CONCEPT FORMATION THROUGH THE INTERACTION OF MULTIPLE MODELS

Mark H. Burstein
Yale University

INTRODUCTION

This paper is about some of the processes involved in learning the fundamental concepts of a new domain. In particular, I have been studying how people learn some of the basic concepts used in computer programming, given little or no prior experience with the use of computers, or computer languages. The goal of the research is to build a computer model capable of learning such basic concepts.

The problem is that most current theories are either too weak or combinatorially explosive. Generalization techniques [4,6] do not construct radically new conceptual structures. Concept clustering techniques [2] require correlations over large sets of data. People learning what a computer variable is must learn something radically new to them, but with very limited experience. How can they do this?

The theory I am developing is based on the study of transcripts of 1-on-1 tutorial sessions with several subjects learning the programming language BASIC, collected over a period of several months. The subjects lacked any previous experience with computers, but even the youngest, Perry, age 10, had a number of preconceptions about what was to be learned, and a wealth of knowledge and experience in other domains which he used to provide correlates for experiences and observations in the new domain.

Learning to program a computer is a rather "formal" subject, requiring the development of expertise in symbolic manipulation. Thus one might readily expect prior knowledge of mathematics to be useful in learning to program. Indeed, our subjects often drew parallels to concepts in mathematics when learning programming concepts. Perhaps less expectedly, however, our subjects also invoked a great deal of common sense world knowledge when learning to program.

This common sense knowledge included such diverse areas as putting objects in boxes, moving pieces around on a game board, and writing on blackboards.

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research under contract N00014-75-C-1111, and in part by the National Science Foundation under contract IST7918463.

It also included knowledge of English, and of how agents perform tasks requested of them. In this short paper, I hope to at least give some idea of how these other areas of expertise were used to form some very basic concepts of computer programming.

HOW ARE CONCEPTS IMPORTED?

One of the techniques a teacher can use to explain an unfamiliar concept is to use analogies or metaphors [5]. For example, a common way of introducing the concept of a computer variable is to make an analogy to boxes and their contents. One introductory BASIC manual phrased it this way:

> (1) To illustrate the concept of variable, imagine that there are 26 little boxes inside the computer. Each box can contain one number at any one time. [1]

Another text referred only to "locations" inside the computer, but illustrated the concept with diagrams showing contiguous rectangles containing numbers. Both of these descriptions suggest that essentially the same set of relations and operations that exist between containers and their physical contents can be applied to variables and their "contents".

However, this is not the only way to describe variables. Another common description is based on similarities to concepts in elementary algebra.

> (2) In ordinary algebra, letters of the alphabet are ofter used for terms that can take on varying numerical values. BASIC has a very nice, built-in algebraic feature that allows the user to assign numeric values to any letter of the alphabet... The user can set the value of A by typing A=10. [3]

While each of these models suggests a number of things that are true for variables, neither of them tells the whole truth and nothing but the truth. Although each analogical notion of variable presented suggests enough of the truth to act as an initial, working model of the functioning of variables in computer programs, they also suggest hypotheses which are either irrelevant or wrong, or both.

Since neither model is sufficient to give the learner an understanding of how to use variables, or how they will work in all situations, each model must be "debugged". [9] That is, having seen the metaphor, the student is presented with simple problems to be solved or new situations to be understood in the new domain, which he attempts to solve using hypotheses suggested by the metaphor. This leads to some successes, and a number of failures. Interestingly enough, the failures provide the most useful information about how the developing concept must be modified so that it agrees with first-hand experience. [7,8] The causes of the failure must be fixed must be fixed to produce useful expectations in the future.

USING MULTIPLE MODELS

One kind of modification that can be made to an inaccurate model is to add restrictions to prevent specific erroneous inferences. For example, there are no inherent restrictions on the number or kind of objects that can go in boxes. Variables, on the other hand, can hold only one thing at a time. Hence, a statement of that restriction must be added by the teacher, as in (1) above, or by the student after an error occurs.

A more important kind of modification of the developing concept is to introduce additional models or analogies. These additional analogies can be used to explain and/or to replace faulty inferences made by earlier versions of the new concept. The new analogies also allow additional inferences and expectations to be made.

For example, a second extremely common analogy for computer variables and computer memory is human memory. The following is an example of such a statement:

> (3) The computer "remembers" any values assigned to numerical variables. Of course, the values of the variables can be changed at any time -- simply do the variable = value command again. [3]

This metaphor for computer memory comes from the same text that contained the algebraic description of variables, (2), above. As we will see shortly, such secondary models can and do affect the hypotheses which can be generated.

These examples illustrate the use of multiple models in describing a new concept. This is not unusual. Rather, it may be the norm. Furthermore, the students will add metaphors and models of their own discovery, based on chance word choices, observed consistencies, and inferences they made.

A model of learning must explain how these analogical descriptions can be "combined" when forming new concepts.

GENERATING ERRONEOUS HYPOTHESES

Here are some examples of the kinds of errors and corrections that occur when a student has several models for the same concept. In the following examples, M is myself, and P is the student, Perry. Perry was initially given a short paragraph to read which said that the computer's memory used variables which were like boxes in which one could store numbers.

Perry produced the following summary of that description:

AN INITIAL HYPOTHESIS

P: Yea. This, the variable, is the box, and you tell it to... It's where he puts his memory and everything... and then you ask him to put it in there. You say the word (variable), and then he puts it in.
M: Right, ok. So suppose I wanted to tell him to remember the number 5 in variable X...
P: Oh. What do you mean X? In any box that he wants?

This shows that for Perry, variables were related to (human) memory, and that they were somehow like boxes which you put things in. His last statement also shows that he already had the concept of an algebraic variable, since here he thought X stood for the unknown. This fact will be important in other examples.

Using the box metaphor, Perry tried to investigate directly the class of objects that can be placed in the computer's "boxes". Since physical objects can certainly hold other physical objects, he thought that the boxes in the compter might be able to hold other "boxes".

STORING BOXES IN BOXES
-- another (partially) wrong hypothesis.

P: Can he store a box in a box? Let's say you tell him X=7+4, and then you make him another box and tell him to store X, the box X into the other box.
M: You can't put boxes inside boxes, but you can put what's inside them in another box.

Another error made in trying to specify what can go into the boxes comes from the use of a second metaphor. As this was also supposed to be a model of the computer's memory, Perry often generated hypotheses consistent with his knowledge of human memories. Since people, when asked to solve simple arithmetic problems, will generally remember both the question and the answer, Perry assumed that the computer would as well.

STORING THE PROBLEM WITH THE ANSWER

M: Suppose I wanted to add up 7 and 4 and store it in X.
P: How? You want to put 7+4 or the answer?
M: I want to put the answer in.
P: You want him to tell me the answer and then put it in?
M: Well, just put it in.
P: You write... X equals 11. You can write X=7+4
M: You can do that too. Try it.
P: (types X=7+4)
M: Ok. Now what's inside the box called X?
P: 7 plus 4, ... 11. You want me to ask him?
M: Yea, why don't you ask him?
P: (types PRINT X) oh. 11. It doesn't matter. If you want him to only answer 7+4, and not the answer, can it do that?
M: No, it only stores one number in each box.
P: He thinks you only want the answer.
M: It's not that, it's just that what you put in the box was not "7+4", but 11. It adds it up and then puts the answer in the box
P: Oh. It doesn't put 7+4 in the box.

The addition of semantic cues to the names of variables caused Perry to assume that the computer was even more like a human agent, with human-like memory capabilities. Here, he wonders whether the computer has the ability to querry it's "frame" for a person in several different ways.

THE COMPUTER AS HUMAN AGENT

M: I can say PERRY=10. How old are you?
P: 10
M: Ok, and now if I PRINT PERRY.
P: It will say 10
M: Right.
P: If you tell him... Let's say Perry is 10 years old. Can you say that? And then you ask him "How old is Perry?" or "Who is 10 years old?", will he answer you?

Finally, there is the case in which several of Perry's models make conflicting predictions, and he is asked to choose which one is correct. In this example, the two models are the box metaphor (assignment is like putting something in a box) and presumed similarities to the algebraic notion of equality (A=10 means A is the same as 10), which Perry inferred from the use of the equal sign to denote assignment in BASIC. If assignment is like moving something into a box, then moving something from one box to another should remove it from the first box. However, if assignment is like algebraic equality, then the statement Q=P should mean that the two variables have the same value.

After I had typed P=10 and then Q=P, (causing both Q and P to have value 10), the following dialogue occurred:

GENERATING CONFLICTING HYPOTHESES

M: So, what's in P now?
P: Oh. Nothing.
M: Nothing?
P: 10! and then Q is also.
M: What do you think it is? Is it nothing or 10?
P: Let's find out. First let's see...
M: Well, what do you think it is?
P: If you have two boxes, and you moved...you moved or it equals to? You moved what's in P to Q so there's nothing in it, or did you only put the same number in Q that's in P? I think it's 10.
M: You think it's 10?
P: Because you don't say that, um, move P at all... take P out. You only said that Q equals the same as P. So if it equals, it has to be 10, because if there's no 10 in it, Q wouldn't equal to it.

Clearly, there is great indecision here about which model is going to prove correct. Only after reviewing both choices does he finally chooses the one which makes the correct prediction.

This example clearly shows Perry operating with at least two distinct models of the what a variable is. Each model is capable of making a number of predictions which are useful. Each makes predictions which must be corrected. Thus, corrections are possible not only when a preferred model produces an error, but whenever two models contradict each other. In both cases, more questions are raised to guide inference experiments in the "model debugging" process.

In this example, Perry was forced to reiterate the two models so that he could reevaluate his choice, and find the one that seemed most consistent with the situation and his beliefs. In this case, the answer 10 seemed preferable due to the use of the phrase "Q equals P".

My goal is to develop a computer model capable of learning concepts radically different from those it already has, without requiring huge amounts of exemplary data. The technique suggested by the protocols is to maintain a set of models that are all partially useful, but which occasionally contradict each other. Much of the learning arises from the correction of the failed expectations of these models. However, these failures can occur even without the student "making a mistake". Contradictory expectations generated by competing models may provide an important alternative source of errors for the debugging process.

REFERENCES

[1] Albrecht, R., Finkel, L, and Brown, J.R. (1978). BASIC for Home Computers. John Wiley & Sons, Inc., NY.

[2] Dietterich, T.G. and Michalski, R. S. (1979). Learning and generalization of characteristic descriptions. Proceedings of IJCAI-79, Tokyo, Japan.

[3] Heiserman, D.L. (1981). Programming in BASIC for Personal Computers. Prentice-Hall, Inc., Englewood Cliffs, NJ.

[4] Kolodner, J.L. (1980). Retrieval and organizational strategies in a conceptual memory: A computer model. RR#187. Ph. D. Thesis, Yale University New Haven, CT.

[5] Lakoff, G. and Johnson, M. (1980). Metaphors We Live By. The University of Chicago Press, Chicago, IL.

[6] Lebowitz, M. (1980). Generalization and memory in an integrated understanding system. RR#186. Ph. D. Thesis, Yale University, New Haven, CT.

[7] Papert, S.A. (1972). Teaching children thinking. Programmed Learning and Educational Technology, Vol. 9, No. 5.

[8] Schank, R.C. (in press). Dynamic Memory: A theory of learning and reminding.

[9] Sussman, G.J. (1973). A computational model of skill acquisition. AI-TR-297. Ph.D. Thesis, MIT, Cambridge MA.