

A COMPARISON OF BLOCK-STACKING HEURISTICS USED BY
PRESCHOOL CHILDREN AND CLASSIC ARTIFICIAL INTELLIGENCE
PLANNING PARADIGMS

by

Alison Virginia Johnson

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF ARTS

in

Education, Curriculum and Instruction

APPROVED:

T M. Wildman, Chairman

J. K. Burton

J. W. Roach

November, 1988
Blacksburg, Virginia

3-1-88
1/20

A COMPARISON OF BLOCK-STACKING HEURISTICS USED BY PRESCHOOL CHILDREN AND CLASSIC ARTIFICIAL INTELLIGENCE PLANNING PARADIGMS

by

Alison Virginia Johnson

Committee Chairman: Terry M. Wildman
Education, Curriculum and Instruction

(ABSTRACT)

Despite the large body of research in Psychology concerning human planning and problem-solving both by adults and children, the study of planning and problem-solving in the field of Artificial Intelligence has proceeded along its own development with very little concurrent exploration of the methods people use to plan and solve problems. Some of the classic planning programs are unable to solve problems which are trivial for children, and it may be that by exploration of the methods children use we will discover certain heuristics which can be incorporated into AI planning paradigms. This thesis explores this possibility. Children aged 3 to 5 years were recorded performing a block-stacking task which simulates the type of problem given to planners to test their efficiency. The data were analyzed in order to determine those heuristics which are common to planners and children as well as those which are unique to the children. Based on this analysis, the psychological validity of the planning programs are evaluated.

Acknowledgements

Many thanks to the members of my committee, Dr. Roach, Dr. Burton, and Dr. Wildman, my advisor, for encouragement, support and suggestions. Thanks also to _____ for her invaluable secretarial expertise. Support for this research was provided by the Ministry of Information and the E. G. F. Foundation.

I dedicate this thesis to _____ .. It is as much his accomplishment as it is mine.

Table of Contents

1. Introduction.....	1
1.1. What Planning Is.....	1
1.2. Central Issues of Planning: Heuristics vs. Representation.....	3
1.3. Planning in Psychology.....	4
1.4. Classic Planning Programs.....	9
1.5. Representation in AI.....	10
1.6. Problem Completeness.....	13
2. Central Issues.....	15
2.1. Representational Assumption: The Piagetian Pre-operational Stage.....	15
2.2. The Link between AI and Psychology.....	15
2.3. The Need for Additional Research.....	17
3. Method.....	19
3.1. Subjects.....	19
3.2. Materials.....	19
3.3. Procedure.....	19
4. Results.....	21
4.1. Coding.....	21
4.2. Analysis.....	22
4.3. Discussion.....	25
5. Conclusion.....	32
5.1. Extension of Psychological Research.....	32
5.2. Relationships Between AI Planners and Children.....	32
5.3. Further Research.....	34
References.....	36
Appendices.....	38

1. Introduction

Considering the amount of research on related topics which is done in the respective fields of Psychology and AI, the amount of research which links the two fields is relatively small. When psychologists employ computer models, they are usually tools to define their theory; when AI researchers study human behavior, it is most often to imitate with a program the external behavior of humans. Only infrequently are the processes employed by humans and programs compared.

It may be that by exploration of the methods children use we will discover certain heuristics which can be incorporated into AI planning paradigms. This thesis explores this possibility. Children aged 3 to 5 years were recorded performing a block-stacking task which simulates the type of problem given to planners to test their efficiency. The data were analyzed in order to determine those heuristics which are common to planners and children as well as those which are unique to the children. Based on this analysis, the psychological validity of the planning programs are evaluated.

1.1. What Planning Is

Problem-solving in the Psychology laboratory, as described in Ellis and Hunt (1983), is "characterized by presenting subjects with an initial state, including assumptions and constraints, and asking the subject how they would go about achieving a particular goal state" (p. 221). In Artificial Intelligence, this same process presented to the computer in the form of a program and input data is called planning. Within this paper the terms planning and problem-solving will be considered equivalent terms and used interchangeably, and they will refer to both the psychological and the AI concept.

Ellis and Hunt (1983) describe several stages which occur in problem-solving. These include understanding the problem, generating solutions using a variety of strategies, and evaluating the solutions. In this research the focus is on the various strategies which are used in the solution-

generation stage. Strategies in this stage are of two types: algorithms, which are sets of rules or procedures which ensure a solution, and heuristics, which are "rules of thumb" which may or may not ensure the solution.

An algorithm will always lead to the solution of a problem. Algorithms are useful when it is absolutely necessary to find the right answer, no matter how long it may take or how much it may cost, or when the problem is known to be of a small size. For example, suppose you are visiting Baltimore and want to call your old high school friend, Mortimer Sobansky. He said to look him up in the book, so you go to the telephone directory and look under "S". There are four Sobanskys listed. In this case, the algorithm you might use would be to call each of those four Sobanskys, knowing that one of them would be your friend Mort.

On the other hand, if you were trying to get in touch with Bob Brown, who now lives in Los Angeles, this type of algorithm would be prohibitively long (and expensive, if you are calling from Baltimore). You would have to call all the Browns in L.A. In order to narrow your search, and minimize your costs, you need to apply some heuristics. You can call the Browns who are listed with the first name B., Bob, R., or Robert. And you can search only in those phone books for the area of L.A. where you think he lives. By using these heuristics, you may not find him; he may be listed under Bobby or Rob, and/or live in a different area of L.A. (Or he may have changed his name and gone to live in a commune; in California one never knows.) But it is likely that you will find him, and it will cost you considerably less than calling every Brown in L.A.

So, algorithms are used when a solution must be found, and time and cost are not a consideration, or when the problem space is known to be small and costs low. Heuristics are used to minimize costs and search time, at the calculated risk of eliminating that portion of the search space which contains the solution to the problem. Both humans and planning programs tend to use heuristics rather than algorithms, for while an algorithm will always lead to the solution, it may be a long time coming, or be very expensive to find. A heuristic is not a sure thing, but it is likely to lead to the solution.

One type of heuristic is the generate-test method. This is a two-step process; first, a possible solution is generated, and then it is tested to see if it works. This trial-and-error method is not very efficient if there is a large number of possible solutions. A better type of heuristic is known as means-ends analysis. In this process, subjects must determine the ends they wish to achieve and the means they need to use to get there. In so doing, they must define subgoals, that is, smaller parts of the problem which may be easier to solve. By completing the subgoals, the subjects will be able to complete the problem. It is also possible to work backward from the goal state to the initial state in setting subgoals to achieve.

1.2. Central Issues of Planning:

Heuristics vs. Representation

In this research the focus is on the solution-generation stage of planning; however, the strategies which are used depend on the understanding of the problem. In other words, the heuristics chosen depend upon the internal representation of the problem, both for people and planning programs. An intuitive example of this dependence is the case of the infant who has not yet learned the concept of object permanence, and behaves as though an object which he can no longer see no longer exists. We can easily believe that this baby sees the world very differently from the way we do. Because of this different representation, the heuristics the baby chooses when his toy is hidden from view differ from those heuristics an infant with object permanence will use. The first child may cry if he wants to have the toy (which for him no longer exists), or turn his attention to another object if he is less interested. The second child may also lose interest and turn to another toy, but faced with a desire for the first toy, he will look for it behind and underneath things. Based on this difference in behavior, we infer a difference in representation.

Klahr and Robinson (1981) discuss this issue of representation and heuristics:

"Throughout the Strategic Analysis, we assumed that children's encodings were isomorphic to the external display.... While this may be a reasonable approach for the 6-year-olds on whom we used it, it is probably not appropriate for the youngest children.

"Indeed, in reviews of the infant search literature, one finds an emphasis not so much on strategies as on the encoding processes with which the child constructs an internal representation of the environment.... We believe that one possible source of difficulty for our youngest subjects was the creation of an internal representation upon which their general problem-solving methods could effectively operate.

"General problem-solving methods manifest themselves in rudimentary form by the end of Piaget's 'sensory-motor' period. ... While much remains to be learned about the developmental trajectory of problem solving methods, we know even less about the development of encoding processes." (p. 146)

1.3. Planning in Psychology

Psychologists such as Siegler (1977, 1978), Richards and Siegler (1981), and Klahr (1978, 1980, 1985) have done much work concerning the development of planning processes in children. Siegler and Richards and Siegler address representational issues, whereas Klahr investigates children's heuristics on a variety of planning tasks.

Siegler (1977) looked at adolescents' performance on the Twenty Questions game as a means of assessing their problem-solving strategies. His study looked at the relationship between children's problem-solving and the questions they ask while trying to solve a problem. The type of problem a child attempts to solve can influence the type of question he or she will ask. The materials for the three experiments were two matrices of 24 cells, either filled in with the numbers 1-24 or the letters A-X. The goal set for the children was to discover the "correct" cell in the matrix. The correct cell was not chosen ahead of time; instead, "feedback was given on the basis of minimal

elimination of alternatives rather than by predesignating a particular number or letter as correct and taking the chance that children who followed unsystematic strategies might stumble onto the answer"(p. 396). For example, if the child asked, "Is it in the first row?" the answer given to the child would be "no," because that would eliminate only six alternatives and leave eighteen cells to guess from. Continuing in this manner, the child would guess and the experimenter answer until only one square was left. With this procedure it is possible to differentiate between children who use a system of questioning and those who question haphazardly.

In the first experiment, each child received both of the matrices, either in the order number-letter or letter-number. The results of this first experiment were that the children who received the matrices in the presentation order numbers-letters consistently performed better on both matrices (discovered the answer in less moves) than the children who received the matrices in the presentation order letters-numbers. Apparently there is a difference in the mental representation of the problem when it is presented with numbers rather than letters, and this difference carries over into the second presentation. The children are able to recognize the isomorphism between the two problems, and therefore apply the same strategies to the solution of the second matrix that they applied to the first.

In the second experiment, the children were given special instructions intended to deepen their search through their existing knowledge of the problem, or in other words, to broaden their representation of the problem. These included instructions on planning the questions they would ask, as well as information on the nature of the task and the desired solution strategy. The special instructions resulted in the elimination of the difference between the numbers-letters and the letters-numbers order of presentation. "However, it was unclear whether the changed instructions actually promoted a deeper search of existing knowledge or whether reductions in the previous differences among experimental groups were due simply to ... additional information about the task that was provided in the instructions" (p. 400). Therefore, in the third experiment, Siegler presented the children with the part of the instructions which clarified the nature of the task, but eliminated

that part which described the type of planning which they should do. Differences similar to those in Experiment 1 appeared, indicating that it was the instructions to plan which reduced that difference in Experiment 2 and not the extra information about the nature of the task or the desired solution strategy.

This series of experiments shows that the problem representation can influence the type of heuristic used-- in other words, the same type of problem represented two different ways can influence the questioning strategy of adolescents. However, further information on the solution strategy to use did not appear to influence the performance of the children-- only representational changes had this effect.

Richards and Siegler (1981) looked at the development of problem-solving heuristics in very young children. The focus of their investigation was on two types of experiences which might lead to children's problem-solving strategies becoming more systematic. The first type of experience a child may encounter is encouragement from others to adopt analytic attitudes. The second type of experience which might cause a child to use a more systematic strategy is an exaggeration in the relevant dimension which would emphasize the type of strategy to be used. For example, in a balance problem which asked "which way would they tip?" a very large child would be drawn on one side of a teeter-totter picture and a very small child drawn on the other. Using a balance scale problem, three-year-olds were either (1) presented with very large differences in weight, (2) asked to try to figure out why one side went down, (3) given both types of aids, or (4) given neither. They found that the children produced more correct answers when encouraged to take a more analytic attitude (57% vs. 42%) and when presented problems with highly salient weight differences (55% vs. 44%). The use of strategies that were more systematic than those used by children who were given no encouragement persisted when the children were given a post-test a week later.

As with the previous experiments, this study demonstrates that modifying the representation of the problem can improve humans' performance on a task, just as it can for planning programs. Also, encouragement to plan can

improve performance-- however, in this case, not to a level significantly higher than the level achieved when the representation was modified.

In investigating children's planning and problem-solving, Klahr (1978) makes use of a computer model called SOLVE as a predictor of children's planning processes. Klahr constructed a variant of the Towers of Hanoi puzzle which consisted of three nested cans inverted on pegs. Three-, four- and five-year old children were asked to compare their version of the puzzle to an experimenter's version and explain how the experimenter should move the cans so as to make the two versions match. Results showed that the 3-year-olds were able to solve two- to four-move problems, 4-year-olds could solve four- to six-move problems, and 5-year-olds were evenly divided between five- and six-move problems. The differences in performance related to differences in strategy as predicted by the SOLVE model. Klahr extended this work in his 1980 study, modifying the procedure slightly and conducting a more detailed analysis of the data. He found that various models of move selection strategies corresponded highly with individual children's performance.

It was found in the two previous studies that certain of the problems presented to the children had subgoal orderings implicit in their structure, whereas other problems were ambiguous in terms of their ordering of subgoals. For those problems where the subgoal ordering was implicit, children were able to apply some type of means-ends analysis to the problem. Klahr (1985) investigates the other methods young children use when presented with problems having ambiguous subgoal ordering. For this experiment, Klahr created a puzzle called the "dog-cat-mouse game" which allows the child to match the animal with its favorite food via a series of constrained moves. The puzzle has ambiguous subgoal ordering, in that the order in which the animals will reach their food is not obvious; it has easily remembered rules and a natural way to represent the goal state; and third, the puzzle has a wide range of levels of difficulty.

By studying the moves made by the children on the puzzle, three main results were found. First, they searched two or three moves ahead for the goal. This was determined by finding the level at which a child was able to complete the puzzle correctly 100% of the time. Two-thirds of the children could

produce perfect answers when the problem was two moves from the goal state, and another third were able to perform perfectly when the problem was three moves from the goal.

Second, the children were sensitive to incremental progress toward the goal. The children had a tendency to favor moves that increase the number of pieces in their goal locations. The children would even make such moves when it led them "up the garden path", seeming to take them closer to the solution when in fact the move deviated from the minimum path. This relates to the third finding, which was that the children tended to avoid backing up. Once an animal was in its goal position, the child was very reluctant to move it from that position. These heuristics led to a pattern of hill-climbing: arriving at a state in the solution to a problem where all the next possible moves lead to a state further from the goal than the current state. In this study the scores varied widely and were not correlated with age.

Klahr created two different computer models of the solution process in order to test the accuracy of his hypotheses. The first model, Model A, had three important features: an all-or-none evaluation of the problem (either it was at the goal state or was not), no backing up, and a two-step look-ahead for the goal state. This model explained 50% of the variance. However, this model oversimplified the children's behavior. A second model, Model B, was capable of partial evaluation of the problem (in effect, seeing how close it was to the goal), and it did back up occasionally. Model B accounted for over 70% of the variance in performance. Since path length was not taken into account when looking at the performance of Model B, a multiple regression was done on the model's performance and path length, and this accounted for 97% of the variance.

These studies on planning processes in children address both the issues of representation and heuristics. Siegler (1977, 1978) showed that representation affects the heuristics chosen during the solution-generation stage of problem-solving. Richards and Siegler (1981) demonstrated that this also holds true for young children. Klahr's 1978 and 1980 experiments showed that children's planning skills develop with age, enabling older preschoolers to solve longer problems than younger children. Klahr (1985) identifies some of the

heuristics children use on a particular problem. These studies form the basis for the type of problem presentation and analysis that will be done on the children's block-stacking behavior in this study.

1.4. Classic Planning Programs

There are several different types of AI programs which can, when given a list of descriptions of an initial state and a list representing a goal state, plan a set of steps to proceed from one to the other. Each of these programs incorporates a set of heuristics. Those that use a means-ends analysis compare the initial state and the goal state and find the differences between the two to create a list of subgoals, called a "difference list". They then choose a move to make that will (hopefully) lead to eliminating one of those differences, thereby achieving a subgoal. When all subgoals are achieved, then the problem is solved.

An example of a planner which uses this type of means-ends analysis is STRIPS (Fikes, 1971). STRIPS tries to eliminate items from the difference list one item at a time. If it finds that it can't reduce the problem one way, it tries reordering the list and achieving the subgoals in a different order. The problem with STRIPS is that if the steps to solve "Subgoal 1" have to be interwoven with the steps to achieve "Subgoal 2" then it cannot solve the problem. It can only consider one subgoal at a time.

Another planner, Noah (Sacerdoti, 1975), was developed later and uses a very different approach. Noah does not look at subgoals as states of the world, but rather at all the possible moves which can be made, given the current state and the goal state. If Noah finds that making one move interferes with making another necessary move later on, it rearranges the moves according to the constraints they put on each other. The trouble with this heuristic is that for some problems it may lead to hill-climbing. If there is more than one possible move that can be made, Noah chooses one at random and discards the others. Discarding alternative moves once a choice has been made does not allow for the possibility that the random choice may have been the wrong one

for this particular problem. It therefore becomes stranded on a hill, and is incapable of solving the problem.

An adaptation of the Noah planner is called Nonlin (Tate, 1977). Nonlin improves on Noah by incorporating the heuristic of backtracking. When choosing between possible moves to be made, Nonlin chooses one but remembers the other possible moves. Later, if it turns out that it reaches a dead-end, it can back up to where it was before and try one of the other moves, rearranging the order in which it makes the moves necessary to solve the problem. However, "The use of backtracking by this algorithm is a mixed blessing. On the one hand, it 'solves' the problem of making planning decisions. Any decision at all can be made correctly by repeatedly guessing and backtracking until the right guess is stumbled upon.... On the other hand, backtracking alone does not address the problem of choosing among feasible plans; it merely rules out completely infeasible ones." (Charniak & McDermott, 1985).

1.5. Representation in AI

Research about problem representation in Artificial Intelligence fills books; this topic comprises approximately half of all AI research. However, the most common type of representation is that used by STRIPS, Noah and NonLin. These programs use first-order predicate calculus, which consists of: (1) a language for expressing propositions or assertions about the world and (2) a set of rules for how to infer new propositions based on those already present. Some examples of assertions in predicate calculus for a simple block-stacking problem might be:

(on block-1 table) [block-1 is on the table]

(clear block-2) [block-2 is clear]

(achieve (on block-2 block-1)) [put block-2 on top of block-1]

There is again a wide latitude in terms of how this predicate calculus is used. The previous section described the methods which STRIPS, Noah and NonLin used to solve problems. There are variations in these heuristics due to the differing representations. STRIPS uses a representation based on

descriptions of the state of the world. If, for example, block A is on block B and block B is on the table, and it chooses to put block A on the table, it must first make sure block A is clear (this is called a precondition), and if the precondition is true, it then adds the two predicates **(on block-a table)** and **(clear block-b)** to its list of facts about the current state of the world and deletes from that list **(on block-a block-b)**.

Noah and NonLin use predicate calculus differently. In their case, a "plan library" is given to them as part of their knowledge of how to operate on the world. This plan library consists of plans and subplans, and sub-subplans, until the level of "primitive actions" is reached. The goal of these two programs is to reduce all the plans down to their primitive actions, and then to put the primitives in the proper order for solution of the problem. There are also safeguards built into this system, called protections, which help to determine the order in which the plans will be carried out. Here is an example of how Noah and NonLin might solve the problem **(achieve (and (on a b) (on b c)))**:

1. **(achieve (on a b))**
2. **(achieve (on b c))**

The plan library shows that a problem of this type can be broken into two parts, achieving one and then the other. No order is put on the problem as yet.

1. **(achieve (on a b))**
 - 1.1 **(achieve (space-for a b))**
 - 1.2 **(achieve (clear a))**
 - 1.3 **(move a b)**
2. **(achieve (on b c))**
 - 2.1 **(achieve (space-for a b))**
 - 2.2 **(achieve (clear b))**
 - 2.3 **(move b c)**

The program finds in the plan library that each of these subproblems can be solved by accomplishing the three primitive actions listed for each one. The next step is to order the primitives based on their protections. The program puts them in the following order, because in order to achieve primitive 2.2, it would be necessary to undo primitive 1.3, thereby violating its protection.

2.1 (achieve (space-for a b))

2.2 (achieve (clear b))

2.3 (move b c)

1.1 (achieve (space-for a b))

1.2 (achieve (clear a))

1.3 (move a b)

The major difference between Noah and NonLin's representation and that of STRIPS is that Noah and NonLin do not "know" anything about the different world states. They simply use the plan library to find all the possible moves which can be made, given the initial problem and the goal.

First-order predicate calculus, although the most common, is not the only type of representation used. An example of a radically different type of representation of the blocks-world is used by Brian Funt (1980). Funt's program, WHISPER, incorporates a high-level reasoner, a "retina" which can "look at" diagrams, and a set of procedures for modifying what the retina sees as its view of the object changes. The retina is structured like the human eye in that the resolution is clearest at the center and decreases toward the edges.

This type of representation is useful for solving such problems as determining when two objects are going to collide, when an object is going to fall off another object, and whether an object will be stable when placed upon another object. STRIPS, Noah, and NonLin are not capable of solving WHISPER'S problems. This is perhaps the clearest example of representation's influence on the types of problems which can be solved.

1.6. Problem Completeness

It was mentioned in the previous section that the plan library was the key to Noah's and NonLin's solutions to the blocks-world problems. They simply must reduce a problem to its primitive actions, order the primitives, and the problem will be solved. This, of course, assumes that the plan library is correct and complete. STRIPS has the same problem: the description of the world which it is given, the preconditions, and the set of rules for manipulating the blocks must all be correct and complete. If the representation is incorrect or incomplete, the program may not be able to solve the problem.

Yet it is nigh unto impossible to determine the dividing line between a complete problem representation and an incomplete one, at least in real life. Fredriksen (1984) divides problems into two categories: "well-structured" and "ill-structured". He points out that "the category in which a given problem falls obviously may depend in part on the problem solver. A problem may be well-structured for the problem solver who possesses the requisite knowledge and has practiced the relevant problem-solving procedures, or it may (be ill-structured) for one who has had insufficient experience or training in solving problems of that type." (p. 367). In effect, the problem solver becomes part of the problem, in terms of the information he (or she or it) contains which can contribute toward solving the problem.

For any problem, the problem solver must be able to call upon the information which it contains which is necessary to solve the problem. It also must be able to determine in some fashion what information presented in the problem is relevant and what is unnecessary to the solution of the problem. Depending on the type of problem, it may be necessary to determine whether time of day, duration of experimentation, availability of certain tools, or interference from other machines or people are needed for the planner to complete a problem. All of these issues are termed resources, and dealing with resources is known as "resource reasoning" in AI. This is an everyday problem for planning humans. Young people in junior high and high school often wrestle with this issue of relevant versus irrelevant information when attempting to solve mathematics word problems. It is the same problem which

arises when encoding the data for this study: what actions of the children shall be considered relevant and what shall be ignored at this time? STRIPS, Noah and NonLin assume that when the information concerning the initial state and the goal state is given, it is complete and sufficient. In other words, no resource reasoning is done by these programs. Because of this, we will not be examining the resource reasoning done by the children as they attempt to solve the blocks-world problems. As with the classic AI planners, they will receive all information necessary in the presentation of the problem, and we will assume a complete representation of the problem state in the minds of the children. This representation is based on Piaget's stages of development.

2. Central Issues

2.1. Representational Assumption:

The Piagetian Pre-operational Stage

In Piaget's stage theory of development, the "pre-operational stage" is the approximate age range of two to six years. A child within this age range has learned to represent objects as symbols, as is evidenced by the development of language, the ability to use a stick as a gun during play, and the delayed imitation of others' acts (sometimes to parents' dismay!). However, a pre-operational child is unable to focus on more than one aspect of a stimulus at a time. He can imagine an object but cannot imagine performing operations on the object. "One could say that thought at this present level... does not exceed the concrete: the child can think on that which he perceives, but his thought does not exceed this perception. He can represent perceptions and actions, he can internalize them, but they remain... that which they were in the concrete, he does not exceed this simple internalization by logical operations generalizable and composable within themselves. This is the property of the 'intuitive thought' of this stage, masterfully described by Piaget...." (Osterreith, 1983; trans.).

The stage which follows the pre-operational stage is termed the concrete operational stage and is characterized mainly by the acquisition of this ability to internalize actions.

What this means in terms of this study is that for the pre-operational child, doing is thinking. In other words, performing an action is the equivalent of thinking about doing that action. It is because of this externalization of the thinking process that we are able to compare the moves the children make to a trace of a computer program showing the program's "thinking process".

2.2. The Link between AI and Psychology

This thesis creates a link between Artificial Intelligence and Psychology by analyzing the block-stacking behavior of children in order to compare their heuristics to the heuristics common to planners. Other researchers have

bridged this gap from one side or the other. The most notable example of this bridge is the use by psychologists of computer models to give form to their theories and make predictions as to what the behavior of the child will be, based on the program's behavior. Klahr's (1978) SOLVE model described in Section 1.3 is a good example of this use of programming. In the AI domain, there is some work being done to model certain processes of development so that it can be of use to further the development of AI. Drescher (1987) has researched Piagetian learning via the computer; Selfridge (1982) modeled the development of sentence structure; and Elliott and Lesk (1982) compare adults' and planners' heuristics for finding routes on a map.

Drescher (1987) has attempted to model aspects of Piagetian learning in infants through a "Schema Mechanism" which interacts with a small, simplified representation of the world modeled inside a computer. Through this interaction the Schema Mechanism accumulates and organizes knowledge. It then uses the knowledge it has acquired to guide its activity in pursuit of further knowledge, in essence, developing a schema in much the same way as does an infant. He implements this Schema Mechanism in a program named MARCSYST, with some preliminary success. However, after building several hundred basic schemas representing "much of the eventual visual field network, and the visual and haptic proprioceptive networks; and many schemas that designate hand-eye and hand-body coordination," the program slows to several real-time seconds per simulated second.

Studies are also done in the area of natural language understanding which offer viable computer models of the development of human language processes. For example, Selfridge (1982) examines the development of understanding passive sentences in young children. He uses reversible passives, which are sentences in the passive voice (such as "the girl is followed by the boy") which have a different meaning when the word order is preserved but the sentence is changed to the active voice ("the girl followed the boy"). Selfridge's program, CHILD, "learns to understand passive sentences following a progression similar to that children follow, and ... misunderstands reversible passives during this progression." By creating a working model of the hypotheses developed through observations of children, Selfridge attempts

to explain the processes underlying children's performance and make predictions based on this explanation:

"The CHILD model suggests that children misunderstand reversible passives at stage two because they first learn active syntax, and use active syntax to understand reversible passive sentences. They understand non-reversible passives correctly because the semantic requirements override the syntactic features. Children then learn the passive set of features after hearing non-reversible passives, and form a new feature set. When they again hear a reversible passive... they understand the sentence correctly" (Selfridge, 1982, p.253).

Elliott and Lesk (1982) combined the study of human and program behavior in the area of planning. In their study, they compare human and computer processes in order to investigate the practical problem of finding the shortest path between two given points on a city map. They found that when the actual shortest routes discovered by the program were shown to raters, the routes were rejected as being too complicated. The program was then altered to incorporate human preferences such as fewer turns and concise directions, and proceeded to produce much more acceptable routes. Elliott and Lesk make a comparison between the program's strategies and the strategies of human subjects with varying levels of familiarity with the cities involved. They found that humans use a combination of depth-first search and a "divide and conquer" strategy, while for the program, depth-first search proved to be the most efficient.

2.3. The Need for Additional Research

In the psychological studies of children's development of planning strategies discussed above, computer models are simply a tool to define their theory and make predictions as to what children's behavior will be. The focus is not on the program as an independent planner, but on the human behavior

(Klahr, 1978, 1981, 1985). In AI, when a comparison is made between computer and human behavior, the approach is often to observe the behavior directly, write a program which models that behavior, and then hypothesize that since the end results are the same, so too might the means to arrive there be (Drescher,1987; Selfridge,1982). This experiment is an effort to turn both of these approaches around in some measure. From the psychological point of view, rather than using the computer as a model for predicting human behavior, human behavior will be used as a guideline for evaluating programming heuristics (as in Elliott and Lesk, 1982). From the AI view, the strategies used by both program and child will be compared, rather than the end results.

The intent of this experiment is to observe the developmental changes in 3- to 5-year-olds' block-stacking processes in order to discover the types of heuristics used at different ages. These heuristics will be compared to the heuristics known to be used in different planning programs in order to check the representational validity of these programs.

3. Method

3.1. Subjects

Subjects for this study were children at the day-care center in the Child Development Laboratory on the Virginia Tech campus. Forty-eight children between the ages of 3:6 (three years, six months) and 5:9 were available for study. As per regulations of the Human Subjects Research Committee, a consent form (found in Appendix 1) was sent home with the parents of the children which offered them choice of participation in the study.

At the time of the study, the children were also asked if they wished to "play the game". Those that did not were not required to participate. Out of the forty-eight children available, a total of 26 participated. Out of these 26, 22 children completed the task: 7 "threes" (3:7 - 4:1, mean age 3:10), 7 "fours"(4:2 - 4:7, mean age 4:4), and 8 "fives"(4:10 - 5:9, mean age 5:2).

3.2. Materials

Two identical sets of four children's building blocks were used. The blocks were cubes, 1 1/2 inches (4 cm) on a side, and painted in distinctive colors, one red, one yellow, one green, and one blue in each set. They were placed on a low, "child-height" table in front of children's chairs set up at right angles to one another. A video camera to record the procedure was mounted in a niche in one wall of the room where testing took place and was turned on once the child had entered the room.

3.3. Procedure

After entering the room, the child was allowed to choose which chair he or she wanted and to play with the blocks for a few moments. Once introductions and familiarization had taken place, the child was given the following instructions:

We are going to play a game with these blocks. Four of these blocks are yours, and four of these blocks are mine. (Experimenter places

one set of blocks in front of the subject and one set in front of herself.) What I'm going to do is pile up your blocks and pile up my blocks, and then you have to make your pile of blocks look just like mine, okay? You have to make our piles match. Now here are the rules to the game. the first one is easy: you can only touch your blocks. The second rule is, you can only use one hand to move your blocks. (This was to ensure that the child moved the blocks one at a time in succession.) And the third rule is, you have to keep your eyes shut when I pile up the blocks. Okay? Let's practice it a couple times. You close your eyes, and I fix the blocks in two piles, and then when I tell you to open your eyes you can use one hand to make your pile look just the same as my pile.

The child was then given three practice problems, in order to become familiarized with the materials and the rules of the game. During the practice trials, the child was given active assistance. The practice problems were followed immediately by six trial problems, during which the child was reminded of the rules only if he or she violated them. Table 1 shows the problem set, and Table 2 shows the order in which the children were tested and the order in which they received the problems.

4. Results

4.1. Coding

In analyzing the videotapes which have been made of each of the children, a protocol was created which listed in detail the moves the child made. The types of moves which were recorded include moving a block onto the table, moving a block onto another block, picking up a block and then replacing it in its former position, picking up a block and holding it over another block before replacing it in its former position, changing the position of a block on the table, and touching a block. A simple shorthand notation was developed for indicating these moves. Each of the blocks was represented by its capital letter initial, as was the table (R, Y, G, B and T). Types of manipulations were indicated by a small letter: m=move, t=touch, u=up, d=down, o=over. Using these symbols, the notations for the moves described above are:

moving a block onto the table: RT

moving a block onto another block: RY

picking up a block and then replacing it in its former position: Bu

picking up a block and holding it over another block: YoG

changing the position of a block on the table: mR

and touching a block: tB

Occasionally a child would move two blocks at a time. This was indicated by placing the two blocks moved together in parentheses: (BG)T. Other infrequent illegal moves included knocking down the tower or pulling a block from the middle of the tower. These were noted as "pull" or "dump". All illegal moves were undone, returning the blocks to their former position; this was noted as "undo", and all the moves which were undone were placed inside brackets. For example: [pull Y, dump B; undo] [(BG)T; undone by e.]. A time when the child would stop and check whether his or her answer was correct was indicated by a "...".

By using this grammar, it was possible to create an easily translatable string of moves for each problem done by each child. After a first coding, the coded data were cross-checked with the video tapes for errors and any errors found were corrected.

As discussed previously, the two central issues to planning are representation and heuristics. The heuristics which are chosen depend heavily on the type of representation used. It is virtually an impossible problem to decide where the line is to be drawn between items of information which are relevant to the problem representation and those which are irrelevant. Therefore, in creating this coding system, certain coding choices were made based on the assumption of a certain representation of the problem in the minds of the children. Certain moves which the child might have made were not recorded, such as hovering the hand over a certain block before choosing another to move, rotating the final tower to produce either a mirror image or a bilaterally symmetrical image of the goal state, or straightening the blocks in the tower. The resulting representation is based on the assumption that a move is not made unless a child makes physical contact with a block, and only certain kinds of contact are considered "relevant" to problem solution.

4.2. Analysis

Moves required. In the coding system above, each of the moves encoded can fall into one of two categories. The first two move types, placing a block onto another block and placing a block on the table, will be considered "effective moves" since these move types lead to the achievement of the goal. The other move types listed above can be considered "non-essential moves" since they do not affect the problem state in a relevant manner. Table 3 shows the mean number of total moves required and mean number of effective moves required by three-, four-, and five-year-olds to complete six-move, five-move, and four-move problems.

In comparing the number of moves (total and effective) required to complete each of the problem types, across all age groups, the mean number of

moves required follows a six<five>four pattern. For three-year-olds, effective moves required follows the same six<five>four pattern, whereas for four-year-olds and five-year-olds effective moves required follows a six>five>four pattern. All the age groups seemed to find five-move problems more difficult than either four-move or six-move problems in terms of making more moves overall to complete the problem, although they did perform as expected in taking more moves to complete the six-move problems than they did for the four-move problems. Three-year-olds followed this same pattern when we only consider the number of effective moves made. Four-year-olds and five-year-olds conformed much more closely to the ideal score for each problem when their mean number of effective moves is considered. Both of these age groups took more moves to complete six-move problems than five-move problems, and more moves to complete five-move problems than four-move problems.

Comparing the data across age of child, for six-move problems five-year-olds < four-year-olds < three-year-olds both on total and effective moves required. However, the difference between five-year-olds' and four-year-olds' effective moves required for six-move problems is so small (0.04 moves) that it can not be considered significant. For the five-move and four-move problems, however, five-year-olds > four-year-olds < three-year-olds, both on total and effective moves required. In other words, six-move problems had the expected performance result of five-year-olds doing better than four-year-olds, who did better than three-year-olds, on total moves required. Five-year-olds and four-year-olds took about the same number of effective moves to complete the six-move problem, both doing so in less effective moves than the three-year-olds. The five-move and four-move problems, however, produced interesting results: in both cases, five-year-olds did better than the three-year-olds but worse than the four-year-olds in terms of total moves required to complete the problems. These same differences hold true when one considers only the effective moves made; four-year-olds did better than either of the other two age groups, with five-year-olds doing better than three-year-olds.

In summary, five-move problems seemed in general to be most difficult, followed by six-move problems, and four-move problems were generally

easiest. Also, four-year-olds performed the best on four-move and five-move problems, five-year-olds performed best on six-move problems, and three-year-olds in general had the worst performance.

Backtracking. Table 4 shows the move sequences of children who used backtracking. Backtracking is defined as moving a block and then returning it to its original state, or more generally, making a series of moves and then reversing that series so that the blocks are returned to their original position before the move sequence was initiated. One move type, picking up a block and then putting it down again, can be considered the most basic instance of backtracking observable under this coding method.

For six-move problems, three-year-olds had five sequences which included backtracking(35%; total sequences=14); four-year-olds had one "backtracking sequence"(7%; total sequences=14); and five year-olds had two backtracking sequences(13%; total sequences=16). For five-move problems, three-year-olds had seven backtracking sequences (50%); four-year-olds had two backtracking sequences (14%); and five-year-olds had five backtracking sequences (31%). For four-move problems, three-year-olds had 3 backtracking sequences (21%); four-year-olds had two backtracking sequences (14%); and five-year-olds had three backtracking sequences (19%).

By examining Table 4, we can see that three-year-olds tended to use the most backtracking, and also had more instances of backtracking per backtracking sequence (an average of 1.8 overall) than either the four-year-olds (average 1.2 overall) or the five-year-olds (average 1.2 overall). Also, all of the three-year-olds at some point used backtracking, as opposed to two of the seven four-year-olds and four of the eight five-year-olds. Backtracking was used more frequently on five-move problems than on four- or six-move problems, which relates directly to the increase in total moves required to complete five-move problems noted above.

"Flattening". From observations made during the course of the study, it seems that there may be a distinct strategy which the children use which is not used in planning programs. This heuristic, termed "flattening," is defined loosely as unstacking the pile of blocks so that each block is on the table, with

nothing on top of it. The child then proceeds to build up the tower to the goal state.

Of the eighteen problems offered to the children, two of the four-move problems (3c and 3f; see Table 1) required non-flattening as part of their minimum-move solution. In other words, to complete these problems in the shortest possible move sequence, it was necessary that the children not flatten the tower. By random roll of the dice, four of the fourteen, five of the fourteen, and four of the sixteen problem presentations (to three-year-olds, four-year-olds and five-year-olds respectively) were "non-flattening" problems, for a total of thirteen out of 44 presentations of four-move problems. Of the thirteen sequences produced, only five were not flattened (one three-year-old, three four-year-olds, and one five-year-old). For eight out of the thirteen problem presentations, the children applied the flattening strategy to complete the move sequence for the problem.

As a further test of this strategy use, all of the five-move problems (2a-2f) were "flattening-optional": there were two possible minimum-move solutions, one of which required flattening and one of which required non-flattening of the tower. In every case, the child chose to use a flattening strategy. So, for eight out of eighteen problems presented, where non-flattening was either necessary or sufficient to complete the problem in the minimum number of moves, the children chose overwhelmingly (52 out of 57 times) to use a flattening strategy, even when it required more than the minimum number of moves required to complete the problem.

4.3. Discussion

Throughout this discussion it will be essential to remember that the small sample size ($n=22$, 3-year-olds: 7, four-year-olds: 7, five-year-olds: 8) permits only descriptive comparisons to be made between the children's and the planners' behavior. Nevertheless, several interesting results have appeared using this relatively simple form of analysis.

Problem length. First, the data concerning the number of moves required to complete a problem can be used to extend the information which

Klahr (1978, 1981) has given us. He found that the older children in his experiment are able to plan more moves at a time, and that the five-year-olds were able to solve six-move Towers of Hanoi problems. If Klahr's findings hold true, then based on their better ability to plan, the five-year-olds should complete more of the problems in the minimum number of moves. They should also take less circuitous routes in general, and therefore end up making less moves overall than the four- and three-year-olds. However, our results showed that the four-year-olds made less moves overall than either of the other two age groups. Two main factors contributed to these results: non-essential moves and backtracking. The four-year-olds made less non-essential moves than did the three-year-olds and five-year-olds, and they used much less backtracking than the other two groups. Since there was a developmental trend of improvement from the three-year-olds to the five-year-olds for both of these factors, it is tempting to point to the small sample size as the reason for the unexpectedly good performance of the four-year-olds. The use of less non-essential moves by five-year-olds than by three-year-olds supports Richards and Siegler's (1981) observation that young children become more systematic in their problem-solving strategies as they grow older, as well as Klahr's (1978) result that older children performed better than younger children on planning tasks.

Another result, which may have come about due to small sample size, concerns the difficulty of five-move problems as opposed to four-move problems. The five-move problems were more difficult for all age groups than either the six-move or the four-move problems. Again, the same two factors were the main reason for these results: the five-move problems contained more non-essential moves overall, and more instances of backtracking overall than either the six-move or the four-move problems.

The occurrence of non-essential moves in the move sequences of children contrasts sharply with the general type of move sequence of an AI planner, and brings up once again the issue of representation. For STRIPS, a blocks-world problem will result in a search through different states of the problem; for Noah and NonLin it will result in references to a library of plans. In all of these cases, someone (the programmer) has given the program the

rules for constructing a problem state, or the primitives of which plans are made. The relevant and irrelevant aspects of the problem are decided ahead of time and the structure of the program and data contains that information. Non-essential moves-- moves that have no relevant effect on the state of the world-- will not be produced.

For children (and adults as well), the dividing line between information critical to a problem and information irrelevant to a problem is unclear. Therefore, a child's representation of the blocks may include orientation of the stack, relative proximity of blocks (as opposed to either touching or not touching; the example would be holding a block over another block), and other moves, both recorded and unrecorded. All of these moves, defined as non-essential in the experiment, may in fact be an aid to the child to perceive the block stack in a different manner. For this age group, it would be extremely difficult to induce a closer approximation of the desired representation without inducing a great deal of confusion as well. It shall suffice to say that, based on the types of moves produced, the children use a different representation than either STRIPS, Noah or NonLin.

Different representations lead to the use of different heuristics. In this case, it would appear that the representation may not be too different, for there are some heuristics which the programs use which the children also use, including backtracking and creating subgoals.

Backtracking. As shown above, the use of backtracking influences directly the number of moves required to complete a problem. Making less moves to solve a problem correctly is generally considered a sign of competent planning; however, the total number of moves can be affected by the strategy used. First, it would seem that the older children would be more prone to use backtracking due to their ability to understand the reversibility of moves, thereby increasing the number of moves they would make, and younger children, being unable to understand the reversibility of moves, would tend toward hillclimbing in those cases where the older children would backtrack. Therefore, they would end up with less moves per problem than the older children (a $5 > 4 > 3$ pattern of results). This initial hypothesis regarding backtracking was not supported. Four- and five-year-olds were less

prone to use backtracking than the three-year-olds, and every three-year-old employed the strategy at some point. It appears that all the children within this age group are able to understand the reversibility of moves, at least at a concrete level.

Second, Klahr has shown in his 1985 study that children of this age tend to avoid backtracking. These findings are only partially supported in this study. Although older children were less likely to use backtracking than the youngest children, all age groups used backtracking, and the three-year-olds seemed to rely on it. Children from all age groups are able to use backtracking on the block-stacking problem. It may be that the complexity of the problem is correlated with the ability to apply the heuristic of backtracking. For example, consider the Rubik's cube: many adults can manage to solve one side, but are extremely reluctant to dismantle that side in order to complete more of the puzzle. This is the case of a highly complex problem which inhibits the ability to backtrack. The more complex structure of Klahr's (1985) problem as opposed to the present block-stacking problem may account for the difference in the amount of backtracking used in each of the two studies.

This use of backtracking is very similar to NonLin's reordering of primitives. When NonLin produces an output, it appears as a partially ordered list of moves; for example, if moves x, y and z need to be done, it might say, "do x before z; do x before y". It discovers this ordering by examining the moves' constraints as they are listed in the plan library. The final ordering will not show that at first NonLin might have attempted to put z before x, and then backtracked upon finding a constraint. However, this can be seen if the program is traced, showing step by step what it does as it runs. We cannot "trace" a child's mind as it "runs" a blocks-world problem, but because we assume behavior is the externalization of the thinking process, as mentioned in earlier sections, we can draw a parallel between NonLin's and the children's backtracking heuristic.

Another method which is used frequently as an attempt at "tracing" the human mind at work is introspection, or the description of the thought process. (See Klahr, 1978, 1981.) On a few occasions in this study a child would volunteer comments such as "I have to get the green on the bottom", or even

more complex, pointing at the various blocks in his stack and saying "This goes here, and that goes there, and this one goes up here," indicating the order and location he was to place the blocks. This verbalization of move sequences is possible evidence of a strategy like Noah's and NonLin's of looking up possible moves in their own internal "plan library" and even going so far as to impose some order on their plans.

The consistent ability to keep an internal plan library, or hold a series of moves in one's head, is a characteristic of the Piagetian concrete operational stage. A child enters the concrete operational stage after the pre-operational stage, at approximately five or six years of age. This type of verbalization contrasts with the behavior of some children who said they did not know what to do when confronted with the stack, but when encouraged to "try something" were able to solve the problem fairly easily. This shows the externalization of the thinking process characteristic of the pre-operational stage.

Flattening and STRIPS. Noah and NonLin-type strategies were not the only ones evidenced by the children. The strategy of flattening is comparable in a general sense to using the STRIPS strategy of creating a subgoal in order to achieve the goal state. It can be summarized as follows: "I have to get from Point A to Point B. It's too hard to remember everything I have to do to get there, but I know I can get from Elea to Point B. Therefore, I will go from Point A to Elea, and then I will go from Elea to Point B, because I can figure out the steps I have to make to travel a small distance much more easily." The children progress from the initial state to the flattened state, often pausing at this point to reconfigure the blocks on the surface of the table, and then proceed from the flattened state to the goal state. It is necessary to note that the specific heuristic "flattening" is not used by any of the three programs; only the basic strategy of creating a subgoal is similar to STRIPS's general heuristic.

In nearly every case where there was a choice between flattening a tower to achieve the goal state and following a non-flattening move sequence, children chose to use the flattening heuristic. Since ten of the eighteen problems required flattening as part of their move sequence, it is likely that the children were able to generalize the subgoal of "all blocks on the table" to

other problems as well. This generalization of simple steps may be the beginnings of systematic problem-solving in other areas, such as the problems of interest to Richards and Siegler (1981). Generalization of rules is also under investigation by AI researchers, such as Selfridge's (1982) program which learns to misunderstand reversible passives, mentioned in section 1.4.

Representational issues. Although the data above suggest the use of heuristics similar to those of the classic AI programs, there is also evidence in favor of a difference in representation between the children and AI planners. We have already discussed the issue of non-essential moves earlier in this section. Illegal move types and incorrect answers point out further representational differences.

The representation of a problem depends upon two main factors: prerequisite procedural and structural knowledge contained by the planner, and external information including problem presentation and resources. As closely as possible, we have tried to model the presentation of the problem to the children in a way that corresponds to the representation given to the planners. Also, the procedural rules given to the children were designed to correspond to the types of moves the programs are permitted to make. Despite these rules given to the children, during their trial problems many of them attempted to move two blocks as a unit, when those blocks were in the correct position relative to each other but not with respect to the rest of the stack. There are two possibilities for this type of behavior. First, the child may be drawing on additional procedural knowledge, using the following heuristic: "if there exist two or more blocks in the correct relative positions, maintain their correct relative positions at all costs." In effect the child has noted that a subgoal of the problem is already achieved and is acting accordingly to preserve the finished portion of the problem.

The second possibility for moving two blocks as a unit is that the child represents those two blocks not as "red block" and "yellow block" but as "red-and-yellow subtower" (for want of a better term). He moves the two blocks together because they are an inseparable segment of the problem. The distinction between these two types of representation may not be great, but there is a difference. In the first case, the child sees a "red part", a "green

part", a "blue part" and a "yellow part" and makes a decision about the manipulation of the stack based on all the parts' relative positions. In the second case, the child sees a "blue part", a "green part", and a "yellow-and-red part" and moves each of those parts according to the rules given.

Neither of these possibilities parallels the programs' behavior. The heuristic used in the first case is not programmed into the set of rules contained by the programs, and the programs are built to consider only one block at a time, not a pair or any other combination of blocks as a unit. It must be noted that the first case is very similar to flattening in that in broad terms, it can be described as recognizing the achievement of a subgoal and acting to maintain that subgoal, similar to the action of STRIPS. However, in neither case is the specific heuristic "flattening" or "moving together" used by that planner.

Another case of differing structural representation appeared in the data of a three-year-old girl who finished the problem incorrectly and, even after checking her answer against the experimenter's tower, believed her answer to be correct. Below is a diagrammatic representation of her answer and the intended goal state:

Betsy's tower:	RG	Experimenter's tower:	RG
	BY		YB

Betsy pointed out that red was next to green, and blue was next to yellow, and red-and-green was above blue-and-yellow. According to her representational requirements, this is a correct answer. Since she used the rules given to her at the beginning of the session correctly, it may be assumed that she organizes the goal state in a different manner; that her concepts of grouping and spatial relationship are different from those implemented in the programs (as well as those used by this experimenter). Because of this different representation of the problem state, the same manipulative rules produce different results.

5. Conclusion

5.1. Extension of Psychological Research

This research has extended the work of psychologists who have studied children's planning processes. The task used in this study differs from the tasks used in Richards and Siegler (1981) and Klahr (1978, 1981, 1985) and therefore has opened a new window into the types of strategies children will use when planning. Unlike the previous studies, there was not a linear improvement in performance as related to age; our four-year-olds performed best, followed by five-year-olds, followed by three-year-olds. Also, length of problem was not correlated with problem difficulty: all children had more trouble with the five move problems than they did with the six- or four-move problems. It is possible that these results are due to small sample size. Further study using a larger sample is indicated in order to check the reliability of these results, but as the saying goes, "This is beyond the scope of this research."

It has also been demonstrated that task structure relates to the strategy used. Klahr (1985) found evidence of resistance to backtracking on his highly constrained "dog-cat-mouse" game. With this less structured task, backtracking was used more freely, especially by the younger children. This is in support of the results found by Siegler (1978) with an older age group. With these findings, this research makes a good contribution to the body of psychological literature on planning and problem-solving.

5.2. Relationships Between AI Planners and Children

There are three categories of strategies which have come to light as a result of this study. The first comprises those strategies which are used by both the AI planners and the children; the second, those which the planners use which are not used by the children, and the third, those which the children use, and which the planners do not.

First, children use some heuristics similar to those used by planners. The specific strategies of "flattening" and moving two blocks at the same time are evidence for a more general strategy of subgoal formation, which is a basic

operating strategy of STRIPS. This type of general heuristic was used by most of the children, regardless of age. Noah and NonLin both impose a partial ordering on primitive moves; two four-year-olds and one five-year-old recited all or part of their plans before making a move, indicating that they were fixing the order of their actions by the use of such words as "first... then..." and "do...before...." Finally, NonLin backtracks when an error in the order of moves is found. All of the three-year-olds, two four-year-olds, and four five-year-olds used backtracking, indicating that there may be a developmental trend in the use of this strategy, as well as the relationship between backtracking and problem representation mentioned above.

Of the second category of strategy, it may be noted that in this study there was no evidence of Noah's hillclimbing behavior. All children were able to complete the problems given. The child who completed the problem incorrectly cannot be considered a case of hillclimbing; hillclimbing is "getting stuck" or being unable to finish the problem, and this child finished the problem correctly according to her description of the goal state, even though her final representation did not coincide with the "correct" representation.

Children also use specific strategies not used by planners. The two observed in the behavior of these children were flattening and moving two blocks at same time. The usefulness of these strategies depends on the larger goal of the planning process. If the intent is to produce a correct answer in relatively little time, without accounting for the number of moves, then flattening is extremely useful. A planner could, for example, instruct a robot arm to place all items on the floor so that they are all equally accessible. Then the robot could assemble the parts of a machine in the proper order with less risk of neglecting to put the widget under the whatzit because the widget was under the whozit to begin with (!). However, if it is necessary to minimize the number of moves made, such as when there are delicate parts to be moved as little as possible, moving two pieces at once may be more helpful. If a frimfram and a fribbin are already together and calibrated, It would be wiser to keep them together rather than risk an error in recalibration.

Preliminary evidence exists for a difference in internal representation of the problem. Non-essential moves, introducing illegal move types, and the single instance of an incorrect answer all point to a different structure used by the children than is used in any of the programs discussed. Based on all the data, for the blocks-world problem, STRIPS, Noah and NonLin have low psychological validity where representation is concerned. STRIPS paints a fairly accurate picture of large-scale problem-solving strategies for all ages, NonLin uses one heuristic similar to that of younger preschoolers (backtracking) and another, more systematic, heuristic (subgoal ordering) evidenced by a few older children. Noah has the lowest validity of all in terms of block-stacking heuristics.

5.3. Further Research

There are many directions to go from this point; as usual, one piece of research has opened a Pandora's Box of additional research questions which, interesting though they may be, cannot all be answered within these pages. The first step to take from this point would be to replicate the study with a larger sample size and perhaps more problems presented to each child. After this, the next step depends on the particular interests of the next researcher. Here are some ideas for consideration.

It was mentioned that flattening was necessary for the completion of ten out of the eighteen problems given to the children. It is likely that from this, the children were able to generalize the use of flattening to almost every problem. How many examples of a heuristic are necessary before that heuristic will be used for all problems similar to the example problems? The generalization of a strategy (such as flattening) will likely depend on the complexity of the strategy, the complexity of the problem, and the amount of exposure to the problem.

On the trial problems, many of the children tried to "bend the rules" in order to arrive at the goal state. Rather than giving the child a rigid set of rules, simply present the initial and goal states and observe the types of heuristics the child creates. From this point one could compare these

heuristics to heuristics implemented in these and other AI planners, or investigate the results of their actual implementation if one is so inclined.

The dividing line between representation and heuristic is not clear; in particular, it is not possible from the data in this study to determine whether attempting to move two blocks at once is a result of the representation or of the heuristics used (see Section 4.3, "representational issues"). Is it possible to tell in what way, and to what extent, representation and heuristics each influence this strategy?

Much of the other research mentioned in this work considered the external behavior of humans and from this developed an inference as to the heuristics behind that behavior (Drescher, 1987; Klahr, 1978; Osterreith, 1983; Richards and Siegler, 1981; Selfridge, 1982). This research has looked at move sequences in order to directly determine the patterns which can be considered heuristics (similar to Elliott & Lesk, 1982, and Klahr, 1985). From this point, it appears that there is a more basic level for comparison, which has turned out to be problem representation. Until there is a more accurate way to determine the actual representations used by humans, it will not be possible to compare and/or emulate that representation in order to improve the state of the art in Artificial Intelligence.

While psychologists continue to investigate problem representation in humans, it remains necessary to continue to compare our heuristics to programming heuristics, to evaluate programs' efficiency and validity, and to implement human heuristics. What this will mean for researchers in Artificial Intelligence is a greater awareness of the limitations of the different types of programs, a greater awareness of useful heuristics, and in the global sense, a greater awareness of the meaning of the term "intelligence" as it applies to the field.

References

- Charniak, Eugene and McDermott, Drew (1985). *Introduction to Artificial Intelligence*. Reading, MA: Addison-Wesley Publishing Company.
- Drescher, Gary L. (1987). A mechanism for early Piagetian learning. *Proceedings of AAAI 1987*, 290-294.
- Elliot, R. J. and Lesk, M. E. (1982). Route finding in street maps by computers and people. *Proceedings of AAAI 1982*, 258-261.
- Ellis, Henry C. and Hunt, R. Reed (1983). *Fundamentals of Human Memory and Cognition*. Dubuque, IA: Wm. C. Brown Company Publishers.
- Fikes, Richard E. and Nilsson, Nils J. (1971). STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2, 189-208.
- Fredriksen, Norman (1984). Implications of cognitive theory for instruction in problem-solving. *Review of Educational Research*, 54, 363-407.
- Funt, Brian V. (1980). Problem-solving with diagrammatic representations. *Artificial Intelligence*, 13, 201-230.
- Klahr, David (1978). Goal formation, planning, and learning by preschool problem-solvers or: "My socks are in the dryer". In R. S. Siegler, (Ed.), *Children's Thinking: What develops?* Hillsdale, NJ: Erlbaum, 181-212.
- Klahr, David (1985). Solving problems with ambiguous subgoal ordering: preschoolers' performance. *Child Development*, 56, 940-952.
- Klahr, David and Robinson, Mitchell (1981). Formal assessment of problem-solving and planning processes in preschool children. *Cognitive Psychology*, 13, 113-148.
- Lerner, Richard M. (1986). *Concepts and Theories of Human Development*. New York: Random House.
- Osterreith, Paul (1983). *Introduction à la Psychologie de l'Enfant*. Paris: Presses Universitaires de France.
- Richards, D. Dean and Siegler, Robert S. (1981). Very young children's acquisition of systematic problem-solving strategies. *Child Development*, 52, 1318-1321.

- Sacerdoti, Earl (1975). The nonlinear nature of plans. Proceedings of IJCAI 4, 206-214.
- Selfridge, Mallory (1982). Why do children misunderstand reversible passives? The CHILD program learns to understand passive sentences. Proceedings of AAAI 1982, 251-253.
- Siegler, Robert S. (1977). The twenty questions game as a form of problem-solving. Child Development, 48, 395-403.
- Siegler, Robert S. (1978). Children's Thinking: What Develops? Hillsdale, NJ: Erlbaum.
- Tate, Austin (1977). Generation project networks. Proceedings of IJCAI 5, 888-893.

Appendix 1

Letter of consent sent home to parents of children who were offered participation in the study.

To the parent/guardian of _____

From: Alison Johnson

Dear parent/guardian,

I am currently working on my master's thesis in Educational Psychology at Virginia Tech, and will be conducting my research at the school your child is attending. In my research I am looking at the way children 3 to 5 years old plan a series of steps to accomplish a task. I will be using a simple block-stacking game, in which I show the child two stacks of blocks and tell him or her to try to "make this stack look just like that stack." By having the child use just one hand to move the blocks one by one, I will be able to see the order of the moves he or she makes to match one stack to another.

Each child and I will play this "game" alone together in a quiet room at the school, and the entire session will be videotaped. In some preliminary trials of the research, children have enjoyed playing the game with me, and since there is no "right " or "wrong" way to do it, there is very little pressure. If a child does happen to become uncomfortable during the session and doesn't wish to participate, he or she will be returned to the classroom without delay. Also, should there be any changes in the procedure, you will be informed and have the option of withdrawing your child from participation if you wish.

We will assume that you have given permission for your child to participate in this research project unless you indicate otherwise. If you do not wish your child to participate in this activity, please contact the Child Development Lab at ..., before Wednesday, April 27, 1988.

Should you have any questions, I can be reached at the above address and phone number every evening except Wednesday. This project is being supervised by Dr. Terry M. Wildman. You may contact him if you have any questions you do not wish to ask me. His phone number is This project has been reviewed and approved by the Division Human Subjects Research Committee.

Thank you very much for your consideration.

Table 1

Problem set. The initial and the goal states may be switched. The problem number in parentheses refers to the problem when reversed. Letters do not refer to any specific color of block, except that letter-color consistency must be maintained.

Group & Problem Pair	Initial State	Goal state
Group I: Six-move Problems	D	A
	C	C
Ia (Id)	A	D
	B	B
	D	D
	B	A
Ib (Ie)	A	C
	C	B
	D	A
	A	B
Ic (If)	B	C
	C	D
Group II: Five-move Problems		A
	C	D
	B	C
IIa (IIId)	AD	B
	B	C
	D	B
IIb (IIe)	AC	A
		D
	B	A
	D	C
IIc (IIIf)	CA	B
		D
Group III: Four-move Problems		D
	AC	B
IIIa (IIIId)	BD	CA
	B	C
	C	A
IIIb (IIIe)	AD	DB
	C	
	B	
	A	AD
IIIc (IIIIf)	D	BC

Table 2

Name, age, and problem set for each child who participated in the study, in order of participation. Problems are counterbalanced by group (1, 2 or 3); problem pair (a, b, c, d, e or f) is determined by the roll of a six-sided die.

NAME	AGE (5/1/88)	PROBLEM SET
DANIEL B.	4:3	1a,3a,2a, 1e,3e,2e, 1f,3f,2f
ASHLEY C.	3:9	2c,3c,1f, 2a,3a,1a, 2b,3b,1b
STAVROULA H.	4:7	2b,3b,1b, 2a,3a,1a, 2f,3f,1f
WIL D.	5:9	3f,2e,1f, 3a,1a,2a, 3b,1b,2e
MARK G.	5:0	3f,2f,1f, 3a,2d,1a, 3b,2b,1b
ENMAR H.	5:1	1d,3d,2d, 1b,3b,2b, 1f,3f,2f
EMILY B.	4:4	2a,3a,1a, 2e,3e,1e, 2c,3c,1c
BETSY C.	3:7	3a,1a,2a, 3b,1b,2b, 3c,1c,2c
GEORGE G.	4:4	1c,2c,3c, 1a,2a,3a, 1e,2e,3e
KENDRA S.	4:2	3f,2f,1f, 3d,2d,1d, 3b,2b,1b
MICAH V.	3:10	2a,1a,3a, 2e,1e,3e, 2c,1c,3c
BRIAN C.	3:9	3f,2c,1c, 3a,2a,1a, 3b,2b,1b
KAREN G.	4:10	1b,3b,2b, 1d,3d,2d, 1f,3f,2f
EMILY A.	5:0	3c,2c,1c, 3d,2d,1d, 3b,2b,1b
HANNAH W.	5:3	2c,3b,1b, 2a,3a,1a, 2b,3c,1c
SCOTT G.	5:6	1b,2b,3b, 1a,2a,3a, 1c,2c,3c
ALICIA S.	4:11	1c,2c,3c, 3a,2a,1a, 3e,2e,1e
TRAVIS W.	3:10	2c,3c,1c, 2a,3d,1d, 2b,3b,1b
KYLE J.	4:6	3e,1e,2e, 3d,1d,2d, 3f,1f,2f
WILLIAM B.	4:3	1b,3b,2b, 1a,3a,2a, 1c,3c,2c
BETH D.	4:1	2a,1a,3a, 2b,1b,3b, 2c,1c,3c
BRIAN W.	4:0	2a,1a,3a, 2b,1b,3b, 2c,1c,3c

Table 3

Mean number of total moves required and mean number of effective moves required by three-, four-, and five-year-olds to complete six-move, five-move, and four-move problems.

<u>Problem type</u>	Age of child			Mean overall
	3 years	4 years	5 years	
six-move				
total	8.50	7.71	6.88	7.70
effective	6.50	6.29	6.25	6.35
five-move				
total	9.29	7.71	8.06	8.35
effective	6.86	5.14	5.69	5.90
four-move				
total	7.76	6.57	7.44	7.26
effective	4.86	4.21	4.69	4.59
Mean overall				
total	8.52	7.33	7.46	
effective	6.07	5.21	5.54	

Table 4

Move sequences of children who used the heuristic of backtracking.
Backtracking is underlined.

SIX-MOVE PROBLEMS

Three-year-olds		
BETSY C.	1b:	GT mG <u>GY GT</u> YT mY mG mY m(RB) <u>RY RT BY GB GT</u> RB GR
BETSY C.	1c:	GT RT mG <u>Rud (YB)ud</u> YG YT BG <u>Rud</u> YB RY
ASHLEY C.	1a:	GT BT RT GY BG <u>Rud</u> tB RB
MICAH V.	1e:	GT RT BT <u>YoB YT</u> RoB RB tG YR GY
MICAH V.	1c:	GT RT <u>GY GT (YB)G (YB)T</u> YT BG <u>Rud</u> YB RY
Four-year-olds		
GEORGE G.	1e:	GT RT BT <u>GB GT RB GR GT</u> YR GY
Five-year-olds		
ENMAR H.	1f:	RT YT BT tG <u>GB GT</u> YB RY GR
WIL D.	1a:	GT BT RT <u>RY RT</u> GY BG RB

FIVE-MOVE PROBLEMS

Three-year-olds		
BETSY C.	2b:	<u>BY ... BT</u> YT GT RG m(RG) <u>BR BT</u> mB YR BY
BETSY C.	2c:	YT <u>RG RT YG RY ... RT YT</u> GT mR mB <u>RG RT YG RY</u> <u>RT BY RB</u>
BRIAN C.	2a:	mG <u>BG YB RY ... RT YT BT</u> tY BY GB RG
TRAVIS W.	2a:	tR BT YT mR tB <u>GY tR BG BT GT</u> BY GB RG
BRIAN W.	2b:	<u>(YG)T (YG)R</u> YT GT mR <u>YG YT</u> RG YR BY
BRIAN W.	2c:	YT GT mR <u>Bud</u> YG BY RB
BETH D.	2c:	mR YT GT tR <u>BG BT</u> YG BY RB
Four-year-olds		
DANIEL B.	2f:	RT BT t(YG) mB YT mB <u>Yud</u> GB YG mR
GEORGE G.	2e:	BT YT RT mB m(RY) mG mB mR GR <u>BG BT</u> YG mB
Five-year-olds		
MARK G.	2d:	RT mR [(BG)T; undone by e.] GT BT mY mG mR <u>Bud</u> YR BY
EMILY A.	2b:	mB YT GT YoR <u>YR YT</u> RG ("get G on bottom") YR BY
ENMAR H.	2b:	YT GT mR ... <u>BG BT</u> RG YR BY
ENMAR H.	2f:	RT BT YT GB <u>RG RT</u> YG
WIL D.	2a:	tR BT YT <u>RG tB RT GB GY GT</u> BY GB RG

FOUR-MOVE PROBLEMS

Three-year-olds

ASHLEY C. 3a: RT mY YB YT tB BT YG YT mG mB mG YB GY

TRAVIS W. 3d: GT YT tR mY mR RY GB ... GT BG

TRAVIS W. 3b: YT BT RG BR YB YT

Four-year-olds

DANIEL B 3e: BT YT RT Yud GR mB YG

DANIEL B. 3f: GT tY (RYB)ud RG YR BY

Five-year-olds

ENMAR H. 3b: YT BT YR YT mR mY BY BT RY BR(gs:GT, YT,RY, BR)

WIL D. 3a: RT BT GY mB [(GY)B: undone by e.] GB GT YB GY mR

WIL D. 3b: YT BT Gud mG tY mR mB RY BR

**The vita has been removed from
the scanned document**