange book. Part III. Communications security. 7. Encryption. 8. Communications and network security. Part IV. Other types of security. 9. Physical security and biometrics. 10. TEMPEST. Part V. Appendices. A. Acronyms. B. Computer security legislation. C. Orange book and other summaries. D. Government security programs. E. A security source book.

Volume Eight: X Window System Administrator's Guide. By Linda Mui and Eric Pearce. O'Reilly & Associates, Inc. Sebastopol, CA. (1992). 346 pages. \$59.95 with CD, \$29.95 without.

Contents:

1. An introduction to X administration. 2. The X user environment. 3. The X display manager. 4. Security. 5. Font management. 6. Color. 7. X terminals. 8. Building the X window system. A. Useful things to know. B. Compiling public domain software. C. X on non-UNIX platforms. D. Resources and keysym mappings. E. The components of X products. F. Getting X11. G. Error messages.

Understanding DCE. By Ward Rosenberry, David Kenney and Gerry Fisher. O'Reilly & Associates, Inc. Sebastopol, CA. (1992). 233 pages. \$24.95.

Contents:

Part I. Components. 1. DCE: The network as computer. 2. Cells: The domain of the distributed environment. 3. Remote procedure call: The foundation of distributed computing. 4. Threads: Improving program performance. 5. DCE security service: Protecting resources. 6. DCE directory service: Locating resources. 7. DCE time service: Synchronizing network time. 8. DCE distributed file service: Providing cellwide access to files. 9. A look at writing DCE applications. Part II. Configuration and management considerations. 10. Getting started with DCE. 11. Determining your cell's boundaries. 12. Initial cell configuration guidelines. 13. Setting up security in a new cell. 14. Distributing and replicating core DCE services. Part III. Appendices. A. DCE client/server examples. B. Common DCE questions ... and some answers. C. External time providers and services. D. Registering a name: GDS and DNS.

Learning the Korn Shell. By Bill Rosenblatt. O'Reilly & Associates, Inc. Sebastopol, CA. (1993). 338 pages. \$27.95.

Contents:

1. Korn shell basics. 2. Command-line editing. 3. Customizing your environment. 4. Basic shell programming. 5. Flow control. 6. Command-line options and typed variables. 7. Input/Output and command-line processing. 8. Process handling. 9. Debugging shell programs. 10. Korn shell administration. A. Related shells. B. Reference lists.

Optimal Pricing, Inflation, and the Cost of Price Adjustment. Edited by Eytan Sheshinski and Yoram Weiss. The MIT Press. Cambridge, MA. (1993). 520 pages. \$39.95.

Contents:

I. Nominal rigidities: Microeconomic evidence and macroeconomic implications. 1. Inflation and price adjustment: A survey of findings from Micro-Data. (Yoram Weiss). 2. Individual inertia and aggregate dynamics. (Andrew Caplin). II. Tools of analysis: Mathematical theory of inventory control. 3. The optimality of (S, s) policies in the dynamic inventory problem. (Herbert Scarf). 4. Quasi-variational inequalities and impulse control. (Agnés Sulem). 5. A simplified treatment of the theory of optimal regulation of Brownian motion. (Avinash Dixit). III. Optimal pricing policies under inflation. 6. Inflation and costs of price adjustment. (Eytan Sheshinski and Yoram Weiss) 7. Optimum pricing policy under stochastic inflation. (Eytan Sheshinski and Yoram Weiss). 8. Staggered and synchronized price policies under inflation: The multi-product monopoly case. (Eytan Sheshinski and Yoram Weiss). IV. Aggregation and the effects of money on aggregate output. 9. Menu costs and the neutrality of money. (Andrew S. Caplin and Daniel F. Spulber). 10. Dynamic (S, s) economics. (Ricardo J. Caballero and Eduardo M.R.A. Engel). 11. State-dependent pricing and the dynamics of money and output. (Andrew Caplin and John Leahy). V. Search and the welfare costs of inflation. 12. Search, price setting, and inflation. (Roland Benabou). 13. Inflation and efficiency in search markets. (Roland Benabou). 14. Search, sticky prices, and inflation. (Peter Diamond). VI. Pricing policies under inflation: The empirical evidence. 15. Why are prices

sticky? Preliminary results from an interview study. (Alan S. Blinder). 16. The frequency of price adjustment: A study of newsstand prices of magazines. (Stephen G. Cecchetti). 17. The behavior of prices and inflation: An empirical analysis of disaggregated price data. (Saul Lach and Daniel Tsiddon). 18. Inflation and relative prices: Evidence from Argentina. (Mariano Tommasi).

Guide to Writing DCE Applications. By John Shirley. O'Reilly & Associates, Inc. Sebastopol, CA. (1992). 251 pages. \$29.95.

Contents:

1. Overview of an RPC application. 2. Using a DCE RPC interface. 3. How to write clients. 4. Using pointers and arrays. 5. How to write a server. 6. Using a name service. 7. Using context handles. 8. Using pipes for large quantities of data. A. IDL and ACF attributes quick reference. B. DCE RPC runtime routines quick reference. C. The arithmetic application. D. The inventory application. E. The remote_file application. F. The transfer_data application.

A book report is an informative piece of writing. Book reports are similar to book reviews but there are some important differences. Let's compare reports and reviews. A book review is a critical account of a fiction or non-fiction book where you have to give a summary of the content, assess the value of the book, and express your personal opinion whether you recommend or don't recommend it to other potential readers. Typically, book reviews are college assignments. Book Reports and Chapter Summaries. These resources are meant for teaching the parts of a book or story such as title, setting, characters, plot, climax, resolution, literary devices, and criticism. The resources are intended for elementary or possibly middle school students, both regular and ESL. Book Reports Templates: A collection of book report templates with different graphics to keep it interesting. The template contains room for title, author, setting, characters, character description, plot, and climax and resolution.