AN ARTIFICIAL INTELLIGENCE SYSTEM TO MODEL AND GUIDE CHEMICAL SYNTHESIS PLANNING BY COMPUTER: A Proposal

by: N.S. Sridharan

DCS-TR #43

Department of Computer Science
Rutgers, The State University of New Jersey
New Brunswick, New Jersey 08903

Department of Computer Science Technical Report #43
June 1976

AN ARTIFICIAL INTELLIGENCE SYSTEM TO MODEL AND GUIDE CHEMICAL SYNTHESIS PLANNING BY COMPUTER: A Proposal

N.S.SRIDHARAN

Rutgers University, Computer Science Department
New Brunswick, New Jersey 08903

One of the central problems in applying computer methods to chemical synthesis planning is Search Guidance. The ability of a chemist to guide an interactive computer in its search for syntheses and the possibilities for self-guidance in Artificial Intelligence programs are both limited by the form and content of the search information that is made available as the exploration proceeds. A system for Search Modelling is proposed in this paper which can augment existing systems for synthesis planning and serve to gather, analyze and amplify the information generated during controlled exploration. The search management model is specified in a simple descriptive form and two example models are included in the paper. The guidance of search using the information gathered is specified by a rule set in the simple syntax of Condition=>Action pairs. A chemist can interactively modify this rule set.

Further, an advanced search model is presented which, by introducing the powerful concept of a Planning Space, allows the search for syntheses to go forward, backward and to leap into the middle under controlled conditions.

[Invited paper presented at a Symposium on "Computer-Assisted Organic Synthesis Planning" held at the Centennial Meeting of the American Chemical Society, New York City, N.Y., April 6-9, 1976. This Symposium was sponsored jointly by the ACS Divisions of Chemical Information and Computers in Chemistry.]

INTRODUCTION

Planning chemical synthesis routes for known molecular structures is a rich problem area offering a challenge that is being met in inventive and imaginative ways not only by chemists, but also by computer scientists and mathematicians. I bring to this task the perspective of a specialist in the methods of developing Artificial Intelligence within a computing system, and a persistent concern for using the mechanisable aspects of human knowledge and human problem solving techniques in the medium of a machine. I bring to this task no expertise in organic chemistry or in synthesis. But I do have the benefit of several years of intimate contact with the problem of mechanizing the search for syntheses and with expert chemists, especially Prof. W.F. Fowler, who foresaw such possibilities and who worked with us. My application to this task under the guidance of Prof. Gelernter culminated during 1971 in the running version of SYNCHEM I, the first computer program to perform successful multi-step synthesis explorations automatically without on-line guidance or intervention. This first version of SYNCHEM employed several key ideas and techniques that were discovered by others early in the research on mechanical problem solving. These key ideas will be reviewed below.

Since my main interest lies in the direction of artificial intelligence, I have spent the five years following my work in SYNCHEM in working on the application of artificial intelligence methods to problems in Mass Spectrometry with the Heuristic Dendral project at Stanford University and also with Professor Ivar Ugi at Munich in assimilating his algebraic approach to the representation of reactions. Upon Todd Wipke's invitation to participate in this Symposium I have elected to set forth in a very specific manner my proposals on how I would tackle the task of synthesis planning in the light of my current understanding of the advances that have been made in the methods of artificial intelligence.

Thus, I have three aims in writing this paper:

a) Propose a system to augment existing synthesis search systems by focusing on the issues of search management. To this end the concept of search modelling is introduced and two search models are presented.

b) Clarify the fine distinction between selection of transform by Relevance criteria and by Applicability criteria. Selection by Applicability defines what is usually called the State Space and search in the state space usually grows the synthesis path uniformly from one end only. Selection by Relevance is not used by any existing system which when used yields the powerful Planning Space. The search in a planning space can take "leaps" along the synthesis sequence.

c) Indicate that a combination of search in both the state space and the planning space is possible and that this a function of the search model employed. The second search model described in the paper allows the search for synthesis to go forward, backward and to leap into the middle under controlled conditions.

It is hoped that the advantages of this way of developing a system will include: upgrading the role of the chemist from one of rating, pruning and selecting subgoals or precursors to that of giving injunctions to the system in the form of rules added, removed or modified as the search proceeds a few steps at a time; the introduction of the strategic and judgmental knowledge in the program in a manner that decoupled from the knowledge of reaction chemistry; and the ability to experiment easily with various models of search management that are qualitatively different from each other. The payoffs foreseen are of genuine importance to those of us concerned with the techniques of artificial intelligence and when proven practicable should be exciting and challenging to chemists as well.

It must be stated at the outset that these new ideas on search modelling are an outgrowth of my attempts to develop a system for common sense reasoning about human actions using a psychological theory of act interpretation [Schmidt, 1976; Sridharan, 1975; 1976]. This system has capabilities

of selecting and applying transforms associated with act names, structuring them into a plan and reasoning both forward and backward along the plan structure. The strategy of act interpretation is coded in the form of rule sets. The framework adapted for this work is called Meta Description System (MDS) [Srinivasan, 1973; 1976], designed and developed by Srinivasan. I shall explore here in detail the use of this framework in the management of search for chemical synthesis. The connections between the psychological theory and synthesis planning strategies will be left for later exposition.

There are three major conceptual approaches [See Sridharan, 1974 for a review] that have been taken on the issue of a computer mediated design of chemical synthesis plans and they share a central idea among them - that of searching a space of possibilities in a <u>systematic manner</u> using empirical knowledge as appropriate. The very power of these approaches stems from the systematization of search of the space of possibilities.

For those approaching synthesis as fertile grounds for designing and building computer programs that solve difficult intellectual problems [Sridharan, 1971, 1973, 1974; Gelernter, 1973, 1976] the main problems are twofold. First, the acquiring and packaging of knowledge of the reactions in a form suitable for use within a program has to be done carefully and the success of the program depends upon the correctness and extent of the knowledge base. Second, the techniques of conducting search with an incomplete, uncertain and possibly inconsistent knowledge base have to be customized to the task of chemical synthesis.

The interactive problem solving concepts developed [Corey, 1969; Wipke, 1973, 1976] are attractive because of their thoroughness and the chemist user tends to approach the system with hopes that the disciplined exploration of pathways will ensure him that he has not overlooked any reasonable alternative.

The third approach taken to computer methods in chemical synthesis is one of formalizing the set of possible reactions [Ugi, 1976], limiting the use of empirical knowledge to the selection rules for these reactions. This method seeks out novel reaction schemes and can suggest to the chemist routes that could not be found by the other methods. However, a reasonable synthesis program is yet to emerge from this approach. The algebraic characterization of the search space however offers the potential of highly interesting structures to be defined on the search space and might be very successful in the long run.

All of these methods currently utilize a search structure generally called the Heuristic Search Method.

THE HEURISTIC SEARCH METHOD

The Heuristic Search Method [Nilsson, 1971; Newell & Simon, 1972] is a usual first approach to problem solving if the specification of the problem itself is given precisely as a Goal Situation, and the solution required is some sequence of Transforms i.e. Operators that can effectively transform the current situation to the goal situation. It is important that the allowable operators are finite and are well-defined. The problem solving procedure involves the use of a Search Model that is used in guiding the search for a solution. The heuristic nature of the procedure arises from the use of approximate methods of evaluating progress towards a solution and of assessing the merit and potential of any partial solution sequence. The user gives up in principle the completeness and optimality of the search process gaining in fact more frequent demonstrations of successful problem solving [Newell & Simon, 1972].

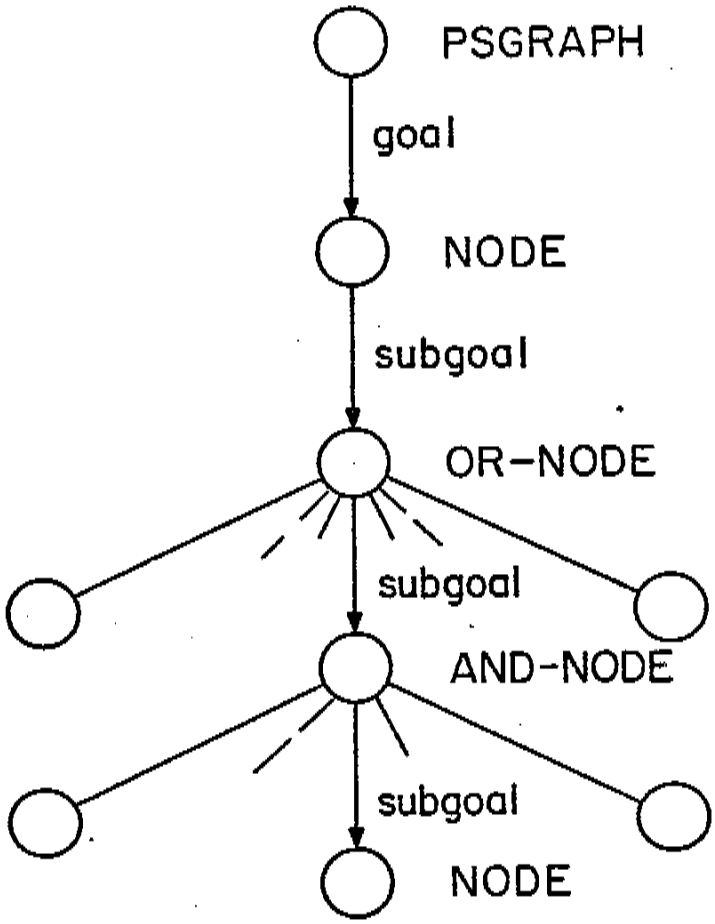
In chemical synthesis the goal state is the target molecular structure and the operators to consider for constructing a solution sequence are molecular reactions. The search for a synthesis plan proceeds backwards from the target molecular structure by considering and applying reactions in the retro-synthetic direction. The process is a serial one and has an "inner loop" that repeatedly asks

itself the questions "Should the process stop now?" and "What next?". The process can be successful, interesting and powerful depending on the use of a proper Search Model to answer these questions.

The answer to the question "What next?" comes in two parts: the choice of a molecular structure that can be set up as a subgoal to complete the synthesis, and the choice of the operator to try on that subgoal. The information available to the process in arriving at the answers is of two kinds, both of which must be included in the search model. The first kind comprises the catalog of starting materials, the library of reactions, tests of applicability of the reactions, the a priori merit rating for the goodness (yield, specificity etc.) of the reaction and includes in general information that is made available to the system prior to the actual statement of the problem to solve. The nature and extent of this prior information determines the structure of the search possibilities. The other kind includes information that becomes available as the search proceeds and constitutes a rich body of information that is specific to the problem at hand. Only limited examples of the use of the second kind of information can be shown in current programs, perhaps the clearest one is the placement of protection reactions for sensitive functional groups. The manner in which such problem specific information is collected, organized and used to guide search determines the character of the actual search performed for a given problem.

THE PROBLEM SOLVING GRAPH AS A MINIMAL SEARCH MODEL.

The Problem Solving Graph (PSG) [Gelernter, 1962] is a graphical model of the dynamics of the search that is conducted on a given problem, and consists of a root node representing the target molecule linked to a cascade of descendants which are one-level precursors to those at a level higher in the PSG. The Heuristic Evaluation function assigns numerical weights (or in some cases elements of an ad hoc scale of discrete values) to each node in the PSG. Typically, the node evaluation is based not only on properties of the situation designated by the node and



AND-OR PROBLEM SOLVING GRAPH MODEL

FIGURE 1

the operator applied to generate the node, but also on the entire path from the goal node to the node in question. The PSG model in combination with the heuristic value assignments provides the search mechanism ready answers to the "What next?" question.

The structure of the PSG for the chemical synthesis problem, shown in Figure 1, consists of AND-nodes when operators generate multiple precursors all of which need to be made available for the reaction to be successfully executed, and OR-nodes designating the choice available among several operators applicable to a node. The selection criteria that are well entrenched in the Theory of Heuristic Search [Slagle, 1971] for handling such AND-OR problem solving graphs are:

> At an OR-node select the most promising subgoal;
>
> At an AND-node select subgoals starting from the most likely to fail to the least likely to fail.

INFORMATION-GATHERING AS COMPLEMENTARY ACTIVITY TO HEURISTIC SEARCH.

The planning activity involves a variety of decisions that require information not customarily included in the initial specifications of a transform. Such decisions involving <u>reasoning about the search process</u> beyond the information provided by the PSG model call for a more elaborate search model and is considered in this section.

The PSG model may be viewed as a collection of ready answers to the following set of questions:

a) Does a node have subgoals? How many? What are they?

b) What is the status of a node? Was a successful path found? Was it tried and failed on all paths? Are there descendant subgoals still open?

c) Does the situation (i.e. molecule) represented in

this node occur elsewhere in the PSG? Is the situation circular i.e. calling for synthesizing X to synthesize X higher up?

d) At an OR-node what is the best subgoal to take up? At an AND-node what is the precursor that should be tackled first?

Now consider the following set of questions that go beyond the information maintained by the PSG model:

a) When there is an operator whose relevance to synthesizing a molecule is clear, but the applicability conditions are not satisfied, under what set of circumstances should the unsatisfied preconditions be made into subgoals? Under what conditions should the operator be rejected altogether?

b) For a given molecule what is the most strategic sequence for the introduction of functional groups?

c) When is the sequence of functional group introduction immaterial?

d) Given a subgoal, can the paths explored for another structurally homologous structure be considered valid here?

e) Given a synthesis route (or partial route) involving a protection/unprotection reaction pair, should an attempt be made to derive a revised route not involving protection by resequencing some of the reactions?

Answering these questions calls for maintaining a richer and better organized base of information about the search paths than that provided by the PSG and its heuristic value assignments to the nodes.

As an alternative to the Heuristic Search process, consider the following two-stage process:

a) Perform some exploration in the search space (be it the State Space of the Heuristic Search described above, or the Planning Space to be discussed below).

b) Gather the search information, analyze, amplify and

use it to guide further exploration.

A great deal of flexibility and investigative power comes to us if we separate the total system into two components giving explicit charge of the exploration by search to one, and the analysis and assimilation of the search information to the other. We gain conceptual clarity in thinking about the rules for search guidance and set about designing novel Search Models with a new ease and vigor. I will describe briefly the information gathering system which has been developed at Rutgers and show by example a novel form of search model in the remaining sections of the paper.

META-DESCRIPTION SYSTEM:
A PARADIGM FOR INFORMATION-GATHERING The structure of a system described in the facility of MDS [Srinivasan, 1973 & 1976; Sridharan, 1975] concepts is radically different from the procedure based systems to which we are accustomed. It is more favorable, therefore, to introduce the system directly by an example.

Table I presents the STRUCTURAL DESCRIPTIONS of the classes of entities that are involved in building the PSG model. The class PSGRAPH designates the Problem Solving Graph whose ELEMENTS are NODES. There are two important classes that help structure collections of NODES into OR-NODES and AND-NODES. The latter three classes have MERIT and STATUS relations associated with them, shown in the Table relating these nodes to INTEGER values for MERIT, and a class called STATUS for the STATUS relation. There are three values of STATUS defined as constants viz., OPEN, FAILED and SUCCEEDED. The PSGRAPH has a GOAL which is a NODE and a collection of open nodes and failed nodes. The TRIALNODE designates the node to take up as subgoal when the search process is set to explore the space again. The PSGRAPH has a relation STATUS which is intended to indicate the conditions under which the process should terminate.

NODES are related to the OR-NODEs via the SUBGOAL relation, the OR-NODEs indicate CHOICEs of AND-NODEs and the AND-NODES in turn INCLUDE any number of NODES

thus clearly exhibiting the AND/OR nature of the graph.

Let us turn our attention briefly to the SENSE DEFINITIONS which are specifications of the Logical conditions that are to be met for asserting the various relations and are at the same time specifications of the computations to be performed if the system is given the responsibility to fill in values for certain relations. Simple definitions are given for the OPENNODES and FAILEDNODES relations of the PSGRAPH class. The OPENNODES are the set of all NODEs N which are ELEMENTS of X (denoting the PSGRAPH) whose STATUS is OPEN. The STATUS of the PSGRAPH is defined to be the status of the goal node of the PSGRAPH written
[(STATUS S) ¦ (X (goal status) S)].
The nature of the AND-NODES and the OR-NODES is clearly spelled out in the definitions for the STATUS relations on these classes. The definition for the status of the AND-NODE may be paraphrased into English as follows:

a) If X includes a node whose status is FAILED then the status of X is also FAILED;

b) If X status is not FAILED and X includes an OPEN node then the status of X is OPEN;

finally, c) If X is neither OPEN nor FAILED then it must be SUCCEEDED.

The information displayed in Table I is the description the user provides to the Information-gathering system as the specification of the Search Model. The system accepts the Structural Descriptions and sets up Data Structures and access functions for each of the relations. The Sense Definitions are analyzed to compile a Network of Information Flow [Sridharan, 1976] that prescribes the data flow paths when a new piece of information is made available to this system. This much could be termed the "compile-time" activity of the system.

The planning and problem solving is initiated and controlled by a set of rules that we shall examine presently. The initialization is straightforward and involves creating an instance of PSGRAPH and filling its goal node. This indicates that it is the specification of the goal that triggers the process. The specification of the goal node involves submitting the structure of the molecule to be synthesized.

Turning our attention away from the Rule set that controls the problem solving process, let us consider the information gathering activity caused by the addition of a subgoal node for some operator explored by the search component of the system. This may be specified as a conjunction of precursor molecules which are INCLUDED in an AND-NODE. Consider the action taken when one of the precursors is asserted as a SUBGOAL of the GOAL node. The check of the condition for the (NODE subgoalof NODE) relation indicates that this AND-NODE needs to be introduced as a choice in the OR-NODE pointed to by the GOAL node [this is indicated by the expression (A (subgoal choice includes) X)]. The consequent assertion of the CHOICE relation flows along its data flow path to the STATUS relation of the OR-NODE in question. It is appropriate to point out here that this data flow link was "compiled" when the definition of STATUS was scanned and it was established then that any additions/changes to the CHOICE of an OR-NODE was to take effect in turn on the STATUS relation. A verbal description such as this one cannot describe all the data flow that takes place but hopefully the above explanation conveys the concept of the data flow and consequent "information-gathering" proceeding as per the structural and sense definitions of the Search Model given by the user for this problem domain.

At this point, if the reader will grant that as the results of the exploration conducted by the search component gets fed to the Information-Gathering component the requisite search model will be created or updated as appropriate, we can turn our attention back again to the Search Guidance Rules.

The Rules are written in what is known in the computer science lingo as the "Production Rule Form" [Davis & King, 1975].

The production system takes a sequence of conditional action specifications of the form:

RULE: (Condition to check)=>(Action to take)

In applying one of these rules the left-hand side of the Rule is first tested against the current state of the model and if the test is satisfied the actions of the right side of the rule are performed. There are a variety of rule sequencing methods conceivable, but we shall use only the simplest of them here. The control starts from the beginning of the rule sequence and tries each rule and cycles back to the first rule after the last rule is tried. The execution of a (HALT) in some action component of a rule terminates the entire process.

The rule set for the PSG model is very simple. Initially, the status of the PSGRAPH is examined to see if the process should terminate. The rules written here specify that if the status of the PSGRAPH is FAILED or SUCCEEDED then the graph is output and the process halts. Otherwise, If there is a node marked as a possible node to sprout, the goal node is picked first. On the other hand, the presence of a trynode which has no subgoals indicates that the selection is completed and the action to take is to SPROUT the trynode and EVALUATE it. The responsibility of SPROUT is to choose a transformation, test its applicatbility and give the set of precursors to Modelling System. The Modelling System then updates the model and posts the tree hierarchy, the circularity and status relations. The evaluation also submits its merit rating of the nodes to the Modelling System which in turn, reassigns the merits of the affected AND-NODES and OR-NODES. The implications of the new information so entered are followed by the Modelling System and the information needed by the rule set is provided in a ready form. The control repeatedly flows in a SELECT-SPROUT-EVALUATE loop until halted.

The rule set is flexible and can be changed if a specific synthesis problem suggests a different form of control. It is not difficult to enter syntactic guidance based on the model graph using the distance of a node from the goal, the number of conjuncts at an

AND-NODE etc.. It is also possible to guide the search based on chemical information, for example, to disregard subgoals involving a seven-membered heteratomic ring. It is conceivable that when the system runs interactively the guidance will be changed and experimented with as the search proceeds.

SEARCH IN A PLANNING SPACE

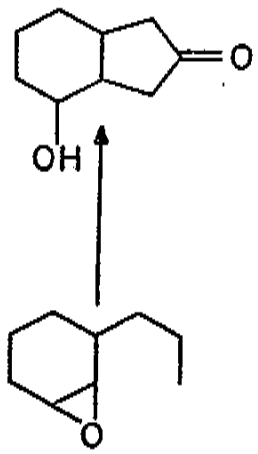
The structure of the search space is determined not only by the collection of transforms available to the system but also by the rules for selecting the transforms to be tried for any subgoal. There are two basic ways for selecting transforms.

a) Selection by Applicability. If the transforms are selected because they guarantee that the target molecular structure will be produced upon their application, the required precursors become subgoals. This is the method of selecting transforms by their applicability in the retrosynthetic direction. The synthesis sequence grows a step at a time ensuring that the target molecular structure will be a product of the reaction sequence developed so far, once the precursors are made available. The space of possibilities determined by this criterion of applicability is termed the STATE SPACE.

b) Selection by Relevance. In some cases a transform is selected because it produces a molecule only similar to the target molecular structure but not exactly the same. When such a transform is applied to a target structure T, it may synthesize some structure S that is similar to T, from a set of precursors P. This breaks the original problem of synthesis into two subproblems,

i) Synthesize P. S(P)
ii) Transform S==>T. TR(S,T)

The transform selected specifies a reaction that converts P to S, and this transform will, in general, constitute an intermediate step in the synthesis sequence. The space of possibilities determined by the criterion of relevance is termed the PLANNING SPACE and in several situations the search toward a



REACTION USED IN TRANSFORM

SCHEME I

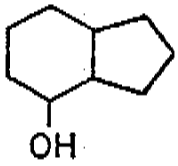
solution can be briefer in this space than in the State Space. This definition of a Planning Space is a variation of the concept introduced in the General Problem Solver system (GPS) [Ernst, 1969].

The distinction between the Selection of transforms by Applicability and by Relevance is an important one when considering the strategies one might employ to search for synthesis sequences. The search in a Planning Space has the characteristic that the search "leaps" into some intermediate point in the synthesis sequence and establishes an "island" and the solution search could then proceed from the island to the target molecule in the forward direction or from the island backward in the retrosynthetic direction toward available molecules. The significance of this ability to leap has been explored in other task areas than synthesis search and has been found to be a powerful tool in converging on solutions rapidly. Its utility for synthesis search remains to be shown and for now can be illustrated only in terms of examples.

The following sketch of an example is offered to illustrate the idea of planning space. The example has not been checked by any chemist and thus its chemical correctness cannot be assured.

Wipke [Wipke, 1976] uses the example of a reaction that synthesizes an alcohol group in 1,4 relation to an electron withdrawing group, say C=O, by the opening of epoxide by a stabilized ion. Consider the target structure shown in Scheme I.

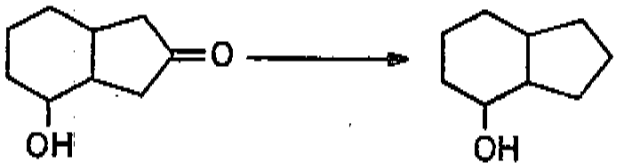
The criterion of applicability would require that the target structure contain the -C(OH)-C-CO- substructure and would not be applicable in the State Space search. In search conducted in the Planning Space, if the transform indicated is considered relevant to the target structure (e.g., if the presence of the alcohol and an unsubstituted 1,4 carbon is sufficient) then this transform may be used. The original synthesis problem is replaced with two subproblems shown in Scheme II.



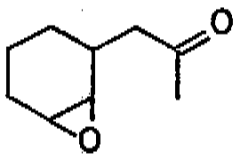
TARGET

a.

TRANSFORM



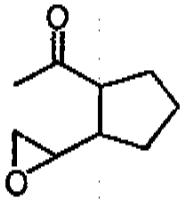
SYNTHESIZE



b.

TRANSFORM

SYNTHESIZE



PLANNING SPACE MANEUVERS

SCHEME II

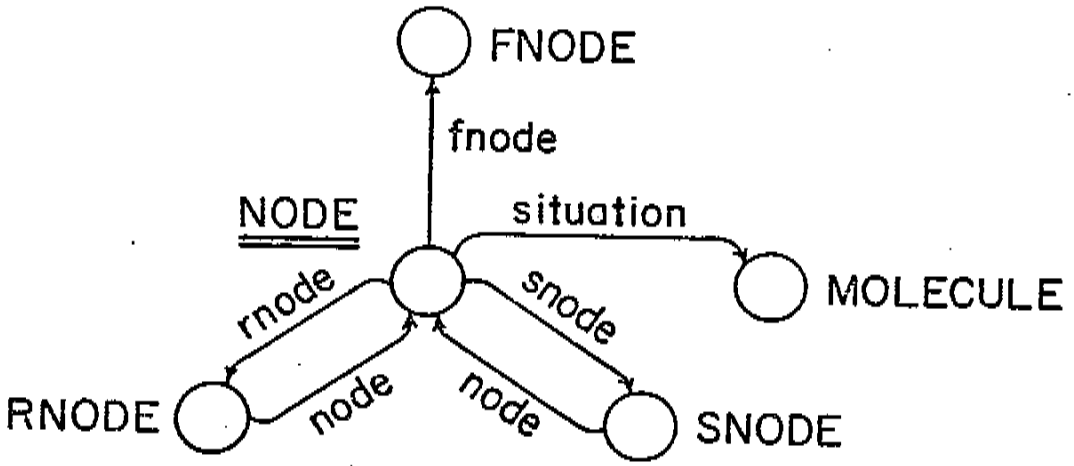
The above example could be successfully completed by working forward reducing the difference between S and T, and working backward in synthesizing P.

COMBINING SEARCH IN PLANNING AND STATE SPACE

The search in a planning space should be conducted by taking repeatedly the synthesize S type problem for each P splitting it each time into a 'Synthesize' and a 'Transform' type problem, deferring all the Transform TR type problems till some available molecular structure is reached along a path. This will generate a Skeletal Plan where many of the intermediate steps are lacking in detail but each one is given as a TR type problem. If an evaluation function can designed for these skeletal plans much useless search can be avoided by using the planning space.

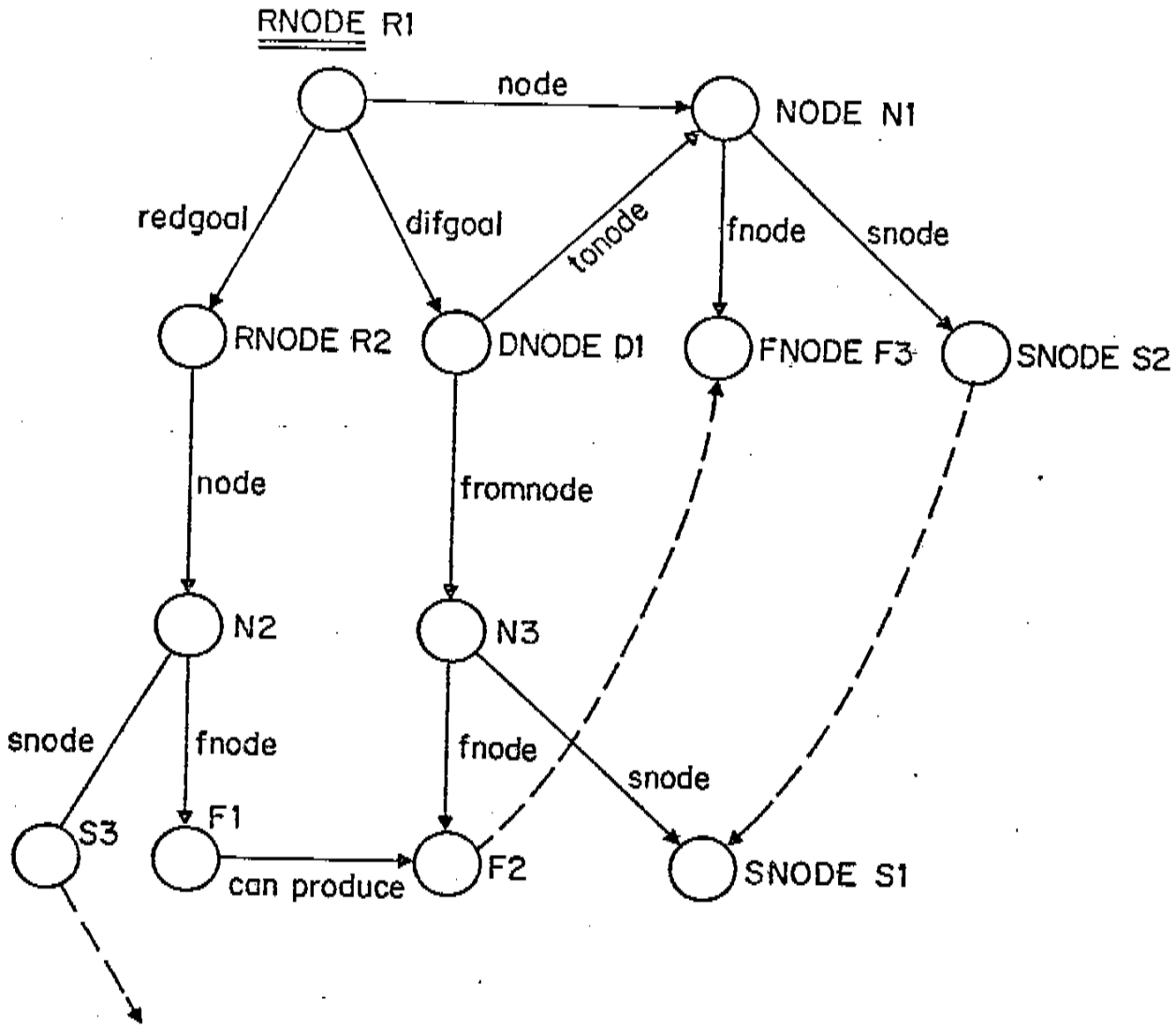
The search in the state space is conservative and takes small steps attempting to make steady progress towards completing a set of solutions. This can cause either aimless wandering because small changes in the merit values assigned to subgoals cause no significant shifts of attention or because the changes in the merit values cause large abrupt changes in behaviour. This has been given the graphic name of the Mesa Phenomenon by Minsky [Minsky, 1963]. The planning space structures the solution sequences quite differently and causes a more goal-directed search to proceed. The method of transform selection allows the program to "leap" into the solution sequence and decide upon one of the intermediate reactions and permits the solution to grow in both directions. It appears that a judicious combination of both the State Space and Planning Space search methods might be able to overcome some of the difficulties found in each of the methods [Amarel, 1969] With these considerations in view the next section introduces a framework in which to combine the two spaces, letting the chemist user supply the search guidance rules customized to the particular problem at hand.

FLEXI: A FLEXIBLE ADVANCED SEARCH MODEL



THE "NODE" BUILDING BLOCK

FIGURE 2



THE "RNODE" BUILDING BLOCK

FIGURE 3

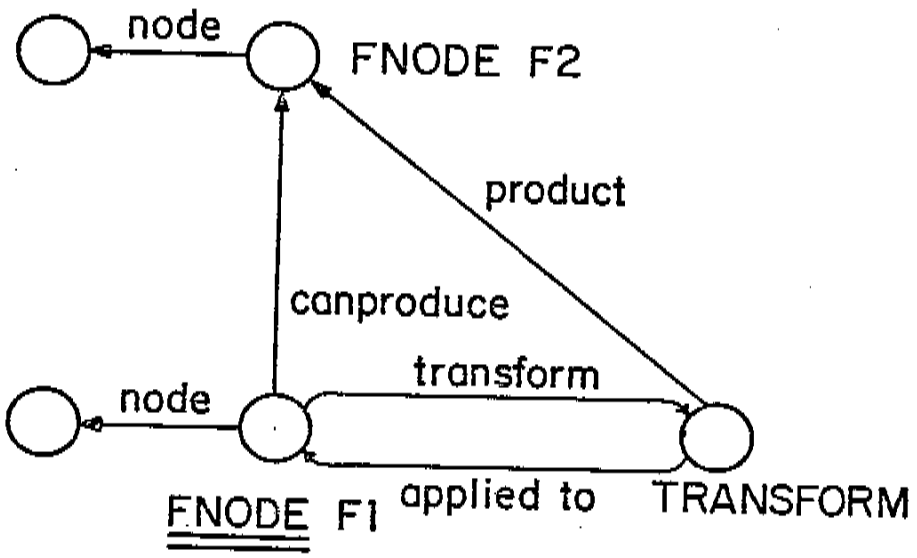
Given a collection of nodes designating molecules there are the following types of tasks one can generate:
a) For a given node whose molecule is not yet synthesized, develop a synthesis by working backwards in the state space using applicable transforms.
b) For a given node whose molecule is not yet synthesized, develop a synthesis by problem reduction using relevant transforms.
c) For a given ordered pair of nodes develop a synthesis route that transforms one molecule to the other.
d) For a given node execute a brief non-goal-directed exploration forwards using reactions in their conventional directions.

A search model is introduced here, called FLEXI, in which four types of structures are used to symbolize the above four categories of tasks and these four nodes form the BUILDING BLOCKS of the search management model. Figure 2 shows that a NODE designates a molecule and individually can be set up as an SNODE for searching retrosynthetic sequence, or as an RNODE for search for a planning route. The FNODE is used to set up the node for forward exploration without a goal guidance.

The sprouting of an SNODE generates a piece of the familiar AND/OR problem solving graph and the status relations on SNODE, OR-NODE and AND-NODE are posted similar to that given earlier.

The sprouting of an RNODE R1, see Figure 3, constitutes a step in the planning space and generates two tasks by problem reduction - an RNODE R2 and a DNODE D1. R2 calls for the synthesis of a molecule by further problem reduction and the DNODE sets up a problem of transforming one molecule into another. The transform used in the sprouting of R1 is used to establish that it canproduce the fromnode N3 of the DNODE from the node N2 of R2. This is exhibited in Figure 3.

The new task set up in the RNODE could of course be restructured as an SNODE task by some rule in the production system. The RNODE will be considered successful as soon as the NODE connected to it has



THE "FNODE" BUILDING BLOCK

FIGURE 4

SUCCEEDED, whether by its RNODE or the SNODE tasks or even by its designating a molecule which is available in the Catalog of Starting Compounds.

Under certain circumstances a synthesis for the fromnode N3 of the DNODE could be attempted independently and its success will disallow the RNODE R2 from further consideration during search. In that case, the success of the DNODE is sufficient to guarantee the success of RNODE R1 and thereby of NODE N1.

The DNODE can succeed either when a path is found by working forwards from FNODE F2 to F3 or by working backwards from SNODE S2 to S1.

Figure 4 illustrates the canproduce relation among pairs of FNODES. Working forwards from F1 if a molecule M2 results from a transformation involving M1 then the corresponding FNODE F1 can produce F2.

The structures exhibited here are only the building blocks of the search model. By suitably controlling and guiding the exploration, the search can take on great variety traversing the state space forwards or backwards or traversing the planning space. The rule set can be made to contain broad injunctions such as "Do not explore a node in the forward direction if it was created by instantiating a SNODE, i.e. a node to proceed in the retrosynthetic direction" or "When you add an FNODE immediately instantiate a revised DNODE type task" as shown in Figure 5. Of course, these two rules are used here only as examples and in given situations one might wish to advise the system otherwise. The essential point is that a flexible form of search guidance specification is available and can be used to bring to bear on a given problem a wide variety of hints, suggestions and advice that would be difficult in a standard Heuristic Search program.

Overcoming some difficulties of Heuristic Search.

One cause of the Mesa Phenomenon in the case of chemical synthesis is the use of functional group substitution reactions while working in the

retrosynthetic direction. If a molecule containing several functional groups is selected for sprouting then performing functional group substitutions often yields

precursors that have nearly the same merit as the target molecule. The presence of several functional groups only aggravates the situation. Within FLEXI this class of reactions could be used mainly for transforming a molecule to another when they are structurally similar, i.e. a DNODE type task which might be carried out most favorably using functional group substitutions. Thus, avoiding the use of these reactions in the retrosynthetic direction should prevent the problem. As further specific problems are isolated and solved, the framework of FLEXI may help us to submit the proper rules of search guidance to the system.

CONCLUSION

The Heuristic Search method is basically very simple. It involves a serial processor working backwards that selects subgoals and transforms by asking "What next?". A good implementation of the heuristic method includes a SEARCH MODEL which is a symbolic representation of the progress of search. The Problem Solving Graph that is commonly used to guide search is a useful but limited search model.

Symbolization is the key to Reasoning and the computer can reason only about things it can handle symbolically. Furthermore, the richness of the search model contributes to the conduct of an intelligent search.

In this paper a search modelling system is described by examples. This system allows the user to describe rather than to program the search model and to associate constraints that govern the growth of the model. The system provides a Rule Language based on the user described search model and the user may then prescribe rules in the form of Production Rules. The rules the user writes can be general, being valid over a wide variety of task specifications, or can be bits of advice and hints about a given problem. The paper concludes by showing how some of the standard difficulties with heuristic search such as getting locked into a plateau can be overcome by suitable techniques of search modelling. The modelling ideas shown here can be used to control and specify protection reactions and for sequencing functional group introduction. It is also possible to carry out different styles of exploration varying the emphasis on the directionality and space in which the search is conducted. The applicability of the model to complement a heuristic search process is now made clear, however its actual use in chemical synthesis planning awaits the participation of a chemist collaborator!

ACKNOWLEDGMENT

I wish to thank Prof. Srinivasan for his stimulating discussions of his modelling ideas represented in MDS without which my understanding of

MDS would be meager. A subset of MDS adequate to deal with PSG and FLEXI is being programmed and made available but so far no synthesis problems have been tried out in these frameworks. The present system is implemented in the language FUZZY [LeFaivre, 1974]. I look forward to the continued cooperation of Prof. LeFaivre in the future and his help in the past is hereby gratefully acknowledged. I wish to thank Prof. Saul Amarel for his expert advice upon reading this paper.

```
(* TABLE I *)

(* CONSTANTS OF THE DOMAIN PSG *)

(CONSTANTS (YESNO (YES NO)))
(CONSTANTS (STATUS (SUCCEEDED FAILED OPEN)))


(* STRUCTURAL DESCRIPTIONS *)

(TDN: [PSGRAPH (element NODE elementof)
               (goal NODE goalof)
               (opennodes NODE opennodeof)
               (failednodes NODE failednodeof)
               (status STATUS statusof)
               (trynode NODE|AND-NODE|OR-NODE
                        trynodeof)])


(TDN: [NODE (subgoal OR-NODE subgoalof)
            (subnode NODE subnodeof)
            (descendant NODE descendantof)
            (status STATUS)
            (situation MOLECULE)
            (merit INTEGER meritof)
            (repeated NODE repeatedby)
            (circular NODE)])


(TDN: [OR-NODE (subgoal AND-NODE)
               (status STATUS)
               (merit INTEGER)])


(TDN: [AND-NODE (subgoal NODE)
                (status STATUS)
                (merit INTEGER)])


(TDN: [MOLECULE (structure CHEMICAL-GRAPH)
                (available YESNO)])


(* Flags specifiable on the relations
   have been left out for simplicity *)
```

```
                    (* SENSE DEFINITIONS *)

(QSCC: [((NODE N) | (X elem N) (N status OPEN))
        PSGRAPH
        opennodes])
(QSCC: [((NODE N) | (P elem N) (N status FAILED))
        PSGRAPH
        failednodes])
(QSCC: [((STATUS S) | (X (goal status) S))
        PSGRAPH
        status])
(QSCC: [((STATUS S) |
          ((X (subgoal status) SUCCEEDED)  =>
                    (X status SUCCEEDED))
          (((NOT [X (subgoal status) FAILED])
            AND
            (NOT [X status SUCCEEDED])) =>
                    (X status OPEN))
          (((ALL NODE N) (N status FAILED)) =>
                    (X status FAILED)))
        OR-NODE
        status])
(QSCC: [((STATUS S) |
          ((X (subgoal status) FAILED) =>
                    (X status FAILED))
          (((X (subgoal status) OPEN)
            (NOT [X status FAILED])) =>
                    (X status OPEN))
          (((NOT [X status FAILED])
            (NOT [X status OPEN])) =>
                    (X status SUCCEEDED)))
        AND-NODE
        status])
(QSCC: [((NODE A) | (A (subgoal subgoal subgoal) X))
        NODE
        subgoalof])
(QSCC: [((NODE A) |
          (X subgoalof A)
          OR
          (X (descendantof subgoalof) A))
        NODE
        descendantof])
(QSCC: [((STATUS S) |
          ((X (situation available) YES) =>
                    (X status SUCCEEDED))
          ((X (subgoal status) SUCCEEDED) =>
                    (X status SUCCEEDED))
          ((X circular Y) => (X status FAILED)))
        NODE
        status])
(QSCC: [((NODE R) | (X (situation situationof) R))
```

```
        NODE
        repeated])
(QSCC: [((NODE C) | (X repeated C) (X descendantof C))
        NODE
        circular])
(QSCC: [((INTEGER I) |
         (X (subgoal merit) I)
        (NOT [X (subgoal merit >=) I]))
        OR-NODE
        merit])
(QSCC: [((INTEGER I) |
         (X (subgoal merit) I)
        (NOT [I (>= meritof subgoalof) X]))
        AND-NODE
        merit])


        (* PRODUCTION RULES GUIDING SEARCH *)

[INITIALIZE (IT (PSGRAPH P))
            (INPUT (P goal G))
            (IR (P goal G))
            (IR (P opennodes G))]

(G status SUCCEEDED) =>
        (OUTPUT G)(HALT)

(G status FAILED) =>
        (OUTPUT G)(HALT)

(G element X)(X circular Y) =>
        (ASSERT (X status FAILED))

(NOT [G trynode X]) =>
        (ASSERT (G trynode (G goal)))

(G trynode X)(NOT [X subgoal Y]) =>
        (ASSERT (G trynode NIL))(SPROUT X)(EVALUATE X)

(G trynode X)(X subgoal Y)(X (merit meritof) Y) =>
        (ASSERT (G trynode Y))

        (* End of Table I *)
```

```
(* TABLE II *)

(* STRUCTURAL DESCRIPTIONS *)

(TDN: [SEARCHGRAPH
                (* This is the search graph of FLEXI)
                (elements NODE)
                (goal NODE)
                (trynode RNODE|SNODE|DNODE|FNODE)])

(TDN: [SNODE
          (* State Space Structure
           (Backward Search))
          (node NODE snode)
          (subgoal OR-NODE)
          (merit MERIT)
          (status STATUS)
          (subgoal NODE)
          (circular SNODE)
          (features FEATURE)
          (descendant SNODE)])

(TDN: [RNODE (* Planning Space Structure)
          (node NODE rnode)
          (redgoal RNODE)
          (difgoal DNODE)
          (merit INTEGER meritof)
          (status STATUS)
          (relevantfeatures FEATURE)])

(TDN: [DNODE (* State Space Structure
             (Forward Search))
          (node NODE dnode)
          (tonode NODE)
          (fromnode NODE)
          (differences FEATURE)
          (merit MERIT)
          (status STATUS)])

(TDN: [FNODE (* Non-Goal-Directed
             Forward Search)
          (node NODE fnode)
          (transform TRANSFORM appliedto)
          (reactivegroup FEATURE)
```

```
                (canproduce FNODE canbeproducedfrom)
                (merit MERIT meritof)
                (status STATUS)])


(TDN: [TRANSFORM
                    (* One is needed for
                       every growth)
                    (appliedto FNODE transform)
                    (product FNODE)
                    (reagents MOLECULE)])


                   (* SAMPLE PRODUCTION RULES *)


(ADDED (SNODE S)) =>
        (ASSERT ((S node) fnode NIL))
                   (* No forward exploration if S was
                      given as a retrosynthetic goal)


(ADD (FNODE D))=>
        (FILLIN (INSTANTIATE
          (DNODE (fromnode &(N node))
                 (tonode &(N (canbeproducedfrom
                              node dnode tonode)))))))

                   (* If N was generated by forward exploration
                      try asserting a DNODE subgoal if the
                      informaion needed is there)


                  (* SENSE DEFINITIONS *)

(QSCC: [((DNODE D) | (D tonode (X goalnode))) RNODE difgoal])

(QSCC: [((DNODE D) |
          ((X (node status) SUCCEEDED) =>
                 (X status SUCCEEDED))
          (((X (difgoal status) SUCCEEDED)
            (X (redgoal status) SUCCEEDED)) =>
                  (X status SUCCEEDED))
          (((X (difgoal status) FAILED)
            OR (X (redgoal status) FAILED)) =>
                  (X status FAILED)))
         RNODE
         status])

(QSCC: [((NODE Y) | (X (difnodeof goalnode) Y)) DNODE tonode])
```

```
(QSCC: [((STATUS S) |
         ((X (fromnode canproduce* node) (X tonode))
          =>
          (X status SUCCEEDED))
         ((X (goalnode goalnode descendant node) (X fromnode))
          =>
          (X status SUCCEEDED)))
        DNODE
        status])

(QSCC: [((STATUS S) |
         ((X (node status) SUCCEEDED) => (X status SUCCEEDED))
         ((X (subgoal status) SUCCEEDED) => (X status SUCCEEDED)))
        SNODE
        status])

(QSCC: [((STATUS S) |
         (((X (rnode status) SUCCEEDED) OR
           (X (snode status) SUCCEEDED) OR
           (X (situation available) YES)) =>
                (X status SUCCEEDED)))
        NODE
        status])
```

REFERENCES

1. Charles Schmidt [1976]
Understanding Human Action: Recognizing the Plans and Motives of Other Persons, in Cognition and Social Behavior, J. Carroll & J. Payne (editors), Lawrence Earlbaum Press, 1976.

2. N.S. Sridharan [1975]
The Architecture of BELIEVER: A System for Intepreting Human Actions. Technical Report RUCBM-TR-46, Department of Computer Science, Rutgers University, New Brunswick NJ.

3. N.S. Sridharan [1976]
The Architecture of BELIEVER-Part II. The Frame and Focus Problems in AI. Technical Report RUCBM-TR-47, Department of Computer Science, Rutgers University, New Brunswick NJ.

4. Chitoor Srinivasan [1973]
The Architecture of a Coherent Information System. Advanced Papers of the Third International Joint Conference on Artificial Intelligence, Stanford, 1973.

5. Chitoor Srinivasan [1976]
An Introduction to the Meta-Description System. Technical Report SOSAP-TR-18, Department of Computer Science, Rutgers University, New Brunswick NJ.

6. N.S. Sridharan [1974]
A Heuristic Program to Discover Syntheses for Complex Organic Molecules. Proceedings of the IFIP74 (International Federation for Information Processing) Congress, Stockholm, August 1974.

7. N.S. Sridharan [1971]
An Application of Artificial Intelligence to the Discovery of Complex Organic Synthesis. Doctoral Dissertation, Department of Computer Science, SUNY at Stony Brook, 1971.

8. N.S. Sridharan [1973]
Search Strategies for the task of Organic Chemical Synthesis. Advanced Papers of the Third International Joint Conference on Artificial Intelligence, Stanford, 1973.

9. H. Gelernter [1973]
The Discovery of Organic Synthetic Routes by Computer. Topics in Current Chemistry, Volume 41, Springer-Verlag, 1973.

10. Herbert Gelernter [1976]
Empirical Explorations of SYNCHEM, An Application of Artificial Intelligence to the Problem of Computer-Directed Organic Synthesis Discovery. This volume.

11. E.J. Corey & W.T. Wipke [1969]
Computer-assisted Design of Complex Organic Synthesis, Science, Volume 166 (178), 1969.

12. Todd Wipke [1973]
Computer-Assisted Three Dimensional Synthetic Analysis. In Computer Representation and Manipulation of Chemical Information. W.T. Wipke et. al. (editors), John Wiley (New York) 1974.

13. Todd Wipke [1976]
SECS--Simulation and Evaluation of Chemical Synthesis: Strategy and Planning. This volume.

14. Ivar Ugi [1976]
The Synthetic Design Program MATSYN as a part of MATCHEM, A System of Computer Programs for the Deductive Solution of Chemical Problems. This volume.

15. Nils Nilsson [1971]
Problem Solving Methods in Artificial Intelligence. McGraw Hill (New York) 1971.

16. Allen Newell & Herbert Simon [1972]
Human Problem Solving. Prentice-Hall (New Jersey) 1972.

17. Herbert Gelernter [1962]
Machine-Generated Problem Solving Graphs. Proceedings of a Symposium on the Mathematical Theory of Automata Polytechnic Institute of Brooklyn Press (New York) 1962.

18. James Slagle [1971]
Artificial Intelligence: The Heuristic Programming Approach. McGraw-Hill, New York, 1971.

19. Randall Davis and Jonathan King [1975]
An Overview of Production Systems, Stanford Artificial Intelligence Laboratory Memo AIM-271, October 1971.

20. George Ernst & Allen Newell [1969]
GPS: A Case Study in Generality and Problem Solving. Academic Press (New York), 1969.

21. Marvin Minsky [1963]
Steps Toward Artificial Intelligence. In Computers and Thought, by Edward Feigenbaum and Julian Feldman (editors), McGraw-Hill (New York) 1963.

22. Saul Amarel [1969]
Problem-Solving and Decision-Making by Computer: An Overview. In Cognition: A Multiple View, Garvin (editor), Spartan Books (Washington) 1970.

23. Richard LeFaivre [1974]
Fuzzy Problem-Solving, Ph.D. Dissertation, Computer Science Department, University of Wisconsin, Madison, 1974.
The Representation of Fuzzy Knowledge, Journal of Cybernetics, volume 4, p57-66, 1974.