CSC: Classic Paper Review/Analysis #2

Title and Author

Title: Computer Science as Empirical Inquiry: Symbols and Search

Author: Allen Newell and Herbert A. Simon

Summary/Hook

This lecture by Allen Newell and Herbert A. Simon was presented at the 1975 ACM Turing Award and is well known throughout the fields of cognitive science, artificial intelligence, and computer science. The lecture begins with Newell and Simon explaining their Physical Symbol Systems Hypothesis (PSSH), articulating the idea that a digital computer can have sufficient means for intelligent action. After explaining the requirements and different aspects of their hypothesis and the common objections that others have used to criticize their hypothesis, Newell and Simon spend the remainder of the lecture discussing heuristic. The heuristic search hypothesis helps in explaining problem-solving in humans and symbol systems alike and suggests a general explanation for human intelligence. After reading through this lecture, the reader will have a better understanding of human cognition and also other various phenomena found in cognitive and computer science.

Knowledge Relating to the Cognitive Science Program Learning Outcomes

1. **Formal Systems and Theories of Computation**

A Turing machine consists of two memories: an unbounded tape and a finite state control. The tape holds data, i.e. the famous zeroes and ones. The machine has a very small set of proper operations--read, write, and scan operations--on the tape. The read operation is not a data operation, but provides conditional branching to a control state as a function of the data under the read head. As we all know, this model contains the essentials of all computers, in terms of what they can do, though other computers with different memories and operations might carry out the same computations with different requirements of space and time. In particular, the model of a Turing machine contains within it the notions both of what cannot be computed and of universal machines--computers that can do anything that can be done by any machine

2. **Symbol Systems**

A physical symbol system consists of a set of entities, called symbols, which are physical patterns that can occur as components of another type of entity called an expression (or symbol structure). Thus, a symbol structure is composed of a number of instances (or tokens) of symbols related in some physical way (such as one token being next to another). At any instant of time the system will contain a collection of these symbol structures. Besides these structures, the system also contains a collection of processes that operate on expressions to produce other expressions: processes of creation, modification, reproduction and destruction. A physical symbol system is a machine that produces through time an evolving collection of symbol structures. Such a system exists in a world of objects wider than just these symbolic expressions themselves.

3. Algorithms and Automata

The contents of the data structures were now symbols, in the sense of our physical symbol system: patterns that designated, that had referents. Lists held addresses which permitted access to other lists--thus the notion of list structures. That this was a new view was demonstrated to us many times in the early days of list processing when colleagues would ask where the data were--that is, which list finally held the collections of bits that were the content of the system. They found it strange that there were no such bits, there were only symbols that designated yet other symbol structures. List processing is simultaneously three things in the development of computer science. (1) It is the creation of a genuine dynamic memory structure in a machine that had heretofore been perceived as having fixed structure. It added to our ensemble of operations those that built and modified structure in addition to those that replaced and changed content. (2) It was an early demonstration of the basic abstraction that a computer consists of a set of data types and a set of operations proper to these data types, so that a computational system should employ whatever data types are appropriate to the application, independent of the underlying machine. (3) List processing produced a model of designation, thus defining symbol manipulation in the sense in which we use this concept in computer science today.

4. Consciousness and Controversies

With the development of the second generation of electronic machines in the mid-forties (after the Eniac) came the stored program concept. This was rightfully hailed as a milestone, both conceptually and practically. Programs now can be data, and can be operated on as data. This capability is, of course, already implicit in the model of Turing: the descriptions are on the very same tape as the data. Yet the idea was realized only when machines acquired enough memory to make it practicable to locate actual programs in some internal place. After all, the Eniac had only twenty registers. The stored program concept embodies the second half of the interpretation principle, the part that says that the system's own data can be interpreted. But it does not yet contain the notion of designation--of the physical relation that underlies meaning

5. Language and Culture

A second active possibility for raising intelligence is to supply the symbol system with a rich body of semantic information about the task domain it is dealing with. For example, empirical research on the skill of chess masters shows that a major source of the master's skill is stored information that enables him to recognize a large number of specific features and patterns of features on a chess board, and information that uses this recognition to propose actions appropriate to the features recognized. This general idea has, of course, been incorporated in chess programs almost from the beginning. What is new is the realization of the number of such patterns and associated information that may have to be stored for master-level play: something of the order of 50,000. The possibility of substituting recognition for search arises because a particular, and especially a rare, pattern can contain an enormous amount of information, provided that it is closely linked to the structure of the problem space. When that structure is "irregular," and not subject to simple mathematical description, then knowledge of a large number of relevant patterns may be the key to intelligent behavior. Whether this is so in any particular task domain is a question more easily settled by empirical investigation than by theory. Our experience with symbol systems richly endowed with semantic information and pattern-recognizing capabilities for accessing it is still extremely limited.