BOOK REVIEW

Annual Review of Computer Science, vol. 2, 1987, pp. 565, (ed.) J. F. Traub, (Published by Annual Reviews Inc., 4139 El Camino Way, Palo Alto, CA 94306, USA), Price: US$ 39, elsewhere $ 42.

The second volume of *Annual Review of Computer Science* is a collection of good papers providing up-to-date information in the most important areas of research in computer science, including artificial intelligence (AI). The papers provide enough background material so that they can be read, understood and appreciated independently. They also provide a good list of references for further information.

Lisp evolved to serve the world of evolutionary, extrapolatory programming. A modern Lisp system contains a lot of built-in facilities compared to other languages. It is this extensive base that separates a professional Lisp implementation from a toy implementation.The paper 'Common Lisp' (S. E. Pahlman) explains the need for common, stable Lisp, facilities offered by a typical Lisp environment, its role in evolutionary programming, and hardware essential for such an environment.

Designing, understanding and reasoning about distributed systems are complicated because of the uncertainties inherent in the system and lack of global knowledge. The paper 'Using reasoning about knowledge to analyse distributed systems' (J. Haplern) describes how to formally use knowledge in distributed systems and the gains of such formalisms. The systems could be computer systems, robots or people. Common knowledge is a prerequisite in the co-ordinated actions and agreement, and this seems to be a useful tool for specifying, synthesizing and verifying protocols.

The discovery of various constraints and techniques for using them are essential to guide inferences from 2-D image data to 3-D scene descriptions. The paper 'The emerging paradigm of computational vision' (S. Zucker) describes computational vision along with the related issues of biological and engineering vision systems. General computational theories hold abstractly for all applications and provide the means for integrating the diverse constraints.

The intelligent robot must be able to reason symbolically. The mobile robot is always encountering new and unexpected events. So static models or preloaded maps are inadequate to represent the robot world. The paper 'Vision and navigation for the Carnegie-Mellon Navlab' (C. Thorpe *et al.*) describes the tools a mobile robot (Carnegie-Mellon Navlab) uses to bridge the chasm between the external world and its internal representation, such as sensors, image understanding to interpret sensed data, geometrical reasoning, concepts of time and of motion over time.

An intelligent artefact must be capable of reasoning about the world it inhabits. The technical problem of AI is to characterize the pattern of reasoning required of such an intelligent artefact. One such reasoning technique is nonmonotonic reasoning. The paper 'Nonmonotonic reasoning' (R. Reiter) describes various nonmonotonic abstractions such as default logic. Circumscription along with some objections such as probability theory may be more appropriate. Nonmonotonicity appears to be the rule rather than the exception in much of what passes for human common-sense reasoning.

Logic and deduction is related to problem solving in more than one way. The paper 'Logic, problem solving and deduction' (D. V. McDermott) describes the work on such relationships. The role of logic in AI depends on several issues such as how effective the automated deduction algorithm can become, how many problems can be cast as deductive problems, and how important logic is to the study of knowledge representation.

Reasoning about actions and plans is fundamental to the development of intelligent machines that are capable of dealing effectively with real-world problems and is one of the areas of investigation in AI research. The paper 'Planning' (M. P. Georgeff) discusses the issues that are relevant in reasoning about actions and plans and describes how to represent actions and events, and plan synthesis.

Language generation and explanation systems allow computer systems to respond to their users in natural language. Language generation can aim to produce short answers, paragraph-length responses or summaries. Explanation is a subtype of language generation. The paper 'Language generation and explanation' (K. R. McKeown and W. Swartout) describes the problem of language generation, the range of system organizations and solutions used, and applications for which generation systems have been built.

Search is a universal problem-solving mechanism in AI. Typical search tasks range from solving Rubik's cube to playing games such as chess, to solving complex practical problems such as medical diagnosis. The paper 'Search techniques' (J. Pearl and R. E. Korf) discusses various search techniques such as brute-force techniques, heuristic techniques and game-tree searching techniques.

Software development requires knowledge about software engineering and programming techniques, the application domain, the machine upon which the software will be run, and the decisions already made about the implementation of the software under development. The current software tools are adept at dealing with the clerical details of programming and are of little support in knowledge-intensive aspects of software development. The paper 'Knowledge-based software tools' (D. R. Barstow) describes the various research efforts in the development of software tools that can exploit the knowledge to some extent like a programmer. These tools can assist the programmer in the development of systems and programs and may offer substantial productivity gains.

Protocols provide communication between local as well as remote processes. Protocols can also be used to integrate a partitioned complex function on a single processor system. There is a need for standard protocols so that teleprocessing-equipment manufacturers will be able to interconnect various equipment efficiently. The paper 'Network protocols and tools to help produce them' (H. Rudin) describes the formal protocol specification concepts and tools that may be driven from a machine-readable formal specification.

Temporal logic can be used for specifying and verifying concurrent programs and for verifying network protocols. The construction of proof in hand is very tedious. The paper 'Research on automatic verification of finite-state concurrent systems' (E. M. Clarke and O. Grumberg) describes the syntax and semantics of temporal logic and its application in verifying concurrent systems.

Many computational problems have benefitted greatly from the study of the mathematical structure underlying the problems and this may help in devising more efficient algorithms than naive enumeration. So it may be fruitful to study the mathematical structure underlying the NP-hard problems and develop new techniques for solving them. The paper 'Algorithmic geometry of numbers' (R. Kannan) provides a self-contained introduction to algorithmic geometry of numbers and surveys its application to cryptography and diophantine approximations among others.

Computational algebra is helpful in the construction of computer systems that enable scientific and engineering users to carry out mathematical manipulations automatically. The paper 'Computer algebra algorithms' (E. Kaltofen) describes the development of efficient arithmetical algorithms and the relationship of computer algebra to other fields such as logic programming and AI.

Linear programming has enormous practical importance. Improvements by orders of magnitude in the efficiency of methods would open up many new application areas, with substantial economic gains. The paper 'Linear programming 1986' (N. Megiddo) describes various algorithms such as Karmarkar's algorithm and linear rescaling algorithm, and surveys recent algorithms for linear programming.

The life-threatening consequences and significant economic impact of computer failure are ponderous. System reliability is the major concern of manufacturers, semiconductor fabricators and software engineers. One of the principal mechanisms for achieving high reliability and high availability in digital systems is fault tolerance. The paper 'Techniques and architectures for fault-tolerant computing' (R. A. Maxion *et al.*) provides a selected review of fault-tolerant computing.

Computers are transforming the way we learn and the way we work. Computers are being used for special education, adult literacy programmes and training in industry and the military. The paper 'Computers in education: a historical overview' (D. M. Kurland and I. C. Kurland) describes the development of educational computer applications and their impact on education at the pre-college level.

In a nutshell, the volume has something for everyone presented in concise fashion.

G. KRISHNA

Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560 012