Acta Cryst. (1977). A33, 13–18

A Correlation between Crystallographic Computing and Artificial Intelligence Research

BY E. A. FEIGENBAUM AND R. S. ENGELMORE

Computer Science Department, Stanford University, Stanford, California 94305, *USA*

AND C. K. JOHNSON

Chemistry Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37830, *USA*

(*Received* 5 *April* 1976; *accepted* 23 *July* 1976)

Artificial intelligence research, as a part of computer science, has produced a variety of programs of experimental and applications interest: programs for scientific inference, chemical synthesis, planning robot control, extraction of meaning from English sentences, speech understanding, interpretation of visual images, and so on. The symbolic manipulation techniques used in artificial intelligence provide a framework for analyzing and coding the knowledge base of a problem independently of an algorithmic implementation. A possible application of artificial intelligence methodology to protein crystallography is described.

Introduction

Crystallographers have been fascinated by computing devices for many years and have done much pioneering work in the design and utilization of such devices for crystallographic research. Machines such as the 1948 analogue Fourier summation device X-RAC (Pepinsky, Van den Hende & Vand, 1961) are firmly imbedded in the history of crystallography. The potential of the general-purpose digital computer was noted very early, and there is now a substantial store of knowledge in crystallographic computing.

Crystallography has made major advances by using the computer as a fast arithmetic engine in the basic crystallographic steps of data reduction, phase determination, Fourier series inversion, structure factor calculation and refinement of the trial model. These successes have tended to focus the crystallographer's attention on the design and implementation of more and better numerical algorithms, and to obscure the potential utility of the computer as a 'physical symbol system' (Newell & Simon, 1976). We are, of course, aware of the use of symbolic addition in direct methods, but the vast majority of crystallographic programs are based on numerical algorithms. In some applications we may already be at the point of diminishing returns with respect to numerical methods, as for example in the refinement of macromolecular structures. Numerical refinement often does not improve a deficient model. Here one needs a technique for pointing out where the model needs correcting. This problem is basically one of reasoning about relations between elements of the model, employing physical, chemical and crystallographic knowledge, much of which is informal and imprecise. Methodologies now exist for programming such problems, and have proved to be useful in a number of diverse applications.

An exemplary crystallographic problem

The following example is given to illustrate the similarity between a particular crystallographic research problem and a specific research domain in the field of artificial intelligence (AI).

The problem of deriving the coordinates for a trial protein structure, given an electron density map, the amino-acid sequence and the stereochemical principles and constraints known to apply, is one which currently requires a considerable amount of crystallographer time and effort. Work by Greer (1974) has produced some promising results toward an automatic tracing of the polypeptide chain in the density map. The overall problem is particularly interesting because it involves an integration of information from diverse sources of chemical and crystallographic knowledge.

The density-map interpretation problem is closely related to the classical 'scene analysis' problem (Winston, 1972) associated with vision processing in robot control. A common problem domain in scene analysis is the 'blocks world' consisting of a number of polyhedral blocks on a table. The input data are in a digitized two-dimensional array of reflectivity values from a television camera scan of the table and its contents. The problem is to derive a data structure describing the positions and shapes of the three-dimensional blocks so that a plan of action for moving the blocks around on the table can be derived and executed by the robot. Of course, the identification problem is simplified if the robot already has a stored 'situation model' for the various blocks on the table. This information corresponds to the known amino-acid sequence for the protein analysis problem; other correspondences between the two task domains are given in Table 1.

An attractive feature of the protein-crystal world problem is that it lacks the non-linear perspective-projection transformation that is particularly troublesome in inferring three-dimensional information from two-dimensional data. The interpretation of density maps is discussed further in a later section.

Symbolic processing

Artificial intelligence research is that part of computer

science that is concerned with the symbol-manipulation processes that produce intelligent action. By 'intelligent action' is meant an act or decision that is goal-oriented, arrived at by an understandable chain of symbolic analysis and reasoning steps, and is one in which knowledge of the world informs and guides the reasoning. Artificial intelligence research encompasses the following topics.* (1) Symbolic problem solving methodology using heuristic search techniques. (2) Natural language (*e.g.* English) understanding programs. (3) Automatic programming (*i.e.* program-writing programs). (4) Theorem-proving programs. (5) Information processing models of human learning, memory and problem-solving behavior. (6) Visual scene interpretation. (7) Speech-understanding programs. (8) Integrated robotics systems combining manipulation, locomotion, vision and problem solving. (9) Applications programs in chemistry, medicine, mathematics, management science and defence systems. (10) Development of advanced computer programming languages for symbolic processing.

Each of the above topics could also be associated with some other research field such as control theory, cognitive psychology, *etc.*, but the unifying computer science overtone is the symbol manipulation processing needed to accomplish these diverse tasks. The concept that the computer is a symbol processing device rather than purely a number processing device is fundamental to AI research, as it is to most of computer science. Strong in motivating AI research is the hypothesis that activities associated with human cognition can be described in terms of symbolic computation, and therefore that computers can perform such activities.

Symbolic computation is simply a generalization of numerical computation. In numerical computation, the primitive symbols are interpreted as numbers and the operations of addition, multiplication, *etc.*, are accomplished by special hardware devices or software procedures. The most familiar symbolic processors are the compiler programs which translate one set of symbols in a language such as Fortran or Algol to another set of symbols in a language which is closer to the machine instruction level. Much of the computer time, even in a scientific computer center, is spent doing the symbolic manipulation of program compilation. Symbolic manipulation is even more common in business data processing. Airline reservation bookkeeping, for example, is almost exclusively a symbol processing task.

In symbolic calculation, as in numerical calculation, the data structures are of fundamental importance. Matrices of coefficients provide convenient data structures for numerical calculations and the concepts of matrix algebra (Householder, 1964) provide a powerful abstraction for thinking about matrix manipulations. In symbolic processing, the data structures are often generalized networks of symbols which may be thought of as node-link graphs (Harary, 1969) with properties associated with the nodes and interactions associated with the links. Sometimes the problem domain is well structured and can be defined in terms of an abstract algebra such as permutation group theory which has a well defined set of primitive operations for manipulating the symbolic data structures (Leech, 1970). In general problem domains, the primitive operations are less clearly defined and the manipulation of the symbolic data structures are usually

* A handbook of artificial intelligence techniques covering these topics is currently being compiled at Stanford University. Preliminary versions of this handbook will be available to interested scientists *via* the TYMNET computer network in the fall of 1977. Publication in book form is expected in 1978.

Table 1. *Comparison of the blocks and protein-crystal worlds*

Characteristic	Blocks world	Protein-crystal world
Physical object	Polyhedral blocks on table	Protein molecules in crystal
Visual domain	Video picture of illuminated object with shadows	Electron density map of protein unit cell
Data geometry	Two-dimensional perspective projection of three-dimensional opaque polyhedron	Three-dimensional sublattice sampling with cyclic boundary conditions
Discrete data	Two-dimensional arrays of reflectivity and color	Three-dimensional array of electron density
Variables affecting data	Surface reflectivity	Atomic species
	Surface orientation	Thermal motion
	Illumination geometry	Data resolution
	Perspective geometry	Amplitude errors
	Video and digitizing errors	Phase errors
Prominent features	Reflectivity discontinuity at edges	Density maxima near atomic sites
Structural constraints	Polyhedral geometry of edges, vertices, faces and shadows in perspective projections	Stereochemical geometry of polypeptide molecules
Situation knowledge	(*a*) Table dimensions	(*a*) Unit-cell dimensions
	(*b*) Number of blocks	(*b*) Number of monomer units
	(*c*) Size and shape of polyhedra present	(*c*) Type of amino acids and cofactors present
	(*d*) Block stacking sequence	(*d*) Amino-acid sequence
	(*e*) Likely local configurations	(*e*) Helix and pleated sheet stereotypes
	(*f*) Likely global configurations (*e.g.* a block city)	(*f*) Protein family resemblance (*e.g.* hemoglobin fold)
Model resulting from analysis	Geometrical characterization and absolute position for each block	Atomic coordinates for all atoms in protein molecule
Use of model	Planning and execution of robotic manipulations	Examining stereochemical principles

tailored to fit the problem at hand. Heuristic search techniques (Nilsson, 1971) are required quite often in symbolic problem-solving programs.

The what-to-how spectrum

The spectrum of computer applications may be characterized, in a necessarily oversimplified way, along one dimension: the *what-to-how* dimension. *What* the user wishes to have the computer do for him is at one end of the spectrum. Precisely *how* the computer is to do it, step-by-step, is at the other end. In the early days of programming, the 'how' was accomplished with plugboard wiring and machine language coding. The next step involved algebraic formula translation compilers such as Fortran and Algol, and symbol manipulation interpreters and compilers such as Lisp (McCarthy, 1960). There is currently a great deal of activity concerning very high-level languages with vocabularies much closer to natural languages such as English. Moving to the next, higher level, *i.e.* implementing a translation of the ultimate 'what' that is in the mind of the user, is beyond the current research frontier.

In parallel with the above 'engineering' tools designed to implement a particular computational task, there are computer science subdisciplines which attempt to identify and understand the fundamental principles at each level. An approximate correspondence is shown in Table 2. The domain of AI research has moved over the years from the lower to the higher levels shown in the table. At the extreme 'what' level there are some aspirations to build a theory of intelligent information processing with the view that symbolic computing concepts provide a modeling domain for studying those processes which we call intelligent behavior. From a philosophical viewpoint, this aspect of AI might be considered a methodological and conceptual framework designed for the scientific study of thought.

Table 2. *The what-to-how spectrum in computer science*

Program engineering	Computer science
What is in the mind of the user	
An idea for a calculation	Cognitive psychology, theory of computing
Very high-level languages	Automatic programming
Algol, Fortran, Lisp	Theory of formal languages
Machine language, microcode, plugboard wiring	Theory of automata Theory of switching
How the machine does it	

Knowledge-based systems

In a general sense, the day to day working judgements which a knowledgeable practitioner uses to implement his expertise in a field are the empirical rules which form the foundation for the discipline. These rules often are not set down explicitly in textbooks because they are basically common-sense rules and do not convey the elegance one desires in describing his field to someone else. Such rules normally are passed along by the apprenticeship aspect of the formal education process.

Some of the more successful AI applications programs are the 'knowledge based' programs where the codified empirical rules of some very specialized areas are used to drive the program. In such programs a serious effort is made to keep the knowledge base (*i.e.* the heuristic rules) separate from the code which implements the rules. One of the techniques is to enter the knowledge in the form of 'production rules'.

A program based on production rules is called a production system. A production rule contains a PREMISE, consisting of a series of conditions, and an ACTION, which is the conclusion reached if the conditional statements are true. An example of a production system is the *MYCIN* system (Shortliffe *et al.*, 1975). *MYCIN* is an interactive computer program which uses the clinical decision criteria of experts to advise physicians regarding selection of appropriate antimicrobial therapy for hospital patients with bacterial infections. *MYCIN*'s problem-solving strategy is goal-directed, or 'backward chaining'. That is, the first rule to be evaluated is the one which concludes the identity of the infecting agent(s) and calls for appropriate therapy. The rule *MONITOR*, shown in Fig. 1, tries to evaluate each of the premises in this rule by either (1) asking the user directly for information, or (2) evaluating other rules whose conclusions contain the desired information (the *FINDOUT* mechanism shown in Fig. 2). A record of the consultation is kept which the program can reference to explain its chain of reasoning to the user. The *MYCIN* system currently has about 400 production rules and a typical rule is

IF: (1) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
(2) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
(3) THE AEROBICITY OF THE ORGANISM IS ANAEROBIC
THEN: THERE IS SUGGESTIVE EVIDENCE (.6) THAT THE IDENTITY OF THE ORGANISM IS BACTEROIDES.

A number of knowledge-based AI programs have been written as production systems and typically require from 40 to 400 rules for the knowledge base. Production systems are only one of a multitude of different methods used for writing AI programs; however, a surprisingly large fraction of them are coded in the Lisp language (Maurer, 1972) because of its versatility in manipulating general data structures.

Empirical induction in mass–spectral analysis

One of the early major efforts to apply AI principles and techniques to a scientific domain was the heuristic

DENDRAL project (Smith *et al.*, 1972; Smith, Masinter & Sridharan, 1974; Michie & Buchanan, 1974) for analyzing mass-spectral and NMR data. The heuristic *DENDRAL* system of programs is based on the paradigm of hypothesize and test. At the core of the system is a generator capable of producing all possible structural isomers for a given empirical formula, in practice, however, constrained by chemical knowledge. Coupled to the generator is a testing program which can predict the significant peaks in a mass fragmentation table for a given chemical structure, based on a large collection of production rules for mass spectroscopy. Thus a simulated mass fragmentation table is constructed by the predictor for each hypothesis passed to it by the generator, and the hypotheses are ranked by their degree of conformance with the experimental data. For any practical problem the generator must be constrained to produce a plausible subset of the complete list of isomers. Here we encounter a critical issue: how does one represent knowledge of the task domain so that it can be utilized by a solution-searching program? In the heuristic *DENDRAL* system the trial structures and substructures are uniformly represented as atom-bond graphs, with the atoms corresponding to the nodes and the bonds to the edges. All knowledge for constraining the generator is then expressed as lists of required or disallowed subgraphs. The sources of these constraints, whether from general principles of chemistry, specific mass-spectral rules for the class of compounds under investigation, or auxiliary NMR experiments, is immaterial, as long as these constraints are expressed in the language of atom-bond graphs. These structural constraints are contained in a planning program, which is the first phase of the *DENDRAL* system.

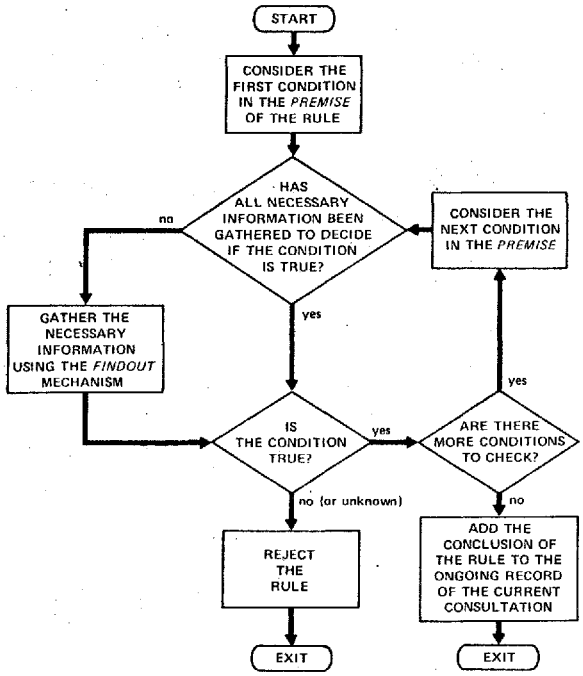

Fig. 1. Flow chart describing the rule *MONITOR* which analyzes a rule and decides whether it applies in the situation under consideration. Each condition in the PREMISE of the rule references some parameter, and all such conditions must be true for the rule to be accepted.

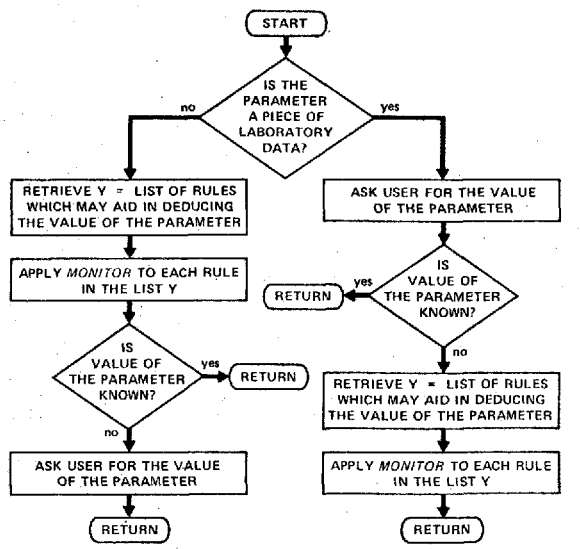

Fig. 2. Flow chart describing the *FINDOUT* strategy for determining which questions to ask the user. The derivation of values of parameters may require recursive calls to the *MONITOR*, thus dynamically creating a reasoning chain.

The programs have been applied to a large number of compounds, taken one family at a time. Its level of performance on carefully selected mass-spectrometry problems is about that of a second or third year graduate student in analytical chemistry. In a few cases the program's behavior is truly exceptional. Rules for constraining the generator (*i.e.* the planning rules) as well as rules for predicting mass spectra, have been elicited from scientists knowledgeable in the field and incorporated in the program on a family by family basis. This procedure for knowledge acquisition is the pace-setting factor in *DENDRAL*'s further development.

Rule acquisition

The performance of a production system rests on the adequacy of its rule set, but the formulation of the rules is often a very time-consuming process since most practitioners in a field do not remember their tricks of the trade as explicit production rules. Some current work in AI is exploring the possibility of generating rules more automatically. An example is the *META-DENDRAL* program (Buchanan *et al.*, 1976) which attempts to discover new inference rules for the *DENDRAL* program discussed above. A key aspect of the *DENDRAL* knowledge base concerns the rules for molecular fragmentation in an electron beam. Since the empirical fragmentation rules are different for each class of chemical compounds, the number of rules is potentially very large. The *META-DENDRAL*

program examines a collection of mass spectra from a series of related compounds for the purpose of noticing and interpreting regularities. Then, heuristic search methods are used to examine possible explanations of the regularities and to generate rules. Finally, the rule set generated is evaluated in the context of other rules and a new self-consistent set of rules is derived.

Chemical synthesis

Another application of AI techniques has been in the development of programs which generate plausible pathways for molecular synthesis, starting from a target structure, a list of available reagents, and a large set of rules for transforming chemical sub-graphs (Blair, Gasteiger, Gillespie, Gillespie & Ugi, 1974; Corey, 1971; Gelernter, Sridharan, Mart & Yen, 1975; Wipke, 1974). These programs begin with the target structure and work backwards, using the target as the goal of a search tree. Each level of the search tree contains a set of compounds which can form the compound at the parent node on the next, higher level by known transformations. A successful synthesis results when the tree has been expanded along at least one path all the way back to known or potentially available reagents.

Protein density map interpretation

The scene analysis example and the comparison of the blocks world and protein crystal world domains was given as an example application of AI to a crystallographic problem. The problem of designing a symbolic reasoning program to interpret protein density maps is currently the topic of a collaborative research project between the Computer Science Department at Stanford University, the Chemistry Department of the University of California at San Diego, and the Chemistry Division of the Oak Ridge National Laboratory.

The system under development attempts to integrate knowledge from three different domains: chemical topology (atomic connectivity), geometrical microstructure (interatomic distances and angles) and geometrical macrostructure (secondary conformation). Corresponding to each domain is a knowledge base encoded as facts, procedures and rules which define the conditions for their applicability. The chemical topology knowledge base is task specified and contains the chemical connectivities of the amino-acid sequence, if known. Connectivities of cofactors and coordination bonding of metals to prosthetic groups are also included. The microstructure knowledge base contains stereochemical facts about the geometry of amino acids and small peptides, hydrogen bonding properties, helix formation propensity, *etc.* The macrostructure knowledge base contains templates for frequently encountered secondary conformations such as alpha helices and pleated sheets. The knowledge bases and control structure are being coded in the Lisp language.

The electron-density map manipulation is being programmed in Fortran by C. K. Johnson. A series of progressively more detailed analytical descriptions of the density functions are planned. The first-order model (Johnson & Grosse, 1976) involves an analysis by a minimum spanning tree calculation (Harary, 1969) of a graph based on the critical points (*i.e.* local maxima, minima and saddle points) in the density map. A formal mathematical theory for critical point analysis is given by Morse & Cairns (1969). Segments of the spanning tree are then checked against steric templates for suspected stereotype configurations such as heavy-atom clusters, alpha helices and pleated sheets.

The chemical model representation for the amino-acid sequence is also encoded as a graph with several levels of edges, corresponding to covalent bonds, coordination bonds, hydrogen bonds and van der Waals contacts. The chemical and density models are manipulated by the knowledge sources until the two representations can be mapped onto one another. The knowledge sources of chemical and stereochemical constraints are coded as production rules, and operate on a hierarchy of representations, the most detailed of which is the atomic positions for the molecular model of the protein.

Summary

A brief overview is given of some applied artificial intelligence research projects in scene analysis, medical consultation, mass-spectral analysis, knowledge acquisition, chemical synthesis and density-map interpretation. These examples are given to illustrate our thesis that symbolic computing techniques are indeed useful in scientific computing applications. We suggest that crystallographic programmers would do well to examine these techniques and make them an integral part of the tools of their trade.

We gratefully acknowledge the informative discussions with B. G. Buchanan and H. P. Nii at Stanford and S. T. Freer of the University of California at San Diego.

References

BLAIR, J., GASTEIGER, J., GILLESPIE, C., GILLESPIE, P. D. & UGI, I. (1974). *Tetrahedron*, **30**, 1845–1859.

BUCHANAN, B. G., SMITH, D. H., WHITE, W. C., GRITTER, R., FEIGENBAUM, E. A., LEDERBERG, J. & DJERASSI, C. (1976). *J. Amer. Chem. Soc.* In the press.

COREY, E. J. (1971). *Quart. Rev.* **25**, 455–482.

GELERNTER, H., SRIDHARAN, N. S., HART, A. J. & YEN, S. C. (1975). *Top. Curr. Chem.* **41**, 113–150.

GREER, J. (1974). *J. Mol. Biol.* **82**, 279–301.

HARARY, F. (1969). *Graph Theory*. New York: Addison–Wesley.

HOUSEHOLDER, A. S. (1964). *The Theory of Matrices in Numerical Analysis*. New York: Blaisdell.

JOHNSON, C. K. & GROSSE, E. (1976). *Amer. Cryst. Assoc. Program Abs., Series* 2, **4**, 480.

LEECH, J. (1970). Editor *Computational Problems in Abstract Algebra*. Oxford: Pergamon.

McCarthy, J. (1960). *Commun. Assoc. Comput. Mach.* **3**, 184–195.

Maurer, W. D. (1972). *The Programmer's Introduction to Lisp*. New York: Elsevier.

Michie, D. & Buchanan, B. G. (1974). In *Computers for Spectroscopy*, edited by R. A. C. Carrington. London: Hilger.

Morse, M. & Cairns, S. S. (1969). *Critical Point Theory in Global Analysis and Differential Topology*. New York: Academic Press.

Newell, A. & Simon, H. A. (1976). *Commun. Assoc. Comput. Mach.* **19**, 113–126.

Nilsson, N. (1971). *Problem Solving Methods in Artificial Intelligence*. New York: McGraw Hill.

Pepinsky, R., Van den Hende, J. H. & Vand, V. (1961). *Conference on Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, J. M. Robertson & J. C. Speakman, pp. 154–160. New York: Pergamon.

Shortliffe, E. H., Davis, R., Axline, S. G., Buchanan, B. G., Green, C. C. & Cohen, S. N. (1975). *Comput. Biomed. Res.* **8**, 303–320.

Smith, D. H., Buchanan, B. G., Engelmore, R. S., Duffield, A. M., Yeo, A., Feigenbaum, E. A., Lederberg, J. & Djerrassi, C. (1972). *J. Amer. Chem. Soc.* **94**, 5962–5971.

Smith, D. H., Masinter, L. M. & Sridharan, N. S. (1974). *Proceedings of the NATO/CNA ASI on Computer Representation and Manipulation of Chemical Information*, edited by W. T. Wipke, S. Heller, R. Feldmann & E. Hyde. New York: John Wiley.

Winston, P. H. (1972). *Machine Intelligence*, edited by B. Meltzer & D. Michie, pp. 431–463. New York: John Wiley.

Wipke, W. T. (1974). *Proceedings of the NATO/CNA ASI on Computer Representation and Manipulation of Chemical Information*, edited by W. T. Wipke, S. Heller, R. Feldmann & E. Hyde. New York: John Wiley.