The Evaluation and Reporting of Discovery Systems

Herbert A. Simon
Department of Psychology and
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 - USA

Raúl E. Valdés-Pérez
School of Computer Science and
Center for Light Microscope Imaging and Biotechnology
Carnegie Mellon University
Pittsburgh, PA 15213 - USA

1 Introduction

For several decades there has been considerable interest within artificial intelligence and cognitive science in building computer simulations of the processes of scientific discovery, in the spirit of constructing expert systems (automated discovery), or directed toward understanding how human scientists make discoveries. Here, we will consider both lines of inquiry, but with emphasis on the AI aspects.

Our central concern will be with specificity and generality. How do we compare the merits of discovery programs that are efficacious in one domain with those that have generality over several or many? How can we develop a common language to describe systems for different domains, thereby carrying the experience gained in systems into explorations of others?

Most of what we have to say is not peculiar to discovery systems, but applies to all kinds of expert systems. Hence, we will not entirely limit our discussion to discovery tasks. We will focus on computer programs as research vehicles, without implying that all worthwhile research on the subject should result in a computer program.

2 Generality vs Specificity

AI history includes an early period of search for general intelligent mechanisms (e.g., generate-and-test, means-ends analysis) followed by a period in which specialized domain knowledge was viewed as the main source of intelligence.

The General Problem Solver (GPS) neither claimed nor possessed a capability for solving problems in all domains [Newell and Simon, 1963]. GPS was called "general" because its basic processes made no reference to the domain to which it was applied. It was applied to a specific domain by specifying (1) a problem space (space of states of the domain, together with operators for moving about in this space); and (2) a "Table of Connections," mapping differences between problem states onto operators that might remove those differences.

The basic ideas in these early problem solving systems were not abandoned in the subsequent era of knowledge-rich, specialized systems. The general "weak methods" became engines for drawing inferences from the knowledge embedded in knowledge-rich systems.

A typical symbolic expert system consisted of a large knowledge base, usually a set of productions, combined with a modest inference apparatus of this sort. New information about a specific situation activated some of the productions in the knowledge base, thereby accessing the associated knowledge. The inference processes incorporated in the system might make further transformations of that knowledge.

In a medical diagnosis system, for example, the new information about symptoms and test results would fire productions accessing knowledge about diseases sometimes associated with them [Davis et al., 1977]. The inference processes would combine information derived from different symptoms into a differential diagnosis.

The expert system shells that soon sprang up attempted to capture the domain-independent components of these systems: specifically, the inferential components and the heuristics for the control of search (for example, [Melle, 1980]). Like GPS, they tried to capture what was common to problem solving in a wide range of domains and to represent it in domain-neutral language.

In the face of this history and experience, one might suppose that further research on domain-rich intelligent systems is of interest only from the standpoint of applications. Building such a system, in this view, consists in assembling the requisite body of domain knowledge in appropriate form. Bringing this knowledge together might well contribute to the compilation and systematization of theory in the application domain, but might not contribute to our understanding of intelligence.

However, there is a flaw in this argument, for it assumes that there are only two kinds of knowledge: general and domain-specific. In point of fact, generality is a matter of degree. For example, the infinitesimal calculus is a rather general, and not too weak, method for solving problems over a wide range of the sciences. Physicists, economists, chemists, demographers, and geneticists all find important uses for the calculus. On the other hand, specialists in literary criticism, anthropology, field geology, and history seldom do. Is the calculus a general or a domain-dependent method? Clearly it is general, but to a degree.

Until we have studied the structures and components of expert systems in a sufficient variety of domains, we will not be sure that we have identified the full range of general components of intelligence, each important for some set of problem domains. An important research task of AI is to build up a good taxonomy of such components for incorporation in domain-specific problem solving systems.

Much more is required than simply building a taxonomy. There are also the important topics of problem representation and learning. Our theory of discovery ought not only elucidate programs that can make discoveries; it should also show how such programs come into existence. Starting with a problem, presented by the external world or in written form, how does an intelligent system ingest the problem description and represent it in a way that makes it accessible to a problem-solver?

Are there general or even universal learning mechanisms (like the chunking of SOAR), or are learning and representation domain specific? To the degree that generality can be found in learning and representation, these become another target of research.

3 Other Sciences of Design

Several other sciences that are also concerned with design have a longer history than machine discovery or even AI: synthetic chemistry, aeronautics, and computer architecture for example. Since computational scientific discovery is itself a domain of science, we can ask what lessons science offers for research on discovery. We will examine briefly the case of organic chemical synthesis.

In their book The Logic of Chemical Synthesis, [Corey and Cheng, 1989] trace the history of their subject. In the 19th century many noteworthy syntheses were developed by exploratory studies of reactions, especially those that began with distillation products of common coal. Deliberate syntheses - posed by the practical need for a certain chemical product - were designed by analogy with the successful reactions contributed by the exploratory studies. More systematic understanding derived from knowledge about what substituents of molecules tended to react most readily. This state of affairs corresponds to AI and machine discovery today, with discovery programs analogous to chemical syntheses.

Corey and Cheng discuss the growing sophistication of organic synthesis, which nevertheless until the 1970's depended almost entirely on case-by-case analyses using specialized techniques. However, by the mid-1960's a more general and systematic method emerged, based on the AI concept of heuristic search [Corey et al., 1985, page 416]. This method involved searching recursively back from the target molecule by attending to structural features that suggested appropriate reactions - operators in AI terms. The analogue in the domain of scientific discovery would be a systematic method to work backwards from a given task to a computer program that automates it, based on recognizing task or subtask features that cue sub-methods to apply.

This analogy suggests that a promising route for research on scientific discovery is to enlarge the space of possible discovery systems by exploratory efforts in new domains. Continuing the analogy, "deliberate syntheses" would likewise serve to assess the state of knowledge by requiring work on problems that are posed beforehand. From such research there could emerge powerful concepts for organizing knowledge about machine discovery, and general, systematic techniques for automating given scientific tasks.

What does this analogy imply for evaluation of discovery systems as research vehicles? Exploratory studies on discovery in specific domains are valuable just as studies of specific reactions are valuable for the science of organic synthesis. Further, their research value is heightened to the extent that they perform on real scientific tasks. Quite apart from its research value, efficacy on real scientific tasks will also produce artifacts of significant worth to outsiders, a desirable goal for any successful and enduring science.

3.1 Validation by domain scientists

There is wide interest in computational methods among domain scientists, as shown by the number of science journals with 'comput-' in their titles. A sound way to validate the efficacy of a discovery system is to submit a report to a journal of the domain science. The DENDRAL project set a high standard for this practice [Lindsay et al., 1980].

Of course, domain scientists are not waiting idly for spinoffs from research on discovery. Scientists, especially chemists, have written quite sophisticated programs for carrying out creative scientific work, often building on ideas imported from AI. A reasonable standard for discovery programs written by AI workers should be that they exceed such work along some dimension of novelty, efficacy, or generality.

4 Languages for the Description of Systems

The generality inherent in components of discovery systems, and hence the interest of those components for AI research, is not easy to detect or exploit if each system, in each problem domain, is developed as an independent entity, with its own programming languages, conventions, and languages of description. Such a pattern of research guarantees that the wheel will be reinvented many times. Knowledge of discovery gained in one problem domain will not be available to other domains.

The problem is mitigated by the use in AI programming of a relatively small number of languages. For many years, LISP served as the common language of AI, and even today, a few languages - LISP, OPS5, PROLOG, C - dominate the field. It is not unreasonable to expect scientists in AI to have command of all of them.

However, even when the programming languages are well understood, the structures incorporated in discovery programs are not always apparent to the naked eye. Moreover, because such programs are typically very large, publications describing them do not usually provide the full code. Programs like AM [Lenat, 1982], BACON [Langley et al., 1987], DENDRAL [Lindsay et al., 1980] and MECHEM [Valdes-Perez, 1992] incorporate many heuristics whose exact nature is not always evident from summary descriptions. We need a standard vocabulary for describing them at a higher level than program code, in order to determine and exploit the generality of its heuristics.

There do, of course, exist rather high-level languages for discussing heuristic search. They can be found in such textbooks as [Nilsson, 1980] and [Pearl, 1984], or in the Handbook of Artificial Intelligence. But the technical vocabulary for such discussion is still very incomplete, and there is still little agreement as to how to communicate the structure and content of large discovery programs without simply printing the code.

There is also little discussion of the scope of generality of the heuristics and other system components, and insufficient attention to processes which, while not fully general, might have application to a considerable range of disciplines. In particular, we need research to determine what are the characteristics of a representation or heuristic that makes it more or less general in its applicability, and how to match particular representations and heuristics against potential domains of application.

References

[Corey and Cheng, 1989] Corey, E. and Cheng, X.-M. (1989). *The Logic of Chemical Synthesis.* John Wiley & Sons.

[Corey et al., 1985] Corey, E., Long, A., and Rubenstein, S. (1985). Computer-assisted analysis in organic synthesis. *Science*, 228:408–418.

[Davis et al., 1977] Davis, R., Buchanan, B., and Shortliffe, E. (1977). Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence*, 8(1):15–45.

[Langley et al., 1987] Langley, P., Simon, H., Bradshaw, G., and Zytkow, J. (1987). *Scientific Discovery: Computational Explorations of the Creative Processes.* MIT Press, Cambridge, Mass.

[Lenat, 1982] Lenat, D. B. (1982). AM: discovery in mathematics as heuristic search. In Davis, R. and Lenat, D. B., editors, *Knowledge-based systems in artificial intelligence.* McGraw Hill, New York, N.Y.

[Lindsay et al., 1980] Lindsay, R., Buchanan, B., Feigenbaum, E., and Lederberg, J. (1980). *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project.* McGraw Hill, New York.

[Melle, 1980] Melle, W. V. (1980). *A Domain-Independent System That Aids in Constructing Knowledge-Based Consultation Programs.* PhD thesis, Stanford University. (Heuristic Programming Project, HPP-80-22).

[Newell and Simon, 1963] Newell, A. and Simon, H. A. (1963). GPS, a program that simulates human thought. In Feigenbaum, E. and Feldman, J., editors, *Computers and Thought*, pages 279–293. McGraw Hill, New York, N.Y.

[Nilsson, 1980] Nilsson, N. J. (1980). *Principles of Artificial Intelligence.* Tioga, Palo Alto, Calif.

[Pearl, 1984] Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving.* Addison-Wesley, Reading, Mass.

[Valdes-Perez, 1992] Valdes-Perez, R. E. (1992). Theory-driven discovery of reaction pathways in the MECHEM system. In *Proceedings of 10th National Conference on Artificial Intelligence*, pages 63–69.