

Advances in **COMPUTERS**

Index Part I

EDITED BY

MARVIN V. ZELKOWITZ

Department of Computer Science
and Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland

VOLUME 50



ACADEMIC PRESS

A Harcourt Science and Technology Company

San Diego San Francisco New York
Boston London Sydney Tokyo

This book is printed on acid-free paper.

Copyright © 2000 by ACADEMIC PRESS

All Rights Reserved.

No part of this publication may be reproduced or transmitted in any form or by any means electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher.

Academic Press
A Harcourt Science and Technology Company
525 B Street, Suite 1900, San Diego, California 92101-4495, USA
<http://www.apnet.com>

Academic Press
24–28 Oval Road, London NW1 7DX, UK
<http://www.hbuk.co.uk/ap/>

ISBN 0-12-012150-6

A catalogue for this book is available from the British Library

This serial is covered by the *Science Citation Index*.

Typeset by Mackreth Media Services, Hemel Hempstead
Printed in Great Britain by Redwood Books, Trowbridge, Wiltshire

00 01 02 03 04 05 RB 9 8 7 6 5 4 3 2 1

Preface

As the century winds down toward the year 2000, the *Advances in Computers* has reached a milestone with the publication of these two index volumes. Ever since its first publication in 1960, the *Advances in Computers*, the oldest continuously published anthology chronicling the developments in the field of computers, has kept pace with this ever changing technology.

Computing today is vastly different from those early days. Computers were enormously expensive machines requiring temples to house their infrastructure catered to by a vast supply of technicians and fed by a constant stream of money to allow them to operate. Compare that to today where machines are ever present in our society and compete with such consumer goods as television sets for the attention and space in our homes. Whereas computing meant payroll calculations or enormously complex algorithms such as weather prediction in 1960, today computing pervades every aspect of our lives, from video games, to controlling our cars, to programming our televisions, refrigerators and toothbrushes.

I am a fan of science fiction, and for me, one of the failures of science fiction was to anticipate the computer. Early science fiction (say from the 1940s and 1950s) all depicted the computer as a large monolithic machine of enormous potential (for both good and evil) for controlling society. No writers anticipated the personal computer, available by the millions, as a relatively low-cost consumer good. None foretold of the laptop, with 'megahertz' of power available for hours at a time on battery power alone. The computer has truly revolutionized society more than anyone in 1960 could have imagined.

My own field of research is software engineering with an emphasis in experimentation and measurement. So it is only fitting that I give you some 'numbers' about the *Advances*. A typical volume contains from 5 to 7 chapters and is written by 8 to 12 authors. From 1960 through 1999, the 49 previous volumes contain 294 articles, as written by 472 authors. Some of the biggest names in computing are among the 472 contributors, and my reluctance of accidentally insulting one of them by omitting them from my list prevents me from listing some of them. Part II of this index gives the names of all these contributors.

In volumes 50 and 51 we present a combined index from the first 49 volumes. This provides an integrated source of information for researching the changes that have occurred in this industry during the latter half of the twentieth century. I hope that you find this index of use to you.

As we go into the next century, I expect the *Advances in Computers* to continue to describe the changes that occur in this ever changing technology.

If you have any suggestions for topics that need to be covered, contact me at mvz@cs.umd.edu. I hope that we will all be around for years to come as this exciting field provides new developments since we still cannot fathom what will happen next.

Marvin V. Zelkowitz
College Park, Maryland

Subject Index

- *-message, **33:96**
- *-property, **29:17, 29:19, 29:42**
 - multilevel access-control system, **29:11**
 - strict integrity, **29:25–26**
- 2×2 switch, **26:159**
- 3B20D, **26:262–267, 26:277**

A

- A++, **45:145**
- A-2 compiler, **7:143**
- A3CE, *see* Analogical and Algorithmic Cost Estimator
- AAA, *see* Automated Academic Advisor
- AAPT architecture, **44:191–192**
- Abacus, **26:50**
 - Chinese, **26:50**
- Abak, **30:273–274**
- Abbreviation, in file organization, **12:121**
- Abbreviations
 - command names, **33:121–123**
 - encoding and decoding, **33:121, 33:124**
- ABC (American Broadcasting Company), **5:15, 5:17–21**
- ABEL, hardware description language, **40:84–89**
- Ability requirements taxonomy, **31:12, 31:18**
- Abnormal scan, **6:258, 6:259**
- Absolute tolerance, and benchmarking accuracy, **41:242**
- Abstract algebra, abstract spaces, **8:323–327**
- Abstract data types (ADT), **35:139**
- Abstract data types, *see also* Data abstraction
 - adequacy and nondeterminism in, **22:327–328**
 - algebraic approach to, **22:304–308**
 - axiomatic specifications for, **22:302–308**
 - consistency and completeness in, **22:323–328**
 - functional and procedural programming with, **22:336–338**
 - future trends in, **22:348–349**
 - high-level user list operations in, **22:347**
 - implementation and verification in, **22:329–334**
 - logical completeness in, **22:324**
 - operational approach to, **22:303–304**
 - specification and implementation in, **22:295–349**
- Abstract factory pattern, **48:87, 48:104–5**
- Abstract machine model, **29:105–107**
 - alternating bit protocol, **29:106–107**
 - timed, **29:146**
- Abstract program, **29:97–99**
 - alternating bit protocol, **29:97–99**
- Abstract machine, defined, **21:99**
- Abstract mathematics, computer and, **10:96–98**
- Abstract model
 - algorithms in, **23:67–70**
 - computational complexity in, **23:72–73**
 - of information and problem solving, **23:63–74**
 - optimal complexity algorithms in, **23:74**

- optimal information in, **23:70–71**
- Abstract models, digital circuits, **37:211–216**
- Abstract OSSAD, for office modeling, **40:214**
- Abstract program, **24:127**
- Abstracting capabilities, of various design languages, **21:101**
- Abstracting, **24:237–238**
- Abstracting, automatic, **3:341, 6:11**
- Abstraction level, **28:47–48**
- Abstraction levels
 - in CDDL design, **21:100–104**
 - choice of, **21:103**
- Abstraction
 - component interconnection, **43:118–121**
 - model checking, **43:160–163**
- Abstraction, **5:198, 6:71, 35:201, 35:270**
 - in box structures, **35:167–168**
 - definition, **35:139**
 - function, **35:245**
 - graphical, **35:220**
 - synthesis, **35:214**
- Abstraction, **45:9–10, 45:24, 49:71,**
- Abstraction, in computer science, **10:10**
- Abstraction, scientific visualization software, **33:267–268**
- Abstractive mechanisms, **5:197**
- Abstractive sorting program, **5:187**
- AC, *see* Information, access control
- Academic materials, methods for creating and disseminating, **45:348**
- Academy of Sciences
 - reemergence, **29:317–318**
 - Soviet computing, **29:307–312**
- Accelerated testing, **42:111–115**
- Accelerators
 - control of, computers and, **25:288–291**
 - design and construction, computers and, **25:286–288**
 - fixed-target and colliding beam, **25:284–286**
 - P+ and NODAL programming languages, **25:290**
- Acceptability, risk, **44:22–23, 44:25–26**
- Acceptance *see* General Acceptance Method
- Accepting states, Turing machine and, **14:103**
- Acceptor-Connector pattern, **48:86, 48:93–6**
- Access control information, **14:238**
- Access control mechanism
 - data base semantics and, **14:245–246**
 - in information security, **14:236–238**
 - logical, *see* Logical access control mechanism
- Access control
 - levels of, **14:236–238**
 - matrix in, **14:239–240**
- Access control, **42:22–23**
- Access control, role-based. *See* Role-based access control (RBAC)
- Access controls
 - descriptors and, **12:26**
 - hierarchical, **12:27–28**
- Access devices, **48:129–31**
- Access matrix
 - capability-list systems in, **14:239**
 - in information security, **14:239–240**
- Access matrix, **24:69, 24:70**
- Access method, **49:335**
- Access modes, **29:3**
 - transactions and, **29:4**
- Access path, defined, **20:89**
- Access path, in query processing, **21:232**
- Access patterns, **49:245**
- Access rights, **35:8**
- Access state bits, **46:310**
- Access time, **9:220**
- Access
 - definition, **44:223**
 - information, *see* Information, access control
 - media, **44:309, 44:310–311, 44:312**
- Access
 - to bibliographic retrieval systems, **24:285**
 - intellectual, **24:280**
 - physical, **24:280**
- Access-control model, **29:42**
 - erasure, **29:22–23**
- Access-list systems, **14:239**
- Access-to-computing delays, at CSRO facility, **21:68**
- Accident, system operation, **42:8**
- ACCION International, **49:51–2**
- Account numbers, **1:20**

- Accounting, network interconnection, **42:140**
- Accreditation, **4:151, 24:334–335**
- Accumulator cells, **30:16**
- Accumulator, **9:338**
- Accuracy of approximation, **2:5, 2:27, 2:31**
- Accuracy tests for integration, **3:35–47**
- Accuracy, **26:408**
- Accuracy, *see* Result accuracy
- Accuracy, cost estimation
limitation, **44:61–62, 44:117, 44:120–122**
model, **44:98**
processes, **44:84**
process steps, **44:66–67, 44:84**
- Accuracy, desired, for bubble chamber measurements, **6:240**; *see also* Precision, resolution
- Accuracy, model, **44:135**
- Accuracy, occurrence probabilities, **42:97**
- Accuracy, SL ranging, **43:245–247**
- ACE, **5:350**
- ACE C++ wrapper facades, **48:89**
- Achorripsis*, **12:77**
- ACI, *see* Information, access control information
- ACID properties, **48:123, 48:124, 48:137, 48:146**
- ACID properties, *see* Atomicity, consistency, isolation, durability properties
- Ack-nack model, **29:156–157**
- ACM Computing Reviews, **13:175**
- ACM Conference on Language Design for Reliable Software, **20:240–241**
see also Association for Computing Machinery
- ACM's Computer Chess Committee, **29:246**
- ACM SIGPLAN Notices*, **20:211–212, 20:237**
- ACM Transactions in Programming Languages and Systems*, **20:211**
- Acoustic waveform, **11:172, 11:179**
- Acoustically controlled machines, **11:217–218**
- Acousto-optic Bragg cell, **28:193–194**
- Acousto-optic processors, **28:206–208**
- Acousto-optic systolic matrix–vector multiplier, **28:206–207**
- Acquisition and Serials Control System, in UTLAS, **21:392**
- Acquisition time, **6:222**
- Acquisition, SL ranging, **43:247, 43:252–258**
- Acronyms
for data bases of interest to chemists and available via local dial-up network terminal basis, **21:314**
for new techniques in molecular spectroscopy, **21:290**
- Acronyms, **27:371**
- Acrylic dependence graph, in parallel processing of ordinary programs, **15:145**
- ACS, *see* American Chemical Society
- ACSLs, **49:181**
- Action classes, **49:167**
- Action function, **5:133**
- Action potential, **36:208**
- Action types, of programmer, **40:8–9**
- Action Workflow system, **45:307–308**
- Action, **24:179**
- Active database functionality, **39:108, 39:116**
- Active database management systems, **39:109–112**
- Active database
concepts and design support, **39:107–189**
database design
design formalism, **39:120–122**
in information system process model, **39:116–120**
reactive system, **39:122–146**
generalized Mealy machine, **39:123–136**
Statecharts, **39:122–146**
technology, **39:107–115**
database management system, **39:109–112**
existing systems, **39:112–115**
technical concepts, **39:112–115**
- Active demultiplexing, **48:28**
- Active design review, **42:46–47, 42:52**;
see also Software inspection
- Active object pattern, **48:86–7**,

- 48:97–100
 Active rules, 39:109; *see also* Triggers
 Active Tester, 29:170
 Active transaction, 48:159
 Active-X, 48:201–2
 Activities perspective, for office modeling, 40:186, 40:190–192
 Activity management system, for office modeling, 40:239–240
 Activity node, 45:8
 Activity segment, CONLAN and, 21:147
 Activity
 management, 26:393–394
 manager, 26:434–436
 Activity, addition, 45:38
 Activity-scanning languages, world view, 33:85
 Actor model, 46:332
 Actor paradigm, for office modeling, 40:192, 40:194
 Actors, 45:5
 Actus, 35:260
 ACU, *see* Array control unit
 Acyclic form protocol validation, 29:116
 Ad hoc exception handling, 49:196
 Ada language, 20:207, 20:232–238
 non-DOD languages and, 20:254
 “package” concept in, 20:237
 Ada Process Programming Language
 with Aspen, 41:31–33
 Ada Programming Support
 Environments, 22:131–138, 22:300
 multilevel structure of, 22:133
 tools for supporting of, 22:131–133
 Ada
 copying and equality testing, 33:33
 efficiency at expense of correctness, 33:43–44
 genericity, 33:47–49
 inheritance, 33:50
 initialization and finalization, 33:27–28
 reusability guidelines, 33:30–31
 reusable software components, 33:18
 Stack_Template, 33:37–39
 swapping, 33:41
 with Time Warp mechanism, 33:105
 Ada, 24:159, 24:164, 27:31–33, 29:99, 35:267, 35:307, 45:36, 45:40
 PDL, 35:219, 43:102–103, 43:112, 43:124, 43:126
 ADABAS 4.1 teleprocessing monitor, 21:367
 Adagen, 35:235
 Adaline, 5:134
 Adams-Moulton formulas, 3:22
 Ada-95, 46:293
 Ada-like syntax, iterator and, 22:341
 Adaptability, 9:35, 49:308
 Adaptation layer, ATM, 44:298–302
 Adaptation, 5:110, 5:166, 8:38
 maturational, 5:133, 8:38
 Adapted mandatory access control
 model, 38:19–33
 advantages, 38:19
 assigned classifications, 38:28–29
 automated security labeling support, 38:21
 catalog relations, 38:30–31
 decomposition, 38:26–28
 fragmentation policy, 38:23–25
 logical design, 38:20
 required security conditions, 38:25–26
 requirements analysis and conceptual design, 38:20, 38:22
 security enforcement, 38:21
 security object, 38:20–21
 select trigger, 38:31–32
 Adapter, 49:323
 Adapter backplane, 49:323
 Adaptive, 5:118, 5:146
 computing machines, 5:130
 machines, 5:212
 threshold systems, 5:135
 Adaption, 7:60
 Adaptive CNAPS, 49:242
 Adaptive conservation equations, 33:197–198
 Adaptive critic method, 36:206
 Adaptive filter, 37:100–102
 Adaptive frames, 47:92, 47:96
 Adaptive genetic algorithm (AGA), 45:188, 45:190
 Adaptive help testbed, 47:84–6
 Adaptive information, 23:55
 integration in, 23:57
 linear, 23:75–76

- Adaptive irregular applications,
 - SPMD execution of, **45:134**
- Adaptive irregular programs, **45:109**
- Adaptive learning
 - algorithms, adjoint-sensitivity theory, **33:223–225**
 - neurodynamics model, **33:208–210**
- Adaptive learning systems, **47:90**
- Adaptive maintenance, **39:4**
- Adaptive mechanisms, **47:132**
- Adaptive mesh refinement (AMR), **45:144**
- Adaptive mutation, **45:173**
- Adaptive networks, **7:107**
- Adaptive neural-network paradigms, *see*
 - Neurodynamics model
- Adaptive parallelism, **46:339**
- Adaptive problems, **45:114**
 - compiler projects, **45:145–147**
- Adaptive program, schedule generation
 - for, **45:113**
- Adaptive time scales, neurodynamics
 - model, **33:210–211**
- Adaptive user help, **47:86–7**, **47:110–22**
 - architecture, **47:90–108**
- Adaptive User Model (AUM), **47:69**, **47:70**, **47:81**, **47:91**, **47:94**, **47:96**, **47:105**, **47:106**, **47:117**
- Adder, **2:405**
- Addition, binary, **1:237 ff**, **1:299**
- Addition, optical computing, **28:155–157**
- Additive scoring models, **39:23**
- Address Capture Device (ACD), **46:311**
- Address indication, **8:290**
- Address masking scheme, **49:255**
- Address modification, **6:123**
- Address Resolution Protocol (ARP), **48:223**
- Address, *see* Memory
- Address, cell, **9:303**
- Address, for cache access, **40:133**
- Addressing modes, ISPS and, **21:120**
- Addressing scheme
 - content-based, **49:335**
 - destination-based, **49:335**
 - source-based, **49:335**
- Addressing
 - network interconnection, **42:131–133**
 - PDU, **42:147–148**
- Address-based memory, **34:162**
 - drawbacks, **34:163**
- ADDS, *see* Amoco Distributed Database System
- ADDUP programming, for random
 - access machines, **14:36–37**
- Adele system, and process
 - representation, **41:34–35**
- Adelman's multi-attribute utility
 - evaluation model, **31:47**, **31:49–50**
- ADEPT-50 system, **12:27**
- Adequacy, office models, *see also* Office models
 - domain, **40:197–198**, **40:225–227**
 - representational, **40:198–201**, **40:227–234**
- Adjacency matrix, **16:106**
- Adjacency
 - list, **26:108**
 - matrix, **26:104–106**, **26:108**, **26:110**
- Adjacency, in color viewing, **16:12**
- Adjacent classes, **39:65**
- Adjoint system of differential equations, **3:39**
- Adjoint-convolution process, **45:90**
- Adjoint-operator algorithms, fast
 - learning, **33:225–230**
 - energy function, **33:226**
 - equilibrium points, **33:228**
 - fixed-point equations, **33:226**
- Adjoint-sensitivity theory application, **33:216–225**
 - adaptive-learning algorithms, **33:223–225**
 - back propagation algorithm, **33:216**
 - computational learning objectives, **33:221–222**
 - energy function, **33:224–225**
 - forward-sensitivity theory, **33:217–218**
 - Lyapunov stability, **33:221–222**
 - neural learning applications, **33:219–225**
 - neurodynamics derivation, **33:219–221**
 - objective function, **33:221**
 - symmetric interconnection matrix, **33:220**
 - temporal evolution, **33:219**
- Adjunct memory, **4:265**
- Adjunct strings, **17:116**

- Adjusted fitness, **45:165**
- ADL (asynchronous design language), **21:101**
 as "pure" architecture description language, **21:131**
 S_A^* and, **21:135**
- Administration on campus, computer, **4:140**
- Administration, network
 interconnection, **42:139–140**
- Administrative Behavior* (Simon), **20:6**
- Administrative roles, **46:262**
- Administrative systems, privacy in, **16:300–301**
- Admission control framework based on multiple segmentation, **47:307**
- Admission controller, **48:18**
- Admissions office, medical information system, **27:215**
- Adoption plan, for CASE, *see* Computer-aided software engineering
- ADP systems, information security and, **14:233**
- ADR, *see* Active design review
- ADSS, *see* Automated development support system
- ADT, *see* Abstract data types
- Advance commitments, **3:115–116**
- Advanced Encryption Standard (AES), **48:227**
- Advanced Micro Devices Am99C10, content-addressable memory, **34:206**
- Advanced Networks and Services (ANS), **48:183**
- Advanced On-Board Signal Processor, **27:17**
- Advanced placement (AP)
 classes in programming, **45:328**
 examinations, **45:330**
- Advanced Research Project Agency, Defense Department, **10:124**
- Advanced Technology Bomber, **27:50**
- Advertising, *see* Unsolicited information
- Advisory agents, **46:410**
- AEC Computing and Applied Mathematics Center, **5:295**
- AED system, **12:211**, **12:220–221**, **12:224**
- AED-O language, **12:211**
- AEI models, *see*
- Agent-entity-information models*
- Aerial photography, mapping and, **13:104–105**
- Aerospace command and control, **9:249**
- Aerospace computer, **9:239**
- AFAP algorithm, **37:255**
- Affinity scheduling algorithms, **45:93**
 comparison of, **45:68–69**
- Affinity scheduling methods, **45:92**
- Affinity scheduling schemes (AFS), **45:64**, **45:67**, **45:70**, **45:92**
 comparison of, **45:67**
- AFFIRM, **20:220**
- Affricates, acoustic characteristics, **31:109**, **31:111**
- AFNS, **47:265**
- AFOTEC Software Maturity Evaluation Guide, **45:200**, **45:241**
- Africa, parallel architecture, **44:206**
- AFTER, **26:131–133**
- AGAT, **29:272–274**
 advantage, **29:273**
 Dos, **29:273**
- AGE language, **22:182**
- Agent-based computing paradigm for MPSEs, **46:408–9**
- Agent-entity-information models, **40:223–225**, **40:227**, **40:230–233**, **40:239**
- Agents in adaptive teaching interface, **47:133**
- Agents perspective, for office modeling, **40:186**, **40:193–194**, **40:216–219**
- Agglomerative approach, **19:131**
- Aggregate objects, **35:283**
- Aggregates, **8:272**, **8:281–284**
- Aggregation
 binary, **39:147**
 cover, **39:147–149**
- Aggregation, component interconnection, **43:114–116**
- Aggression, **10:1**, **10:5**
- Agile manufacturing systems, **49:308–9**
- Aging/key management, **44:236–237**
- AGNET time-sharing services, **21:216–217**
- Agriculture, microcomputers in, **21:215–219**
- AHI group, Stanford University, **10:167**
- AHI system, **10:168**, **10:171**

- AHPL language, **21:95**
- Ahwatukee house control system,
microcomputers in, **21:204–209**
- AI techniques, in 3-D computer
animation, **16:27**
- AI, *see* Artificial intelligence
- AICC, *see* Token-based metrics,
Average Information Content
Classification metric
- Aid to Families with Dependent
Children, **16:230**
- AIDS, *see* American Institute for
Decision Sciences
- AIKR paradigm, *see* Artificial
intelligence knowledge
representation paradigm
- AIMACO, **4:4, 4:5, 7:144**
- AiMail, for office modeling, **40:239**
- AIMDS language, **22:181**
- Air Force Cambridge Research
Laboratory Program, **21:342**
- Air Force, U.S., **17:306**
- Air Traffic Control
displays in, **11:217**
speech in, **11:170**
- Air traffic control, **49:89**
- Airline reservation systems, computers
in, **12:66**
- Airlines flight schedules, gedanken
system and, **17:4–5**
- Air-traffic control, **26:6**
- AIST project, **18:238**
- Aizu architecture, **44:184**
- Akademset'*, **30:263–268**
architecture, **30:264–265**
educational, **30:268–271**
equipment problems, **30:267**
hardware, **30:265**
information resources, **30:264**
MTsNTI, **30:266**
NTsAO, **30:266–267**
objectives, **30:263–264**
sub-networks, **30:266–270**
telecommunications, **30:267–268**
VNIIPAS, **30:266**
- AL/X, **26:13**
- AL/X system, **22:180**
- ALAP scheduling algorithms,
37:237–238
- Alewife, **46:314–16**
- Alewife, cache coherent multiprocessor,
40:162, 40:168
- Alexander's criteria, for prototyping,
40:52–54
- Alexandria Digital Library (ADL),
48:281–2
- Alfred Bell & Co. v. Catalda Fine Arts,
Inc.*, **22:3, 22:11**
- ALGAMS, in Soviet Union, **18:247,
18:257**
- Algebra, program attribute, **42:47**
- Algebraic coding theory, in public key
cryptosystems, **22:93–94**
- Algebraic Data Systems Language,
4:24–27
- Algebraic equations, **7:112, 9:32**
- Algebraic manipulation languages, **9:52**
- Algebraic processors, **28:214–218**
bimodal optical computer,
28:215–218
matrix–matrix multiplication,
28:214–215
matrix–vector multiplication,
28:214–215
- Algebraic reconstruction techniques,
16:166–167
- Algebraic specification, **33:12–16,
36:51–53**
mathematical stack theory, **33:13–14**
versus model-based specification,
33:12–13, 33:19–20
program stacks, **33:15–16**
- Algebraic specifications
of binary tree, **22:307**
meaning of, **22:309–322**
of queue, **22:306**
- Algebraic structure, analysis of, **7:221**
- Algebras
heterogeneous, **22:309–310**
homogeneous vs. heterogeneous,
22:310
initial, **22:315–316**
of natural numbers, **22:310–312**
properties of class of, **22:309**
terminal, **22:316–320**
- Algol block structure, **16:40–41**
- Algol compiler, **12:8**
- ALGOL **10:68, 10:12**
assignment in, **10:46**
identifiers in, **10:46–50**

- information structure model of, 10:45–48
- manifest modes and, 10:53
- “possessing” in, 10:48
- ALGOL 10:60, 10:11, 10:21, 10:48–49
 - binding in, 10:65
 - compile-time invariance and, 10:55
 - dangling reference problem and, 10:57
 - identifiers in, 10:56
 - initial representations in, 10:38
 - instruction pointer in, 10:60
 - lambda calculus and, 10:28
 - semantics of, 10:32
 - structurally invariant data component in, 10:44
 - value of procedure in, 10:60
- ALGOL, 4:15, 4:17, 4:24, 5:24, 5:25, 5:46, 5:48–52, 5:66, 5:70, 5:72, 5:77, 5:83, 5:101, 5:350, 5:352, 5:368–376, 7:84, 7:85, 7:90, 7:126, 7:127, 7:129, 7:136, 7:143–146, 7:212, 8:125, 8:143–147, 8:195, 8:224, 8:229, 8:233–238, 8:254, 8:269, 9:56, 9:58, 9:61, 9:63, 9:64, 9:78
 - deficiencies, 8:238
 - Formula ALGOL, 8:54, 8:55, 8:60, 8:61, 9:56, 9:58, 9:61, 9:63, 9:64, 9:78
- ALGOL
 - context-free grammar and, 14:87
 - for large-scale programs, 14:52
- ALGOL
 - in literary analysis, 13:45
 - machine instructions in, 13:4–5
 - poetry generation with, 13:50
- ALGOL 20:60, 20:206–207, 20:211–212
- ALGOL, in Soviet Union, 18:245, 18:247
- ALGOL, similarity to LEAP language
 - associative extensions, 34:222
- ALGOL 58, 20:205
- ALGOL 60, 20:206–207, 20:211–212
- Algol-60 language, 16:40–41
- ALGOL-60, 14:61
- ALGOL-60, in Soviet Union, 18:236
- ALGOL 68, 20:211–212, 20:228–229, 20:243, 20:253
- ALGOL-68, 14:54, 14:61
- ALGOL-COBOL, in Soviet Union, 18:236
- Algorithm, 7:7
- Algorithm approach, to information problems, 23:38
 - limitations of, 23:61–63
- Algorithm complexity, 23:47
- Algorithm cost, 23:47
- Algorithm error, 23:67
- Algorithm for scheduling, 8:27
- Algorithm generator, 18:143–144
- Algorithm program, vs. heuristic, 15:47
- Algorithm SDLR, level reduction, 32:87
- Algorithm selection problem, 15:65–117, 46:410
 - abstract models in, 15:67–77, 15:82–84
 - algorithm space in, 15:67
 - approximation forms in, 15:101
 - approximation theory machinery in, 15:91–117
 - artificial intelligence in, 15:66–67
 - avoidable error in, 15:108–109
 - best features of algorithm in, 15:71–72
 - “best model” definitions in, 15:73–75
 - “best” scheduler in, 15:85–90
 - characterization question in, 15:113–115
 - concrete application of, 15:77–82
 - existence question in, 15:109–112
 - feature space in, 15:83
 - game-playing problem in, 15:69–70
 - (general) optimization theory and, 15:115
 - mathematical theory questions in, 15:109–115
 - model with variable performance criteria in, 15:75–77
 - open questions about, 15:115–117
 - operating systems in, 15:66
 - operating system schedulers in, 15:82–90
 - performance measure in, 15:67, 15:83
 - problem space in, 15:67–68, 15:78
 - quadratic algorithms in, 15:77–82
 - quadrature in, 15:66
 - scheduling algorithm in, 15:84–85
 - and selection based on features in, 15:70–73

- systematic evaluation in, **15:81–82**
 - uniqueness question in, **15:112–113**
- Algorithm space, in algorithm selection problem, **15:78**
- Algorithm synthesis, heuristic
 - knowledge-based, **15:46–57**
- Algorithm visualization, **49:98**
- Algorithm(s)
 - analysis of, **10:7**
 - attributes of, **10:15**
 - computational processes of, **10:14**
 - formalism in, **10:175–176**
 - mechanical process and, **10:8**
 - mechanical substitution in, **10:32**
 - vs. operator, **10:16**
 - recognition type, **10:175**
 - for specific problems, **10:73–74**
- Algorithm(s)
 - in abstract model, **23:67–70**
 - averaging, **23:46**
 - Batcher's odd-even merge, **23:303–307**
 - Batcher's parallel sorting, **23:303–308**
 - bitonic sorting, **23:309–311**
 - central, **23:44, 23:69**
 - Gauss, **23:61–63**
 - defined, **23:45**
 - generation of, **23:44, 23:68**
 - vs heuristic, **23:52**
 - hybrid merge, **23:328–333**
 - interpolating, **23:44, 23:68**
 - linear, **23:45, 23:70**
 - linear optimal, **23:76–77**
 - odd-even merge splitting, **23:321–326**
 - odd-even transposition sorting, **23:319–321**
 - optimal, **23:67, 23:74–77**
 - optimal complexity, **23:74**
 - overlapped merge-sort, **23:313–317**
 - parallel merge-sort, **23:342–346**
 - parallel sorting, *see* Parallel sorting algorithms
 - probabilistic, **23:53**
 - realizable, **23:50–51**
 - spline, **23:69–70**
 - two-stage parallel sorting, **23:317–318**
 - two-way and 2s-way merge, **23:333–340**
- Algorithm
 - Myers' visible surface, **16:51**
 - in picture growth, **16:16, 16:23, 16:35–38**
 - shade, **16:25**
- Algorithm
 - reliability assessment, **26:424–426**
- Algorithm, **5:214**
 - inductive inference, **5:193, 7:7**
- Algorithm, DIMN, **32:73–74**
 - modification, **32:83–85**
 - precursor, **32:67–72**
- Algorithm, implemented by a prototype, **40:45**
- Algorithm, potential volume and, **18:122–123**
- Algorithm, scheduling, **15:84–85**
- Algorithmic approach, **28:38–39**
- Algorithmic decomposition, **7:96–99**
- Algorithmic graphical representations, **35:202**
- Algorithmic information, **36:260**
- Algorithmic languages, **5:368–374**
- Algorithmic method, **4:149, 4:157**
- Algorithmic model, **44:77, 44:78, 44:114**
- Algorithmic multicomputers, military information processing, **27:24–27**
- Algorithm-specific parallel processing, **38:197–242**
 - algorithm model, **38:199–201**
 - conflict-free mappings, **38:207–211**
 - control logic, **37:234**
 - design space, **37:225**
 - applications, **38:230–240**
 - constraints, **38:222–226**
 - data-input conflict, **38:225–226, 38:233**
 - design method, **38:226–230**
 - parameters, **38:219–222**
 - general parameter method, **38:217–230**
 - mapping problem, **38:204–207**
 - notation, **38:199**
 - relation to nested-loop programs, **38:202–203**
 - time-optimal mappings without computational conflicts, **38:211–217**
- Algorithms
 - adaptive parallel, **26:123–124**
 - all-pairs shortest path, **26:125**
 - alpha-beta, parallelism in,

- 26:139–146
 alpha-beta search, 26:139–140,
 26:142
 parallelization of, 26:143
 ALSP-Kucera, 26:125–126
 Atallah and Vishkin, 26:130
 Awerbuch et al., 26:130
 BAL, 26:138–139
 Bar-On and Vishkin, 26:133
 Cheriton and Tarjan, 26:113
 COM-BAL, 26:137–138
 connected-component (sequential),
 26:105, 26:110, 26:113, 26:114
 connectivity, 26:100–101
 deterministic, 26:104
 parallel, 26:127
 Dijkstra, 26:125
 single-source shortest-path, 26:125
 exploiting parallelism in, 26:97–98
 Galil's, 26:135
 graph-theoretic, 26:131
 Hirschberg, 26:106, 26:108, 26:110,
 26:113, 26:114, 26:115
 Hyafil, 26:134
 Interior, 26:143, 26:145–146
 Leaf, 26:143–144
 minimum spanning tree, 26:115,
 26:123
 Munro and Patterson, 26:134
 NC, 26:127
 parallel
 arithmetic evaluation, for,
 26:131–133
 aspiration search, 26:143
 biconnected-components,
 26:110–112, 26:148
 connected-components,
 26:104–111, 26:148
 connectivity, 26:104–112
 Euler circuits, for, 26:129–130
 Kruskal's, 26:123, 26:148
 mandatory-work-first, 26:143
 matching, 26:127–128
 matrix multiplication, 26:105
 maximal-independent-set (MIS),
 26:128–129, 26:148
 maximum matching, 26:148
 minimum spanning tree,
 26:112–124
 MIS-KW, 26:129
 MIS-SEQ, 26:128
 miscellaneous, 26:131
 multivariate polynomials, for,
 26:134
 planarity-testing, 26:128
 polynomial evaluation, for,
 26:133–134
 Prim-Dijkstra, 26:115–122
 shortest-path, 26:124–126
 string-matching, 26:134–135
 tree-balancing, 26:135–138, 26:148
 tree-splitting, 26:143
 parentheses-matching, 26:133
 POSTFIX, 26:131–133
 PRIM-DIJKSTRA, 26:118, 26:119,
 26:123, 26:124, 26:148
 for a tree machine, 26:119–121
 for MIMD machine, 26:121–122
 PRIM-MIMD, 26:121
 PRIM-TREE, 26:119–120
 PSOLLIN-MIMD, 26:115, 26:116
 PSOLLIN-SIMD, 26:114
 Savage and Ja'Ja', 26:113
 serial balanced tree, 26:97
 shortest-path, 26:124–126
 label-correcting type, 26:124
 label-setting type, 26:124
 SOLLIN, 26:113–115, 26:123
 sorting, 26:103, 26:115
 SP, 26:124–125
 SV, 26:109
 triangular Toeplitz matrix inversion,
 for, 26:134
 Valiant and Skyum, 26:134
 Vishkin, 26:135
Algorithms
 asynchronous iterative, 19:104–105
 for asynchronous multiprocessors,
 19:93–106
 chaotic relaxation, 19:104
 complete-link, 19:142–146
 concurrent data base systems and,
 19:94–102
 correctness and efficiency of, 19:93
 “implicit form” of, 19:242
 intrinsic dimensionality, 19:130
 with large module granularity, 19:92
 mode-seeking, 19:174–175
 multidimensional scaling, 19:122
 parallel, *see* Parallel algorithms

- shuffle-exchange networks in, **19:90–91**
- single-link, **19:142–146**
- for special problems, **19:102–106**
- synchronized iterative, **19:104**
- Algorithms
 - in distributed query processing strategies, **21:237**
 - in query optimization, **21:240–256**
- Algorithms, coding of, **12:211**
- Algorithms, deterministic vs nondeterministic, **14:3**
 - see also* Cocke-Kasami-Younger algorithm; Earley's algorithm; Strassen's algorithm; Valiant's algorithm
- Algorithms, for public key cryptosystems, **22:45–102**
- ALGY, **8:53**
- Alias, program slicing, **43:21–22**
- Aliases, **40:133**
- Aliasing, **18:4–5, 33:44, 36:91–92, 37:63, 37:72–73**
- ALICE language, **22:182**
- Alignment networks, in parallel processing of ordinary programs, **15:170–174**
- All terrain vehicles, **27:51–52**
- Alliance CAD tools, **40:91**
- All-integer integer linear programming, **32:8**
- All-interconnection network
 - formulation, **32:24–25**
- All-magnetic circuits, **4:54, 4:62**
- All-Union Association Soyuz EVM Komplex, **18:254**
- All-Union Institute for Scientific and Technical Information, Soviet Union, **18:259**
- All-uses coverage, of software testing, **41:206**
- Allocate transaction, cache, **40:151**
- Allocation of computer components, **3:105–106; see also** Tape, Disk, Core storage
- Allocation of memory, **8:18**
- Allocation
 - failure intensity objectives, **42:83**
 - flow control, **42:136–137**
 - project resources, **42:83**
 - test, **42:104–106**
- Allophone, **7:186**
- ALMA environment, **43:84–85**
- ALMA system, in music representation, **12:88–89**
- ALOHA network, **21:228**
- ALP (arithmetic logic processor), **21:131–132**
- ALPAC (Automatic Language Processing Committee) report, **11:44–48**
- ALPAK, **8:53, 8:54, 8:56, 8:70, 8:98**
- Alpha rhythm, **6:65**
- Alphabet, **8:226, 8:270, 8:271**
- Alphabet, of a programming language, **7:130**
- Alphabeta function, **26:142**
- Alphabetical operator, in computational theory, **10:16**
- Alphabets, input and output, **39:123**
 - intersecting, in Statechart machine, **39:144–145**
- Alpha-7 architecture, **44:202**
- Alpha-beta algorithm, **18:94–97, 29:206–210**
 - flowchart, **29:207–208**
 - using transposition tables, **29:216–217**
 - iterative deepening, **29:222**
 - number of nodes scored by, **29:210**
 - search backed up to a position, **29:214**
 - search trees, **29:206–207, 29:209**
 - two-pass search, **29:233–234, 29:236**
 - unsynchronized iteratively deepening parallel search, **29:227–228**
- Alpha-beta bounds, **26:142**
- Alpha-beta window, in computer chess, **18:97–98**
- Alpha-Dos, **30:232**
- Alphanumeric constants, **8:271**
- Alphanumeric keyboard, **7:276**
- Alphard-like specifications, vs. algebraic, **22:308–309**
- ALPINE, **7:286, 7:287**
- ALPS, **7:71–73**
- ALSTEN, **25:233–234**
- ALT properties, *see* Transitions model
- Alternating bit protocol, **29:89–90**
 - assumptions, **29:143–144**

- CCS model, **29**:102–103
 CFSM model, **29**:90–91
 CSP model, **29**:100–101
 ETG model, **29**:107–108
 FSA model, **29**:111–112
 global state graph, **29**:112–113
 ITTG model, **29**:153–155
 Petri net model, **29**:96–97
 TG model, **29**:93–95
 Alternating bit protocol, **42**:196
 Alternating-bit protocol, **26**:406, **26**:410
 Alternating direction iterations, **5**:318
 Alternating direction methods, **2**:31,
 2:34, **2**:37 ff, **2**:49
 Alternative hypothesis, **44**:129, **44**:130
 Alternative interpretations, plausibility
 of, **17**:70
 Alternative multimodal user interface,
 47:46
 ALU, *see* Arithmetic/logic unit
 ALUs (arithmetic logic units), in ADL
 language, **21**:131–132
 Am99C10, Advanced Micro Devices
 content-addressable memory,
 34:206
 Amber, **35**:276
 Ambient lighting, SL ranging, **43**:248
 Ambient scene parameters, **34**:95
 Ambiguities
 permanent predictable, **8**:176, **8**:177
 syntactic, **8**:157–163
 Ambiguity of programming languages,
 7:139–141
 Ambiguity
 apparent, in centralization effects of
 computers, **25**:382–384
 resolution in database query
 interpretation, **25**:31–32
 Ambiguity, handling, **38**:79–80
 Ambiguous grammar, **14**:82
 AMBIT, **9**:108
 Ambulatory care department, medical
 information system, **27**:215
 AMCEE, *see* Association for
 Media-Based Continuing
 Education for Engineers
 Amdahl's Law, supercomputer
 performance, **27**:312–315
 American Association for the
 Advancement of Science, **10**:10,
 22:102
 American Broadcasting Company *see*
 ABC
 American Chemical Society
 Chemical Abstracts Division,
 21:311–312
 as "chemical establishment" unit,
 21:318
 chemistry education and, **21**:277,
 21:282
 Committee on Professional Training,
 21:277–278
 computer applications in, **21**:320–321
 Division of Chemical Education,
 21:282
 Division of Computers in Chemistry,
 21:320
 American Civil Liberties Union, **17**:305
 American Council of Education, **16**:229
 American Documentation Institute
 (ADI) (now American Society for
 Information Science (ASIS)),
 48:259
 American Documentation Institute,
 31:339
 American Geophysical Union, **18**:156
American Heritage Dictionary, CRT
 editing of, **10**:167–169
 American Institute for Decision
 Sciences, **20**:25
 American Library Association, **21**:344,
 21:386
 American National Standard for
 Bibliographic Information
 Interchange, **21**:407–410
 basic structure and substructures of,
 21:407–408
 fixed-length data element field of,
 21:409
 American National Standards Institute
 (ANSI), **35**:336
 American National Standards Institute,
 20:212–217, **20**:237, **21**:287
 American Nuclear Society, **5**:290, **5**:296
 American Petroleum Institute, **21**:314
 American Society of Information
 Science, special interest groups,
 31:343–345
 American Standards Association, **20**:213
 American Telephone and Telegraph

- Company, microcomputer services by, **21**:219–220
- American University, **4**:145
- American
 Engineering Standards Committee, **8**:110
 Society for Testing and Materials (ASTM), **8**:112
 Standards, **8**:109–115, **8**:138, **8**:139
 Standards Association *see* ASA
- Americas, parallel architecture, **44**:207–209
- Amethyst, **49**:105
- AMICAL, for system synthesis, **40**:94–95, **40**:110
- Amino acids, in biomedical pattern recognition, **10**:241
- Amoco Distributed Database System (ADDS), **32**:179, **32**:187
- Amoeba system, **49**:311
- Amoeba, **35**:280
- Amortized communication complexity, **44**:344
- Ampere's law, **11**:240
- Ampex TBM, **27**:169, **27**:171
- Amplifier, free jet interaction, **4**:199–202
 turbulence, **4**:202–205
 vortex chamber, **4**:224–229
 wall interaction, **4**:210–224
- Amplifiers, balanced circuit, **11**:266–272
- Amplifiers, vacuum tunnel effect, **2**:190 ff, **2**:286
 electron-beam, **2**:196
- Amplitude *see* Structure
- Amplitude modulation, **7**:60
- AMR++, **45**:145
- AMS, *see* Activity management system
- AN/USQ-67, **20**:90, **20**:97–98
- ANACHIES, **47**:83, **47**:89
- ANALITIK, in Soviet software, **18**:236
- Analog advantages, **9**:30
- Analog computer graphics, **16**:18–19
see also Three-D computer animation
- Analog computer, history, **37**:383–384
- Analog computers (in crystallography), **5**:258
- Analog computers, **2**:360 ff, **2**:364, **4**:136
- Analog control system, **23**:178
- Analog hardware, **9**:26, **9**:46
- Analog mechanization, **7**:48
- Analog operational amplifier, **36**:207
- Analog optical processor, **7**:59
- Analog optics, **28**:216
- Analog processors, **28**:175
 applications
 spectrum analysis, **28**:193–196
 synthetic-aperture radar, **28**:189–193
 correlators, **28**:178–179
 Fourier transform processors, **28**:175–178
 image processors, **28**:182–186
 nonlinear processors, **28**:186–189
 spatial filters, **28**:178–182
- Analog programming, **9**:47
- Analog signals, **23**:178
 vs digital, **23**:179
- Analog signals, transmission of, **10**:116
- Analog software, **9**:46
- Analog storage, **9**:12
- Analog/digital conversion, in local computer networks, **20**:89
- Analogical and Algorithmic Cost Estimator, **44**:119–120, **44**:122
- Analogical model, **44**:71, **44**:78, **44**:103–105, **44**:113–114
- Analogies, method of, in computer chess, **18**:98
- Analog-digital communication, **9**:26
- Analog-digital conversion, **10**:116
- Analog-digital converter, **9**:44
- Analog-digital techniques, **7**:57
- Analog-digital variables, **3**:288–296; *see also* Arithmetic
- Analog-to-digital computers, microcomputers and, **21**:175
- Analog-to-digital conversion system, DSP, **37**:64–66
- Analog-to-digital converter, **37**:64
- Analog-to-digital, *see* Converters
- ANALOGY program, **13**:222–223
- Analysis of Cell Images, The* (Prewitt-Mendelsohn), **12**:368
- Analysis Patterns*, **47**:290
- Analysis of variance, in inventory simulation, **10**:135
- Analysis
 dynamic, **26**:346, **26**:387
 static, **26**:346, **26**:347, **26**:349, **26**:354, **26**:387

- Analysis
 research, *see* Investigation, software engineering
 risk, *see* Risk management, IT
 Analysis, and process models, **41:22**, **41:27**, **41:51**
 Analysis-by-synthesis method of translation, **7:230**
 Analysis-by-synthesis research, musical performance, **36:180–195**
 Analysis–synthesis–evaluation paradigm, **28:30–34**
 problems, **28:31–33**
 Analytic evaluation of programming language processors, **7:173**
 Analytic hierarchy process, **46:134–5**, **46:136**
 Anaphoric reference, in LUNAR system, **17:70–71**
 Anatomic visualization, **47:194**
 Anchor specialists, in CHOLESTASIS program, **22:233**
 AND gate, optical, **28:166**
 AND state refinements, **39:137–144**
 AND/OR
 gates, **26:304**
 implementations, **26:315**
 AND, **24:295–299**
 Andalternate windows, in HXDP system, **20:70**
 Anderson report, **29:6**
 Anderson's language acquisition system, **15:194–195**
 AND-gate, **4:175**, **4:190**, **4:196**, **4:205**
 AND-OR graph, **13:214**
 AND-OR graph search concept, **13:216–217**
 AND-OR logic, in magnetic bubble logic, **17:247**
 Andrew File System (AFS), **48:167**
 Androids, **6:41**
 Anecdotal analysis, **42:51**
 ANESTHETIC frame, **22:289**
 ANGEL model, **44:103–104**, **44:105**
 Anglo-American Cataloging Rules, **21:338**
 ANIM, **49:105**, **49:106**
 Anima II system, **16:44–52**
 digital video information storage and retrieval for, **16:51–52**
 graphics programming language for, **16:47–51**
 hardware for, **16:45**
 options in, **16:49–50**
 preview and edit stages of, **16:51**
 production stages in, **16:44–48**
 ANIMATE, **47:123**
 Animated Film Language, **13:118**
 Animated films, first, **16:7–8**
 Animated help, **47:123**
 Animation, **47:233–6**
 Animation, first, **16:6**
 see also Three-D computer animation
 Animation, in videogames, **23:125–129**
 Animation, scientific visualization software, **33:265–266**
 need for sophisticated, **33:284**
 ANNA, **33:18**
 Annealing schedule, **37:133**
 Annealing, **45:157–159**
 Annotation, design language, **43:159–160**
 Annual Review of Information Science and Technology, **21:342**
 Anomalous problem, **37:359**
 Anomaly
 deletion, **35:67**
 insertion, **35:67**
 update, **35:67**
 Anonymity, *see also* Confidentiality; Privacy
 access control, **44:246–262**
 usage control, **44:276**
 Anonymizer, **48:213**
 Anonymous exceptions, **49:217**, **49:228–9**
 ANSBII, *see* American National Standard for Bibliographic Information Interchange
 ANSI BASIC, **21:287**
 ANSI COBOL, **21:287**
 ANSI FORTRAN, **21:287**
 ANSI MUMPS, **21:287**
 ANSI/SPARC architecture, **46:278**, **46:280–1**
 ANSI, *see* American National Standards Institute
 Ant colony, **45:182–183**
 Ant cycle, **45:186–187**

- Ant system, **45:184**
- Antenna pattern, **7:241**
- Anthropometric harness, **16:19**
- Antialiasing filter, **37:65–66**
- Antimicrobial therapy, **16:171**
- Anti-image filter, **37:66–67**
- Anti-intrusion alarms, **12:33**
- Anxiety, **6:103**
- AnyBoard, for hardware reusability,
40:100, 40:102
- Aortic valve, **47:220**
- AOSP, **27:41**
- AP1000 architecture, **44:182**
- APE (autonomous processing element)
system, **14:202–205**
- APE machine, **26:60**
- APG, *see* Automatic program
generation
- API, *see* Application programming
interface
- APILIT data base, **21:314**
- APIPAT data base, **21:314**
- APL language, **15:4–6, 22:180**
- APL system
vs PPL system, **14:54–55**
for small-scale programs, **14:49–50**
- APL/360, **20:207, 20:211, 20:217, 20:229**
- APL, **10:213**
- AP-120B Floating-Point Systems,
20:118, 20:136, 20:160–164
- AP-120B processor, block diagram of,
20:161–162
- AP Chemistry Newsletter*, **21:279**
- APNs, *see* Augmented Petri nets
- Apollo Mission Simulator, **6:95, 6:129**
- Apollo mission, **9:270**
- Appl/A, *see* Ada Process Programming
Language with Aspen
- APPLE (Associative Processor
Programming Language), for
STARAN, **34:214**
- Apple-II clone, AGAT, **29:272–274**
- Appliances, microcomputer applications
in, **21:198–200**
- Application benchmark, **41:236**
- Application data units (ADUs), **48:30**
- Application domain, **40:13**
- Application domain-based testing, **49:144**
- Application generators, **34:13–15,**
34:41
- Application interconnection graph,
43:65
- Application layer, **48:221**
protocol, **48:220, 48:221**
- Application packages, stripped-down
versions
benchmarking, **27:326**
performance, **27:344–348**
- Application program, **26:9, 47:174**
- Application programming interface,
41:110, 41:133, 44:324
- Application requirements, **35:9**
- Application specific reuse, **34:25–27,**
34:41
- Application-dependent modules,
16:200–214
- Application-programming interface
(API), **48:172**
- Application-sharing technologies,
45:292
- Application-specific integrated circuit,
see ASIC design
- Applications dimension, of
organizational structure, **41:115**
- Applications of residue numbers,
6:179–182
- Applications perspective, for office
modeling, **40:187, 40:195–196,**
40:221, 40:223
- Applications processors, in OCLC
system, **21:386–387**
- Applications program standard, **42:151**
- Applications programmer interface
(API), **45:141–142**
- Applications prototyping, **31:9**
- Applications software, Soviet, **30:285**
- Applicative State Transition System,
20:243
- Applied mathematics, objectives of,
10:87–89
see also Mathematics
- Applied parallelism, **9:289, 9:294**
- Apportionment, **5:4**
- Approximate information, uncertainty
of, **23:83**
- Approximation *see* Accuracy,
Chebyshev,
Least squares, Mixed problems,
Nonlinear
- Approximation forms, in algorithm

- selection problem, **15:91–95**, **15:101–106**
- Approximation techniques, **18:10–12**
- Approximation theory *see also*
 - Algorithm selection problem in algorithm selection problem, **15:91–117**
 - approximation forms in, **15:101–106**
 - characteristic question in, **15:112–115**
 - classification of problems in, **15:95–96**
 - complexity in, **15:98–100**
 - degree of convergence in, **15:96–97**
 - discrete form in, **15:101–102**
 - general nonlinear forms in, **15:106–107**
 - linear forms in, **15:102–105**
 - mathematical theory questions in, **15:109–115**
 - norms and approximation forms in, **15:91–95**, **15:101–106**
 - piecewise linear forms in, **15:105–106**
 - robustness in, **15:100**
 - tree and algorithm forms in, **15:108**
 - uniqueness question in, **15:112–113**
- Approximation theory, **36:238**
- A priori domain analysis, **49:155–71**
- APSE, *see* Ada Programming Support Environments
- APS-48 architecture, **44:205–206**
- APT, **5:352**, **7:125**, **7:127**, **7:144**, **7:146**, **7:282**, **8:149**
- APT language, **20:209**, **20:215**
- APTIC data base, **21:314**
- ARA97 (ability – role assignment) model, **46:266–7**
- Arabic **7:216**, **7:219**
- ARBAC₀, **46:258**
- ARBAC₁, **46:258**
- ARBAC₂, **46:258**
- ARBAC₃, **46:258**
- ARBAC97 administrative models, **46:257–70**
- ARBAC97 model, components, **46:261**
- Arbitrary constants, freedom from, **18:166–168**
- Arbitrary Shuffle, **29:160–161**
- Arcade games, **23:94**, *see also* Videogames
- Arch/maintainer's assistant, **35:225**
- Archetypal register-transfer languages, **21:100–101**
- Archimedes, **26:50**
- Architectural description, **46:334**
- Architectural design level, **49:198**
- Architecture description language, **24:120**
- Architecture
 - ATM, **44:293–303**
 - parallel, *see* Parallel architecture; *specific parallel architectures*
 - quality of service, **44:326**
 - switching, **44:304**
- Architecture
 - backpropagation networks, **37:397–398**
 - cascade-correlation, **37:128**
 - dataflow trends, **37:307–309**
 - early, **37:291–295**
 - neural networks, **37:122–134**
 - functional-link net, **37:125–126**
 - multilayer feedforward networks, **37:126–128**
 - single-layer networks, **37:123–126**
 - perceptron, **37:123–125**
 - multilayer, **37:398**
 - programmable DSP chip, **37:105–106**
 - memory, **37:109–111**
- Architecture
 - block-oriented database, **28:121–123**
 - database machines, **28:109**, **28:118**
 - endo-architecture, **28:35**
 - exo-architecture, **28:14**, **28:24–26**
 - fully parallel associative, **28:120–121**
 - management intelligence systems, **28:253–257**
- Architecture
 - cache coherence impact on, **40:174**
 - IOIS, **40:242**
 - models, **40:84**
 - multistage interconnection network, **40:129**, **40:131**, **40:156–161**, **40:167–168**
 - software prototyping and, **40:45**
 - three schema, **40:200**
 - VHDL and, **40:84**
- Architecture
 - locomotion control, **42:262–264**
 - software reliability engineering,

- 42:82–84
- Architecture
 military information processing,
 27:24–29
 algorithmic multicomputers,
 27:24–27
 trained neural networks, 27:27–29
 vision, 27:307
- Architecture, 9:301
- Architecture, and benchmarking,
 41:247–248
- Architecture, interface issues, 43:52–53,
 43:54; *see also* Component
 interconnection
- ARCHIVE utility, at CSRO, 21:43
- Ardenne tube, 9:7
- Area (in information algebra), 4:42
 set, 4:44
- Area scan, 6:267
- Area, 26:195
- Area, in digital picture, 18:49
- Area-time trade-off, 44:353–354
- Argonne Code Center, 5:296, 5:325
- Argonne National Laboratory, 21:298
- Argument set, 19:38
- Ariadne/HOPLa, 46:349–50
- Arifoglu process, 44:81, 44:84, 44:85
- Arithmetic calculator, 4:295
- Arithmetic expression, defined,
 15:128–129
- Arithmetic expressions
 parallel algorithms for evaluating,
 26:131–133
 postfix form, 26:131–133
- Arithmetic hierarchy, problem
 complexity, 43:217, 43:238–239
- Arithmetic logic processor, 21:131–132
- Arithmetic logic units (ALUs),
 26:228–229
- Arithmetic mean, 39:65
- Arithmetic operations, in LSD base
 language, 12:264
- Arithmetic pipeline, operational
 configuration for, 20:140
- Arithmetic/logic circuit level, in pipeline
 processors, 20:130
- Arithmetic/logic unit, in HXDP system,
 20:58
- Arithmetic
 in burst processing, 14:216
 probability theory and, 14:191
- Arithmetic, analog-digital, 3:283–288
 hybrid, 3:285
- Arithmetic, types of, in formula
 manipulation, 8:57, 8:58, 8:82
- Arithmetical formulas, operators, 5:29,
 5:35, 5:36, 5:74, 5:96, 5:103
- Arithmetic-expression tree-height
 reduction, 15:128–137
see also Parallel processing of
 ordinary programs
- Arizona, University of, 4:145
- ARM2 work stations, 30:276
- Armchair model, 47:43
- ArMen architecture, 44:195
- Army domestic intelligence program,
 16:269
- ARPA computer network, 10:124
- ARPA network, 16:187–188,
 16:195–196, 16:198–199, 16:207,
 16:213
- ARPANET (Advanced Research
 Projects Agency Network), 20:91,
 20:231
- ARPANET computer network, 21:41,
 21:228
- ARPANET network, 17:166, 17:195
- ARPANET, 42:137, 42:148–149, 48:182
- ARPA-SDC Time Sharing System, 8:9,
 8:10, 8:34
- Array assignment axiom schema,
 36:62–63
- Array computer, 9:17
- Array control unit, of MPP processor,
 20:175
- Array dimensions, binding of, 10:67–68
- Array process, *see* Processor array
- Array processor, 6:269, 7:50–53
- Array processors, *see also* Vector
 processors
 BSP-like, 20:121
 future trends in, 20:189–191
 as pipeline processor, 20:180–182
 recent advances in, 20:164–179
 SIMD computers as, 20:118
 vectorization in pipeline type,
 20:124–126
 vector length in, 20:183–184
 vector processing in, 20:120–122
- Array unit interconnection, of 8-PE