THE MODELING OF SIMPLE ANALOGIC AND INDUCTIVE PROCESSES IN A SEMANTIC MEMORY SYSTEM

Joseph D. Becker
Computer Science Department
Stanford University
Stanford, California *

"It is part of our thesis that concepts in the strict sense of the term, as we know them - which, since Euler, the great mathematician (1707-1.783), ore represented by circles, a fact which means far more than meets the eye - are foreign to the Chinese mind." - Gustav Herdan, Linguistics No. 28

Summary

In this paper w. present a general data structure for a semantic memory, and we give a definition of "analogy" between items of semantic information. We then construct an inductive process in which general laws are formulated and verified on the basis of observations of individual cases.

I. Introduction

The model described in this paper represents on attempt to formalize a number of general cognitive processes. Although these processes may be said to be "simple" in the sense of being primitives of cognitive behavior, they are by no means simple to make explicit in their full generality. Within the confines of this paper we could not begin to discuss all of the intricacies of modeling these processes, and if we could, the reader could not begin to sort out the main ideas underlying the model. Therefore we have chosen to present the elements of the model in an oversimplified form designed to bring out the major ideas they embody; then in separate sections (Sections II.C, III.B, and V.B) we indicate what elaborations must be made in order for the model to be truly general. In Section V.B we also discuss the formidable problems that arise in validating a model such as this one.

Relation to Other Research

We will briefly indicate where the present model stands with respect to other semantic systems which are currently under development. Obviously it is unjust to characterize such complex models in a phrase or two, but it is impossible to compare them in detail here.

Most of the semantic memory systems that have been proposed are designed around the problem of dealing with natural language. A primary component of the "understanding" of an input text is its translation into some formal, language-free representation. In the systems of Quillian [1,2] and Simmons et al [3], the input is translated directly into a format consistent with the rest of the general semantic memory. A distinctly different approach is being pursued by Kellogg *[4],* Woods [5], and Kochen [6], who divide their semantic systems into two components: a data base (in a non-general format) and a procedural programming language which operates on this data base. In these systems the input text is translated into an appropriate program in the procedural language, and then this program is applied (rather than added) to the data base. The semantic system which we will present here contains elements of both approaches. The semantic memory is a single structure in a format which is claimed to be general, but this format is itself procedural.

The formalism is also closely allied to the predicate calculus representation adopted by many workers who are concerned with preserving deductive capacity within the system, e.g. Green and Raphael [7], McCarthy [8], and Black [9]. At the same time, we have striven to maintain consistency with what little is known of the psychology of human memory, as discussed by Bartlett [10] and Oldfield [11], and with the belief-system simulations of Colby [12], and Abelson and Carroll [33]•

The notion of "analogy" appears to have received extremely little attention in the technical literature. Evans *[14]* in his analogy-test taker of course deals with the concept, but only in a very constrained context. Much deeper studies

* The research reported here was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (SD-183).

into analogic reasoning have been made by Kling [15] in his analysis of analogies between mathematical proofs. Although Kling's investigations also are limited to a highly restricted problem domain, they are in many respects richer than those reported in the present paper (see Section III.B).

There have been any number of schemes proposed under the name "induction", with the paradigm of sequence extrapolation gaining perhaps the greatest amount of attention (including an empirically-based model by Simon and Kotovsky [16] and an exhaustive analysis by Persson [17]). Unfortunately, it is difficult to find any useful relationship between these "induction" models and the "generalization-over-cases" process described in the present paper. One striking difference is that in our model the general law is manufactured directly out of the instances from which it is inferred, rather than being selected from some narrowly-specified set or grammar of possible laws. The two procedures which seem closest to our generalization-over-cases are the generalization techniques discussed by Evans [14] and Doran [18]. These are certainly noteworthy studies, but again in these models there is a fixed set of dimensions along which generalization can occur.

Finally, there is a considerable literature on the process of "concept formation", including detailed simulations by Hunt et al [19]. Since our generalization process falls sligitly short of being concept formation (see Section IV.A), we will not attempt any comparison with concept-formation models here.

II. The Semantic Memory Structure

In this section we will describe the semantic structure which is the basis of the present model. What is defined here is actually a reduced version of the syntax of the structure; the necessary elaborations are given in Section II.C. Throughout the paper we make the convention of capitalizing a word if it is used as a formally-defined term rather than in its usual English meaning (e.g. "Situation").

A. Consequence and Criteriality

Before introducing the syntax of the memory structure, we will discuss the two notions which most sharply differentiate it from other models which have been proposed: that of "consequence" and that of "criteriality".[1]

Consequence. The principal unit of information in this model, called the Rule, contains an arrow 'W, whose function is to introduce serial order as a primitive of the data structure. The number of reasons for desiring such a primitive is so large that we can give only a sampling of them here. The idea of "consequence" in an expression such as "lightning =» thunder" has at least four possible interpretations: (l) temporal sequence, (2) causal law, (3) logical implication, and *(4)* behavioral response (e.g. "given lightning, predict thunder"). What is important is that we can store this item of data in a noncommittal fashion by means of the "=»" primitive, and leave the precise interpretation up to the processes which operate on the item. Thus, a given Rule may at one time behave as a predicate calculus formula, and at another time as a "pattern-operation" rule or "production-language" procedure for behavior. This use of serial order as a basic feature of memory is of course consistent with almost every psychological observation or theory, from Associationism to Stimulus-Response. A particularly insightful discussion of its importance is given by Lashley [21].

Criteriality. Given formal structures A and B such that B is a sub-structure of A, we will want a measure of the degree to which the presence of B in A is responsible for the distinctive identity of A. This we will call the "criteriality" of B with respect to A. For example, if it is irrelevant whether or not B is present in A, then B may be said to have zero criteriality with respect to A. The formal utility of such a notion will become clear in Section IV; at this point we may give an informal motivation in terms of the phenomenon of "attention". When we perceive the world, various

The idea of criteriality was introduced by Quillian [20], but has been relegated to a minor role in his latest model, Quillian [2].

aspects of the situation at hand receive varying degrees of attention. These variations must be recorded in the semantic memory, for they clearly affect the recall and further processing of our perceptions. Criteriality, at least in its initial assignment, is merely the frozen record of attention.

B. Description of the Memory Structure

The "objects" in this memory system are actually graph structures (i.e. pointer nests), but it is generally more convenient to work with them in a notation which disguises their net-like nature. The reader must bear in mind (as when programming in LISP) that the notation does not tell the full story, and that an "occurrence of an object" is identically the same as a pointer to some graph structure.

Two types of objects, Facts and Rules, are used to encode the content of any assertion, situation, or event. We will begin by describing the pieces from which Facts and Rules are built up.

The Node. A Node is a nest of two-way pointers. Two Nodes are equal if and only if they are identically the same Node. Thus, the Node is the "atom" of the system, like the Atom in IISP except that here the pointers are two-way. Intuitively, Nodes represent "concepts". Some of these are bound to individual, distinct entities (e.g. Eugene-McCarthy, Paul-Bunyan), while others may be considered as classes of other Nodes (e.g. Minnesotan, hero). We will usually denote Nodes by English words, but this notation is for convenience only, and bears no relation to the representation of English words within the model. When we need to discuss Nodes for individual objects which, unlike Eugene McCarthy, do not have names assigned to them already, we will invent names of the form AA, BB, etc. We will also use a special symbol "@", which designates "the Situation in which this symbol occurs" (a "Situation" is defined below).

The Kernel. A Kernel is an ordered n-tuple of Nodes, where to each Node is assigned an integer between 0 and 6, called its (Node-)Criteriality. The Kernel is interpreted as an (n-l)-ary predicate and arguments (i.e. its presence indicates the assertion of that relation), where the first Node is the predicate name. The integers exhibit criterialities on a 7-point scale. A Node of zero Criteriality is called a Dummy. For convenience in the examples presented in this paper, we will conventionally attach a Criteriality of 3 "to a Node unless we have a particular reason to do otherwise. In the notation, Kernels will be enclosed in pointy brackets, and Node-Criterialities will be written as superscripts, e.g. $\langle \text{gives}^3\ \text{Max}^5\ \text{Olga}^0\ \text{AA}^5 \rangle$.

The Situation. A Situation is an unordered set of Kernels, where to each Kernel is assigned an integer between -6 and 6, called its (Kernel-) Criteriality. The Situation is interpreted as the conjunction of the statements made by its Kernels. The negative Criterialities indicate the importance of the absence of a given condition; they serve to introduce a scaled logical negation. In the notation, Situations will be enclosed in wavy brackets, and Kernel-Criterialities will be written after the Kernels, preceded by a colon, e.g.:

$$\left\{ \begin{array}{l} \langle \text{teases}^3\ \text{Willy}^3\ \text{Twinky}^3 \rangle :3 \\ \langle \text{member}^3\ \text{Twinky}^3\ \text{parakeet}^3 \rangle :2 \end{array} \right\} \qquad \text{(i)}$$

On occasion (e.g. Figure 6) we will write a Kernel outside of the Situation to which it belongs, with a two-way arrow joining it to its proper location within the brackets.

Weights. In order for a semantic memory to be complete, it must store with each item of information a considerable number of data about that item, for example: the subjective probability that it is true, its degree of surprisingness or inconsistency given the rest of the information in memory, its trace-strength or possibly the time elapsed since it was last referred to, its degree of pleasantness, etc. In the modeling of cognitive processes, this "meta-information" about an item must play a role second only to its content in determining what is done with the item in a given process. Unfortunately, this role is difficult to describe, and the notation for many weights is messy. Therefore in this paper we will consider only the behavior of the weight for Subjective Probability in any detail. We will note points in

$$\left[\{\langle has^3\ Percy^3\ Fido^3\rangle :3\} \Rightarrow \left\{\begin{array}{l}\langle has^3\ Agnes^3\ Fido^3\rangle :3\\ \langle causes^3\ Percy^3\ @^3\rangle :2\\ \langle wills^3\ Percy^3\ @^3\rangle :1\end{array}\right\}\right] \qquad (2)$$

Section IV at which the absolute likelihood of an item is critical in determining the course of its processing.

We now introduce the two types of unit in which information is stored, the Fact and the Rule.

The Fact. A Fact is, syntactically, simply a Situation. The term "Fact" merely distinguishes those Situations which stand free, from those which occur in Rules.

The Rule. A Rule is an ordered pair of Situations. It is interpreted as asserting that the second Situation (called the Right Half) is a "consequence" of the first (called the Left Half), in the unspecified sense discussed in Section II.A. In the notation, the two Situations are written between square brackets, with an arrow between them, as in Figure 2 (above). Any Dummy which occurs in the Left Half of a Rule is implicitly universally quantified; any Dummy which appears only in the Right Half of a Rule is implicitly existentially quantified. Thus we have translations such as the following:

$$\forall_{x\epsilon A}\ \forall_{y\epsilon B}\ [\varphi_{xy}] \quad \text{becomes:} \qquad (3)$$

$$\left[\left\{\begin{array}{l}\langle member^6\ x^0\ A^6\rangle :6\\ \langle member^6\ y^0\ B^6\rangle :6\end{array}\right\} \Rightarrow \{\langle\varphi^6\ x^0\ y^0\rangle :6\}\right]$$

$$\forall_{x\epsilon A}\ \exists_{y\epsilon B}\ \{\varphi_{xy}\} \quad \text{becomes:}$$

$$\left[\{\langle member^6\ x^0\ A^6\rangle :6\} \Rightarrow \left\{\begin{array}{l}\langle member^6\ y^0\ B^6\rangle :6\\ \langle\varphi^6\ x^0\ y^0\rangle :6\end{array}\right\}\right]$$

(See Step 4.3, Section IV.B, for justification of these conventions.) Finally, attached to each Rule is a number between 0 and 1, interpreted as the Subjective Probability that the Right Half actually is a consequence of the Left Half. In the notation, we may display this "S.P." to the right of the Rule, but in most cases we will suppress it.

C. Elaborations of the Syntax

For the purpose of simplicity in this presentation, we have shorn the data format of the several forms of recursive nesting it must have if it is in fact to be able to represent arbitrary information. A Node must be able to denote any Fact or Rule (in the way that "@" denotes a Situation). A Situation must be able to contain Facts and Rules as well as Kernels, and this leads to the need for a canonical form for Situations. And Rules may be composed out of other Rules, as well as Situations.

The latter form of nesting allows us to define a notion of equivalence of representations, whose role in the completed model is extremely significant. For the present we will content ourselves with giving an example of what can be done. We can essentially define the Node "gives" by creating two Rules which expand a Situation involving "gives" into a Rule composed of more primitive predicates.[2] This "definition" might look something like Figure 4 (below).

1/ Recall that each occurrence of the special symbol "@" denotes the Situation in which it appears. The Rule in Figure 2 records the event expressed by "Percy gives Fido to Agnes". This could also be represented statically as a Fact, namely {<gives Percy Agnes Fido>} (cf. Section II.C).

2/ Such expansions of single predicates into more basic terms raise the interesting issue of the existence of "semantic primitives". The matter is still very much open; see Bendix [22] for the most thorough discussion to date.

3/ In Figure 4 we use a double arrow 'V merely as a notational device to express two distinct Rules (one each way) in the same diagram.

$$\left[\{\langle gives^6\ x^0\ y^0\ z^0\rangle :6\} \Leftrightarrow \left[\{\langle has^6\ x^0\ z^0\rangle :6\} \Rightarrow \left\{\begin{array}{l}\langle has^6\ y^0\ z^0\rangle :6\\ \langle causes^6\ x^0\ @^6\rangle :5\\ \langle wills^6\ x^0\ @^6\rangle :4\end{array}\right\}\right]\right] \qquad (4)$$

As an additional extension of the syntax, the Rule must be redefined to be a sequence (of Situations and Rules) of any length.

III. Analogy between Situations

We will give a semi-formal definition of the notion of an analogy between two Situations, discuss the features and inadequacies of this definition, and then consider the role of analogy formation in cognitive processing.

A, Basic Definition of Analogy

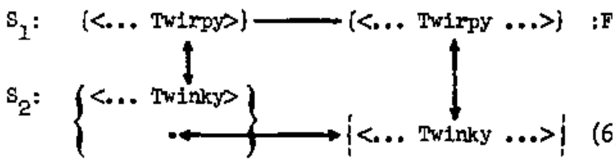
The intuitive idea we are striving to capture is that an analogy between two situations is a motivated correspondence between the elements of the situations. In essence, then, an Analogy between two Situations S and S_2 is defined to be a one-to-one mapping of the Kernels of S_1 onto the Kernels of S_2 . Each Kernel-to-Kernel pairing induces a mapping of the Nodes in one Kernel onto the Nodes in the other; we will denote such mappings in set-theoretic notation of the form $\{(a,b); \ldots\}$. We require that all of the Node-to-Node mappings be consistent (i.e. that their union be one-to-one). To formalize the idea that these mappings be "motivated", we further require that for each non-identical pair of Nodes $n_1 \epsilon S_1$ and $n_2 \epsilon S_2$ mapped into each other, we be able to exhibit some further information which will "justify" the identification of n_1 with n_2 . More precisely: we require the existence of a Situation Σ_1 containing n_1, and a Situation Σ_2 containing ru, such that, given the identification (n_1,n_2), Σ_1 and Σ_2 are themselves Analogous. There are two possible sources of each item of Justifying Information Σ_i, namely as a sub-Situation of the given Situation S_i, or as an independent Fact retrieved from memory.

An example may serve to clarify this discussion. Suppose that memory contains the Fact F = {<member^3 Twirpy^3 parakeet^3> :3}, and we are presented with:

S_1: {<teases^3 Willy^3 Twirpy^3> :3}

S_2: {<teases^3 Willy^3 Twinky^3> :3 ; <member^3 Twinky^3 parakeet^3> :2} (5)

A reasonable Analogy between S_1. and S_2 will match the Kernel <teases^3 Willy^3 Twirpy^3> of S_1 with the Kernel <teases^3 Willy^3 Twinky^3> of S_2 . This match induces the identification (Twirpy,Twinky). According to the definition, we must now seek further information about Twirpy and Twinky which will justify our mapping these Nodes into each other. The Fact F presents itself as information about Twirpy; the sub-Situation of S_2 consisting of the Kernel <member^3 Twinky^3 parakeet^3> constitutes a Fact about Twinky. The definition requires that these two Facts be Analogous, given the identification (Twirpy,Twinky), which is the case since they in fact become identical under that substitution. We may diagram the Analogy between S_1 and S_2 as:



(6)

Here the Justifying Information is on the right, and the vertical arrows denote correspondences between Nodes. Expressed in English: Willy's teasing Twirpy is analogous to Willy's teasing Twinky in that both Twirpy and Twinky are parakeets.

It is important to note that our definition of Analogy is recursive (at the point where Σ_1 and Σ_2 are required to be Analogous, given the proposed identification). The insistence that the Node-to-Node mapping be one-to-one makes "is Analogous to" a symmetric relation by giving each Analogy a well-defined inverse. The definition can easily be extended into a definition of analogy between Rules.

B. Elaborations of the Definition

As complex as the definition above may seem, it is still far too simple to be adequate for the analogy-formation situations that arise in the actual modeling of cognitive processes. Below we will mention several extensions which must be made to this definition. None of these elaborations will be pursued in the present paper.

It may occur that two Situations will match closely except for a corresponding pair of Nodes such that no additional information is available for one or both of these Nodes. In such a case one would probably want to risk identifying the two Nodes, especially if they were of low Criteriality. It will also occur in general that only a subset of one Situation can be mapped into a subset of the other. Whether or not the presence of unmatched Kernels voids the analogy must depend on their number and their Criterialities. We see, then, that a working definition of analogy must actually be based on a complex scoring function involving the number of matched and unmatched objects, their Criterialities, the depth to which the recursion must be pushed, and so on.

The intricacies of this definition multiply when we consider the need for recognizing similarities which transcend simple Kernel-to-Kemel matching. For example, the conjunction of the Kernels <eats^3 AA^3 BB^3> and <property^3 AA^3 hungry^3> should certainly match a single Kernel of the form <devours^3 CC^3 DD^3>. In fact, a single Kernel may even paraphrase a whole Rule. Clearly the notion of equivalence illustrated in Section H.C must be used to supplement the simple one-to-one mapping of Kernels on which our original definition was based.

There are other ways in which situations may be said to be analogous, besides corresponding directly. They may, for example, have similar consequences: eating cyanide and jumping off a bridge are very different activities, but they are analogous inasmuch as they have similar results. Or two situations may have similar antecedents: a rainbow and a puddle are quite dissimilar, yet both betoken the occurrence of rain. Since these other types of analogy involve the notion of consequence, they too may be defined in our system.

An inherent limitation of all the various definitions we have discussed is that they are only syntactic; they cannot ensure that the analogies produced will be semantically meaningful - i.e. that the Justifying Information will in fact be "relevant". These definitions must be regarded merely as the syntactic "necessary conditions" for analogy, where the mustering of truly relevant information is the responsibility of the larger process which makes use of analogy formation as a subroutine. The investigations of Kling [15] seem to offer the best insight so far available into what makes an analogy "meaningful" and useful.

C. Uses of Analogy Formation in Cognition

We may distinguish two major functions which analogies can perform in cognitive processing: they provide a means of dealing with novel situations, and they serve to arrange semantic information in an organized structure suitable for further processing. These two uses of analogy formation are of course aspects of one and the same process, but we may discuss them separately.

Response to Novel Situations. We have mentioned (Section II.A) that a Rule may be regarded as a routine for behavior written in a "pattern-operation" language. If we pursue such a notion, we soon realize that "left-half matching" for Rules is not at all a straightforward process, since we very seldom encounter the same situation twice, and we are often called on to respond to situations which are only vaguely similar to those we have met before. Evidently this idea of "vaguely similar situations" can be made precise by our definition of Analogy between Situations. This suggests the following prediction paradigm for dealing with novel Situations:

```
Given Situation: {---}        {---} :Predicted Situation
                   |M           ^M-1
                   v            |
                [{---}   =>   {---}]                      (7)
```

We make an Analogic mapping M between the given Situation and the Left Half of a Rule taken from memory. We then apply the inverse of the Analogy to the Right Half of the Rule, to obtain a prediction of what will happen next.

Exactly the same operation can be applied to derive overt responses, as soon as we are provided with a formalism for representing them. That is, if we let "(—-)" denote an instruction to a perceptor or effector, then we have:

```
Given Situa {---}        (---) tion Performed
              |M           ^M-1
              v            |
Existing Rule: [{---}  =>  (---)]                      (8)
```

We might consider this as a crude model of "Stimulus-Response" behavior, including the phenomenon of "stimulus generalization". The paradigm might also be said to model the "assimilation of schemata" central to Piaget *[23],* where we identify the notion of "schema" with that of "Rule".

Organizing Information from Memory. As can be seen from Figure 6, an Analogy, once formed, presents a goodly amount of information in a very organized structure. This structure grows in an orderly way as levels of recursive Analogies are applied to the Justifying Information; the sets Σ become strung out in what we may call the Path of the Analogy. For example, a diagram such as Figure 6, if extended by two levels of recursion, would look like:

$$\begin{array}{ccccccc} S_1 & - & \Sigma_1 & - & \Sigma_1' & - & \Sigma_1'' \\ \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow \\ S_2 & - & \Sigma_2 & - & \Sigma_2' & - & \Sigma_2'' \end{array} \qquad (9)$$

The simplest way of gleaning information from such a Path is to take the unions $S_1 \cup \Sigma_1 \cup \ldots$ and $S_2 \cup \Sigma_2 \cup \ldots$ Note that semantically, $S_i \cup \Sigma_i \cup \ldots$ has the meaning of $S_1 \wedge \Sigma_1 \wedge \ldots$, since a Situation is interpreted as the conjunction of the Kernels it contains. We will call the process of taking such unions Path Compression.

The information provided by an Analogy is structured enough to serve as a starting point for many cognitive processes, including deduction. In the next section we will examine in detail the role played by Analogies in a process of inductive generalization.

IV. The Modeling of an Inductive Process

A. A Sketch of the Process

The cognitive behavior which we would usually term "generalization" is in general a complex problem-solving process, involving strategies of guessing, deductions from "general principles", and so on. In this section we will be discussing an operation which is very much simpler and more primitive, but which nevertheless seems to merit being called "inductive". We may describe this process very schematically as follows:

Suppose that and are two properties, behaviors, etc., and that for a number of entities X we observe φ_X and ψ_X to co-occur. If enough such cases accumulate, and especially if 9 and t are *a.* priori unlikely individually, we might attempt to explain their co-occurrence by postulating that one entails the other, e.g. that $\varphi \rightarrow \psi$. That is, we generalize over the individual cases X to postulate: $\forall x\ [\varphi_x \rightarrow \psi_x]$. This proposed law may be tested by finding new cases Y for which φ_Y , and noting whether or not the prediction of ψ_Y is satisfied. The formation and testing of such generalized implications will be called the Generalization-over-cases process.

If the induced rule is in fact successful, a logical next step is to consider the two new entities: $\Phi = \{x |\ \varphi_x\}$ and $\Psi = \{x |\ \psi_x\}$. The generalization may now be rewritten as $\forall x\ [x \in \Phi \rightarrow x \in \Psi]$, which can be neatly compressed to: $\Phi \subset \Psi$. Since Φ and Ψ are classes of entities, they are "concepts" in the traditional psychological sense, and the process we have just described is evidently a meaningful form of concept formation. Note that in the expression $\Phi \subset \Psi$ we have attained a "higher-order" relation which may be considered without reference to the individual cases from which it arose. This compression of an implication into a single higher-order predicate (in this case, "$\subset$") is mediated by an equivalence of precisely the sort illustrated in Section II.C . We will not discuss concept formation further in this paper, but it clearly can be represented within our formalism.

Although the above description may seem rather straightforward, the complete modeling of Generalization-over-cases is actually a very complex matter. Rather than attempt to present an immense algorithm abstractly, we will content ourselves with following in detail how the process might work in a particular example. This form of exposition has its perils, of course. The example probably would not work as described. It is sometimes difficult to distinguish properties of the particular example from the general behavior of the algorithm. And it is often not clear why, at a given point, one thing is done next and not another. For these deficiencies we can only beg the reader's indulgence. With respect to the last objection we may note that in cognitive processes the factors which determine precisely what is operated on next need not be closely relevant to the operation itself (nor very interesting: e.g. the precise connectivity of the semantic memory at the time the process takes place).

B. Example of Generalization-over-cases

For the purpose of clarity in our example, we will not indicate Node-Criterialities except where we have particular interest in them. Where they are omitted, the reader may assume they have the conventional value 3.

Phase I: The first structuring of information.

Synopsis: Three Facts F_1, F_2, F_3 exist in memory. A new Fact F_4 is found to be Analogous to F_2, with F_1 and F_3 serving as Justifying Information. By Path Compression two new Facts are created:

$$F_5 = F_2 \wedge F_1 \text{, and } F_6 = F_4 \wedge F_3 .$$

Steps 1.1-1.3: At various times the following three Facts find their way into our semantic memory:

F_1: {<member Wilfred fireman> :3}

F_2: {
<wears Wilfred AA> :3
<member AA suspenders> :2
<property AA red> :2
<time @ Thursday> :2
}

F_3: {<member Cyrus fireman> :3} (10)

Step 1.4: At some time, presumably while visiting Peoria, we notice that:

F_4: {
<wears Cyrus BB> :3
<member BB suspenders> :2
<property BB red> :2
<location @ Peoria> :2
} (11)

It would be possible simply to record this new Fact without analyzing it farther, but we seldom record input without trying to "understand" it. One aspect of understanding a novel situation is to relate it to something already known. In our model, this means to find an Analogy with some Situation already in memory.

Step 1.5: We seek an Analogic match for F_4 in memory, and propose F_2 as a candidate, with M = {(BB,AA);(Cyrus,Wilfred)}. The correspondence (BB,AA) is justified by <member BB suspenders> and <member AA suspenders> (also by the shared property of redness), while the identification (Cyrus,Wilfred) is justified by F_3 and F_1. Sad to say, the Kernels <time @ Thursday> of F_2 and **<location @ Peoria>** of F_4 map neither into each other nor into anything else.

When a Kernel is not immediately matched in what is an *otherwise* Promising Analogy, a search is instituted for information that will correspond — we would for example search memory if possible, to determine whether or not F_4 was occurring on a Thursday. If such information is unavailable and the Kernel must remain unmatched, this fact must duly be taken into account in scoring the Analogy (recall Section III.B). Even if the Analogy is accepted, this failure to match will affect further processing (Step 3.3). Let us assume that in our example these two Kernels go unmatched, but the Analogy is accepted anyhow.

The mere construction of the Analogy does not in itself produce anything that we would be willing to call the "understanding" of the new Situation. What is significant is the structuring of the four

```
F4: {... Cyrus ...} ———— {... Cyrus ...} :F3
          ↕                      ↕
F2: {... Wilfred ...} ——— {... Wilfred ...} :F1   (12)
```

Step 1.6: Our Analogy, in relating F_2 to F_1, suggests that these two Facts about Wilfred may somehow be "relevant" to each other. It seems reasonable to commemorate this relationship at least by recording a new Fact which is their conjunction: $F_5 = F_2 \wedge F_1$. We note that, syntactically F_5 is merely the union of F_2 and F_1, and so we have just performed the operation of Path Compression defined in Section III C. Written explicitly, F_5 is:

F_5: {
<wears Wilfred AA> :3
<member AA suspenders> :2
<property AA red> :2
<time @ Thursday> :2
<member Wilfred fireman> :3
} (13)

Analogously, we form $F_6 = F_4 \wedge F_3$.

Phase II: The Analogy perpetuates itself.

Synopsis: A new Fact F_8 is encountered and found

Analogy is facilitated by a remnant of the old Analogy (namely the copy of F_1 embedded in F), which allows the prediction of a Fact F_7 which serves as Justifying Information. A new Path Compression forms $F_9 = F_8 \wedge F_7$.

Step 2.1: At some point we record that:

F_7: (<member Rupert fireman> :5) .

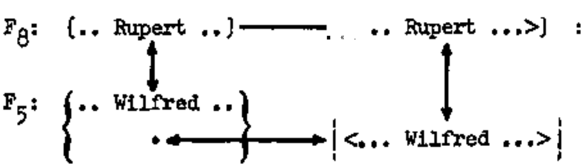
Step 2.2: Later, at the Firemen's Ball, we are keeping tabs on Rupert when we suddenly notice that:

F_8: { <wears Rupert CC> :3
<member CC suspenders> :2
<property CC red> :2
<dances-with Rupert Maude> :2 } (14)

Presumably this observation is sufficiently striking that we will attempt to "understand" it.

Step 23: A search is made through semantic memory for Situations resembling F_8 . Let us suppose that F_5. happens to present itself as a candidate for matching.

Step 2.4: In constructing an Analogy between F_8 and F_5 there are three unmatched Kernels: <dances-with Rupert Maude> in F_8, and <time @ Thursday> and member Wilfred fireman> in F_5 . As noted in Step 1.5, each of these induces a search through memory for corresponding information. In this case, the first two left-overs presumably find no match, but happily the Kernel <mnember Wilfred fireman> does retrieve an item from memory, namely F_7 . The Path which justifies the identification (Rupert,Wilfred) is therefore:



(15)

There is a significant improvement in the way in which the "firemanhood" Justifying Information was found here, in contrast with Step 1.5 • In "the Phase I Analogy, the Facts relating Cyrus and Wilfred were turned up by a poorly-guided or non-guided search. In this new Analogy, on the other hand, the Kernel <member Wilfred fireman> served as a clue to search for a specific item, namely the corresponding Fact {<member Rupert fireman>} . Recall that this Kernel was descended of F_1 via the Phase I Analogy. Thus the old Analogy has greatly facilitated the formation of a new Analogy similar to it.

Step 2.5: A new Path Compression gives us yet another Fact: $F_9 = F_8 \wedge F_7$.

Phase III; Conjunction is restructured into implication. Synopsis: The successful prediction of F triggers a re-examination of the Facts F_5, F_6, and F_9, which are found to be mutually-Analogous conjunctions. The conjunction is split up and reorganized as two tentative implications (Rules) R. and R_2, one going each way. The Criterialities within R_1 and R_2 are based on information provided by the Analogies among F_5, F_6, and F_9 .

Step 5.1: The successful search for F_7 in Step 2.4 may be regarded as a "prediction" that Rupert is a fireman. Since "Rupert" and "fireman" are both low-frequency concepts, the success of this prediction may be so surprising as to cause us to re-examine the conjunctions which F_5 and F_9 represent. In particular, we search memory for other similar (Analogous) items, and turn up F_6 .

Step 5.2: We reconstruct the three Analogies among F_5, F_6, and F_9 . The three are parallel to each other, thus:

$F_5 = F_2 \wedge F_1$	Wilfred	AA	
$F_6 = F_4 \wedge F_3$	Cyrus	BB	
$F_9 = F_8 \wedge F_7$	Rupert	CC	(16)

Step 3.3" Picking one of these Facts as typical, say F_9, we now convert its conjunction into a pair of implications, thus implicitly following the reasoning of Section IV.A • These implications will of course be represented as Rules. We will have basically, but not exactly: $R_1 = F_7 \rightarrow F_8$ and $R_2 = F_8 \rightarrow F_7$. We must now consider the modifications to be made to F_7 and F_8 before they are

-< The arbitrariness of this choice, and of many other aspects of the process we describe here, reflect the almost total lack of psychological data on which to base the algorithm. Hopefully, future studies along the lines of Posner and Keele *[2k]* will eventually enable us to make far more accurate models of when and how generalization takes place.

$$R_1: \left[\{\langle member^4\ Rupert^2\ fireman^4 \rangle :4\} \Rightarrow \left\{ \begin{array}{l} \langle wears^4\ Rupert^2\ CC^2 \rangle :4 \\ \langle member^4\ CC^2\ suspenders^4 \rangle :3 \\ \langle property^4\ CC^2\ red^4 \rangle :3 \\ \langle dances\text{-}with^4\ Rupert^2\ Maude^4 \rangle :1 \end{array} \right\} \right] \quad (17)$$

---

combined into R_1. (the story is identical for R_2).

Analogies, as we have said, are a superb source of information, and there is a great deal of infor mation left in the Analogies made in Step 3.2 that can contribute to the content of R, In particular, examination of these Analogies will determine the Criterialities of the various parts of R_1, as follows.

Node-Criterialities: The three Facts F_5, F_6, and F_9 differ significantly only in the two triplets of corresponding Nodes: Wilfred, Cyrus, Rupert; and AA, BB, CC . Clearly these are the cases which the Analogies generalize over. It appears that the presence of one or another of these Nodes is not crucial to the general law which unites the three Facts. Therefore, by definition these Nodes are less criterial to R_1 than those Nodes which are constant over all three Facts.

Kernel-Criterialities: The Kernels which never found Analogic matches have shown themselves to be dispensable in, or perhaps even irrelevant to the generalization which underlies the three Facts. They thus are less criterial to R_1 than those Kernels which found mates in all three Analogies.

Thus, in assembling R from F and F_9 we increase the Criterialities of those structures which through their constancy give evidence of being relevant to the Rule, and decrease the Criterialities of those structures whose presence shows signs of being inessential. Given that all of the Node-Criterialities were originally 3> "the new Rule R will be as shown in Figure 17 (above).

We should mention that these adjustments of Criterialities, like everything else in life, are fallible. It might have happened, for instance, that by coincidence both Wilfred and Rupert were observed to wear red suspenders on a Thursday. We assume that with the accumulation of evidence (as in Phase IV below), such coincidences will wash out of the inductions.

Step 3.4: The data structure requires that the new Rules R_1 and R_2 be assigned Subjective Probabilities, Since we have as yet no reason to prefer one of these Rules over the other, their initial S.P.'s should be equal. We might assign initial values of 1/4 , and thereafter treat the S.P. as the ratio of $\frac{\text{number of successes}}{\text{number of predictions}}$ when the Rule is used predictively (as below). This particular treatment of S.P.'s, and the particular scheme we use in adjusting Criterialities, were concocted for illustrative purposes in presenting this example; the actual manipulations must of course be more subtle.

Phase IV; New evidence argues for a generalization.
Synopsis: A new Fact F is matched Analogically with the Left Half of R_1 . The inverse of the Analogy is applied to the Right Half of R_1 to obtain a predicted Situation. This Situation is in fact observed, and R_1 is rewarded for its success.

Step 4.1: At some arbitrary time after Phase III has taken place, a new Fact comes to our attention: F : {<member Otis fireman> :3} .

Step 4.02: In an attempt to understand F_{10}, we seek an Analogic match for it in memory. Suppose that the Left Half of R_1 comes up as a candidate for matching. Although we have no Justifying Information for the identification (Otis,Rupert), the Situations are otherwise identical, so we may assume that the Analogy is accepted.

We are now in a position to follow the "prediction paradigm" of Figure 7. That is, we may apply the inverse of our Analogy, namely the mapping $M^{-1} = \{(Rupert, Otis)\}$, to the Right Half of R_1 to obtain a Situation which we may expect to observe or to find already recorded in memory.

Step 4.3. The application of M^{-1} to the Right Half of R_1 encounters an interesting difficulty. This Right Half contains a Node, CC, whose low Criteriality indicates that it has been generalized over. We would expect the Analogy M^{-1} again to map CC into some other Node, but M^{-1} in fact provides no such Node. Hence in this and similar cases we are led to predict the existence of an entity in the new Situation, corresponding in this case to CC. (Note that the translations of quantifiers in terms of Dummies (Figure 3) arise from just this sort of argument.) We may optimistically create a name for our new entity, say XX. We thus predict a Situation of the form:

$$\left\{ \begin{array}{l} \langle \text{wears Otis XX} \rangle : (4) \\ \langle \text{member XX suspenders} \rangle : (3) \\ \langle \text{property XX red} \rangle : (3) \\ \langle \text{dances-with Otis Maude} \rangle : (1) \end{array} \right\} \qquad (18)$$

The Kemel-Criterialities here, which are taken from those in the Right Half of R_1, assume a new role: they indicate the zeal with which we should seek a realization of the given Kernel. That is, the dominant Criteriality of <wears Otis XX> tells us that our main job is to look for something Otis wears; the low Criteriality of <dances-with Otis Maude> suggests that we should not give much concern to finding such a condition, since this Kernel is under suspicion of being irrelevant to the prediction.

Step 4.4: We may search for the predicted Situation in our semantic memory or in the real world. If it is in fact true that all firemen wear red suspenders, then we will indeed find such a Situation. We will find an object to which we can attach the name XX, and in all likelihood there will be no match for the Kernel <dances-with Otis Maude> .

Step 4.5: This successful prediction gives us valuable information with which we may adjust the Rule R_1 . We may raise and lower Node- and Kernel-Criterialities in accordance with the Analogy formed between the Right Half of R_1 and the predicted Situation. In particular, the unmatched Kernel <dances-with Rupert Maude> attains a Criteriality of 0 and disappears, since a Kernel of zero relevance has no place in a Situation. The successful prediction of course increases the Subjective Probability that the Rule is a valid one. Thus from R_1 we derive a new Rule R_1', shown in Figure 19 (below).

Steps 4.6+: After enough recurrences of Steps 4.1-4.5, our successful induction will approach the form shown in Figure 20 (below). Here the zero Criterialities of the Nodes "Rupert" and "CC" indicate that these occurrences have completely lost their identities and become Dummies. Like dummy variables in mathematical notation, these Nodes could be replaced (consistently, of course) by any symbols, e.g. "x" and "y". In view of the translation between Dummies and quantifiers given in Figure 3, R_1'' expresses precisely the proposition:

$$\forall_{x \in fireman} \; \exists_{y \in suspenders} \; [Wears(x,y) \wedge Property(y,Red)] \qquad (21)$$

Which is to say, "Firemen wear red suspenders."

Phase V: New evidence may argue against a generalization. We recall that a Rule R_2 was formed along with R_1, and is its converse. The statement made by R_2 is that "If a person wears red suspenders, then he is a fireman". Although this proposition is false in the absolute, it will be worth retaining if its statistical validity is significantly greater than zero. Therefore we will reward this Rule as per Phase IV when it succeeds, simply adjust its Subjective Probability when it fails, and expunge it if the Subjective Probability

$$R_1': \left[\left\{ \langle \text{member}^5 \; \text{Rupert}^1 \; \text{fireman}^5 \rangle : 5 \right\} \Rightarrow \left\{ \begin{array}{l} \langle \text{wears}^5 \; \text{Rupert}^1 \; \text{CC}^1 \rangle : 5 \\ \langle \text{member}^5 \; \text{CC}^1 \; \text{suspenders}^5 \rangle : 4 \\ \langle \text{property}^5 \; \text{CC}^1 \; \text{red}^5 \rangle : 4 \end{array} \right\} \right] \quad S.P. = \frac{2}{5} \qquad (19)$$

$$R_1'': \left[\left\{ \langle \text{member}^6 \; \text{Rupert}^0 \; \text{fireman}^6 \rangle : 6 \right\} \Rightarrow \left\{ \begin{array}{l} \langle \text{wears}^6 \; \text{Rupert}^0 \; \text{CC}^0 \rangle : 6 \\ \langle \text{member}^6 \; \text{CC}^0 \; \text{suspenders}^6 \rangle : 5 \\ \langle \text{property}^6 \; \text{CC}^0 \; \text{red}^6 \rangle : 5 \end{array} \right\} \right] \quad S.P. = 1 \qquad (20)$$

falls below some threshold. In this way our model becomes capable of retaining "half-truths" - a capacity which is very valuable to a semantic memory (among other things, it allows the storage of contradictory information).

V. Discussion

A. The Relation Between Analogy and Generalization

If we examine the Generalization-over-cases process closely, we find that an interesting statement can be made of the relation between analogy and generalization. In Step 3.3 we saw how a great deal of information supplied by Analogies was incorporated into the representation of the generalized Rule. Thus, analogy takes part in generalization. But in Phase IV we found that the induced Rule led to the search for a Situation which was in fact Analogous to previously-known Facts, and which might have gone unnoticed if the inductive generalization had not existed. Thus, generalizations facilitate the finding of new analogies. In fact, the search for the new Analogy was guided by precisely that information which had been contributed to the Rule by the old Analogy (i.e. Criterialities, see Steps 3.3 and 4.3), so we might say that analogies perpetuate themselves via generalizations. On the other hand, we could also summarize Phase IV by saying that generalizations perpetuate themselves via new analogies. In any case, we have certainly shown that analogy and generalization are mutually reinforcing processes which can hardly be separated from each other.

B. Problems in Modeling Cognitive Processes

We are at present attempting to construct a computer implementation (in the LISP language) of the processes outlined in this paper. Such an effort necessarily leaves one sadder but wiser with regard to the prospects for formulating and testing explicit cognitive models. In this section we will discuss some of the more forbidding obstacles we have encountered. We feel that the problems brought out below correspond not to deficiencies in our particular model (although there are enough of those, heaven knows), but rather to major dilemmas attending the construction of any general cognitive model in a large semantic memory system.

We have already mentioned that the choice of "what to do next" in a cognitive process is often poorly specified. Because of the extreme scarcity of psychological data regarding such choices, the model builder is confronted with a small infinity of arbitrary decisions in designing an algorithm. The cumulative effect of these low-level choices may well wash out the central theoretical propositions that the model was designed to test.

There are other factors which complicate the issue of what should be done when and for how long. In the first place, many cognitive processes contain no inherent termination condition. Like memory search or the construction of Analogies according to a recursive definition, they are bounded only by the size of semantic memory. In the second place, a cognitive operation is seldom totally successful or totally unsuccessful. As in our discussion of Analogies, success must be defined by a scoring function and threshold. These considerations would seem to imply that a general cognitive process cannot be represented as an orderly succession of tidy operations, but instead must be couched in a welter of effort-limiting and evaluation heuristics.

Of all of the issues we have sidestepped in Sections III and IV, certainly none is more worrisome than the problem of memory search. Not only must relevant information be brought forth, but this must be done without exhaustive search (i.e. rapidly), despite the fact that 99% of the contents of memory will be irrelevant to the given search. Moreover, the phenomena of "set" and "effect of context" show us that in human memory the memory structure (or, equivalently, the means of search access into it) is continuously adapting in response to ongoing cognitive activity. Certainly no process involving a large, general semantic memory can be adequately modeled until some progress is made on this most refractory set of problems.

The necessary size and intricacy of a semantic memory create a host of methodological problems in validating the algorithms which operate in such a system. It becomes extremely time-consuming to construct a suitable data-base on which to test an algorithm - especially if that data-base is to be realistic in being 99# irrelevant to the test problem. Sometimes it becomes very difficult to distinguish which properties of a program's behavior are inherent in its algorithm, and which stem from its interaction with the particular test data-base used. In addition, any algorithm will contain dozens of arbitrarily-set parameters and arbitrarily-made decisions. Ideally, one would evaluate these by varying them one at a time, using a large number of test data-bases, but such a procedure is out of the question in practice.

The ultimate problems in validation arise when one strives, as we have striven, to characterize general "subroutines" of cognitive behavior, rather than attempting to build a beginning-to-end model of a particular type of performance in a well-defined cognitive task. The processes of Analogy formation, Generalization-over-cases, and the prediction paradigm of Section III.C are not by themselves sufficient to model any particular cognitive behavior. They are intended rather to represent elementary sub-processes which may be observed to participate in a very wide range of psychological phenomena, from sensory perception to natural language understanding. Our approach is in accord with the venerable programming dogma that the best and often the only way to come to grips with a complex process is to decompose it into easily-conceptualized subroutines. Certainly this law must apply to that most complex of processes, human cognition. But the question immediately arises of how one is to validate a proposed algorithm for a "cognitive subroutine'[1]. It is essentially impossible for experimental techniques to provide data on a single cognitive sub-process taken in isolation from all others. But with no data as to how the subroutine is supposed to perform, one cannot even debug a proposed algorithm, much less validate itl

Is it not intolerable, in a scientific investigation, to be asked to consider models for which empirical validation is next to impossible? We think not. Consider the situation in linguistics. The linguist (not to be confused with the metatheorist, or prophet) spends his time trying to model particular aspects of a particular language, e.g. negation or nominalization. He does not have a complete grammar of the language available to him, nor does he attempt to construct one. He knows that his limited model is guaranteed not to be fully consistent with empirical observations of language, because in language too it is impossible in reality to isolate one aspect from all the others. In the face of a host of counter-examples, exceptions, and phenomena not covered by his model, the linguist calmly decides to judge the worth of his theory by subjective criteria such as internal elegance and explanatory power. He is happy with a model if it gives him a better understanding than he had before.

We hope that the model presented in this paper gives the reader a better understanding than he had before.

Acknowledgments

Many of the ideas put forth in this paper rightfully belong either to Jerome Feldman of Stanford University, or to Ross Quillian of Bolt Beranek and Newman Inc. I would also like to acknowledge my great personal indebtedness to these two gentlemen. This research was originally inspired by Kenneth Colby of Stanford, and he has contributed much to the model. I am most grateful to Daniel Bobrow of Bolt Beranek and Newman, and to Nils Nilsson, Bertram Raphael, and Charles Rosen of Stanford Research Institute for their encouragement and for the opportunity to use their computing machines in developing the LISP implementation of the model.

References

[1] Quillian, M. Ross, "Word concepts: a theory and simulation of some basic semantic capabilities," Behavioral Science, 12(5), Sept. 1967, 410-430

[2] Quillian, M. Ross, "The Teachable Language Comprehender: a simulation program and theory of language," Jan. 1969 (forthcoming)

[3] Simmons, Robert F.; Burger, John F.; and Schwarcz, Robert M., "A computational model of verbal understanding," SP-3132, System

Development Corporation, Santa Monica, Calif,, April 1968

[1*] Kellogg, Charles H., "A natural language compiler for on-line data management," Proceedings, Fall Joint Computer Conference, AFIPS, Dec, 1968, 473-492

[5] Woods, William A., "Procedural semantics for a question-answering machine," Proceedings, Pall Joint Computer Conference, AFIPS, Dec. 1968, 457-471

[6] Kochen, Manfred, "Automatic question-answering of English-like questions about simple diagrams," Journal of the ACM, I6(I), Jan. 1969, 26-1*8

[7] Green, C. Cordell and Raphael, Bertram, "The use of theorem-proving techniques in question-answering systems," Proceedings, 23rd National Conference, ACM, Aug. 1968, 169-181

[8] McCarthy, John, "Programs with common sense," Memo No. 7, Stanford Artificial Intelligence Project, Stanford, Calif., Sept. 1963

[9] Black, Fischer, "A deductive question-answering system," in Semantic Information Processing, Marvin Minsky ed., MIT Press, Cambridge, Mass., 1968, *354-402.*

[10] Bartlett, Sir Frederic C, Remembering, Cambridge University Press, Cambridge, England, 1932

[11] Oldfield, R. C, "Memory mechanisms and the theory of schemata," British Journal of Psychology, 45, 195U,"IE3g

[12] Colby, Kenneth M., "Computer simulation of change in personal belief systems," Behavioral Science, 12(3), May 1967, 248-253

[13] Abelson, Robert P. and Carroll, J. Douglas, "Computer simulation of individual belief systems," American Behavioral Scientist, 8(9), May 1965, 24-30

[14] Evans, Thomas G., "A program for the solution of a class of geometric-analogy intelligence-test questions," AFCRL-61-884*, Air Force Cambridge Research laboratories, Hanscom Field, Mass., Nov. 1964

[15] Kling, Robert E., "Reasoning by analogy with applications to logical pattern recognition," Ph.D. Thesis, Department of Electrical Engineering, Stanford University, Stanford, Calif, (forthcoming)

[16] Simon, Herbert A. and Kotovsky, Kenneth, "Human acquisition of concepts for sequential patterns," Psychological Review, 70(6), 1963, 534~*546*

[17] Persson, Staffan, "Some sequence extrapolating programs: a study of representation and modeling in inquiring systems," Technical Report No. CS 50, Department of Computer Science, Stanford University, Stanford, Calif., Sept. 1966

[18] Doran, James E., "Planning and generalisation in an automaton/environment system," Presented to Fourth Machine Intelligence Workshop, Edinburgh, Scotland, Aug. 1968

[19] Hunt, Earl B.; Marin, J.; and Stone, F., Experiments in Induction, Academic Press, New York, 1966

[20] Quillian, M. Ross, "A notation for representing conceptual information: an application to semantics and mechanical English paraphrasing," SP-1395, System Development Corporation, Santa Monica, Calif., Oct. 1963

[21] Isahley, Karl S., "The problem of serial order in behavior," in Cerebral Mechanisms in Behavior, L. A. Jeffress ed., Wiley, New York, 1951, 112-036

[22] Bendix, Edward H., "Componential analysis of general vocabulary: the semantic structure of a set of verbs in English, Hindi, and Japanese," International Journal of American Linguistics, 32(2), Part II, 1966

[23] Piaget, Jean, The Psychology of Intelligence, Humanities Press, New York, First published 1947

[21*] Posner, Michael I. and Keele, Steven W., "On the genesis of abstract ideas," Journal of Experimental Psychology, 77(3), Part 1, July 1968, 353-363