Deriving Descriptions of the Mind

Technical Report PCG-7

Steve Garlick and Kurt VanLehn

Departments of Psychology and Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213 U.S.A.

7 July 1987

This research was supported by Personnel and Training Research, Psychological Sciences Division, Office of Naval Research, and DARPA under Contract Number N00014-86-K-0349. Approved for public release; distribution limited.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
Distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
PCG-7

5. MONITORING ORGANIZATION REPORT NUMBER(S)
Same as Performing Organization

6a. NAME OF PERFORMING ORGANIZATION
Carnegie-Mellon University

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Personnel and Training Research
Office of Naval Research (Code 1142 PT)

6c. ADDRESS *(City, State, and ZIP Code)*
Department of Psychology
Pittsburgh, PA 15213

7b. ADDRESS *(City, State, and ZIP Code)*
800 N. Quincy St.
Arlington, VA 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Same as Monitoring Organization

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-86-K-0349

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS 442a558-02/12-15-86

PROGRAM ELEMENT NO	PROJECT NO	TASK NO.	WORK UNIT ACCESSION NO
N/A	N/A	N/A	N/A

11. TITLE *(Include Security Classification)*
Deriving Descriptions of the Mind.

12. PERSONAL AUTHOR(S)
Steve Garlick and Kurt VanLehn

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM 86Jun15 TO 87Oct31

14. DATE OF REPORT *(Year, Month, Day)*
87 July 7

15. PAGE COUNT
42

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Protocol analysis, cognitive psychology, machine learning, procedure induction

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

See reverse side.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL
Susan Chipman

22b. TELEPHONE *(Include Area Code)*
(202) 696-4322

22c. OFFICE SYMBOL
1142 PT

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Table of Contents

Abstract

Cirrus is a tool for protocol analysis. Given an encoded protocol of a subject solving problems, it constructs a model that will produce the same protocol as the subject when it is applied to the same problems. In order to parameterize Cirrus for a task domain, the user must supply it with a problem space: a vocabulary of attributes and values for describing spaces, a set of primitive operators, and a set of macro-operators. Cirrus' model of the subject is a hierarchical plan that is designed to be executed by an agenda-based plan follower. In this paper, the philosophical and mathematical foundations of Cirrus are explored.

1. Introduction

1.1. Function and Structure: Two Alternatives in Describing Cognition

In this section we outline a distinction between *Functional* and *Structural* viewpoints in psychological theory. We postulate that the source of the distinction is in the source of constraints that inform theory building. Subsequently, we propose that functional level theories should hypothesize only functional relationships and we present a computational model describing this process.

Dreyfus and Dreyfus (1987) trace the history of two alternative approaches in the developement of the new sciences of cognition. One approach, which we label the *functional* approach, saw that there was a common level of description between the symbol-manipulating capacities of the digital computer, and the apparent symbol-manipulating capacities of the human information processor. This common level was described by the *Physical Symbol System* hypothesis (Newell, 1980, 1982), and in short, stated that the necessary and sufficient ingredient for intelligence was a system capable of manipulating symbols, regardless of whether this system was implemented in silicon or organic substrates. The analogy drawn by Newell is on a formal level, that is, the bridge between brains and computers lies in the their representational ability.

The second approach, which we label the *structural* approach, drew its inspiration not from the capacities of the digital computer, but from the emerging neurological sciences. Rosenblatt (1962) considered that cognition should begin in the processes and organization of the physical system underlying intelligence.

> It is both easier and more profitable to axiomatize the *physical system* and then investigate this system analytically to determine its behaviour, than to axiomatize the *behaviour* and then design a physical system by techniques of logical systhesis.

(Rosenblatt, 1962; in Dreyfus and Dreyfus, 1987). This approach is currently represented by the burgeoning school called connectionism, which utilizes massively parallel connections of simple computational neuron-like units to model cognitive phenomena (McClelland and Rumelhart, 1986).

These two approaches have different points of view as to what constitutes an adequate basis for describing cognition. Currently there is much debate over the veracity, purpose and usefulness of one over the other of these viewpoints. The difference has been described as a symbolic-subsymbolic distinction (Smolensky, 1984, 1986) although both classes of models are clearly symbolic in that both seek to represent (symbolize) certain aspects of cognition. Others have argued that the difference is one of levels of description, an *implementational* level as constrasted to an *algorithmic* level of description (Anderson, 1987; Marr, 1982). We prefer to avoid this terminology since it implies a definite causal ordering; i.e., that the implementational level merely exists to implement pre-existing algorithms. To avoid this implication, we have termed the two viewpoints *functional* and *structural.*

We wish to argue that the difference between the two classes of theory stems from the different sources of constraint that informs the two classes of theory. The *functional* approach argues that representation is the basis of cognition. Cognition should be described in terms of the formal structures capable of manipulating the relationship between such symbolic objects to achieve the logical requirements for computation. Smolensky (1986) describes this as follows: we have theories with descriptive entities such as formal logic, that capture human information processing in some high level domains (such as mathematics and circuit analysis). Symbolic theorists attempt to extend this high level of description "down the abyss", to attempt to describe the vast middle ground of cognition, for which there is no formal domain theory, in terms of rules and effective procedures. Thus, the source of constraints for these types of theories is the actual task domain. This appeals to a more *functional* level of analysis, where the constraints for a computational theory are drawn from the functional requirements of the task. (c.f., Chomsky's notion of competence).

The connectionists on the other hand, stated that cognition was an *emergant* property of the arrangement of the physical units of the brain. Their paradigm is that intermediate cognition is of the same kind as low level perceptual processing and is well described by reference to low-level constraints on the type of computation that can be performed by the brain. These constraints are dictated by structural considerations, such as the parallel nature of neuronal structure and computational abilities of richly interconnected units with only local information. Thus, connectionists attempt to climb "up the abyss" to the mid-ground of cognition from a structural level of description whose constraints derive from neural considerations.

Our current concern is not to contrast these two theoretical approaches, but to draw an implication from the distinction. The functional approach describes the competencies required by certain tasks, while implementational level theories give structural accounts of how such functional competencies may be achieved in a neurally plausible way. We argue that this distinction should be taken seriously in cognitive science, and should dictate the kind of theoretical entities that are appropriate at each level. We further wish to argue that the model of cognition presented here, which certainly falls within the functional approach, demonstrated how this distinction might be taken seriously. It does so by presenting a method of inducing the functional competancy of a particular domain skill without invoking hypothesized structure.

The argument is as follows: Some domains of human competence are rule-like, they embody a theory or an effective procedure. Usually this rule structure is something derived from the environment and serves to describe the structure of the task. For example, there are rules for performing subtraction procedures. These rules are codified in the teaching environment and the succesful learner is one who suceeds in internalizing these rules.

Functional theories are capable of describing this level of rule-using competence. They are descriptive, detailing at the level of logic and computation, the abilities required for executing a particular task. Descriptive devices to describe such abilities are rules and representations. For example, consider the production rule formulation from Anderson's well-known ACT* theory (1983).

IF the goal is to subtract the digits in a column
and the subtrahend is larger than the minuend
THEN set a subgoal to borrow.

This rule formalizes a particular piece of competency required to accomplish that task. Note that as such, it is descriptive of the *task domain* rather than of the performer of the task.

A collection of such rules can describe the competency required to perform a particular skill in a certain domain. To verify completeness and sufficiency, such rules can actually be "run" in a computational formalism called a production system, which effectively provides the control structures to determine the order of firing of these rules, and provides the storage facilities to keep track of inputs and partial answers. We argue that the production system itself is useful in determining the sufficency of the postulated set of rules.

One problem with this approach to determining functional knowledge is that the rule set must be deduced apriori. For domains which have a clear formal domain theory, this is not a huge problem (c.f., the rule sets for geometry proofs in Anderson et al, 1981). In effect, determining the knowledge that a problem-solver has in a domain then becomes a generate and test cycle. We generate a set of rules and then test them in a production system to determine the sufficiency of the postulated rule set. Note that no claims for necessity can be made for such a derivation method.

An alternative to this approach would be to attempt to determine this functional knowledge from the actual performance of a problem solver. Rather than extract the proposed competence from the task domain, it may be instructive to derive this knowledge from the student. We may be able to look through the students eye's at the acquired domain knowledge, rather than just postulating necessary knowledge.

A secondary concern with the functional approach is that the mechanism for determining the completeness of the rule set (e.g., the production system interpreter) may start to look attractive as the structural basis with which such competence is achieved. By eliminating the generate and test cycle, and deriving the rule set that characterizes a task domain from actual performance data, we eliminate the need for an executing mechanism to gaurantee sufficiency, and thus avoid the need for statements about structural hardware employed by the problem solver. This avoids the problem of making structural propositions based on a functional source of constraints. We return to this problem in our conclusions.

The model of cognition presented here directly confronts the need to extract the functional knowledge of a problem solver, without being seduced into making structural assumptions. (We will later argue that structural hypotheses should be made within structurally constrainted theories). This paper presents a theory of how the description of the mind should proceed. We describe how a theory of cognition at the symbolic level should be descriptive of what is the mental competence of the subject under exploration. In the next section, we detail what we see as the principle assumptions required by models of cognition that recognize their inherantly descriptive nature.

2. Fundamentals of Descriptive Cognition

At the "symbolic timescale" (when we consider cognitive actions in .5 to 10 second chunks), some forms of cognitive behavior[1] can be meaningfully described as a series of *operators* that move the cognizer through a sequence of *states*, leading to a more desired state, namely the goal of the sequence of actions. This formulation was proposed initially by Newell and Simon (1972) and was termed the *Problem Space hypothesis*. They characterized problem solving and other phenomena as search within the problem space, which consisted of all the possible knowledge states in a particular domain, and which was traversed by the application of operators to transform one state into the next. The path so traced through the hypothesized problem space was termed a *solution path*. The problem space hypothesis suggests that we may understand problem solving behavior in terms of operator application sequences leading to the goal state.

How can we understand what gave rise to such a sequence of behaviours? We posit that the task for a functional theory of cognition is to describe the states that lead to such a sequence of behavior. Historically, this was also the task that the behaviorist tradition set for itself. They proposed that to understand the responses made by an organism, one only need know the stimulus conditions holding at the time of a response, since what the cognizer *really* knew was an association between this stimulus configuration and the response. This notion is embodied in Clark Hull's formulation of the famous rule Behavior = F (Stimulus Intensity, Drive, Habit Strength, Incentive Reinforcement), stating this association.

In line with this behaviorist credo, we accept that there will be features of the stimulus situation that will be causally related to the particular response of the organism. Further, understanding such an association is a fruitful way to describe and understand the

[1] That is, behavior sequences for which there exists a formal domain theory.

competence of the subject. This leads to our first principle of functional cognitive theory.

Cognitive behavior can be described as a sequence of State (S_i) - Operator (O_i) pairs such that there exists a function F that maps from S_i to O_i. We term this the *Regularity* principle.

However, since we are proposing a theory of cognition rather than behavior, we need to consider more than just the *external state* at the time of the behavior. We must also consider features of the *internal state*, attributes describing the state of the information processor. Internal state attributes may describe features relating to goals of the organism, partial results, past history and the state of the processor itself. Thus, the *Regularity* principle needs to be extended to include internal state such that any response of the organism is best described by the salient features of *internal state* and/or *external state*. This leads to our next assumption about function theory.

Our second principle concerns the nature of the response of the organism. Just as a cognitive theory needs the capacity to represent internal state, it also needs the capacity to represent internal operators that may change internal state. This is just another way of saying that observed behavior is not necessarily the result of a unitary internal act, but that multiple internal states and operators, in a sequence of state-operator changes, may have preceded the externally observed behavior. Much has been written about the decomposability of complex skills, generally under the rubric of hierarchical goal structured behavior. The most convienient way of describing such decomposability is by way of a grammar. We title our second principle the *Decomposition* principle.

To summarize the two principles that we have suggested form the basis of a descriptive functional-level theory of cognition, we have suggested firstly, that cognitive behavior at the symbolic level, may best be understood by knowing features of the internal and external world that were significant at the time of the behavior (the Regularity principle). Secondly, we acknowledge the need to describe internal operations. Note that we do not feel constrained to say how these operations may be implemented. Rather they exist because of task demands and external theories that the learner has internalized. They correlate with theories of task decomposition. These internal task decompositions may best be described as a grammar, and so we label our second principle the decomposition principle. These two principles in themselves do not consititute a functional theory of cognition. In fact, we label them principles since they are not falsifyable. However, they do establish the framework for a descriptive functional theory of cognition, which we consider in the next section.

2.1. A Functional Theory: Fundamental Hypotheses

We have just considered how a functional theory of cognition should determine the relevant parameters of behavior which are *Operator* = *F* (*State* $_{internal}$, *State* $_{external}$). In this section we consider some of the assumptions necessary to mechanistically determine this relationship. We do this by detailing the assumptional basis for a computational implementation of this theory called *Cirrus*.

External state is given as data, but how are we to determine internal state? Internal state is determined by a theory of domain structure. That is, the specification of internal state is a theory given by the experimenter as to how a certain competency may be achieved. In the procedural skills that we will discuss, the internal state theory describes how the top-level task is decomposed into smaller task steps. It is reasonable to assume that complex tasks, particularily those taught in formal education, have an associated method for "divide and conquer"; specifications how difficult problems can be reduced to simpler problems for which the student has already acquired competence. This task decomposition can be described as a *grammar hypothesis*, which describes the hierachical structure of the task. The grammar specifies how a given skill is decomposed hierarchically to give internal goal states.

We can decompose the knowledge implied by the grammar into two components. One component, the *goal component* is the actual task decomposition, its goal structure. This component is given by the rewrite rules of the grammar. However, grammers may be non-deterministic. That is, one left hand side (LHS) clause may expand into multiple right hand side (RHS) rules. This component may be labelled the *method* knowledge, which of a set of possible operations to apply. We can extend the grammar to be an *annotated grammar* to include this information. In the present model, we assume this information to be included in the operators, which contain general base restrictions for when they are appropriate. Adopting this approach simplifies the models constructed. For example, a grammar rule might specify that the operator *Sub1Col* could be replaced by either *ShowTop* or *Difference*. Which operator was produced by expanding this rule is dictated by conditions for applicability for the operator (i.e., that the bottom digit equals 0 for *ShowTop*).

In addition to a specification of the task decomposition rules, we need to specify an *ordering hypothesis*. This hypothesis covers the way we theorize that the skill performance is ordered, that is, what determines the order of subgoal expansion. As a model of how such a goal tree may be ordered, we could appeal to a stack model of subgoal processing (c.f., Soar, (Laird, Rosenbloom and Newell, 1984, 1985, 1986; Rosenbloom, Laird, McDermott, Newell and Orciuch, 1985) with its universal subgoaling). That is, the order of expanding subgoal nodes is given by a deterministic order.

However, we feel that the stack model is a special case of the more general *agenda* model, in which the order of subgoal expansion is non-deterministic. Roughly, the agenda ordering hypothesis states that the subgoal operators expanded from a goal are executed in an order that is not specified by the goal tree. Note that the assumption of an agenda mechanism does not imply that we hypothesize that there is an "agenda scheduling device" hardwired into the brain. Rather, we feel this is a suitably general descriptive formalism to describe how internal states may be ordered. That they need ordering is assumed from the fact that skills are described by a hierarchical task structure (decomposition principle).

Finally, we assume that states can be adequately described by *attribute-pair* representations. That is, for each state, we can specify the universe of attributes that adequately represent the complete set of properties and relations and their value. We interpret an "attribute" rather loosely; any piece of information at all can be expressed, limited only by what is considered relevant to the domain. For example, we could postulate an attribute that a cetain piece of information was in memory with a value true or false, or that certain actions had just taken place, or that something existed or had existed in the external environment. It is considered crucial to actually encode all such facts that may bear on the task domain.

All these assumptions, the grammar hypothesis, ordering hypotheses and representation assumption, come under the *decomposition* principle These allow us to specify a theory of internal operators, that we can combine with the data-given external operators. We can also specify internal state given this theory. Of course, the task theory and the hypotheses we use to operationalize it may be wrong, and later we discuss how theories may be accepted or rejected in a proposal for hypothesis testing.

However, our descriptive task is not yet complete. Even though we now have a way to specify which states go with which operators, this information is not useful. A state description contains the universe of attributes applicable to a domain which may be true or false (or valued, at that given time. The regularity principle leads us to infer that out of this universe of attributes, some will be meaningfully (perhaps causally) related to the operator application. Thus the second function of a descriptive theory of cognition must be to define a method to extract the most informative attributes from the complete universe. This is clearly an inductive task over the universe of attributes. The question is, which attributes are most highly predictive of an operator application. Thus the basis of the model is a general induction algorithm, in that the most predictive features will also be the most general features associated with an operator application. The model employs an induction formalism to determine which features of internal and external state are the most predictive. There is a parallel between the notion of concept formation, with its selection of the most general features of the instances of a concept, and the inductive task as it is performed by Cirrus. This parallel is discussed subsequently in the section on Induction and Psychological Modelling. The way in which the inductive generalization is accomplished is discussed below under Attribute Selection.

3. Induction and Psychological Modelling: a brief review

Cirrus utilizes an induction formalism known as decision trees (*Dtrees*). This formalism has a rich psychological background which is traced in this section. Dtrees had their first psychological application in the work on concept formation. The paradigm for this line of research was established by Bruner, Goodnow and Austin (1956). They studied how subjects form hypotheses about a concept on the basis of positive and negative exemplars. In these studies, concepts and thereby concept attainment was defined by critical attributes of the set of objects to be classified. Subjects were presented cards containing objects that could vary along four dimensions; number of objects, their shape, their colour and the number of borders surrounding them. Subjects had to discover a concept (i.e., cetain values on 1 or more of the four attributes) that covered the positive instances presented but none of the negative instances. To do so, they had to identify which attributes were relevant (attribute identification) and the kind of rule which connects those attributes (conjunctive, disjunctive or relational) (Anderson, 1980). Cirrus applies the notion of critical attributes of a concept formation rule to the critical features of a rule of operator application.

The combination of attributes selected was used as a rule for the application of a particular concept. As an extension of this, concepts can also be represented as a sequence of tests of the values of individual attributes. Simon and Feigenbaum (1979) employed such a formalism in their EPAM model of perceptual recognition. Tests on attributes of the stimuli led naturally to a decision tree representation, where attributes form a test at each node of a tree, whose branches are the values of that attribute. A concept is thus sorted down successive branches of the tree, until it arrives at a *leaf* of the tree. The *name* of the leaf is the label for the concept identified while the *denotation* of that name is the set of concepts sorted to that leaf. Thus the *concept* is the decision rule formed by tracing that path in the dtree. Cirrus adopts an EPAM-like tree representation, except that operator classes instead of concept classes are the labels of the leaves of the tree.

The EPAM model was an incremental model of concept formation unlike the model formulated by Hunt, Marin and Stone (1966) in their *Concept Learning System.* CLS was intended to solve single-concept learning tasks, the learned procedure then being capable of classifying new instances. They presented the decision tree process with a complete set of examples initially. For Hunt, Marin and Stone (1966) CLS was a device that discovered rules for combining previously learned concepts (attributes) to form a new decision rule. Similarily, Cirrus is presented with all the positive and negative instances at one time. Thus Cirrus is not a model of the acquisition process for operator-state pairing knowledge, rather it represents that knowledge of the problem solver as defined by their performance at that point in time. Quinlan (1983, 1986) extended the CLS model to deal with noise, n-ary attributes rather than binary, and with complex relational attributes rather than the feature-value representation used in CLS. These improvements in the Dtree formalism were adopted in Cirrus, except that the feature-value representation format was retained.

Langely, Ohlsson and Sage (1984a, 1984b) have applied the decision tree formalism to modelling of student data in a similar manner to that presented here. However there are some important differences between the two models. Their model (ACM) takes as input, the answers to a problem domain rather than the actual solution paths, and ACM determines the solution path taken to arrive at that answer. It chooses the minimum cost solution path, which is an untenable assumption for students. For example, even in the simplified world of subtraction they employ, actual students show marked variation in solution paths while arriving at the correct answer. In more complex domains such as algebra, the varience in solution paths is quite astounding, even given students with the same learning history.

ACM forms separate decision trees for each operator, in contrast to Cirrus which forms only one Dtree to classify all the operators for a particular skill. In ACM's case, attributes are given positive or negative values according to whether they were present when the particular operator was applied. States are then sorted down the tree according to whether they were a positive instance of operator application or a negative instance. In Cirrus, the decision tree is formed over the whole problem space rather than over an individual operator. For Cirrus, adequate classification is assured since discrimination is incomplete (and marked as such) unless each leaf of the dtree contains only one class of operator. In Langley et al.'s formalism, inadequate discrimination between operators is not addressed. Finally, ACM employs a different mechanism for selecting the next attribute in building the tree. This point will be discussed in more detail under Attribute Selection. However, the major distinction between the two models is in the fact that Cirrus employs real protocol data rather than hypothesized or ideal data. This motivated many of the features of Cirrus, such as the noise-filter added to the attribute-selection mechanism. Overall, the inductive formalism that Cirrus uses to extract information from the universe of state features has a long history in the psychological literature.

3.1. Attribute Selection

The essential element of the regularity principle is to extract the most informative aspects of the external and internal state that predict the application of an operator. This section details the process of selecting these features from the universe that characterizes the state.

Dtrees work inductively to make classificatory decisions. In this case the tree contains nodes for the critical attributes only, rather than specifying the universe of attributes *U*. Such a partial description may be realized by several equivalent trees. Different attribute sets may adequately classify the universe of objects, but some will be better than others. The mechanism that determines the 'quality' of the generalization is that which chooses the attribute to discriminate at sucessive roots of the Dtree. Several methods have been employed in the previously discussed models and these are compared further in this section.

Hunt et al. (1966) proposes criteria based on cost minimization, costs associated with measurement, complexity of the tree and understandability for their CLS model. The optimal attribute for the root of the tree was chosen by a look-ahead method. This was accomplished as follows.

1. A search was made for an attribute value which appeared in *all* the positive instance descriptions and never in the negative instance descriptions. The dtree-building method halted if such an attribute was found, since it completely differentiated the world of objects.

2. If this search failed, the above procedure was then applied to negative objects.

3. If both steps 1. and 2. fail, then the attribute which has the highest frequency for a given value was selected.

Such a procedure assumes that only binary attributes are encoded. Since all n-ary attributes could be recoded in a binary format, this is conceptually non-problematic, but psychologically is less plausible. The procedure used by the CLS falls into the class of *category validity* methods of attribute selection, after Rosch's (1975) notions of family resemblance. Category validity methods seeks to maximize the coverage of each critical attribute over all the positive instances (Smith and Medin, 1981). That is, a feature is high in category validity to the extent of the number of instances of the category contain that feature. Although much research supports the category validity model (Rosch and Mervis, 1975 ; Medin, Wattenmaker and Michalski; in press), the CLS procedure for attribute selection is computationally intractable, since it can involve up to 3 serial searches, which seems uneconomic if the attribute space is large.

A more psychologically satisfying method was employed by Langley et al. (1984) in ACM. Their system computes the number of positive instances matching a given test (**M+**) and the number of negative instances failing that test (**U-**), and the total number of positive (**T+**) and negative instances (**T-**). ACM calculates the sum of the proportion of instance class to total class $S = \frac{M_+}{T_+} + \frac{U_-}{T_-}$ and then computes $E = max(S, 2 - S)$. It is easy to see that an optimal test would be one that completely discriminated all the positive and negative examples. Such a test would receive a score of 1 + 1 = 2. Similarily, a test with no discrimination would match only half the positive as well as half the negative examples, and so would score one. 0 - S gives the discriminability of negated tests.

This method works in the case of single attribute selections but fails when *multiple attribute selections* are considered. To illustrate the shortcomings of the above evaluation function, we shall treat the dtree as an information source, that is, the dtree can be seen as providing a classification and hence information. Information theory tells us that an act is informative to the extent that it reduces uncertainty, and the amount of information received is proportional to the extent of uncertainty about the information to be received. Consider the following example taken from Garner (1962). If a coin is tossed in the air, then the uncertainty of that message is obviously determined by the number of possible outcomes, in this case two. If instead, a die is rolled, then the uncertainty would be greater since there are six possible outcomes. Thus, the roll of the die contains more information potential than the toss of the coin, since the outcome of the die is more uncertain.

Thus one measure of information would be the uncertainty of the event, measured by the number of outcomes for that event. This is the measure suggested by Langley et al. (1984) for ACM, since an attribute is selected as more informative if it accounts for more of the total uncertainty. (i.e., the sum of postive and negative outcomes it correctly discriminates). However, consider the case when two coins or two die are tossed in the air. Two coins have 4 possible outcomes whereas two dice have 36 possible outcomes. Further, three coins do not provide us with 6 units of information potential (i.e., states of uncertainty) but 8.

Thus a simple linear additive model of accumulated information is clearly inadequate. The number of possible outcomes of an event does not give a measure of the uncertainty of that event. The measure which satisfies the above state of affairs must be a logarithmic model, rather than the linear model employed by ACM. This is so because a logarithmic function is monotonically related to the number of outcomes and each successive event adds the same amount of uncertainty as preceeding events.

This leads to the definition of the *bit*, the basic measurement unit of information. The uncertainty U of an event, and hence its potential for carrying information, can be measured by $U = c \log k$ where **k** is the number of outcomes of an event and **c** is a proportionality constant. It is accepted practise in information theory to utilize base 2 logarithms and to define the unit of measurement so that c = 1. Thus $U = \log_2 k$. Intuitively, one bit of information enables us to decide between two outcomes, whereas one bit of uncertainty involves doubling the number of categories of an outcome.

The above formula for uncertainty assumes that each value in **k** has an equal chance of occuring. Since the probability **p(x)** of any one event occuring is the reciprocal of the total number of values in **k**, then $U = log_2 \frac{1}{p(x)}$ (Garner, 1962). Thus $U = -log_2 p(x)$ is the measure of average uncertainty when all categories of x are equally likely. When there is a discrete probability distribution for x, the average uncertainty is computed by determining the uncertainty for each item and obtaining a weighted sum of these uncertainties. This weight will be just the probability of that category occuring. This transformation thus gives us Shannon's measure of average information: $U(x) = -S\, p(x)\, log_2\, p(x)$.

The preceding formula allows us to measure how informative any particular feature of internal or external state will be in deciding the correct operator to apply. A derivation of this measure is used in Cirrus to select the next attribute, the details of which will be discussed in a later section. The point here was to demonstrate the superiority of an information theoretic measure over the linear probability method employed by Langley et al., particularily when combinations of attributes must be considered. With regard to Smith and Medin's (1981) classification of attribute classes, this measure provides a basis for attribute selection according to the *cue validity* class of models, such that each attribute seeks to maximize the discriminability of the concept. Thus, a feature is selected according to how well it differentiates the application of an operator (c.f., concept) from the space of all the domain operators. We will return to a discussion of cue validity vs. category validity models of attribute selection in discussing shortcomings of the model.

3.2. Summary

In this section, we have argued that we could understand the process of operator selection, termed *search control* if we knew what were the critical features of the *internal state* (features related to a domain theory of task decomposition) and the *external state* (the state of the problem or task). We claimed that although external features are given by the data, internal features need to be generated by a domain theory of task performance. This theory related to the decomposition of the top-level task into a subgoal hierarchy that is specified by the grammar hypothesis. Ordering of the task was scheduled by an agenda (agenda hypothesis) and the states so produced were encoded by an attribute-value representation.

To determine which of these feature were most informative, we argued that an information theoretic measure is the most appropriate to inductively select the critical features from the universe of internal and external features present at the time of each operator application. These assumptions, grouped under the grammar principle and regularity principle, allow the mechanization of determining the functional association of states and operators. In the following sections, we first discuss the actual implementation of Cirrus as a computer program, and then we illustrate its use in analysing solution path protocols from the domain of subtraction.

4. Computational Details of Cirrus

Cirrus is a multi-stage serial process with the following stages.

- Data Encoding
- State Parsing
- Attribute Encoding
- Decision Tree construction
- Application

The following section describes each of these processing stages in turn. First, some general comments will be made on the computational basis of Cirrus. Overall, Cirrus is an example of the class of *similarity-based* induction formalisms (Mitchell, Keller and Kedar-Cabelli, 1986). It uses multiple training instances (solution path protocols) and an information theoretic inductive bias (Shannon's measure of average information, Quinlan, 1983). The output is expressed as a *decision tree*. It is not limited to conjunctive generalizations, as are some popular inductive formalisms, such as Version spaces (Mitchell, Utgoff & Banerji, 1983). Decision trees can handle disjunctive concepts equally well. This merely means that one concept (operator) will occupy multiple leaves of the tree. With that definition of the class of inductive methods, we can turn to examining the inputs or parameters that need to be given to Cirrus. To illustrate some of the computational mechanisms, examples from the domain of subtraction will be employed.

4.1. Data Encoding

As noted earlier, a problem space consists of states, particularily an initial state, and operators that transform those states. The sucession of states from the initial to the goal state is termed the *solution path* for that problem. Cirrus accepts as input the solution path employed in solving a given problem. The sequence of states are displayed and the operator is asked to name the *primitive operators* corresponding the state change, transforming the data to a series of state-operator tuples. Note that the set of standard operators may need to be supplemented with various *buggy operators* if the student has non-standard primitives. Figure 1 illustrates the set of state-operator tuples that result from encoding a subtraction protocol.

INSERT FIGURE 1 ABOUT HERE

4.2. Parsing

Parsing converts the sequence of primitive operators into a goal structure according to a *grammar* that can successfully cover the input string. Figure 2 illustrates the *standard order* grammar that corresponds to the algorithm most commonly taught in schools for performing subtraction. Figure 3 shows a parse tree resulting from parsing a subtraction problem. It is clear from this diagram how a grammar can specify the goal hierachy implicit in an instructed skill. Note that the adequacy of a grammar (and hence of the domain theory) is determined by its sufficency to parse the input string, thus giving an empirical validation of the task decompostition theory.

INSERT FIGURES 2 AND 3 ABOUT HERE

Cirrus employs a bottom-up parsing algorithm with a variable constraint mechanism. The need for variable constraint parsing deserves closer attention as it illustrates some of the features of protocol data. An ordinary context-free parser builds parse trees that obey several constraints:

1. Constituents of a phrase appear in the order specified by the rule that sanctions building the phrase.
2. Constituents of a phrase abut each other; you can not skip over pieces of the string.
3. Constituents of a phrase do not overlap each other, they have to abut.
4. Constituents of a phrase have a category/type that is specified by the grammar rule that sanctions building the phrase.

These constraints are too confining for analyzing protocols, if one wishes to consider the possibility of non-deterministic order of sub-goal expansion, as implied by agenda scheduling The Cirrus parser relaxes the first two constraints to allow for such possibilities. That is, constituents need not be ordered nor abutting but they are still required to be nonoverlapping and to obey type restrictions. If the *grammar hypothesis* specifies that constituents abut or appear in a certain order, these requirements can be represented as predicates attached to the grammar rules. These predicates act as constaints such that the user can specify added requirements beyond those required by Cirrus.

As discussed previously, the *grammar hypothesis* specifies the task decomposition. This grammar must be extended to an annotated grammar to include specification of which method, if there are alternatives, is to be applied. For simplicity, the model assumes that in the case of a non-deterministic grammar rule (i.e., with multiple RHS), the operator itself encodes the information necessary to determine its applicability.

The output of the parsing stage is a goal-tree structure of state operator tuples. This tree is converted to an episodic sequence of tuples by a *tree walk* proceedure that generates the hypothesized internal states (such as stack or agenda contents, and focus of attention contents). Thus the tree walk procedure specifies the remainder of the *grammar hypothesis*, the internal informational states resulting from the specified task decomposition. In this way, the internal state is built from the primitive operator sequence specified by the external solution path protocol.

4.3. Attribute Encoding

Each operator-state tuple is converted to an operator-attribute/set tuple by running the set of all domain attribute predicates over each state description. Recall that the state description includes the actual scratch marks made by the subject as well as the hypothesized internal state given by the parsing mechanism. Additionally, some attributes have access to the previous history of the problem solver (e.g., to episodic STM traces). The set of attributes classes that we determined adequate for the domain of subtraction can be classified as follows.

- *Internal State* attributes.
  These attributes encode features of the internal state and are thus specific to the hypothesized model being explored. For example (*add/10-on-agenda, top-of-stack*).

- *Problem State Constant* attributes.
  These encode features of the problem that are constant from state to state. For example (*number-zeros, number-columns, number-blanks*).

- *Focus of Attention* attributes.
  These attributes encode features of the external state with respect to the column of the subtraction problem that the student is currently attending to. For example (*top<bottom, top<bottom-originally, bottom=blank*).

- *Rightmost Unanswered Column* attributes.
  These attributes encode features of the external state with respect to the rightmost column whose answer row is still a blank. The actual features encoded are identical to the focus of attention attributes.

- *History* attributes.
  These attributes encode previous events in the problem solving history. Most of these concern the previous operator applied, but some are flags indicating once only events. For example (*just-borrowed, borrowed-ever*).

Of course, new attribute sets need to be constructed for each task domain. In general, a task analysis is usually sufficent to establish the universe of necessary attributes. Note that some of the complexity in the above description of thef attribute set stems from the use of a *propositional* encoding of attribute-value pairs as opposed to a more powerful first-order predicate encoding. Thus attributes cannot take arguments, such as column-type. This explains why it is necessary to specify *Focus of Attention* problems distinct from *Rightmost Unanswered Column* problems. The output of this stage is a sequential list of operator-attribute tuples for each problem.

4.4. Decision Tree Construction

The final stage recursively constructs a decision tree. At each node. the most informative attribute, as defined subsequently, is selected. The operators sorted to this node are further discriminated according to their value on the selected attribute. The process halts if all the operators sorted to the node are identical (e.g.. the unary set) or if there are no attributes remaining which can discriminate the operators.

4.5. Attribute Selection

An information theoretic measure is utilized to determine the choice of the root and subsequent attributes in the Dtree. It is based on Shannon and Weaver's dictum that an event is informative to the degree that it permits one to decide among a set of alternative possibilities as to what it might have been. The justification for the following selection criteria is outlined in detail in Quinlan (1983, 1986) and is only briefly summarized here.

At any particular node, let *C* be the set of objects sorted to that node. *C* will contain the classes of operator *P..N* with *p..n* objects per class. Let *N* be the total number of objects in *C*. If a decision tree were to classify a random object at this point in the tree, it would assign the object to class *P* with the probability $\frac{p}{N}$. Thus, the information required to classify an object as one of *P..N* is then:

$$I_{(p..n)} = -\sum_{i=p}^{n} \frac{i}{N} \log_2 \frac{i}{N}$$

An attribute *A* with values $\{A_1 \ \ldots \ A_v\}$ will partition *C* into $\{C_1 \ \ldots \ C_v\}$ leaves, where C_i contains objects in *C* that have value A_i on attribute *A*. This set of objects will contain *p* items of class *P* and so on through class *N*. Thus the information that is given by the attribute *A* over the class of *C* objects is a weighted average of the total expected information, where the weight for each branch is the proportion of objects in C that belong to that branch:

$$E(A) = \sum_{i=1}^{v} \frac{p_i + \ \ldots \ + n_i}{p + \ \ldots \ + n} \, I \, (p_i \ \ldots \ n_i)$$

Thus, the gain in information given by attribute *A* is the actual information needed to generate a classification minus the expected information:

$$gain(A) = I(p..n) - E(A)$$

So, at each node in the Dtree, this quantity is computed for all the attributes that are not parents of the current node. The attribute which maximizes the gain of information at this node in the tree is selected, and the operators that have been sorted to this node are further discriminated by their values on the attribute so selected.

Two other factors must be considered in attribute selection; the influence of *noisy data* on attribute selection, and the *inflation of information content* by multiply-valued attributes (Quinlan, 1986). The source of noise could be misclassified objects on the basis of incorrect attribute value assignation or classification. Thus the tree building mechanism must know when the attribute set is unable to fully distinquish the classes of objects, and also when not to needlessly complicate the tree to classify incorrectly valued objects.

The chi-squared method (χ^2) suggested by Quinlan (1986) is employed as a noise filter. If an attibute is useful in classifying an object, then there will be a correlation of the values of the attribute with the class of objects in C. If an attribute is irrelevant to the classes of object, then the expected value p'_i of p_i will be:

$$p'_i = p \cdot \frac{p' + \ldots + n'}{p + \ldots + n}$$

This value can be utilized as the *expected value* in a normal chi-squared equation, and the resulting value checked against a stringent confidence interval ($p < .01$). In this way, attributes whose value distribution is unassociated with class distributions will not be selected, thus giving some immunity to noise in the data.

Kononenko, Bratko and Roskar (1984; in Quinlan, 1986) report that the gain criteria suggested by Quinlan (1984) is sensitive to the number of values in an attribute, favouring attributes with more values. Rather than limiting feature sets to binary attributes, Cirrus implements Quinlan's (1986) suggestion for overcoming this selection bias.

The output from Cirrus is a graphic tree representation of the decision tree. Each node of the tree represents an attribute which, at that node, carried the greatest information content in the overall discrimination of the operators. The arcs of the tree are the values of the discriminating attribute. The leaves of the tree represent the operator/s, thus the path from the root of the tree to that leaf represents the rule for applying that particular class of operators. Note that the most influential attributes (i.e., most informative) for an operators application will be high in the tree whereas the more trivial attributes, along with any remaining "noise" will be lower in the tree. This concludes our discussion of the implementation details of Cirrus. In the following section, we apply Cirrus to protocol data from the domain of subtraction, to illustrate this style of induction-based protocol analysis.

5. An Analysis of Subtraction Protocols

In this section, we apply the Cirrus method of descriptive analysis to subtraction protocol data. We will utilize as our Grammar hypothesis the standard order grammar depicted in Figure 3.

5.1. Procedure

The data analysed were 12 subtraction problems solved by P.D., a 3rd grade student. This student solved these items by paper and pencil test. In order to collect the exact writing actions, the test page was taped to an electronic tablet and PD filled out the test with a special pen. Tablet data was then converted to a sequence of character writing actions, separated by measured pauses. (Vanlehn, 1982, 1985; VanLehn and Ball, 1987) Each scratch mark made on the page defined an external state transition, hence each problem could be encoded as a sequence of state transitions. The problem set is illustrated in Figure 4. The set of problems was designed so that *Borrow* and *Borrow from Zero* procedures were tested as well. Figure 5 illustrates the sequences of external states that PD wrote in the solution of one problem.

INSERT FIGURES 4 AND 5 ABOUT HERE

These sequences of solution path protocols were encoded as cartesian-coordinate problem representations. The first stage of Cirrus sequentially displayed the external state changes of the page and queried the operator for which of a predefined set of primitive operators could produce such a state change. The set of operators unambigously corresponded to these external state changes. The list of normal and buggy operators that were found sufficient to encode a number of different subject's subtraction protocols is given in Table 1, along with the action implied by the operator's name. Each scratch mark made on the page defined an external state transition, hence each problem could be encoded as a sequence of state transitions.

INSERT TABLE 1 ABOUT HERE

The standard order grammar illustrated earlier was used to parse these external state/ primitive operator pairs. The internalized state descriptions produced by the grammar hypothesis, along with the attribute set described previously complete the input parameters to Cirrus. From this input, Cirrus produced *Dtrees* that collapse across the 12 examples fed to the program.

5.2. Results

Figure 6 diagrams the results of Cirrus's analysis of the data[2] This tree structure requires some explanation before the actual results can be discussed. The nodes of the tree propose an attribute whose values are maximally informative in discriminating operators in the subtree below the node. Attributes that are higher in the tree are more informative than those lower in the tree. Values of the attribute (often just true or false) label the links between nodes. The tree can be seen as "sorting" operators to its leaves. The operator appears in a box under the leaf. When two or more operators appear together in a box, that means either no state features were capable of discriminating the operators or senseless attributes were being used in a "last ditch" effort to seperate operators that reasonably should not be discriminated. The one case where this occured is discussed below.

This tree needs to be considered in conjunction with the goal tree (grammar hypothesis) proposed for subtraction (Figure 2). First consider the leftmost branch of the dtree. The operators *Sub*, *Sub1Column*, *ShowTop* and *Decrement* can be correctly "scheduled" by reference only to the state of the agenda. Note that primitives pop themselves from the agenda when completed, as do goals when their subgoals have all been completed. Also, *ShowTop* is assumed to have been placed on the agenda rather than *Diff* due to method information encoded in the operator, as to its applicatility. So if we know that we want to do a subtraction problem and we have not done anything yet (i.e., the agenda is empty) then we apply the goal operator Subtract. Likewise, if the subgoal to take the difference has not yet appeared on the agenda, then we want to subtract one column, and so on. What is significant here is that there is sufficent information just in what has appeared on the

---

[2] For purposes of simplifying the output, the cases where only one operator instance is sorted to a node separated in a non-intuitive way from other class members, have been treated as "noise". Most likely these few instances (3 instances out of 215 operator applications) are caused by *slips* (Norman, 1981).

agenda to schedule these operators with out reference to any external state or subgoal ordering.

INSERT FIGURE 6 ABOUT HERE

The remainder of the tree divides into two main subtrees. Note that if the goal to take the difference of two numbers (*Diff*) has been put on the agenda, then we are either going to be able to just take the difference, or we will need to borrow (called *Regroup* here). This is expressed by the middle branch of the tree, and of course, we decide between these two alternatives according to whether the top digit is less that the bottom digit in the right-most unanswered column (*RUC.T<B*). Note that this is the only feature of external state chosen by the tree. If *Regroup* has already been chosen and expanded, this means that the operator Add 10 must be on the agenda (*Add/10.ON.AGENDA*). This attribute discriminates at a higher level of the tree whether to process a column (Diff or Regroup) or complete the borrowing process.

The rightmost branch of the tree deals with scheduling the operators to complete the borrow procedure. Interestingly, the ordering of these operations (Decrement, Add/10, ScratchMark, From and Regroup) is probably the least determined by the standard order algorithm as taught by schools. This is reflected in the Dtree attribute nodes, which refer to a variety of internal states, including memory states. For PD, *Scratchmark* is the first operator executed whenever it appears on the agenda. This is sensible since this mark is a temporary reminder to decrement. If we have made a scratch mark (*Just/Scratchmark*) then we are going to *Add/10* or *Decr.* the remaining two operators left in the borrow sequence. The actual output dtree could differentiate these operators by appealing to psychologically implausible attributes. (Number of top zero's and whether the top digit was originally zero). Subjects themselves show greater variability in scheduling these two operators both between subjects and even within subjects in the same problem. Since the order of applying these does not affect the outcome of the procedure (i.e., its correctness) perhaps the dtree is just reflecting a basic Indeterminacy in this scheduling and selecting only spuriously associated state features. (Recall that the attributes lower in the tree are less important and predictive). For this reason, we show the operators together.

Finally, the remainder of the tree uses a memory trace to schedule the remaining *Add/10* operators directly after a decrement operation. These Add/10 operators are different from the previous group of Add/10 operators in that these do not occur in the context of borrowing. Alternatively, if one has not yet performed the basic operators for borrow. (*Decr. Add/10, ScratchMark*) then one executes the planning operators for borrowing; *From* and *Regroup*. These operators are just ordered by their appearance on the agenda, Regroup being performed first (which indeed it must before *From* can be executed).

Thus the analysis of this Dtree suggests that the Cirrus method of descriptive analysis tells us the information used by this student in performing the complete range of subtraction skills. It has certainly presented a coherant theory of the skills required to control the execution of a subtraction procedure (**search control**). Subtraction for multi-column problems is viewed as a procedure rather than a skill requiring search. This is reflected in the Dtree in that only one external state feature is needed to effectively sequence a complex collection of skills. The top-most ordering for this skill, derives from the task decomposition given in the grammar. This analysis also predicts that the only intermeadiate results that PD needs to store in working memory (*Just/Sratchmark* and *Just/Decr*) are involved in sequencing the more complex Borrow operation. One could speculate that the reason students have more trouble in acquiring borrowing skills is due to this extra memory demand. Interestingly, the results demonstrate that relativley unstructured ordering principles such as agenda's are nevertheless sufficient to schedule complex operator sequences with very little additional information.

In the next section, these results are related to a wider view of what can be accomplished with the use of such an automated method of analysis.

6. Discussion

We take the previous example as evidence for the utility of a descriptive theory of cognition. However, how might such results be utilized? Cirrus is seen as having three major applications. These three applications, *data reduction/analysis, student modelling* and *architectural hypothesis testing* are described with regard to the *problem space hypothesis*, using the data just analysed.

Recall that for the problem space hypothesis, problem solving and other phenomena are viewed as search within the domain of the problem space. One way we could describe the process of operator selection (*search control*) is to know which features (attributes) of the internal and external states are most predictive (and hence, presumably causal) of operator application. By knowing these features, we would know what external cues and internal states of the information processor "drove" the sequence of operators observed. Previous approaches to protocol analysis (Bhaskar and Simon, 1977; Ericsson and Simon, 1984) offered no mechanistic way that such an analysis could be conducted. The model Cirrus described in this paper offers an automated method for exploring these kinds of data analysis. In the next section, we propose three difficulties in the traditional approach to protocol analysis, and describe how Cirrus might answer these difficulties.

6.1. Limitations of Protocol Analysis of Solution Paths

As a method of enquiry, protocol analysis of solution path data has several unresolved issues that stem from the above noted need to describe the internal states of the processor and the external cues that predict operator application.

- To date, most empirical studies of the problem space hypothesis have concerned the protocol analysis of single subjects. No adequate technology as yet exists for comparing and contrasting the solution-paths of multiple data-sets. Nor has there been a means of reducing the data of solution paths so that only the essential features are considered. There is not the means for conducting hypothesis tests of group differences, a standard resouce for data comparison. This limits protocol-based studies to descriptive rather than evaluative formats. We term this the need for *data reduction/analysis*.

- The output of protocol analysis is potentially useful in many facets of student modelling, from assesment of learned procedures to intelligent tutoring system uses. However, it is at present not feasible to utilize such data in any "real time" sense, since protocol analysis involves tedious handcoding of large amounts of data, and the promulgation of many "intuitive" but untested assumptions. There is a need to automate this analysis process so that student models can be built in real-time, with a clearly defined set of assumptions. We term this the need for *automated student modelling.*

- The problem space hypothesis leaves under-determined the *ordering* principle, control processes that schedule the application of operators to achieve the goal state. There are many applicable models in the artificial intelligence literature, from planning models to blind search models, but there has been no easy way to test the architectural assumptions with respect to their fit to the data. Anderson (1983) has argued that such an issue may in fact be undecidable, since an infinitude of machines can model arbitary input/output relationships. Nevertheless, such models of internal architecture have been proposed in the literature. We feel that there is a need to evaluate such descriptive models against real solution path data. We term such an evaluation of *ordering principle adequacy.*

These three issues, of *data reduction/analysis, ordering principle testing* and *automated student modelling* are elaborated next in greater detail, along with an indication as to how Cirrus may solve these problems.

6.2. A Basis for Data Reduction.

There are two criterial ways in which solution path data may be utilized. Firstly, we wish to know which of the universe of features of a problem state are the relevant ones, that is: what aspects of the data are causal or correlated with the outcome? Secondly, how can different sets of data, once the critical aspects of the data have been found be compared.

The first problem is one of induction over the universe of attributes. That is, which attributes are most highly predictive of an operator application. This information is directly available as the result of Cirrus's analysis. Refering back to Figure 6, we can see what is important in learning how to execute a subtraction procedure. For the operators that establish the topmost goals (*Sub*, *Sub1Col*) in subtraction, only their order of appearance on the agenda is required. Similarily, if one does not take the difference in a column (i.e., *Diff* not on agenda), then one will just write down the top digit (*ShowTop*). From this, it is tempting to hypothesize that learners learn just the sequence of activities for these operators.

Obviously, the most crucial piece of information is whether top < bottom, in order to decide whether to take the difference or to initiate borrowing. Thus the borrowing procedure is executed when we have two conflicting pieces of evidence. If *Diff* is on the agenda, but top < bottom, then we must delay executing *Diff* and do *Regroup* first. Thus for this operator, learned sequences are contingent on external states. Likewise, ordering of operators may be contingent on the history of previous actions, as it is for Add/*10* or decrementing, which occur after making a scratch mark on the page.

Cirrus then, is capable of seperating from state representations what is important knowledge in skilled performance. For further analysis, one may wish to assess the extent of differences between subjects solution paths as a function of different treatment or sampling conditions. Previous protocol approaches that have tried to compare across groups have assumed a metric scaling space and rated the protocols on various dimensions within this space, assessing similarity or difference by traditional statistical methods. However, this approach is unsatisfactory for several reasons. Firstly, the measures that are inferred from the data are those that fit the subjective view of the experimenter as to what is interesting to examine. Secondly, Tversky (1977) has questioned the validity of the geometic approach in that the assumption of a metric scaling space is often untenable where judgements of similarity are concerned.

Tversky suggests instead a contrast model of analysis, employing a linear combination of common and different feature sets to assess similarity. Such a matching function can measure the degree to which two objects, viewed as a set of features, match each other. Such an analysis needs a comparison of feature set commonalities and differences rather than a computation of metric distances between points of inferred constructs. However, before such a matching function can apply, there is a prior necessary process of the extraction and compilation of the relevant set of features maximally associated with an operator application. The model Cirrus performs the preliminaries to such an analysis, by extracting the critical features from the universe of attributes.

In summary, we perceived a need to discern the critical features of external and internal state that maximally predict an operator's application. Such an analysis can firstly be employed for data reduction, so that complex protocol data can be understood by noting only the critical features. Futher, this type of data reduction could be futher employed as input to an analysis of feature similarities and differences, that would serve as a basis for within or between subject comparision. Cirrus exactly meets these requirements.

6.3. Automated Student Modelling

There is a further extension of the model beyond a theoretical basis for data reduction. If the assumptions of the model (the grammar hypothesis and the ordering hypothesis) are appropriate for human problem-solvers, then Cirrus can serve as a model for the process of operator selection, that is, a student model. This is a stronger theoretical claim than the previous one of determining the state features that "cause" an operator application. Here, we are saying that the process utilized by Cirrus to discriminate when a particular operator should be applied may be analogous to the process a student utilizes when solving a problem. That is, Cirrus in this role, is describing the actual search control mechanism employed by a problem solver. Thus, the output of Cirrus (decision trees), would be hypothesized to be control knowledge employed by the problem solver.

Domains such as subtraction, algebra and physics fall readily into the *Problem Space hypothesis* approach, where the basic manipulations constitute the operators within the problem space. The task for the student is to apply the correct sequence of operators to move from the initial to the goal state. Such learning may be termed *Search control*. How this knowledge is induced from examples, and the form of this knowledge are interesting questions. We imagine that the form of this knowledge could be described as falling along what may loosely be described as an external-internal state descriptor dimension. At one pole is the kind of search control knowledge described by Lewis and Anderson (1985) as schema abstracted from problem examples that predict when and when not a particular operator will work. These schemata contain certain problem features that are properties of the problem diagram and information contained in the problem description. They conclude that search control can be described as a correlation between surface features of the problem state (external state) and the correct rules of inference for these problems.

At the other pole are models which propose that what is learnt as search control is an internal structure which guides the sequence of operator application. As an example, VanLehn (1985) argues that problem solvers learn a goal hierarchy which specifies the order of operator application. Thus, there are internal features of the processor state (internal state) that when learnt, can guide the execution of a complex skill. Inevitably, search control will consist of a mixture of external state features and internal state features. In terms of modelling the search control acquired by a learner, it would be instructive to know what these critical features are.

Cirrus exactly describes what the balance is between search type attributes and internal procedural attributes. Given solution path protocols, Cirrus can now determine the features of the internal and external state that the student has most likely employed in reaching a solution. This data can then be employed in student modelling applications, be it evaluation of the search control strategies employed by the student for research purposes, diagnosis of incorrect operator application caused by attention to incorrect features of the problem-state in teaching situations, determining differential procedures following differing teaching histories, or building representations of the student in ICAI applications.

6.4. Evaluation of Decomposition and Ordering Principles

The two most significant free parameters of Cirrus are the ordering hypothesis and the task decomposition hypothesis. Both of these can be varied and the resultant models evaluated on a "what if?" basis. We have argued for an *agenda* ordering structure on the grounds of its generality. However, we could postulate more constrained ordering principles. For example, we may postulate that a subject uses a stack architecture. That is, the plan or procedure that the subject follows is represented following a stack regime, such that when all the sub-goals of a goal are accomplished, then that goal can be popped off the stack. Note that the stack model is a more restricted case than the *agenda* regime used here. The primary feature of stack regimes is deterministic ordering of subgoal expansion, against non-deterministic expansion order in an agenda.

The choice of ordering principle suggests the set of internal attributes to supplement the already established set of external attributes. For example, in the case of a stack architecture, relevant attributes may be *top of stack* and *depth of stack*. Finally, the set of operator states (primitives plus sub-goal operators determined by the hypothesized architecture) resulting from the parsed problem, together with the concommittant encoding of attribute sets are utilized to construct a decision tree. The decision tree represents the *control structure* that directs the execution of the procedure on the hypothesized architecture

Different architectural models and the resultant control structures may then be compared with each other. We have no formal way of contrasting the resultant Decision trees, instead we rely on the following heuristics to evaluate "goodness of fit". The ordering principle that utilizes a minimal set of features to produce decision trees of maximal simplicity and psychological plausibility is selected on the grounds of parsimony. Further bases for comparision arise from the fact that building such an "executable" model of search control demonstrates the limitations inherant in the proposed architecture. For example, the above illustrated *stack model* architecture, with its order determined sequence of subgoal expansion may be too inflexible to account for actual solution path data in a plausible manner.

The ability to manipulate such parameters and estimate their effect on the knowledge structures depicted by the resultant Dtrees seems advantageous in the construction of theoretical models.

6.5. Limitations and Problems

The heart of the Cirrus model is the inductive method employed in selecting attributes. We have noted that the information theoretic model employed falls into the *cue validity* class of induction models. That is, attributes are selected on their ability to *discriminate* between operator classes. (Here, the basis of discrimination is information content). However, we have no guarantee that this is the strategy employed by human subjects. Indeed, there is evidence that in the domain of concept formation, humans utilize *category validity* methods. Here, attributes are selected on their correlation with the set of other attributes defining a concept (Medin, Wattenmaker and Michalski, in press; Rosch and Mervis, 1975). The difference between the knowledge structures arising from cue versus category methods has yet to be explored, and an extension of the Cirrus model would be to incorporate an alternative category-validity attribute selection method.

The model uses an attribute-value encoding, a relatively weak propositional formalism for knowledge representation. More compact and powerful state descriptions may result from adopting a first order predicate state representation, particularily in expressing relational information. The extent of the limitation from a propostional representation is not yet clear.

6.6. Conclusions

In the introduction, we argued that theories of cognition that drew their sources of theoretical constraints from functional analyses of task domains should limit their hypotheses to functionally based statements about competence. In the analysis presented here, we have taken a theory of how subjects decompose the task of subtraction (the *decomposition* hypothesis) and generated a representation of a problem-solvers internal and external state constrained by this theory. From this, we have been able to extract the knowledge structures utilized by the problem solver, that adequately explain the competence demonstrated.

In doing so, we have not needed to hypothesize an *architecture* that would serve to execute this knowledge. We believe that this is an important advantage of the current method. Not only does it obliviate the need for ad hoc test and generate cycles to determine the "rule structure" of a particular task domain (as would a conventional production system analysis), but it does not build structural mechanisms based only on functional constraints. Thus there is no temptation to reify descriptive mechanisms to the status of an actual structural mechanism. Neither have we elevated the regularities found in a subject's performance to the status of a *rule*, that is, something that is explicitly represented in the neural substrate. Instead we have described observed competence, and can leave theories concerning how such competence may be represented and processed to theories that concern the structural requirements for computation.

Finally, we have sketched how such a method of analysis may be applied to interesting problems. We considered the problems of automated analysis of protocols, particularily online protocols of say, a student in a tutoring situation. We demonstrated how Cirrus could meaningfully extract a model of the student's competence. The applications of this to intelligent tutors that need to build models of that same competence are apparent. We briefly considered how the induction method of Cirrus might be used to perform group analyses of protocol data, and suggested how different domain theories might be compared.

-o0o-

References

Anderson, J. R. (1980). *Cognitive Psychology and its Implications.* San Francisco, CA: Freeman.

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89,* 369-406.

Anderson, J. R. (1983). *The Architecture of Cognition.* Cambridge, MA: Harvard Press.

Anderson, J. R. (1985). *Cognitive psychology and its implications.* New York: Freedman.

Anderson, J. R., Greeno, J., Kline, P. J. & Neves, D. M. (1981). Acquisition of problem solving skill. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition.* Hillsdale, NJ: Erlbaum.

Bhaskar, R. & Simon, H. A. (1977). Problem solving in a semantically rich domain: An example from enginneering thermodynamics. *Cognitive Science, 1,* 193-215.

Bruner, J. Goodnow J., and Austin, G.A. (1956). *A Study of Thinking.* New York: Wiley.

Ericsson, K. A., & Simon, H. A. (1984). *Protocol Analysis: Verbal Reports as data.* Cambridge, Mass.: MIT Press.

Garner, W.R. (1962). *Uncertainty and Structure as psychological constructs.* New York: Wiley.

Hunt, E.B., Marin, J., and Stone, P.J. (1966). *Experiments in Induction.* New York: Academic Press.

Laird, J. (1983). *Universal Subgoaling.* Pittsburgh, PA: Carnegie-Mellon University Computer Science Department.

Laird. J. E., Newell, A., and Rosenbloom, P. S. (1985). Soar: An Architecture for General Intelligence. in preparation.

Laird, J. E., Rosenbloom, P. S., & Newell. A. (1984). Towards Chunking as a General Learning Mechanism. In *Proceedings of AAAI-84. National Conference on Artificial Intelligence.* Los Altos, CA: Morgan Kaufman.

Laird, J. E., Rosenbloom, P. S., and Newell, A. (1986). Chunking in Soar: The anatomy of a general learning mechanism. *Machine Learning, 1*(1), 11-46.

Langley, P. & Ohlsson, S. (1984). Automated cognitive modeling. In *Proceedings of AAAI-84.* Los Altos, CA.: Morgan Kaufman,

Langley, P., Ohlsson, S. & Sage, S. (1984). *A machine learning approach to student modeling* (Tech. Rep. CMU-RI-TR-84-7). Pittsburgh, PA: Carnegie-Mellon University.

Lewis, M. W. & Anderson, J. R. (1985). Discrimination of Operator Schemata in Problem Solving: Learning from examples. *Cognitive Psychology, 17,* 26-65.

Marr, D. (1982). *Vision.* Cambridge, Mass.: MIT Press.

Medin, D.L., Wattenmaker, W.D. & Michalski, R.S. (1987). Constraints in Inductive Learning: An experimental study of Human and Machine Performance. *In press,* .

Mitchell, T.M., Keller, R.M. & Kedar-Cabelli, S.T. (1986). Explanation-based generalization: A unifying view. *Machine Learning, 1*(1), 47-80.

Mitchell, T. M., Utgoff, P. E. & Banerji, R. B. (1983). Learning problem-solving heuristics by experimentation. In R. S. Michalski, T. M. Mitchell & J. Carbonell (Eds.), *Machine Learning.* Palo Alto, CA: Tioga Press.

Newell, A. (1973). Production systems: Models of control structures. In W. C. Chase (Ed.), *Visual Information Processing.* New York: Academic.

Newell, A. (1980). Reasoning, problem solving and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.), *Attention and Performance VIII.* Hillsdale, NJ: Erlbaum.

Newell, A. (1982). The Knowledge Level. *Artificial Intelligence, 18,* 87-127.

Newell, A. & Simon, H. A. (1972). *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall.

Norman, D. A. (1981). Categorization of action slips. *Psychological Review, 88,* 1-15

Quinlan, J. R. (1983). Inductive inference as a tool for the construction of high-performance programs. In R. S. Michalski, T. M. Mitchell & J. Carbonell (Eds.), *Machine Learning.*

Palo Alto, CA: Tioga.

Quinlan, J. R. (1986). The effect of noise on concept learning. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Ed.), *Machine Learning: An Artificial Intelligence Approach. Volume II.* Los Altos, CA: Morgan Kaufman.

Rosch, E. (1975). Cognitive Respresentation of semantic Categories. *Journal of Experimental Psychology, 104*, 192-233.

Rosch, E. & Mervis, C. B. (1975). Family resemblences: Studies in the internal structure of categories. *Cognitive Psychology, 7*, 573-605.

Rosenblatt, F. (1962). *Principles of Neurodynamics.* New York, NY: Spartan.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. E. (1986). Schemata and Sequential Thought Processes in PDP Models. In D. E. Rumelhart & J. L. McClelland (Ed.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2.* Cambridge, MA: MIT Press.

Simon, H. A. (1979). *Models of Thought.* New Haven, CT: Yale University Press.

Smith, E.E. & Medin, D.L. (1981). *Categories and Concepts.* Cambridge, Mass.: Harvard University Press.

Smolensky, P. (1986). Information processing in Dynamical Systems: Foundations of Harmony Theory. In D.E. Rumelhart & J.L. McClelland (Ed.), *Parallel Distributed Processing: Explorations in the Microstructure of cognition.* Cambridge, Mass.: MIT Press.

Smolensky, P. & Riley, M. S. (1984). *Harmony theory: Problem solving, parallel cognitive models and thermal physics* (Research Report ICS-8404). California University, La Jolla.

Tversky, A. (1977). Features of Similarity. *Psychological Review, 84*(4), 327-352.

VanLehn, K. (1982). Bugs are not enough: Empirical studies of bugs, impasses and repairs in procedural skills. *The Journal of Mathematical Behavior, 3*(2), 3-71

VanLehn, K. (1983). *Felicity conditions for human skill acquisition: Validating an AI-based* *theory* (Tech. Report CIS-21). Xerox Palo Alto Research Center.

VanLehn, K. (1983). The representation of procedures in repair theory. In H. P. Ginsberg (Ed.), *The Development of Mathematical Thinking*. Hillsdale, NJ: Erlbaum.

VanLehn, K. (1983). Human skill acquisition: Theory, model and psychological validation. In *Proceedings of AAAI-83*. Los Altos, CA: Morgan Kaufman,

VanLehn, K. (1985). *Acquiring Procedural Skills from Lesson Sequences* (Tech. Rept. ISL-9). Xerox Parc.

VanLehn, K. (in press). Toward a theory of impasse-driven learning. In H. Mandl & A. Lesgold (Ed.), *Learning Issues for Intelligent Tutoring Systems*. New York, NY: Springer.

VanLehn, K. & Ball, W. (1987). *Flexible execution of cognitive procedures* (Technical Report PCG-5). Carnegie-Mellon University, Dept. of Psychology.

List of Figures

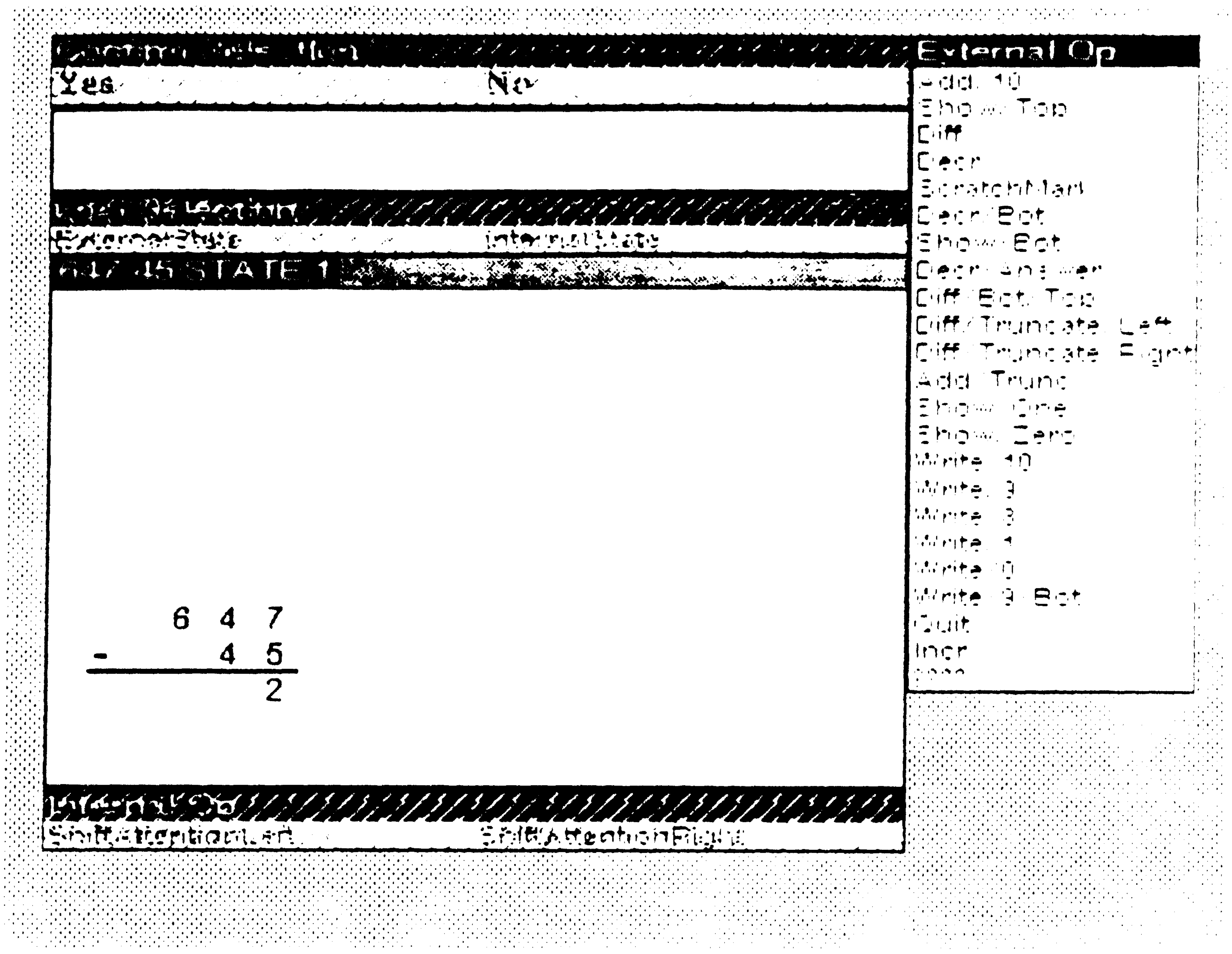

Figure 1: State Operator Encoding Window

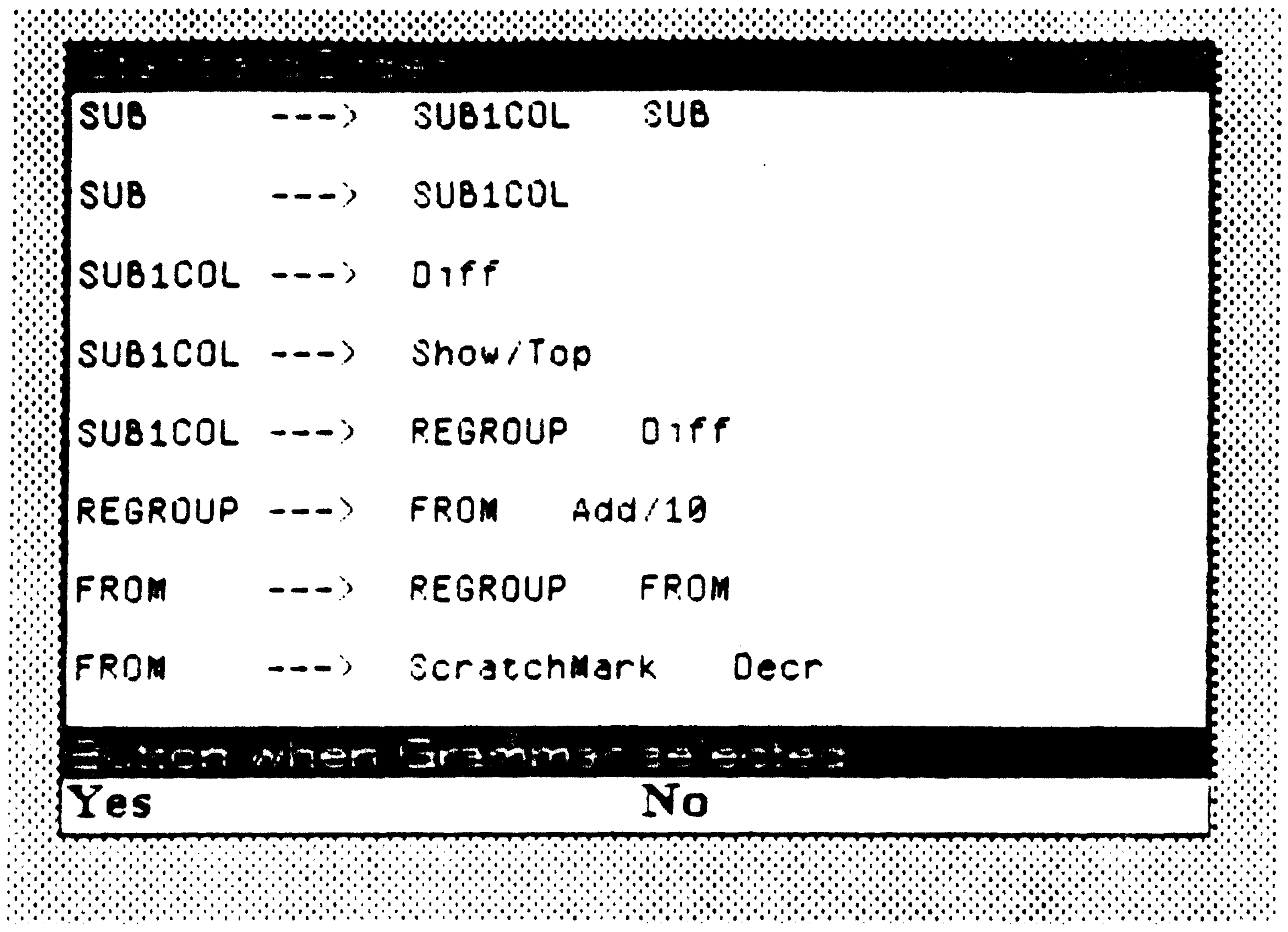

Figure 2: Standard Order Grammar for Subtraction

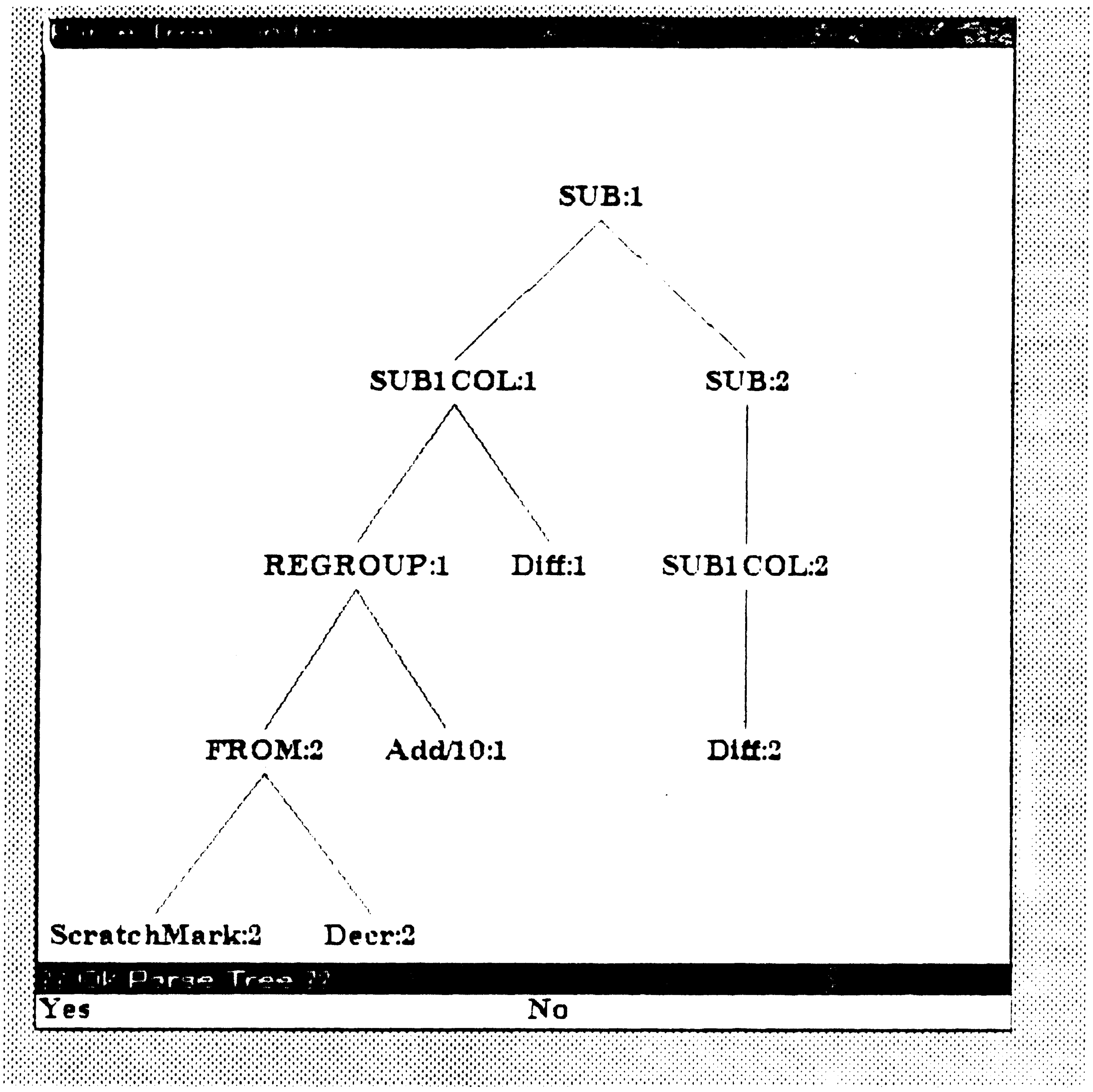

Figure 3: Parse Tree for Subtraction with Borrow

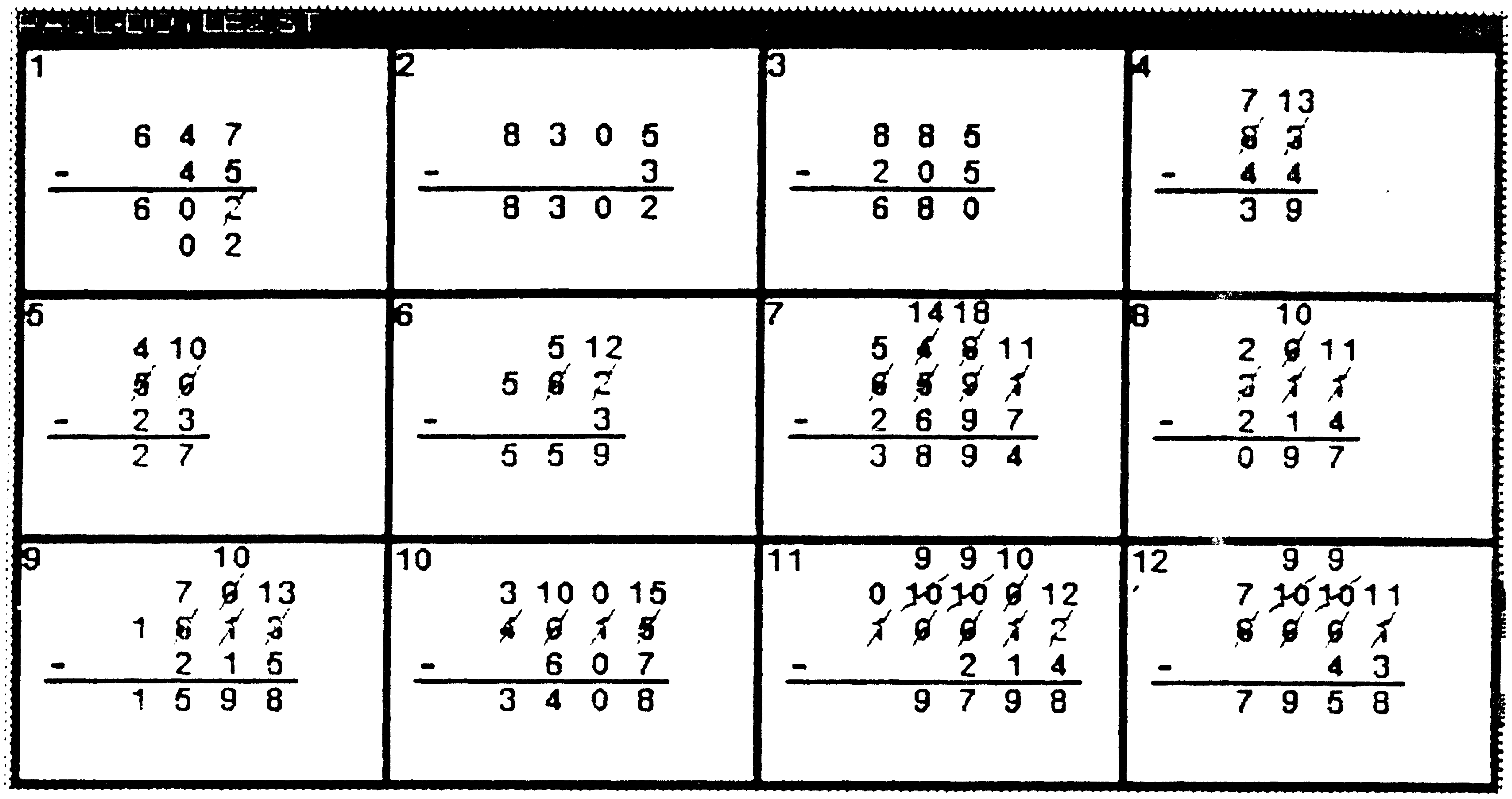

Figure 4: Subtraction Problem set for P.D.

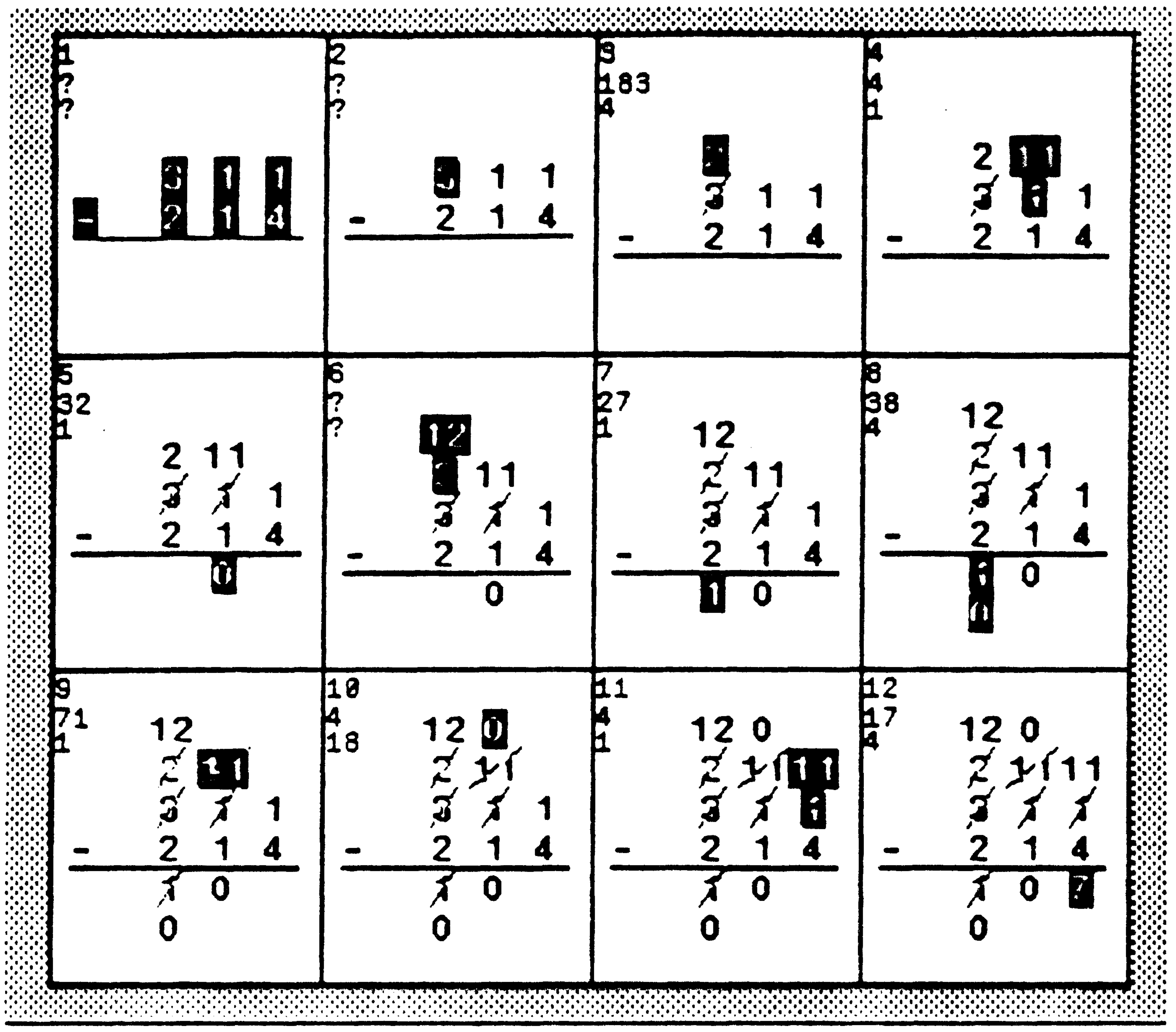

Figure 5: Sequence of External State Changes

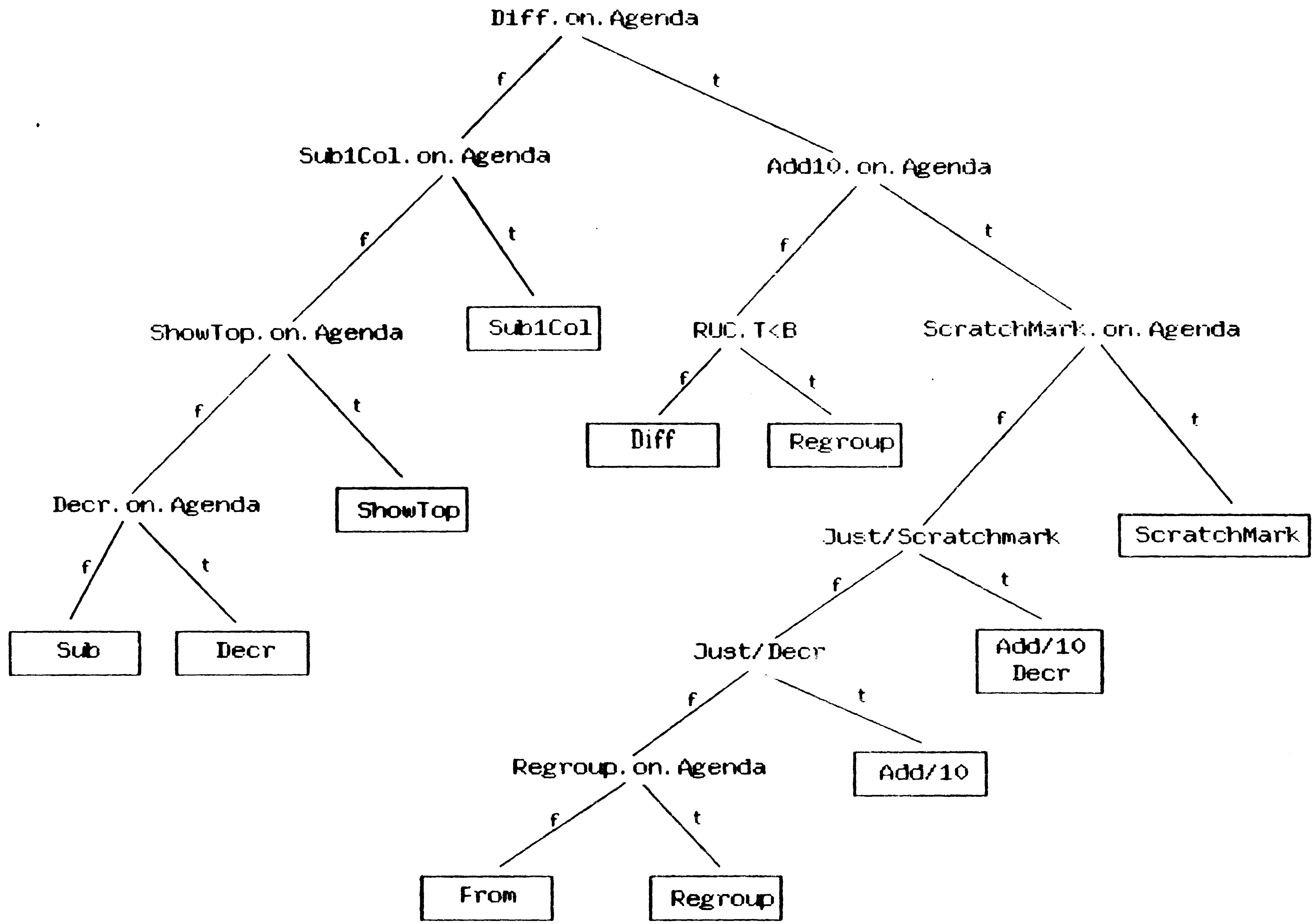

Figure 6: Decision Tree for P.D.

Dr Phillip L Ackerman
University of Minnesota
Department of Psychology
Minneapolis, MN 55455

Dr Beth Adelson
Department of Computer Science
Tufts University
Medford, MA 02155

Dr Robert Ahlers
Code N711
Human Factors Laboratory
Naval Training Systems Center
Orlando, FL 32813

Dr Ed Aiken
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr Robert Aiken
Temple University
School of Business Administration
Department of Computer and
Information Sciences
Philadelphia, PA 19122

Dr James Algina
University of Florida
Gainesville, FL 32605

Dr John Allen
Department of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr William E. Alley
AFHRL/MOT
Brooks AFB, TX 78235

Dr John R Anderson
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr Thomas H Anderson
Center for the Study of Reading
174 Children's Research Center
51 Gerty Drive
Champaign, IL 61820

Dr Steve Andriole
George Mason University
School of Information
Technology & Engineering
4400 University Drive
Fairfax, VA 22030

Technical Director, ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr Alan Baddeley
Medical Research Council
Applied Psychology Unit
15 Chaucer Road
Cambridge CB2 2EF
ENGLAND

Dr Patricia Baggett
University of Colorado
Department of Psychology
Box 345
Boulder, CO 80309

Dr Eva L Baker
UCLA Center for the Study
of Evaluation
145 Moore Hall
University of California
Los Angeles, CA 90024

Dr Meryl S Baker
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr Isaac Bejar
Educational Testing Service
Princeton, NJ 08450

Leo Beltracchi
United States Nuclear
Regulatory Commission
Washington DC 20555

Dr Mark H Bickhard
University of Texas
EDB 504 ED Psych
Austin, TX 78712

Dr. John Black
Teachers College
Columbia University
525 West 121st Street
New York, NY 10027

Dr Arthur S Blaiwes
Code N711
Naval Training Systems Center
Orlando, FL 32813

Dr Robert Blanchard
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr R Darrell Bock
University of Chicago
NORC
6030 South Ellis
Chicago, IL 60637

Dr Jeff Bonar
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr Richard Braby
NTSC Code 10
Orlando FL 32751

Dr Jomills H Braddock II
Center for the Social
Organization of Schools
The Johns Hopkins University
3505 North Charles Street
Baltimore, MD 21218

Dr Robert Breaux
Code N-095R
Naval Training Systems Center
Orlando, FL 32813

Dr Ann Brown
Center for the Study of Reading
University of Illinois
51 Gerty Drive
Champaign, IL 61280

Commanding Officer
CAPT Lorin W Brown
NROTC Unit
Illinois Institute of Technolo
3300 S Federal Street
Chicago, IL 60616-3793

Dr John S Brown
XEROX Palo Alto Research
Center
3333 Coyote Road
Palo Alto, CA 94304

Dr John Bruer
The James S McDonnell
Foundation
University Club Tower, Suite 1
1034 South Brentwood Blvd
St Louis, MO 63117

Dr Bruce Buchanan
Computer Science Department
Stanford University
Stanford, CA 94305

Dr Patricia A Butler
OERI
555 New Jersey Ave NW
Washington DC 20208

Dr Tom Cafferty
Dept of Psychology
University of South Carolina
Columbia, SC 29208

Dr Joseph C Campione
Center for the Study of Readin
University of Illinois
51 Gerty Drive
Champaign, IL 61820

Joanne Capper
Center for Research into Pract
1718 Connecticut Ave., N W
Washington, DC 20009

Dr Susan Carey
Harvard Graduate School of
Education
337 Gutman Library
Appian Way
Cambridge, MA 02138

Dr Pat Carpenter
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

Dr John M Carroll
IBM Watson Research Center
User Interface Institute
P O Box 218
Yorktown Heights, NY 10598

LCDR Robert Carter
Office of the Chief
of Naval Operations
OP-01B
Pentagon
Washington, DC 20350-2000

Dr Alphonse Chapanis
8415 Bellona Lane
Suite 210
Buxton Towers
Baltimore, MD 21204

Dr Davida Charney
English Department
Penn State University
University Park, PA 16802

Dr Paul R Chatelier
OUSDRE
Pentagon
Washington, DC 20350-2000

Dr Michelene Chi
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

Dr L J Chmura
Computer Science and Systems
Code 7590
Information Technology Division
Naval Research Laboratory
Washington, DC 20375

Mr Raymond E Christal
AFHRL/MOE
Brooks AFB, TX 78235

Dr William Clancey
Stanford University
Knowledge Systems Laboratory
701 Welch Road, Bldg C
Palo Alto, CA 94304

Dr Charles Clifton
Tobin Hall
Department of Psychology
University of
Massachusetts
Amherst, MA 01003

Dr Allan M Collins
Bolt Beranek & Newman Inc
50 Moulton Street
Cambridge, MA 02138

Dr Stanley Collyer
Office of Naval Technology
Code 222
800 N Quincy Street
Arlington, VA 22217-5000

Dr Lynn A Cooper
Learning R&D Center
University of Pittsburgh
3939 O Hara Street
Pittsburgh, PA 15213

LT Judy Crookshanks
Chief of Naval Operations
OP-112G5
Washington, DC 20370-2000

Phil Cunniff
Commanding Officer, Code 7522
Naval Undersea Warfare Engineering
Keyport, WA 98345

Dr Cary Czichon
Intelligent Instructional Systems
Texas Instruments AI Lab
P O Box 660245
Dallas, TX 75266

Brian Dallman
3400 TTW/TTGXS
Lowry AFB, CO 80230-5000

Dr Natalie Dehn
Department of Computer and
Information Science
University of Oregon
Eugene, OR 97403

Dr Gerald F DeJong
Artificial Intelligence Group
Coordinated Science Laboratory
University of Illinois
Urbana, IL 61801

Goery Delacote
Directeur de L informatique
Scientifique et Technique
CNRS
15, Quai Anatole France
75700 Paris FRANCE

Dr Thomas E DeZern
Project Engineer, AI
General Dynamics
PO Box 748
Fort Worth, TX 76101

Dr Andrea di Sessa
University of California
School of Education
Tolman Hall
Berkeley, CA 94720

Dr R K Dismukes
Associate Director for Life Sciences
AFOSR
Bolling AFB
Washington, DC 20332

Dr Stephanie Doan
Code 6021
Naval Air Development Center
Warminster, PA 18974-5000

Defense Technical
Information Center
Cameron Station, Bldg 5
Alexandria, VA 22314
Attn TC
(12 Copies)

Dr Thomas M Duffy
Communications Design Center
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr Richard Duran
University of California
Santa Barbara, CA 93106

Edward E Eddowes
CNATRA N301
Naval Air Station
Corpus Christi, TX 78419

Dr John Ellis
Navy Personnel R&D Center
San Diego, CA 92252

Dr Jeffrey Elman
University of California,
San Diego
Department of Linguistics C-008
La Jolla, CA 92093

Dr Susan Embretson
University of Kansas
Psychology Department
426 Fraser
Lawrence KS 66045

Dr Randy Engle
Department of Psychology
University of South Carolina
Columbia, SC 29208

Dr William Epstein
University of Wisconsin
W J Brogden Psychology Bldg
1202 W Johnson Street
Madison WI 53706

ERIC Facility-Acquisitions
4833 Rugby Avenue
Bethesda, MD 20014

Dr K Anders Ericsson
University of Colorado
Department of Psychology
Boulder CO 80309

Dr Beatrice J Farr
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr Marshall J Farr
Farr-Sight Co
2520 North Vernon Street
Arlington, VA 22207

Dr Paul Feltovich
Southern Illinois University
School of Medicine
Medical Education Department
P O Box 3926
Springfield, IL 62708

Mr Wallace Feurzeig
Educational Technology
Bolt Beranek & Newman
10 Moulton St
Cambridge, MA 02238

Dr Gerhard Fischer
University of Colorado
Department of Computer Science
Boulder, CO 80309

J D Fletcher
9931 Corsica Street
Vienna VA 22180

Dr Linda Flower
Carnegie-Mellon University
Department of English
Pittsburgh, PA 15213

Dr Kenneth D Forbus
University of Illinois
Department of Computer Science
1304 West Springfield Avenue
Urbana, IL 61801

Dr Barbara A Fox
University of Colorado
Department of Linguistics
Boulder, CO 80309

Dr Carl H Frederiksen
McGill University
3700 McTavish Street
Montreal, Quebec H3A 1Y2
CANADA

Dr John R Frederiksen
Bolt Beranek & Newman
50 Moulton Street
Cambridge, MA 02138

Dr Michael Genesereth
Stanford University
Computer Science Department
Stanford, CA 94305

Dr Dedre Gentner
University of Illinois
Department of Psychology
603 E Daniel St
Champaign, IL 61820

Lee Gladwin
Route 3 -- Box 225
Winchester, VA 22601

Dr Robert Glaser
Learning Research
& Development Center
University of Pittsburgh
3939 O Hara Street
Pittsburgh, PA 15260

Dr Arthur M Glenberg
University of Wisconsin
W J Brogden Psychology Bldg
1202 W Johnson Street
Madison WI 53706

Dr Marvin D Glock
13 Stone Hall
Cornell University
Ithaca, NY 14853

Dr Sam Glucksberg
Department of Psychology
Princeton University
Princeton, NJ 08540

Dr Joseph Goguen
Computer Science Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Dr Susan Goldman
University of California
Santa Barbara, CA 93106

Dr Daniel Gopher
Industrial Engineering
& Management
TECHNION
Haifa 32000
ISRAEL

Dr Sherrie Gott
AFHRL/MODJ
Brooks AFB, TX 78235

Jordan Grafman, Ph D
2021 Lyttonsville Road
Silver Spring, MD 20910

Dr Wayne Gray
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr Bert Green
Johns Hopkins University
Department of Psychology
Charles & 34th Street
Baltimore, MD 21218

Dr James G Greeno
University of California
Berkeley, CA 94720

Prof Edward Haertel
School of Education
Stanford University
Stanford, CA 94305

Dr Henry M Halff
Halff Resources, Inc
4918 33rd Road, North
Arlington, VA 22207

Janice Hart
Office of the Chief
of Naval Operations
OP-11HD
Department of the Navy
Washington, D C 20350-2000

Mr William Hartung
PEAM Product Manager
Army Research Institute
5001 Eisenhower Avenue
Alexandria VA 22333

Dr Wayne Harvey
Center for Learning Technol
Educational Development Cen
55 Chapel Street
Newton, MA 02160

Prof John R Hayes
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr Barbara Hayes-Roth
Department of Computer Sci
Stanford University
Stanford, CA 95305

Dr Joan I Heller
505 Haddon Road
Oakland, CA 94606

Dr Shelly Heller
Department of Electrical E
neering & Computer Scie
George Washington Universi
Washington DC 20052

Dr Jim Hollan
Intelligent Systems Group
Institute for
Cognitive Science (C-015
UCSD
La Jolla CA 92093

Dr Melissa Holland
Army Research Institute for
Behavioral and Social Sc
5001 Eisenhower Avenue
Alexandria VA 22333

Ms Julia S Hough
Lawrence Erlbaum Associates
6012 Greene Street
Philadelphia, PA 19144

Dr James Howard
Dept of Psychology
Human Performance Laborator
Catholic University of
America
Washington DC 20064

Dr Earl Hunt
Department of Psychology
University of Washington
Seattle, WA 98105

Dr Ed Hutchins
Intelligent Systems Group
Institute for
Cognitive Science (C-015)
UCSD
La Jolla, CA 92093

Dr Dillon Inouye
WICAT Education Institute
Provo, UT 84057

Dr Alice Isen
Department of Psychology
University of Maryland
Catonsville, MD 21228

Dr R J K Jacob
Computer Science and Systems
Code 7590
Information Technology Division
Naval Research Laboratory
Washington, DC 20375

Dr Zachary Jacobson
Bureau of Management Consulting
365 Laurier Avenue West
Ottawa, Ontario K1A 0S5
CANADA

Dr Robert Jannarone
Department of Psychology
University of South Carolina
Columbia, SC 29208

Dr Claude Janvier
Directeur, CIRADE
Universite du Quebec a Montreal
P O Box 8888, St "A"
Montreal, Quebec H3C 3P8
CANADA

Dr Robin Jeffries
Hewlett-Packard Laboratories
P O Box 10490
Palo Alto CA 94303-0971

Margaret Jerome
c/o Dr Peter Chandler
83, The Drive
Hove
Sussex
UNITED KINGDOM

Dr Douglas H Jones
Thatcher Jones Associates
P O Box 6640
10 Trafalgar Court
Lawrenceville, NJ 08648

Dr Marcel Just
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr Ruth Kanfer
University of Minnesota
Department of Psychology
Elliott Hall
75 E River Road
Minneapolis, MN 55455

Dr Milton S Katz
Army Research Institute
5001 Eisenhower Avenue
Alexandria VA 22333

Dr Dennis Kibler
University of California
Department of Information
and Computer Science
Irvine, CA 92717

Dr David Kieras
University of Michigan
Technical Communication
College of Engineering
1223 E Engineering Building
Ann Arbor, MI 48109

Dr Peter Kincaid
Training Analysis
& Evaluation Group
Department of the Navy
Orlando FL 32813

Dr Paula Kirk
Oakridge Associated Universities
University Programs Division
P O Box 117
Oakridge, TN 37831-0117

Dr David Klahr
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr Stephen Kosslyn
Harvard University
1236 William James Hall
33 Kirkland St
Cambridge, MA 02138

Dr Kenneth Kotovsky
Department of Psychology
Community College of
Allegheny County
800 Allegheny Avenue
Pittsburgh, PA 15233

Dr Benjamin Kuipers
University of Texas at Austin
Department of Computer Sciences
T S Painter Hall 3 28
Austin, TX 78712

Dr Pat Langley
University of California
Department of Information
and Computer Science
Irvine, CA 92717

M Diane Langston
Communications Design Center
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr Jill Larkin
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

Dr R W Lawler
ARI 6 S 10
5001 Eisenhower Avenue
Alexandria VA 22333-5600

Dr Alan M Lesgold
Learning Research and
Development Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr Jim Levin
Department of
Educational Psychology
210 Education Building
1310 South Sixth Street
Champaign, IL 61820-6990

Dr John Levine
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr Clayton Lewis
University of Colorado
Department of Computer Science
Campus Box 430
Boulder CO 80309

Library
Naval War College
Newport RI 02940

Library
Naval Training Systems Center
Orlando FL 32813

Dr Charlotte Linde
Structural Semantics
P O Box 707
Palo Alto, CA 94320

Dr Marcia C Linn
Lawrence Hall of Science
University of California
Berkeley, CA 94720

Dr Frederic M Lord
Educational Testing Service
Princeton, NJ 08541

Dr Sandra P Marshall
Dept of Psychology
San Diego State University
San Diego CA 92182

Dr. Richard E. Mayer
Department of Psychology
University of California
Santa Barbara, CA 93106

Dr. Jay McClelland
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Joe McLachlan
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. James S. McMichael
Navy Personnel Research
and Development Center
Code 05
San Diego, CA 92152

Dr. Barbara Means
Human Resources
Research Organization
1100 South Washington
Alexandria, VA 22314

Dr. Arthur Melmed
U. S. Department of Education
724 Brown
Washington, DC 20208

Dr. George A. Miller
Department of Psychology
Green Hall
Princeton University
Princeton, NJ 08540

Dr. James R. Miller
MCC
9430 Research Blvd.
Echelon Building #1, Suite 231
Austin, TX 78759

Dr. Mark Miller
Computer Thought Corporation
1721 West Plano Parkway
Plano, TX 75075

Dr. Andrew R. Molnar
Scientific and Engineering
Personnel and Education
National Science Foundation
Washington, DC 20550

Dr. William Montague
NPRDC Code 13
San Diego, CA 92152-6800

Dr. Randy Mumaw
Program Manager
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. Allen Munro
Behavioral Technology
Laboratories - USC
1845 S. Elena Ave., 4th Floor
Redondo Beach, CA 90277

Dr. T. Niblett
The Turing Institute
36 North Hanover Street
Glasgow G1 2AD, Scotland
UNITED KINGDOM

Dr. Richard E. Nisbett
University of Michigan
Institute for Social Research
Room 5261
Ann Arbor, MI 48109

Dr. Mary Jo Nissen
University of Minnesota
N218 Elliott Hall
Minneapolis, MN 55455

Dr. A. F. Norcio
Computer Science and Systems
Code 7590
Information Technology Division
Naval Research Laboratory
Washington, DC 20375

Dr. Donald A. Norman
Institute for Cognitive
Science C-015
University of California, San Diego
La Jolla, California 92093

Director, Training Laboratory,
NPRDC (Code 05)
San Diego, CA 92152-6800

Director, Manpower and Personnel
Laboratory,
NPRDC (Code 06)
San Diego, CA 92152-6800

Director, Human Factors
& Organizational Systems Lab.
NPRDC (Code 07)
San Diego, CA 92152-6800

Library, NPRDC
Code P201L
San Diego, CA 92152-6800

Technical Director
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Harold F. O'Neil, Jr.
School of Education - WPH 801
Department of Educational
Psychology & Technology
University of Southern California
Los Angeles, CA 90089-0031

Dr. Michael Oberlin
Naval Training Systems Center
Code 711
Orlando, FL 32813-7100

Dr. Stellan Ohlsson
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

Director, Research Programs,
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1133
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142PS
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142CS
800 N. Quincy Street
Arlington, VA 22217-5000
(6 Copies)

Office of Naval Research,
Code 11R
800 N. Quincy Street
Arlington, VA 22217-5000

Director, Technology Progr
Office of Naval Researc
Code 12
800 North Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 125
800 N. Quincy Street
Arlington, VA 22217-5000

Psychologist
Office of Naval Research
Branch Office, London
Box 39
FPO New York, NY 09510

Special Assistant for Mari
Corps Matters,
ONR Code 00MC
800 N. Quincy St.
Arlington, VA 22217-5000

Psychologist
Office of Naval Research
Liaison Office, Far East
APO San Francisco, CA 9650

Office of Naval Research,
Resident Representative
UCSD
University of California,
San Diego
La Jolla, CA 92093-0001

Assistant for Planning MAN
OP 01B6
Washington, DC 20370

Assistant for MPT Research,
Development and Studies
OP 01B7
Washington, DC 20370

Dr Judith Orasanu
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

CDR R T Parlette
Chief of Naval Operations
OP-112G
Washington, DC 20370-2000

Dr James Paulson
Department of Psychology
Portland State University
P O Box 751
Portland, OR 97207

Dr Douglas Pearse
DCIEM
Box 2000
Downsview, Ontario
CANADA

Dr James W Pellegrino
University of California,
Santa Barbara
Department of Psychology
Santa Barbara, CA 93106

Dr Virginia E Pendergrass
Code 711
Naval Training Systems Center
Orlando, FL 32813-7100

Dr Nancy Pennington
University of Chicago
Graduate School of Business
1101 E 58th St
Chicago, IL 60637

Military Assistant for Training and
Personnel Technology,
OUSD (R & E)
Room 3D129, The Pentagon
Washington, DC 20301-3080

Dr Ray Perez
ARI (PERI-II)
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr David N Perkins
Educational Technology Center
337 Gutman Library
Appian Way
Cambridge, MA 02138

Dr Steven Pinker
Department of Psychology
E10-018
M I T
Cambridge, MA 02139

Dr Tjeerd Plomp
Twente University of Technology
Department of Education
P O Box 217
7500 AE ENSCHEDE
THE NETHERLANDS

Dr Martha Polson
Department of Psychology
Campus Box 346
University of Colorado
Boulder, CO 80309

Dr Peter Polson
University of Colorado
Department of Psychology
Boulder, CO 80309

Dr Michael I Posner
Department of Neurology
Washington University
Medical School
St Louis, MO 63110

Dr Joseph Psotka
ATTN PERI-IC
Army Research Institute
5001 Eisenhower Ave
Alexandria, VA 22333

Dr Mark D Reckase
ACT
P O Box 168
Iowa City IA 52243

Dr Lynne Reder
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr Wesley Regian
AFHRL/MOD
Brooks AFB, TX 78235

Dr Fred Reif
Physics Department
University of California
Berkeley, CA 94720

Dr Lauren Resnick
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

Dr Gil Ricard
Mail Stop C04-14
Grumman Aerospace Corp
Bethpage, NY 11714

Mark Richer
1041 Lake Street
San Francisco, CA 94118

Dr Linda G Roberts
Science, Education, and
Transportation Program
Office of Technology Assessment
Congress of the United States
Washington, DC 20510

Dr Andrew M Rose
American Institutes
for Research
1055 Thomas Jefferson St , NW
Washington, DC 20007

Dr David Rumelhart
Center for Human
Information Processing
Univ of California
La Jolla, CA 92093

Dr James F Sanford
Department of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr Walter Schneider
Learning R&D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr Alan H Schoenfeld
University of California
Department of Education
Berkeley, CA 94720

Dr Janet Schofield
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Karen A Schriver
Department of English
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr Marc Sebrechts
Department of Psychology
Wesleyan University
Middletown CT 06475

Dr Judith Segal
OERI
555 New Jersey Ave , NW
Washington, DC 20208

Dr Colleen M Seifert
Intelligent Systems Group
Institute for
Cognitive Science (C-015)
UCSD
La Jolla CA 92093

Dr Ramsay W Selden
Assessment Center
CCSSO
Suite 379
400 N Capitol, NW
Washington, DC 20001

Dr Sylvia A S Shafto
Department of
Computer Science
Towson State University
Towson, MD 21204

Dr Ben Shneiderman
Dept of Computer Science
University of Maryland
College Park, MD 20742

Dr Lee Shulman
Stanford University
1040 Cathcart Way
Stanford, CA 94305

Dr Randall Shumaker
Naval Research Laboratory
Code 7510
4555 Overlook Avenue S W
Washington, DC 20375-5000

Dr Valerie Shute
AFHRL/MOE
Brooks AFB, TX 78235

Dr Robert S Siegler
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr Zita M Simutis
Instructional Technology
Systems Area
ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr H Wallace Sinaiko
Manpower Research
and Advisory Services
Smithsonian Institution
801 North Pitt Street
Alexandria, VA 22314

Dr Derek Sleeman
Dept of Computing Science
King's College
Old Aberdeen
AB9 2UB
UNITED KINGDOM

Dr Richard E Snow
Department of Psychology
Stanford University
Stanford, CA 94306

Dr Elliot Soloway
Yale University
Computer Science Department
P O Box 2158
New Haven, CT 06520

Dr Kathryn T Spoehr
Brown University
Department of Psychology
Providence, RI 02912

James J Staszewski
Research Associate
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr Robert Sternberg
Department of Psychology
Yale University
Box 11A, Yale Station
New Haven, CT 06520

Dr Albert Stevens
Bolt Beranek & Newman, Inc
10 Moulton St
Cambridge, MA 02238

Dr Paul J Sticha
Senior Staff Scientist
Training Research Division
HumRRO
1100 S Washington
Alexandria, VA 22314

Dr Thomas Sticht
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr John Tangney
AFOSR/NL
Bolling AFB, DC 20332

Dr Kikumi Tatsuoka
CERL
252 Engineering Research
Laboratory
Urbana, IL 61801

Dr Robert P Taylor
Teachers College
Columbia University
New York, NY 10027

Dr Perry W Thorndyke
FMC Corporation
Central Engineering Labs
1185 Coleman Avenue, Box 580
Santa Clara, CA 95052

Dr Sharon Tkacz
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr Douglas Towne
Behavioral Technology Labs
1845 S Elena Ave
Redondo Beach, CA 90277

Dr Paul Twohig
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr Kurt Van Lehn
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr Jerry Vogt
Navy Personnel R&D Center
Code 51
San Diego, CA 92152-6800

Dr Beth Warren
Bolt Beranek & Newman, Inc
50 Moulton Street
Cambridge, MA 02138

Dr Barbara White
Bolt Beranek & Newman Inc
10 Moulton Street
Cambridge MA 02238

LCDR Cory deGroot Whitehead
Chief of Naval Operations
OP-112G1
Washington, DC 20370-2000

Dr Heather Wild
Naval Air Development
Center
Code 6021
Warminster, PA 18974-5000

Dr William Clancey
Stanford University
Knowledge Systems Laboratory
701 Welch Road, Bldg C
Palo Alto, CA 94304

Dr Michael Williams
IntelliCorp
1975 El Camino Real West
Mountain View, CA 94040-2216

Dr Robert A Wisher
U S Army Institute for the
Behavioral and Social Scie
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr Martin F Wiskoff
Navy Personnel R & D Center
San Diego, CA 92152-6800

Dr Dan Wolz
AFHRL/MOE
Brooks AFB, TX 78235

Dr Wallace Wulfeck, III
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr Joe Yasatuke
AFHRL/LRT
Lowry AFB, CO 80230

Dr Joseph L Young
Memory & Cognitive
Processes
National Science Foundation
Washington, DC 20550