2

DTIC FILE COPY

AD-A225 800

MODELLING CONCEPTUAL DEVELOPMENT: A COMPUTATIONAL ACCOUNT OF CONSERVATION LEARNING

Technical Report AIP - 121

Tony Simon
Department of Psychology

The Artificial Intelligence and Psychology Project

DTIC
ELECTE
AUG 23 1990
S B D

Departments of
Computer Science and Psychology
Carnegie Mellon University

Learning Research and Development Center
University of Pittsburgh

029

(2)

MODELLING CONCEPTUAL DEVELOPMENT: A COMPUTATIONAL ACCOUNT OF CONSERVATION LEARNING

Technical Report AIP - 121

Tony Simon
Department of Psychology

Carnegie Mellon University
Pittsburgh, PA 15213 U.S.A.

February 28, 1990

This is the final report on the research supported by the Personnel and Training Research Program, Psychology Sciences Division, Office of Naval Research, under contract N00014-86-K-0349.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AIP - 121	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Carnegie Mellon University 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Computer Sciences Division Office of Naval Research (Code 1133)
6c. ADDRESS (City, State, and ZIP Code) Department of Psychology Pittsburgh, PA 15213	7b. ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street 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-0678
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS p40005ub201/7-4-86 PROGRAM ELEMENT NO: N/A PROJECT NO.: N/A TASK NO.: N/A WORK UNIT ACCESSION NO: N/A

11. TITLE (Include Security Classification)
Modelling conceptual development: A computational account of conservation learning

12. PERSONAL AUTHOR(S)
Tony Simon

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 86Sept15 TO 91Sept14	28Feb1990	26

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
COGNITIVE DEVELOPMENT
CONSERVATION
SOAR

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 Dr. Alan L. Meyrowitz	22b. TELEPHONE (Include Area Code) (202) 696-4302 22c. OFFICE SYMBOL N00014

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

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

This chapter presents the view that concept formation is a critical mechanism in cognitive development. To illustrate this point I present a novel explanation of the formation of Piagetian conservation concepts whose plausibility is *supported by an early implementation of a computational model. This model* relies heavily on the impasse driven performance and learning nature of the Soar architecture (Laird, Newell & Rosenbloom, 1987) and on an Explanation-Based Learning style generalization process to explain the critical transition that characterizes the acquisition of conservation knowledge. As an account of a developmental phenomenon, this model takes the first steps towards demonstrating unsupervised, incremental learning over an extended range of *input and levels of competence.*

MODELLING CONCEPTUAL DEVELOPMENT: A COMPUTATIONAL ACCOUNT OF CONSERVATION LEARNING

Tony Simon

Department of Psychology
Carnegie Mellon University
Pittsburgh PA 15213 U.S.A.

tonys@cs.cmu.edu

ABSTRACT

This chapter presents the view that concept formation is a critical mechanism in cognitive development. To illustrate this point I present a novel explanation of the formation of Piagetian conservation concepts whose plausibility is supported by an early implementation of a computational model. This model relies heavily on the impasse driven performance and learning nature of the Soar architecture (Laird, Newell & Rosenbloom, 1987) and on an Explanation-Based Learning style generalization process to explain the critical transition that characterizes the acquisition of conservation knowledge. As an account of a developmental phenomenon, this model takes the first steps towards demonstrating unsupervised, incremental learning over an extended range of input and levels of competence.

Acknowledgements

I would like to thank Allen Newell, David Klahr and Kurt VanLehn for ideas and comments about this work. A special acknowledgement goes to Richard Young for comments and discussions.

This work was supported by the Personnel and Training Research Program, Psychological Sciences Division, Office of Naval Research under contract number N00014-86K-0349.

Table of Contents

1. INTRODUCTION 1
2. COGNITIVE DEVELOPMENT AS CONCEPTUAL CHANGE 1
3. ANALYSIS-BASED LEARNING TECHNIQUES 3
    3.1. Soar 5
    3.2. Mapping EBL into Soar 6
4. A MODEL OF CONSERVATION DEVELOPMENT 7
5. ABC-Soar: An Illustrative Example 9
    5.1. Problem spaces for ABC-Soar 11
    5.2. An Example Trace. 15
6. DISCUSSION 16
    6.1. Psychological Implications 17
    6.2. The Role of Soar 18
7. Conclusions 21

1. INTRODUCTION

Conceptual change has long been regarded as an important *part of* children's cognitive development but in this chapter it is presented as a *mechanism underlying* the advancement of cognitive competence. The perspective presented here is that concepts are highly adaptive as a way to maximize the usefulness of knowledge that learners have. This is because, by categorizing instances, other knowledge associated with a relevant concept can be brought to bear to support inferences and expectations about the situation at hand. For example, a novice car buyer may be overcome with the complexity of all the diverse options of the individual cars on the market and have no idea about how to map these onto judgements of potential reliability. However, having a concept that defines reliable cars as those that are Japanese means that, merely being able to categorize a car as Japanese allows the novice to pick just one critical detail to attend to and make reasonable inferences from.

Cognitive development is a process of incrementally forming such concepts and revising them as a function of their usefulness in any given domain. Although some form of external guidance is sometimes available, concept formation in development is largely an unsupervised process. Children face the inductive task of detecting and explaining regularities in the world in terms of knowledge that they already have. In this vein, I present in this chapter a novel account of the formation of what are referred to as conservation concepts (Piaget, 1952). A part of this model is implemented in the Soar architecture (Laird, Newell & Rosenbloom, 1987) since it depends heavily on the impasse-driven performance and learning of that architecture which arises from the theory of cognition for Soar (Newell, 1990). The second key feature is the use of a variant of the machine learning technique called Explanation-Based Learning (DeJong & Mooney, 1986, Mitchell, Keller & Kedar-Cabelli, 1986) to construct generalizations about the conservation effects of transformations out of pre-existing knowledge. The account presented here turns on the fact that the way into understanding conservation is by generalizing from the effects of transformations on small numbers of objects and that this involves the use of young children's pre-existing ability to quantify small numbers and do causal attribution.

2. COGNITIVE DEVELOPMENT AS CONCEPTUAL CHANGE

There has recently emerged in the cognitive development literature a view that concepts and categories play a critical role in enabling children to manage the task of acquiring cognitive maturity. Researchers such as Susan Carey (Carey, 1985), Susan Gelman (Gelman, 1988), Frank Keil (Keil, 1989) and Ellen Markman (Markman, 1989) have all argued for and demonstrated the utility of concepts and their use by children from early childhood onwards. To see why concepts, and the categorization capability they support, play such an important role let us examine the problem facing a child trying to comprehend a novel domain. All novices encounter an apparent chaos of information in novel tasks while trying to achieve the goals that they have. By definition the novice will not have any deep understanding of this new domain and yet must determine the critical information to attend to. Further, they must learn to combine it in useful ways and to produce the appropriate responses to a range of situations. The categorization of incoming information enables the child to bring to bear any relevant existing knowledge that may help to understand the new situation so that, instead of treating each new input as totally novel, it can be responded to *as if it were like* a known case.

As Smith & Medin (Smith & Medin, 1981) put it, "Concepts give our world stability. They capture the notion that many objects and events are alike in some important respects, and hence can be thought about and responded to in ways we have already mastered". This is an important functionality for a non-expert learner in a given domain. As existing knowledge is brought to bear in comprehending new instances, it can support inferences about other aspects of the input or environment and expectations arising from the use of such knowledge in previous cases. The critical function of concepts then is that they afford generalization from old to new cases. It is this generalization that enables the novice learner to manage the task of obtaining understanding of a new domain.

The inferences that arise from such generalizations are not guaranteed correct. However, they do provide useful heuristics for reasoning and their shortcomings can guide learning when provided with feedback. This is because when inferences and expectations prove insufficient the learner can focus on gaining information that will resolve the problem. Imagine our car-buyer encountering the new range of Geo cars. The immediate reaction is that they should be discounted on the grounds of reliability because, being sold by the General Motors, an American company, they cannot be Japanese. (This itself is an inference from a concept of US car manufacturers). However, the novice is told that Geos are in fact Japanese cars that are being sold by GM but is not told if these cars are made in the US to a Japanese design or are imported directly. Nevertheless, this information has a significant effect on the novice's concepts that are relevant to this problem. S/he now knows that the definition of US car manufacturers was too general; they do market Japanese cars. What is now unclear is whether they are Japanese cars made in Japan or American cars made to a Japanese design? This uncertainty may degrade rather than improve any inferences that may be made until further knowledge is gained. This is because it is not known if there is a difference between where the car is made or who it is designed by. As a result, the status of the Japanese car (and thus reliable car) concepts are uncertain. Thus, concept formation in development is an incremental process of integrating new information with existing knowledge and modifying that knowledge on the basis of it's usefulness on future occasions. As such it can be seen as an instance of a classic inductive task (Holland et al, 1986).

The above example is also meant to show that what are referred to as concepts are really dynamic associations between knowledge which are constantly being modified rather than adhering to the classic structural view that previously dominated psychology (Smith & Medin, 1981). As new information is comprehended, associations are triggered and inferences and expectations are created from existing knowledge. Where they are incorrect the knowledge base is augmented with new information and associations are revised. This view is consonant with a recent re-characterization of the nature of concepts (Carey, 1985, Medin, 1989, Murphy & Medin, 1985) which argues that concepts are associations of pre-existing knowledge that are held together on the basis of naive theories of the interrelationships of entities rather on the more traditional basis of similarity.

Medin has shown that the similarity exhibited by category members is really a function of the "theory" that is used by the learner to group them in the first place. This "theory" is an explanation of why these entities are alike and form an ad hoc category which is usually dependent on the goal that the learner is concerned with at the time (Barsalou, 1983). Thus, objects such as children, jewelry, photograph albums and manuscripts can acquire a high degree of similarity when a goal such as *taking things out of a house during a fire* is active. All of these things share the feature of being irreplaceable, but it is not a salient common feature until the goal of avoiding their impending loss becomes active. Under these circumstances it is the irreplaceability of each entity that defines its concept membership. Adopting such a perspective means that concepts are not be stored as static data structures but rather as distributed knowledge that is dynamically accessed when the circumstances under which it was associated with other knowledge are recreated.

3. ANALYSIS-BASED LEARNING TECHNIQUES

In order to understand the conservation model that will be presented here it is necessary to examine Explanation-Based Learning (EBL) and some of its variants. The standard EBL task can be described in the following way. Given a training instance and an abstract description of a target concept, the goal is to demonstrate that the instance fits the concept description. This is not a trivial task because target concepts are expressed in more abstract terms than instances. Mitchell et al (1986) present an example in which the system is shown a positive instance of the concept "Safe-to-Stack(b,t)" where b is a block and t is a table. The concept definition is described in terms of the stackable object being lighter than the supporting object or the supporting object not being fragile while the instance describes the block and table only in terms of their location, color, density, volume etc.. Since there is no representation of lightness or fragility *per se* the EBL system must solve the problem of mapping the predicates in the instance to those in the definition. It doe this by using a domain theory which consists, in this case, of the knowledge that is needed to translate from the instance predicates to the definition predicates such as how to compute weight from volume and density and then compare two results. Thus, the system "proves" that the instance is indeed a positive example of the Safe-to-Stack concept since the box is lighter than the table. The result of this processing is an explanation of why the instance is a concept member. The explanation is produced from the execution trace of the problem solving and can be seen as a justification of the form, "because it had volume<v>, density <d> etc.". This new concept definition is said to be operational since it is stated in terms that map closely onto the language in which new instances will be described. Any other instance that fits the resultant set of conditions is also a positive example of the concept. In other words, EBL provides a goal directed justification of why any object that fits a certain description would be considered a member of the given concept.

EBL was never intended as a model of human concept formation and, in the form described, too many limiting assumptions based on the domain theory are made. Since the introduction of EBL, considerable effort has been put into the adaptation these systems so that they will be able to augment their own domain theories (Pazzani, 1988), work with incomplete domain theories, combine EBL and similarity-based learning, and many other changes aimed at using EBL in more realistic situations (Segre, 1989). At the same time, psychologically motivated versions of analytical concept formation have been developed to explain human performance. These systems generalize using a weak method such as causal attribution or analogical reasoning instead of a domain theory . The clearest statement of such an approach is EXPL (Lewis, 1988). In the EXPL system, a known procedure from a human-computer interaction task is used to generate new procedures for new goals. Lewis argues that, in the absence of domain knowledge, users are able to subject the known procedure and its outcome to an analysis whereby causal links are created between components of the input procedure and aspects of the computer's response. For example, touching a triangle on the screen and pressing the Delete key, and observing the disappearance of the triangle leads to the attribution that touching the triangle identifies it as the target of some action and pressing Delete causes the system to execute that action. These causal links are created by the application of a small set of psychologically well-founded causal reasoning heuristics (Shultz, 1982). New procedures are created by the "synthetic generalization" process of selecting targets and actions from a knowledge base of sub-units such as the objects "triangle" and "square" and the actions "Delete" and "Move". EXPL can thereby infer that the creation of a "move-square" procedure is constructed by replacing the "triangle" with the "circle" in the Touch(x) sub-unit and replacing "Delete" with "Move" in the <Action>(x) sub-unit.

The model of conservation learning that I will present is implemented in the Soar architecture which will be described next. It adopts an EBL style of analysis to learn and generalize the conservation effects of various actions. I will therefore also describe the mapping of EBL into Soar.

3.1. Soar

Soar was created as an "architecture for general intelligence" (Laird, Newell & Rosenbloom, 1987) and exists as the basis of a "unified theory of cognition" (Newell, 1990). All of Soar's activity is implemented as search in problem spaces. Although currently the knowledge for these spaces is provided by an external agent, eventually Soar will be able to acquire this knowledge for itself by interaction with its environment. (Indeed, this project itself constitutes one of the first steps down that road). During processing, any difficulty or impasse that arises results in the automatic creation of a subgoal where processing is concentrated on the resolution of that problem. Impasses can be caused by a range of situations, from not knowing which of a set of possible operators to select, to not knowing how to apply an operator, or simply having no next step in problem solving be suggested. The successful termination of subgoals by resolving impasses leads to the automatic creation of "chunks" which are rules of the form "If X then Y" and which summarize the processing in the subgoal. The next time a similar situation is encountered the result will be accessed from long term memory without the need for the subgoal processing to be repeated. The situation need only be similar for the chunk to apply because, in creating the chunk, Soar generalizes out only the critical features that were used in the creation of the result and only these, rather than all of the information present become the conditions for the new chunk. In this way, Soar systems exhibit a gradual shift from deliberate, impasse driven learning to automatic recognition-based action. The chunking mechanism is hypothesised to be sufficient for all learning (Rosenbloom, Newell & Laird, 1990) in a cognitive system.

3.2. Mapping EBL Into Soar

The mapping of EBL into the Soar architecture is very straightforward and is described in detail elsewhere (Rosenbloom & Laird, 1986). To review, EBL systems typically comprise four elements. These are the target concept, the training instance, the domain theory and an operationality criterion to ensure that generalizations are usable. The goal or target concept is a rule which defines the concept to be learned. In the Safe-to-Stack example used earlier, the aim is to learn when it is safe to stack one object on top of another. The rule is stated in the form:

```
If y is not Fragile OR x is Lighter-than y
Then it is safe to stack x onto y.
```

The training example is an instance of the concept to be learned stated in terms of predicates such as volume, density, color etc. which do not map directly onto the target predicates such as *Lighter-than*. The domain theory is the knowledge which is sufficient to prove that the training instance fits the definition of the target concept. In this case it is the requisite rules and facts for determining *Lighter-than* from volume and density. The operationality criterion restricts acceptable "rewritings" of the target concept. Here an "operational" definition would be one that defines concept membership in terms of specific volumes and densities.

Rosenbloom and Laird describe how these components can be mapped into the Soar architecture. The target concept is simply a goal to be achieved. The training instance is the representation that exists on the state when that goal is generated. The operationality criterion is that the concept must be expressed in condition predicates that existed before the creation of the goal. The domain theory is a problem space where the satisfaction of the goal can be attempted and where the rules and facts correspond to operators in the problem space. The goal is to implement an operator, Safety?(x,y), that cannot be applied on the basis of existing knowledge. This creates an impasse and a problem space, called *Safe*, is selected for the subgoal to attempt to implement the operator (thereby operationalizing the Safe-to-Stack concept). It does so by examining the state and working until is augmented with enough information to determine whether it is safe to stack one object onto the other; in other words, until the Safety? operator can be applied. The trace of operator applications in the subgoal constitutes the explanation as in standard EBL systems. Generalization is carried out by the backtracing mechanism which determines conditions for the chunks that Soar creates from subgoals. For a fuller description of this process the reader is referred to Rosenbloom and Laird's original article (Rosenbloom & Laird, 1986).

4. A MODEL OF CONSERVATION DEVELOPMENT

As already mentioned, the domain of conceptual development that will be used to illustrate the current model is that of Piagetian conservation concepts. This topic was a major preoccupation of the field of cognitive developmental psychology for a long time mainly because one of the central tenets of Piagetian theory (Piaget, 1970) is that the acquisition of conservation concepts is a critical milestone in the child's development of mature conceptual capabilities. Conservation concepts can be defined as the understanding that, in the face of irrelevant transformations (such as spreading a row of objects) some aspects of those objects (such as numerical value) remain invariant.

In a typical experiment on number conservation a child is shown two rows of objects generally with the same number of objects in each row. The objects are usually lined up in 1-to-1 correspondence. The child is asked if the two rows have the same number of items in them. Having agreed, one of the rows is transformed (usually by lengthening or shortening) and the child is asked the "conservation question", whether the rows still have the same number of items in them. Children who consistently answer that the two rows do have the same number, and can explain why, are said to have acquired the conservation concept with respect to number. Those who are misled by the perceptual features and say that the longer row has more objects are classified as non-conservers.

It is generally agreed that the critical shift in the development of conservation concepts is for the child to stop attending to the details of the objects themselves and to focus on the transformation that was applied to them. Implicitly, at least, this has been thought of as the acquisition of a new behaviour or of the children learning to no longer respond in one way and to replace this behaviour with another, more reliable one. The explanation presented here is very different. It is a failure- or impasse-driven account which says that the children attend to the transformation from the start but have no knowledge based on that input alone with which they can produce a judgement in terms of number. In order to answer the question they are forced to use the only behaviours that they do have that are relevant to making numerical judgements. These are counting where that is possible and estimation where it is not. Therefore, children's initial response to conservation questions is to attend to the specific details of the materials in order to objectify whether any change has occurred. As more is learned about the effects of actions and this is generalized to other actions and other materials then this behaviour becomes unnecessary and a response about number can be made merely by observing and categorizing the transformation and recalling its inferable numeric effects.

This corresponds to the learning of what functions as a class of operations that affect a given dimension (such as quantity) where that class contains members that are related by virtue of the goal of the task. In other words, it is the formation of an ad hoc category of "number-preserving actions". Once such a concept has been learned, it can be applied to objects of any quantity to assess conservation. Many experiments have been carried out on a wide range of conservation concepts for number, mass, area, volume and even existence (Gelman & Baillargeon, 1983). Concentrating on conservation of number and other kinds of quantity one can observe that a set of regularities exist in the literature. By a regularity we mean a finding that is consistently reported and for which there is little or no disconfirming evidence. The four main regularities are presented below.

1. Young children in the 3-6 year-old age range are able to obtain specific quantitative values for small sets (e.g. up to about 5) of objects e.g. (Frye, Braisby, Lowe, Maroudas & Nicholls, 1989).
2. Young children in the 3-6 year-old age range are unable to obtain specific quantitative values for larger sets of objects e.g. (Gelman & Gallistel, 1978).
3. Children who have not acquired the conservation concept nevertheless can still correctly answer the conservation question when they can obtain a specific quantitative value for objects concerned e.g. (Siegler, 1981).
4. Children who have not acquired the conservation concept cannot correctly answer the conservation question when they cannot obtain a specific quantitative value for objects concerned e.g. (Halford & Boyle, 1985).

These regularities support another part of the current explanation of the development of conservation which points to the special role of small numbers in the process of acquiring knowledge about invariance or conservation. I shall call this explanation the Quantification model (or Q-model). The Q-model says that children learn quantity conservation by generalizing the effects of transformations on countable arrays into the classes of quantity-preserving and quantity-modifying transformations. This comes about in the following way. Given the goal of assessing the numeric effects of transformations, the child is unable to respond at the level of transformation and so is forced to rely on some objective means of judging if the number of the objects has changed. For small children concerned with small numbers the most reliable means of doing this is counting[1]. Thus the child counts the specific number a given set of objects and then recounts them after a transformation has occurred (such as spreading out a row of counters or building a tower with a pile of blocks). Then, if the two values match, a plausible inference can be made that the quantity was not affected by the transformation concerned. The Q-model assumes that the child forms conservation concepts by "explaining" such outcomes in terms of the actions and by

[1] I use the term counting to include "subitizing", a capability to perceptually apprehend small numbers.

generalizing over the specific objects and amounts involved in the transformations. Eventually a transformation, such as spreading, will be represented without any reference to specific objects or agents and it will always be expected to have a quantity-preserving effect (Klahr, 1984). As a result the quantitative effect of the analyzed actions can be easily determined, even for material where a specific quantitative value cannot be directly computed.

Further support for this model comes in the form of two more regularities that can be observed in the conservation literature.

5. Correct answers to conservation of quantity are achieved on tests of discrete stimuli (e.g. buttons, coins, shapes) before tests of continuous stimuli (e.g. liquids) e.g. (Siegler, 1981).
6. Training conservation on discrete quantities alone transfers to continuous quantities once the conservation concept has been formed e.g. (Vadhan, 1984).

Both of the above regularities provide evidence for the claim that the learning and generalization of the effects of various transformations takes place only in cases where the materials are countable. Then, once the generalized conservation knowledge has been learned, it transfers to non-countable materials like liquid. Thus we can see that there is a special role for small numbers in the process of coming to appreciate invariance in the face of certain transformations. This is because numbers are objective symbols that can be manipulated in lawful ways, such as counting, adding and subtracting. This is not true of category labels such as "long" or "short" that are the result of estimation. Therefore, numbers are the probably the first way of assigning objective values to things that children have. Secondly, young children have the ability to count small numbers and the capability of plausibly assigning causes to outcomes by selecting actions that are in some way related. In other words children have all that is needed in order to make simple invariance judgments although they may have never used these competencies together for that purpose. Based upon this argument I have created a model of inductive explanation-based learning which answers conservation questions and generalizes the results.

5. ABC-Soar: An Illustrative Example

In order to illustrate the feasibility of the impasse-driven account of conservation learning I have constructed ABC-Soar, a demonstration model of a part of this learning process. The task is the number conservation test described above. It is a primary hypothesis of the model presented here is that the content of the learning is a direct consequence of the tasks that the learner was engaged in when comprehending incoming information. However, it is not currently known precisely what everyday activity that children engage in is responsible for the formation of their numerical understandings. The solution that chosen to enable this modelling to be done was to simulate the learning that has been documented in experiments that train children in the acquisition of number conservation concepts. These experiments provide information about what the children's initial competences are. Further we know just what experiences they are exposed to and what kind of post-training performance was possible. The simulation reported here is only limited in its coverage of the development of conservation concepts but the existing system does demonstrate some of the major principles and provides a principled basis for extrapolation of a scaled-up version that will closely model such training studies. To re-iterate the above point, the developmental account arising from this model does not depend on it being the result of training. This was just a useful to begin to address the question.

ABC-Soar adopts the same methodology to creating and inductive version of EBL as that of Lewis (Lewis, 1988). In place of a complete domain theory it is provided with two competences that have been shown to exist in children who participate in conservation experiments; small number quantification and causal attribution. In effect these are two incomplete domain theories. The ability to count collections of up to five objects is justified from two sources. Apart from the support from the regularities presented above, there are a number of training studies where an explicit count of collections of such a size are executed by the children being trained (Gelman, 1982, Halford & Fullerton, 1970). Behind the inclusion of the domain-independent ability to causally attribute outcomes to actions based on a set of heuristics is much evidence that such causal reasoning abilities are not only well within the capabilities of young children but also that they are an important component of their learning activities (Shultz, 1982).

Thus, ABC-Soar is provided with two problem spaces which implement naive domain theories. Another way to view this knowledge is as a kind of weak method that provides a bootstrapping capability so that a more sophisticated domain theory may be learned. In other words, the ability to count transformed displays and then attribute the cause of numerical outcomes to given actions can produce simple beliefs about the numerical effects of those actions to be acquired. As the inadequacies of that knowledge are discovered, the learner is forced to revise it to take into account the subtleties of numerical invariance. Having learned new facts and rules about the effects on number of a class of actions, the naive problem space operations will be superceded by the new, more sophisticated knowledge.

5.1. Problem spaces for ABC-Soar

The problem spaces that implement Analysis-Based Concept Formation in ABC-Soar are shown in Figure 1. It depicts the operators that exist in each of the spaces and also which of the other spaces implement operators whenever the problem solver has insufficient knowledge to apply them directly. The model does not simulate the entire procedure of the training experiment but focuses on two kinds of learning. One is the initializing and generalizing of concepts that describe the observed transformations. The other is the learning of the inferable effects on number of those actions. The model simulates the processing of the child from the point where the experimenter performs a transformation on one of the two rows of objects. It is assumed that the child has already quantified the two rows of objects and agreed that they have the same number of objects in them.

Figure 1 about here

The **Perform** space has the goal of producing a judgement about the numerical effect of an observed transformation and has three operators. *Observe-Transformation* produces an internal representation of the transformation performed on one of the rows. It is proposed when the external transformation is presented but does not have an internal representation on the state in the **Perform** space. The representation that is created for the transformation is a modified form of that used for operators in the STRIPS problem solver (Fikes & Nillson, 1971). The action representation has ADD and DELETE lists which represent the explicit or perceptually available effects of the action as well as a "subject" slot (who did the action) and a "object" slot (what was it done to). For example, the ADD list of a spreading operation will state that it increases the length of a transformed row while the DELETE list will represent that **same-length**(as the other row) no longer applies. Such a representation enables the non-visible, inferable effects (such as numeric) to be added to these lists selectively depending on the context in which the action is being comprehended.

The *State-Numeric-Effect* operator is applicable when a result concerning relative numerosity is represented on the state in the **Perform** space. When this is so, the operator reports whether or not the number in the transformed row is the same as the untouched row. This satisfies the top goal of ABC-Soar and terminates processing. In a more complete version of the model, processing will not terminate here but will use feedback to assess the accuracy of these judgements and revise its understanding of the transformations as best it can.

The *Comprehend* operator is applied to each new input that is presented. Its effect, after learning has occurred, is that the system is able to automatically bring to bear all of the knowledge that it has related to understanding that input *in the context in which it is experienced.* This generally involves the creation of a data structure that relates the new input to previously learned knowledge and, in most cases, to produce expectations as a result of that input in the current context. Other language systems e.g. (Small, 1979) and non-language systems (Schank, 1982) operate with a similar expectation-based knowledge access scheme. If a new piece of input is not understood then the comprehension operator will fail to apply and, in the resulting subgoal, Soar will apply knowledge in order to comprehend that input in the current situation. Thus, if observed transformations are understood in terms of their numerical effects, that knowledge will be accessed by applying the comprehension operator and thus allowing the *State-Numeric-Effect* operator to apply. If the transformation is not understood then the comprehension operator will create an impasse and ABC-Soar will attempt to learn sufficient information to enable it to apply enabling a numeric effect to be stated for the problem.

To implement a *Comprehend* operator in the context of the conservation task, ABC-Soar first attempts to Categorize the observed transformation and then retrieve the numeric effects of the action. Given sufficient knowledge, this processing is carried out in the **Comprehend** space. The *Categorize-Input* operator uses the description of the represented transformation to access concepts such as "spreading-action" and categorizes the transformation as a member of one of these concepts. Problems that arise in the comprehension process create impasses and are dealt with by operators in the **Generalize** space. Once the represented transformation (called the "instance" hereafter) has been categorized the *Same-amount?(number)* operator becomes applicable. If known, this operator reports the numerical effects of the kind of action that the instance has been categorized as. This knowledge completes the implementation of the *Comprehend* operator and enables the *State-Numeric-Effect* operator to apply in the **Perform** space.

The failure to return a numeric effect for the instance creates a further impasse where the **Explain** space for number will be selected for the subgoal. There are five operators in this space and they constitute the naive domain theory which ABC-Soar uses to compute whether the transformation instance conserves number or not. The *Link-Input-to-Outcome* operator is applicable when there is an outcome represented on the state that is not connected to an observed action. This operator cannot be implemented in the **Explain** space as the requisite knowledge exists in the **Cause?** space. Having been implemented, the *Link-Input-to-Outcome* operator creates links based on casual attribution between actions on objects and new (outcome) representations of those objects. The *Quantify* operator is applicable if there is no numerical outcome represented on the state. The current implementation of the *Quantify* operator oversimplifies the complexities of the counting process. However, it serves to illustrate the role counting in the domain theory. The *Quantify* operator produces a numerical outcome (the number of objects that exist) for the transformed row. Both the *Quantify* operator and the *Link-Input-to-Outcome* operator are default operators. This means that, although applicable, they will only be used if there is no other knowledge available which implements the *Same-Number?*, and ultimately the *Comprehend* operators. As more is learned about such transformations by the system, these operators will become superceded and the more deeply analyzed knowledge will be directly accessed by the higher level operators.

The **Cause?** space has a single operator, *Determine-Causal-Link*, which links actions and outcomes. This is done by attributing cause on the basis of a set of causal attribution heuristics the main of which is the similarity heuristic. This says that if an action and an outcome share some feature (such as one of the rows in the array) then it is likely that the outcome was caused by the action.

The *Greater-Than?* and *Less-Than?* operators are applicable when there is a numerical outcome for the transformed row and a value for the non-transformed row to compare it to represented on the state. These operators simply return true or false values for whether the transformed row has a numerical value that is greater or less than the untransformed row. The final operator is the *Effect?* operator which is applicable when the greater? and less? results have been computed. It operationalizes the *Same-Number?* operator by producing a result for the subgoal. In other words, by checking whether a causally linked effect of the transformation has increased, decreased or preserved the number of the transformed row, this operator determines whether the effect of the transformation is number-conserving or not. Successful termination of the subgoal that causes a chunk to be built. This chunk determines the conditions for the operationalized definition of a naive conservation concept that ABC-Soar will be able to use in response to a small range of similar transformations.

The **Generalize** space, which has three operators, is selected in response to impasses from the *Categorize-Input* operator whose failure to recall an appropriate concept for a given input can be seen as a signal that the input is a member of a novel concept. Otherwise, a concept definition that was appropriate would have been recalled even if some revision might have been required to subsume the current instance. However, if no existing definition matches then the *Initialize-Definition* operator is selected. This creates a new concept definition which is the same as the current instance except that it is provided with a "name" (which is a gensym created by the system) and a "constant" attribute and value. The "constant" slot functions to describe what the concept *is a concept of.* It is necessary because ABC-Soar is an unsupervised concept learner in the sense that presented instances are not preclassified as positive or negative instances of given concepts, such as "spreading action". Thus, if the system did not distinguish by some means which concept an instance was a candidate member of, the generalization process would indiscriminately generalize existing concepts using inappropriate instances. The functions of the two remaining operators, *Compare* and *Generalize* demonstrate why.

The *Compare* operator is applicable when there is both an instance represented on the state that has not been categorized and a recalled concept definition that matches in some way. The *Compare* operator compares all of the values of the attributes that are shared by the instance and definition except the one(s) listed as the "constant". If any of the values are different then the *Compare* operator lists the relevant attribute as requiring generalization. When this has been done the *Generalize* operator is selected and variabilizes the value slots of all the listed attributes. Thus, a definition whose constant is value of the "ADD" list (e.g. the effect of making an object longer) then the ADD slot will not be variabilized. The definition might state that "making-longer" actions are ones done by a person (the "subject" slot) to rows of five buttons (the object slot) while the instance states the subject as a teddy bear (the agent in some conservation experiments) and the object as a row of three cookies. The result would be that the "making-longer" concept would be variabilized to the point where it states that "making-longer" is where length is added by anyone to anything. This approach is similar to many existing inductive generalization schemes in machine learning (Mitchell, 198?).

How does a concept definition have its "constant" value decided? One way is a weak method akin to the psychological phenomenon known as the "orienting response". This states that an organism will orient its attention to aspects of its environment that change. In the case of the transformation representations in the conservation domain, the default value of the constant slot is the same as the ADD list. In the standard conservation task, all that changes in the array is the dimension transformed by the experimenter's actions (to which the child's attention is deliberately drawn) so this is selected as the default referent of the concept. All future actions that have the same transformational characteristic (e.g. increasing length) become candidates for categorization on that basis and for generalization of all values but that one. Another basis for deciding the constant of an instance is to index it by the goal of the task that the learner is engaged in. In the context of a number conservation task there may be a number of operations that involve decreasing and increasing length but for which it is clear that objects are being added. Since the goal of the task is to assess the effects of actions on number, the initial grouping of these actions is likely to be in terms of their numeric effect and not simply the visible physical effects.

5.2. An Example Trace.

Figure 2 shows an example trace of part of the process of the formation of a simple conservation concept. The trace begins after a row of five crosses and five circles have been counted and agreed as having the same number. The goal is to state the numeric effect of an observed transformation. Having observed a transformation which is represented as the experimenter having lengthened a row of five crosses (#1), the *Comprehend* operator is selected (#2). Since there is no existing knowledge of the numeric effects of spreading such a row the operator fails to apply and in the resulting subgoal the *Categorize-Input* operator (#3) is selected in the **Comprehend** space. The spreading transform is novel for ABC-Soar so this operator also fails and in the **Generalize** space the *Initialize-Definition* operator (#4) applies. It creates a concept of spreading actions which is defined as actions carried out by a person on five crosses. The creation of such a concept enables the *Categorize-Input* operator to successfully apply in the **Comprehend** space. The result of this application is to augment the instance with the name of the newly created concept. Chunking enables ABC-Soar to learn the categorization of a transformation with the features of the one currently represented.

Figure 2 about here

Now that there is a categorized instance on the state, the *Same-Number* operator is selected (#5). However, an impasse is created by the lack of known effects of the transformation. This causes the **Explain** space to be selected in the subgoal. The *quantify* operator (#6) produces a new value for the transformed row. The *Link-Input-to-Outcome* operator then assigns its cause to the spreading action via implementation by the *Determine-Causal-Link* operator (#8) in the **Cause?** space. The *Greater-Than?* (#9) and *Less-Than?* (#10) operators establish that spreading has neither increased nor decreased the number of the row and this is returned as a result to the **Comprehend** space by the application of the *Effect?* operator (#11). The form of that result is the augmentation of the ADD list of the observed transformation with the information that it neither increased nor decreased the number of objects. Now the

Same-Number? operator can apply in the **Comprehend** space and return this result to the **Perform** space where the *State-Effect* operator outputs the judgement that the number of crosses is still the same, thereby satisfying the top goal and terminating the run.

Most importantly, the successful termination of processing in the **Explain** space subgoal causes a chunk to be built. The backtrace mechanism in Soar's chunking process determines what are the critical conditions for the future automatic augmentation of instance with information about their numeric effects. In other words ABC-Soar converts the addition of inferable numeric effects of a transformation to a recognitional process . The chunk that is built is presented, translated into English, below

```
If
There is a Same-Amount?(number) operator
and a spreading action has been applied
Then
the transformation did not add or subtract number
```

The processing described in this trace illustrates only how ABC-Soar defines what is in effect a conservation concept. It does this as a side effect of categorizing a new input (transformation) that it sees and of answering a question about the numeric effects of that transformation. In other words, when asked to attend to the numeric effects of an action ABC-Soar creates a chunk that determines the conditions under which, in future, the invariance knowledge that it acquired from that experience will be accessed. The generalization of that knowledge into a full-blown conservation concept is not illustrated here but some suggestions about the nature of that process are addressed in the next section.

6. DISCUSSION

This example has demonstrated some of the principles by which ABC-Soar is able to form simple concepts of transformations and augment them with task-specific inferable knowledge as indexed by the problem solver's goal. Rather than claiming that this is a reasonable model of all of the parts of the conservation learning process that were simulated I would claim instead that ABC-Soar provides support for the view that the EBL-style impasse-driven explanation of the acquisition of invariance knowledge is a plausible framework for the development of a more psychologically accurate version. Nevertheless, we saw how an initial definition of the spreading actions concept was formed and how it was effectively converted into a simplistic conservation concept which states that, if such a transformation is observed while the problem solver is attempting to understand its numerical effects then it is known that this type of action is a number conserving action.

6.1. Psychological Implications

Even such a limited example has some interesting implications in terms of the developmental psychology of acquiring conservation concepts. The most important is that this model demonstrates the claim made in theories of conservation that the critical development is attention to the transformation and not to the details of the stimulus materials. However, ABC-Soar offers a completely novel explanation of how that comes about and shows that the transition is one of the accretion of knowledge and not the rejection of one kind of behaviour and the adoption of another. The model attempts from the start to explain conservation in terms of the transformation. Whenever there is insufficient information to do this it must find some means of computing specific values in order to substantiate the quantitative effects of the transformation. As a result, the majority of initial attempts to answer conservation questions require attention to specific details of the materials. In other words, the model can be observed to engage in counting and comparing. After learning, such attention to detail is not necessary since sufficient knowledge is cued by attention to the transformation. Then the model engages in totally different behaviour, namely automatic categorization and inference of the numeric effects. It is worth restating that, in many cases, discrete values cannot be computed by young children. These include problems concerning liquids, large numbers, mass and almost any conservation materials except for small numbers of discrete objects. Conservation in these domains is dependent on generalization of the effects of transformations that occurred on materials where invariance could be objectively computed. Therefore this model not only models the transition within one domain but provides an explanation of the across domain pattern that can be observed in the empirical regularities that were reported earlier in the chapter. It also explains why experience with small numbers is crucial to the development of the understanding of conservation in general.

The construction of ABC-Soar also highlights some of the issues concerning the speed of learning conservation concepts. Concepts formed from initial exposure to the conservation task constitute real learning but are highly specific. Spreading was defined only in terms of an action carried out by a person (perhaps even the specific experimenter) on a row of five crosses. Thus the conservation knowledge that spreading is a number preserving action will be automatically accessed only for actions with the same feature lists. With exposure to further spreading actions those features will be generalized and the definition will become more inclusive. The fact that this will take some time suggests that, in the tests of conservation the transformations are sufficiently similar that the child's performance looks rather more advanced than it may be if a wider range of actions were to be considered. On a similar point, in the training studies learning is fairly fast whereas in the real world it takes years for children to acquire conservation. The slowness of natural conservation learning may well be due to the fact that it is not number that is seen as the salient dimension when observing transformations of this sort; there are many visible ones such as length, spatial density, height etc. that also change (Klahr, 1982). When, in the training studies the number aspect is made salient then some conservation learning takes place very quickly. One less resolved issue is the hierarchical generalization of transformation knowledge such as from number to quantity (so that transformations concerning liquid will be immediately understood). At present it seems that this requires inductive reflection over the similarity between the effects of actions on different materials.

Other evidence suggests that the explanation embodied in ABC-Soar of trained conservation development accounts for naturally occurring, or at least untrained, conservation development. It can be seen that the pattern of conservation performance in Siegler's study of 5 to 9 year olds (Siegler, 1981, fits the pattern predicted by ABC-Soar. Number conservation tasks were answered correctly by the younger children when there were small numbers of objects although they could not perform correctly on liquid and solid quantity tasks or large number tasks. These findings support the hypotheses that where computation of absolute values is possible it will be used to objectify the quantitative effects of an action on a set of materials. An intermediate form of processing emerged in the number task but not in the liquid or solid quantity task. This involved the use of what looks like an overgeneral conservation rule that states that, "if something is added the number gets larger" without testing if there was the same number of objects in the first place. It was only used on trials where something was added but in cases of initial inequality led to the wrong result. The form of this rule suggests that it was acquired by a process of counting the effects of an addition transformation. In the "real world" (i.e. outside of the psychology laboratory) the only real way to be sure of the quantitative effects of an action on some material is to compare a before and after value. The more objectively comparable that value is (such as numbers and not category labels like "large" or "small") the more certain one can be about the effects of the action.

6.2. The Role of Soar

This chapter has focussed on an impasse-driven account of conservation learning which employs an EBL style of analysis and a particular theory of concept-like knowledge. However, much of what psychological plausibility does exist has been gained by its implementation in the Soar architecture. There are two levels at which this can be stated; the role of the architecture and the role of the theory, although these are not independent dimensions. The cognitive architecture of Soar both constrains the way that any given model of task behavior can be implemented but also provides "for free" some principled guidelines about how, according to the theory of Soar (Newell, 1990), such a task is likely to be carried out by the problem solver. I will illustrate this point with the examples of the contribution of Soar to the integration of deliberate and automatic behavior and the learning and revision events in ABC-Soar.

The primary contribution of modelling this developmental phenomenon in Soar was that the natural way to do this led to the parsimonious impasse-driven account of the classical conservation transition for which there is no obvious counterpart in the psychological literature. The model arises from Soar in the following way. All Soar's problem solving takes place by search in problem spaces. Thus the implementation of ABC-Soar contains a set of operators in the **Perform** problem space that are sufficient to implement the part of the conservation task that we are concerned with. If all the necessary knowledge were stored in long term memory then these would be the only operators that the system would need in order to be able to perform the task. However, if any of these operators fail due to insufficient knowledge then further problem spaces (and operators) are specified in order to create the knowledge to allow that operator to apply. The automatic impasse creation and termination leads this competence to emerge into performance at whatever level the knowledge in the model enables it to do so. When it has little or no knowledge about the numeric effects of transformations, ABC-Soar can been seen to engage in counting and comparing the rows of objects. However, as that knowledge is built into chunks and the subgoals are no longer created then the model's behavior changes to merely responding to the transformations directly. There is no program as such to tell it to do this. The manifest behaviour is a result of an interaction of the knowledge in the system at any time, it's goals and the tasks that the system is currently working on.

In this way, ABC-Soar is able to exhibit a smooth integration of automatic and deliberate behavior for different tasks and subtasks depending on what knowledge it has and what it is working on. This determines when existing knowledge will be augmented and revised and sets the context for the comprehension of new information. In other words ABC-Soar has decisions about when to learn and what the learned information means determined ultimately by the cognitive architecture in which it is implemented. Another effect is a continual move towards automaticity where possible so that the inability to comprehend input and expectation failures trigger deliberation and learning. As a result, context dependent learning creates compiled knowledge whose interrelations are a function of goal-dependent "explanations" as suggested by Medin (Medin, 1989, Murphy & Medin, 1985).

Two more examples concentrate on the subgoaling and chunking mechanism in Soar. Chunking in ABC-Soar not only determines the contextual references of what is learned but also carries out the generalization process. That generalization arises from the selection of critical conditions for later access to the subgoal's results. The reason for this selection is that, when the system arrives at the same situation where previously a subgoal would have been created, the new chunk must match and fire and so it cannot require the existence of predicates that would only arise were the subgoal created. The effect is that the learning is generalized to test only some of the information present at the time of the creation of the results as conditions for its access.

The natural termination and removal of subgoals when processing can continue in the supercontext solves the operationality issue that concerns EBL theorists (DeJong & Mooney, 1986). The operationality issue is stated by DeJong and Mooney as a dynamic criteria where useful generalizations are ones that the performance part of the system can use with the knowledge that is already available; in short the results of EBL must not require further problem solving. This is precisely the case in the Soar architecture. Processing in a subgoal (and possibly even lower subgoals) continues until sufficient knowledge exists for the stalled superoperator to apply and satisfy its terminating conditions. Until this is true subgoal processing just continues to work to achieve that state. Thus, if a subgoal is ever resolved then it is certain that the resulting knowledge is usable by the system because that is the very goal test for the subgoal termination.

The theoretical aspects of Soar's contributions are a little more abstract. They really concern the issues of constraints and degrees of freedom. In the process of building any integrated model of learning and performance all kinds of decisions have to be made about the implementation of parts of the model. Accepting the theoretical structure of the Soar architecture removes the assumptions inherent in implementing a novel learning mechanism or control scheme. Instead one simply assumes Soar's theoretically grounded constraints on how a model can be built, what kind of processing is necessary to make it run, what kind of learning will it do, and so on. In other words, the control structure of a model built in Soar is not *ad hoc* but relies on principles arising from a theory of cognition and gains validation from its use on a wide range of tasks.

7. Conclusions

This chapter described an attempt to provide one the first computational accounts of a mechanism of cognitive development. It proposed concept formation as the basis of the mechanism and presented an instantiation of that mechanism in a computational model, ABC-Soar. This demonstrated how quantitative effects of a class of actions become recognitionally triggered as the result of the problem solving carried out by applying an EBL-like mechanism to knowledge and competence that the learner had already acquired. This knowledge and competence was brought into play because it constitutes the "ground level" abilities for objectifying the numeric effects of transformations that the system "failed back" to when it was unable to produce a judgement of numeric effects merely by observing a transformation. The special role of small numbers and small number competence that is central to ABC-Soar enables it to provide a totally novel explanation of within- and across-task conservation development.

Furthermore, the fact that the inferable conservation knowledge is added when indexed by a certain goal achieves the explanatory coherence characteristic of concept-like knowledge (Medin, 1989). Just as the goal of removing things from a burning house provides an explanatory coherence of why things like children, jewelry and photographs belong in the same category, ABC-Soar explains why certain actions would go together. For example, if it was solving numeric problems, ABC-Soar would effectively Categorize spreading, compressing, piling and distributing actions together since they have no numeric effect. However, if the concern was spatial density then compressing and piling would form a category with the opposite effect of spreading and distributing.

Much needs to be done to expand the ABC-Soar model, first in relation to number conservation development, and later to wider aspects of conceptual development. However, what has been done so far suggests that there is a promising future to the modelling of human developmental mechanisms by the application of machine learning techniques such as EBL and chunking in Soar. ABC-Soar is the first step towards an explanation of how a single system incrementally gains expertise in a diverse range of domains without the need for guidance by an external intelligence.

References

Barsalou, L.W. (1983). Ad hoc categories. *Memory and Cognition.*, *11*, 211-227.

Carey, S. (1985). *Conceptual Change in Childhood.* Cambridge, Mass: MIT Press.

DeJong, G. & Mooney, R. (1986). Explanation-based learning: An alternative view. *Machine Learning, 1*, 145-176.

Fikes, R.E. & Nillson, N.J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence, 2*, 189-208.

Frye, D., Braisby, N., Lowe, J., Maroudas, C. & Nicholls, J. (1989). Young children's understanding of counting and cardinality. *Child Development, 60*, 1158-1171.

Gelman, R. (1982). Accessing one-to-one correspondence: Still another paper about conservation. *British Journal of Psychology, 73*, 209-220.

Gelman, S.A. (1988). The development of induction within natural kind and artificial categories. *Cognitive Psychology, 20*, 65-95.

Gelman, R. & Baillargeon, R. (1983). A review of some Piagetian concepts. In J.H. Flavell & E.M. Markman (Eds.), *Handbook of Child Psychology*(Fourth ed.). New York: Wiley.

Gelman, R. & Gallistel, R.C. (1978). *The Child's Understanding of Number.* Cambridge, Mass: Harvard University Press.

Halford, G.S. & Boyle, F.M.. (1985). Do young children understand conservation of number? *Child Development, 56*, ??

Halford, G.S. & Fullerton, T.J. (1970). A discrimination task which induces conservation of number. *Child Development, 41*, 205-213.

Holland, J.H., Holyoak, K,J., Nisbett, R.E. & Thagard, P.R. (1986). *Induction: Processes of Inference, Learning and Discovery..* Cambridge, Mass: MIT Press.

Keil, F.C. (1989). *Concepts, Kinds and Cognitive Development.* Cambridge, Mass: MIT Press.

Klahr, D. (1982). Nonmonotone assessment of monotone development: An information processing analysis. In S.Strauss & R. Stavy (Ed.), *U-Shaped Behavioral Growth.* New York: Academic Press.

Klahr, D. (1984). Transition processes in quantitative development. In R.J. Sternberg (Ed.), *Mechanisms of Cognitive Development.* New York: Freeman.

Laird, J.E., Newell, A. & Rosenbloom, P.S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence, 33*, 1-64.

Lewis, C.L. (1988). Why and how to learn why: Analysis-based generalization of procedures. *Cognitive Science, 12,* 211-256.

Markman, E.M. (1989). *Categorization and Naming in Children.* Cambridge, Mass: MIT Press.

Medin, D.L. (1989). Concepts and conceptual structure. *American Psychologist, 44,* 1469-1481.

Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence, 18,* 203-226.

Mitchell, T. M., Keller, R. & Kedar-Cabelli, S. (1986). Explanation-Based Generalization: A Unifying View. *Machine Learning, 1,* 47-80.

Murphy, G.L. & Medin, D.L. (1985). The role of theories in conceptual coherence. *Psychological Review, 92,* 289-316.

Newell, A. (1990). *Unified Theories of Cognition.* Cambridge, Mass: Harvard University Press.

Pazzani, M.J. (1988). Integrated learning with incorrect and incomplete theories . J.E. Laird (Ed.), *Proceedings of Fifth International Machine Learning Workshop.* Los Altos, CA, Morgan Kaufman, Ann Arbor, MI.

Piaget, J. (1952). *The Child's Conception of Number.* New York: W.W. Norton.

Piaget, J. (1970). *Structuralism.* New York, NY.: Basic Books.

Rosenbloom, P.S. & Laird, J.E. (1986). Mapping explanation-based generalization into Soar. *Proceedings of Fifth National Conference on Artificial Intelligence.* AAAI, Philadelphia.

Rosenbloom, P.S., Newell, A. & Laird, J.E. (1990). Towards the knowledge level in Soar. In K. VanLehn (Ed.), *Architectures for Intelligence.* Hillsdale, N.J.: Erlbaum.

Schank, R.C. (1982). *Dynamic Memory.* Cambridge, UK.: Cambridge University Press.

A.M. Segre (Ed.). (1989). *Proceedings of Sixth International Machine Learning Workshop.* San Mateo, CA: Morgan Kaufman. Ithaca, NY.

Shultz, T.R. (1982). Rules of causal attribution. *Monographs of the Society for Research in Child Development, 47,* 1-51.

Siegler, R.S. (1981). Developmental sequences within and between concepts. *Monographs of the Society for Research in Child Development, 46,* 1-74.

Small, S. (1979). Word expert parsing. *Proceedings of 17th Annual Meeting of the Association of Computational Linguistics.* Association of Computational Linguistics, August.

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

Vadhan, V. (1984). Induction of conservation by discrimination training. *The Journal of Psychology, 116,* 273-277.

Overview of ABC-Soar

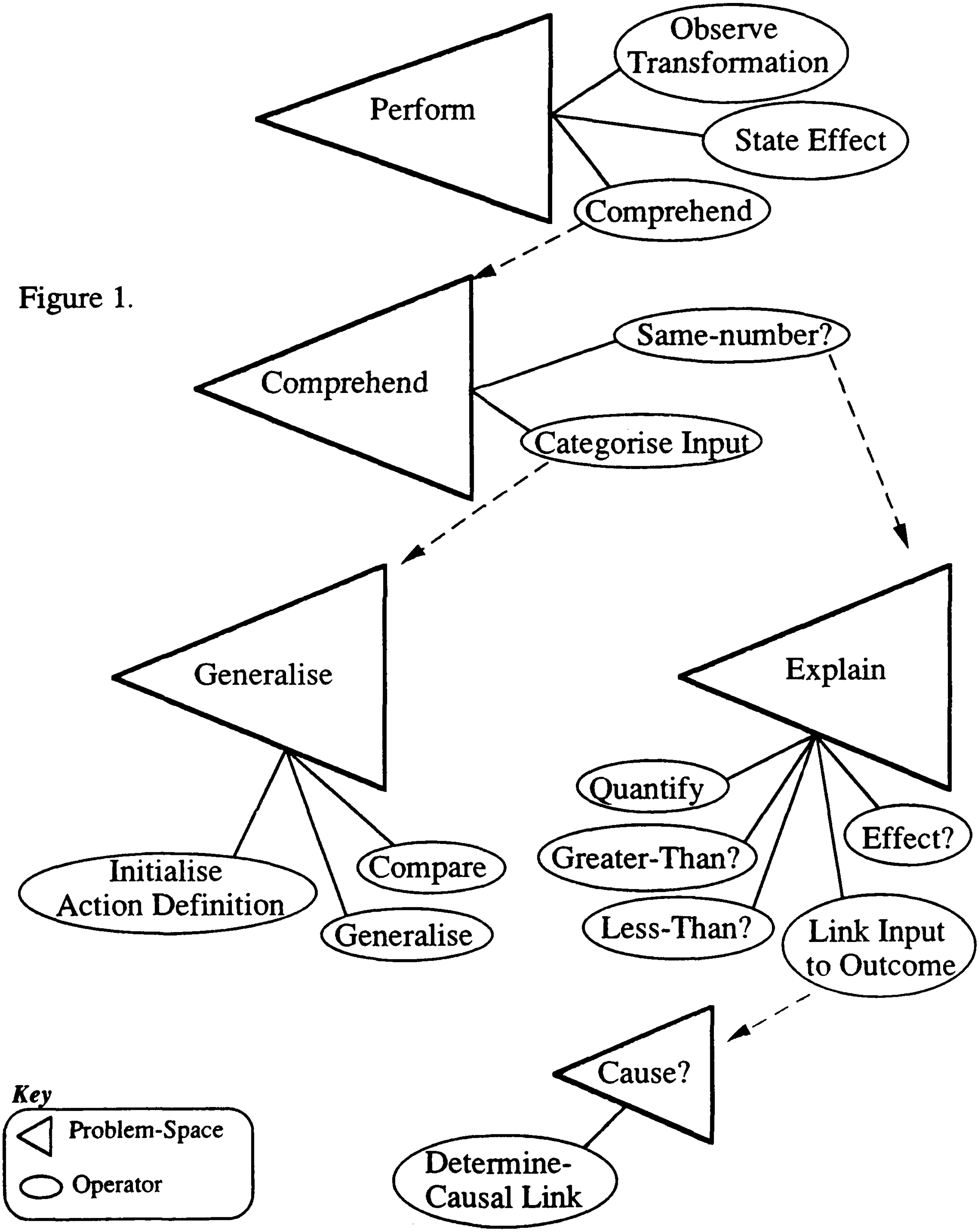

Figure 1.

An Example Trace

Figure 2.