STANFORD ARTIFICIAL INTELLIGENCE LABORATORY
MEMO AIM - 217

STAN-CS-73-391

# SEARCH STRATEGIES FOR THE TASK OF ORGANIC CHEMICAL SYNTHESIS

BY

N. S. SRIDHARAN

SUPPORTED BY

ADVANCED RESEARCH PROJECTS AGENCY
ARPA ORDER NO. 457
& NATIONAL INSTITUTES OF HEALTH
OCTOBER 1973

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

STANFORD ARTIFICIAL INTELLIGENCE PROJECT
MEMO AIM-217

AUGUST, 1973

COMPUTER SCIENCE DEPARTMENT REPORT
NO. STAN-(X-73-391

# SEARCH STRATEGIES FOR THE TASK OF ORGANIC CHEMICAL SYNTHESIS

by

N. S. Sridharan

ABSTRACT: A computer program has been written that successfully discovers syntheses for complex organic chemical molecules. The definition of the search space and strategies for heuristic search are described in this paper.

The publication of this report is supported by the Advanced Research Projects Agency (SD-183) and the National Institutes of Health (RR-612).

This text was presented to the Third International Joint Conference on Artificial Intelligence, Stanford University, August, 1973.

TABLE OF CONTENTS

It is not growing like a tree . . .

. . . In small proport Fans we just beauties see; - Ben Jonson.

## INTRODUCTION

The design of application of artificial intelligence to a scientific task such as Organic Chemical Synthesis was the topic of a Doctoral Thesis completed in the summer of 1971 (Reference 1). Chemical synthesis in practice Fnvolves i) the choice of molecule to be synthesized; ii) the formulation and specification of a plan for synthesis (Involving a valid reaction pathway leading from commercial or readily available compounds to the target compounds with consideration of feasibility regarding the purposes of synthesis); iii) the selection of specific individual steps of reaction and their temporal ordering for execution; iv) the experimental execution of the synthesis and v) the redesign of syntheses, if necessary, depending upon the experimental results. In contrast to the physical synthesis of the molecule, the activity in ii) above can be termed the 'formal synthesis'. This development of the spectfication of syntheses involves no laboratory technique and is carried out mainly on paper and in the minds of chemists (and now within a computer's memory!),

## IMPORTANCE AND DIFFICULTY OF CHEMICAL SYNTHESIS

The importance of chemical synthesis is undeniable and there is emphatic testimony to the high regard held by scientists for synthesis chemists. The level of intellectual activity and difficulty involved

in chemical synthesis are illustrated by Vitamin A (example solved by our program) and Vitamin R12. Roth problems absorbed the efforts of several teams of expert chemists and held them at bay for over 20 years. Professor R.B. Woodward of Harvard University was awarded the nohel prize in 1965 for his numerous and brilliant syntheses and their contribution to science.

## A DESIGN DECISION

A program has been written to execute a search for chemical syntheses (i.e. formal syntheses) for relatively complex organic molecules. Emphasis has been placed on achieving a fast and efficient practical system that solves Interesting problems in organic chemistry.

The choice of design made very early in this project is worth mentioning. We could have aimed at an Interactive system which would employ a chemist seated at a console guiding the search for synthesfs. The merit of this approach, exemplified by Corey (Reference 4), lies Fn this direct interaction between the chemist and computer whereby the designers are afforded rapid feedback allowing the system to evolve into a tool for the chemists, An obvious shortcoming however, Is that it circumvents the questions that are very pertinent to art if icialintelligence. In contrast, our approach was to design a non-Interactlve, batch-mode program with artificial intelligence aspects built into Ft. We have tackled the problem of synthesis discovery chiefly from the vantage point of artificial Intel 1 igence, utilizing the task area only as a vehicle to investigate the NATURE OF AN APPLICATION OF MACHINE REASONFNG

WITH AN EXTENSIVE SCIENTIFIC KNOWLEDGE RASE.

Our choice is perhaps vindicated on three counts:

a) It has freed us from the" distractions of designing a user Interface, which is not a simple task;

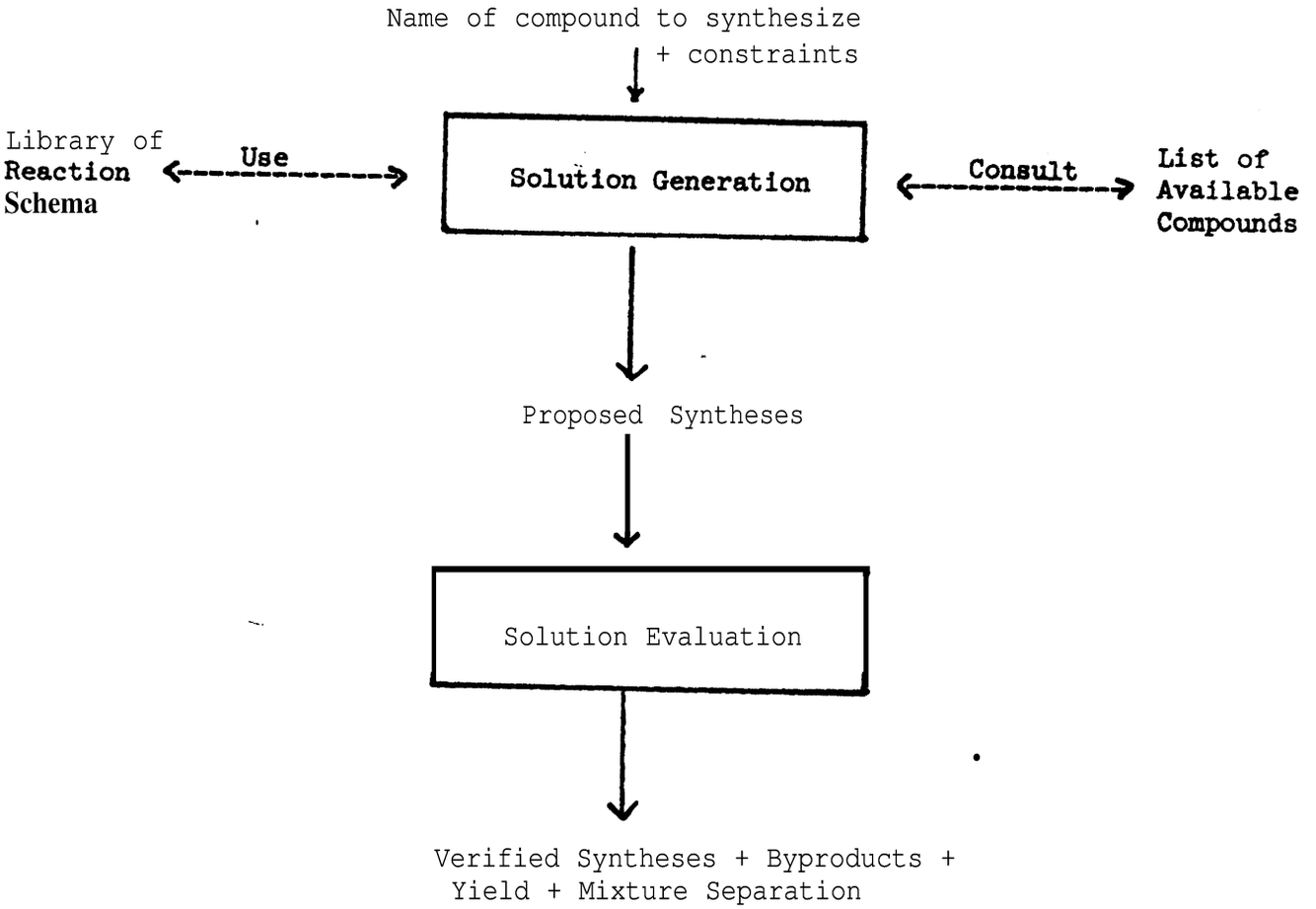
b) it has resulted In a fast system that runs on standard hardware to be found in nearly every medium-sized computation center, and has produced successfully several syntheses for each of several complex molecules;

c) the program works autonomously In searching for solutions and incorporates into its task several key Judgemental capabilities of a competent synthesis chemist.

## TASK ENVIRONMENT

The program accepts as input some representation of the target compound together with a list of conditions and constraints that must govern the proposed syntheses (Figure 1). A list of compounds that are commercially avaflable (along with indications of cost and availability) can be consulted. A reaction library containing general lzed procedures is suppl led to the program. The output is a set of proposed syntheses, each being a valid reaction pathway from available compounds to the target molecule. The syntheses are arrived at by means of strategic exploration of an AND-OR search space. The design of the search strategy concerns us here..

The search space has characteristics that make the problem a novel one. Well known search strategies using AND-OR problem solving trees (Reference 2) concern themselves with either optimal solutions or minimal effort spent In finding a solution. Heurlstlc DENDRAL in its search for a solution has the distinction of knowing that only one answer is 'the correct answer' and fewer number of alternative solutions is commensurate with greater success for the -program. The synthesis program, on the other hand, is not aimed toward any optimal search or toward 'the best' synthesfs (there is not- one). Quite simply, the task of the synthesis search is to explore alternative routes of synthesis and develop a problem solving tree rich in information, having several 'good' complete syntheses. The success of the program is not to be judged sole1 y on the number or variety of completed syntheses, but with the understanding that paths of exploration not completed by the program are very informative as well.




Note: This paper concerns solution generation

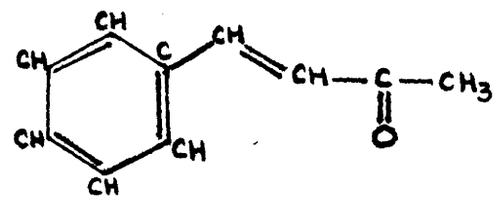
FIGURE 1. PROBLEM SCHEMATIC

The reader is referred to the Thesis (Reference 1) for a detailed exposition of the algorithm, programming details such as chemical structure representat tan, representat ion of react ions, the setup of a reaction library and a catalog of readily available compounds. This brief article describes one aspect of the problem that is of primary significance to those interested in artificial intelligence. Other topics of interest to be found in the Thesis include: Elimination of invalid subgoals, Invalidation of subgoals by cost considerations, Elimination of redundant subgoals and Elimination of unpromising subgoals.

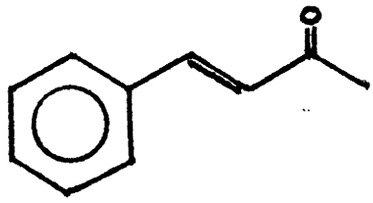
## BASIC CONCEPTS AND TERMS

A sample synthesis problem, deliberately chosen for its simplicity, is now followed partially through the search for a solution. The intent of this example is mainly to introduce some basic concepts and to illustrate terminology. It Is not Intended to explicate the complexity of the task area, In dealing with the example the hypothetical course of problem solution by a chemist is given and the problem solving components related to the program are presented in addition. It should he menttoned that thts problem has been solved by the program (with facility),

Consider a synthesis is required for a compound whose structural formula is as shown below.



Chemfsts also accept a stylized version of the same dfagram:



The usual representation of chemical structures for program manipulation involves a list with each item representing an atom and Its connections to other atoms by bonds. We have designed a variant of the connection list to suit the manipulations relevant to synthesfs; This variant will be referred to as the TOPOLOGICAL STRUCTURE DESCRIPTION for a compound. Details of this representation and manipulation are described in the Thesis (Reference 1) and are not needed to understand this paper.

The chemist examines the molecule and recognizes several structural features such as the presence of the six-membered ring with three internal double bonds (usually called the phenyl group). Other noticeable features *are* the ketone, $-\overset{\overset{O}{\|}}{C}-$ , and olefin bond, -CH=CH- . What is defined as a feature depends upon the purpose of the examination and the chemical knowledge one possesses. We use the term SYNTHEME to refer to the structural features of a molecule that are relevant to its synthests.

The program examines the topological structure description and through graphical pattern matching techniques develops an ATTRIBUTE LIST consisting of a list of synthemes for the molecule,

Among the features of the molecule, the phenyl group is very stable and occurs in many commercially available compounds. Thus, in seeking ways to synthesize this compound the chemist considers the ketone and olefin bond and not the benzene as possible reactive sites.

The chemist knows of several reactions that can synthesize an olefin bond and several that can synthesize the ketone syntheme. He can consider each of these as trial last steps of the synthesis sequence he is seeking.

The program is provided with a collection of reaction schemata called the REACTION LIBRARY. The reaction schemata are grouped into reaction chapters according to the syntheme they synthesize. Each reaction schema is provided with a set of tests to be performed on the target molecule and structural patterns for the target and subgoal molecules. The tests embody many of the chemical heuristics that guide the program. Based on the results of some of the tests the program may reject the reaction schema. Each schema has an a priori assignment of merit rating. Based on the results of other tests the program may alter the merit rating to reflect the suitability of the schema to the specific target molecule.

We may represent the alternative courses of syntheses developed for the target molecule by a PROBLEM SOLVING GRAPH (Figure 3). The target molecule is a node at the top. A serfes of arrows lead from the target through the chapter, attribute and schema layers to the

subgoal layer. Each subgoal consists of one or more conjoined compounds -- implying that they all enter the reaction *to* generate the target molecule. Thus, the compound layer is an AND-layer in this AND-OR graph.

If all the compounds needed for any one subgoal are available commercially we would consider that we know a plausible single-step synthesis for the target molecule. Any compound generated as subgoal which is not commercially available needs to be synthesized and can be considered in turn as a target molecule.

Repeating the above considerations with the new target molecule will open the path for multi-step syntheses. The problem solving graph branches downward like a tree whereby each path represents a possible course of synthesis for the target molecule.

The above presentation is not to imply that a chemist actually follows these steps shown in devising syntheses, The method of reasoning analytically from the target molecule in a sequence of steps, ending up in available compounds is but one technique in the vast repertoire a chemist usually possesses. However, the analytic search procedure is amenable to convenient computer implementation and is suitable for investigating a very large class of synthesis problems The solution scheme is described in the next section.

## SOLUTION SCHEME

The problem lends itself to an analytic search procedure. The search begins at the target molecule and the last step of the synthesis is the first to be discovered, the next to the last step is found second and so on. Thus the discovery sequence is the reverse of the synthesis sequence.

The GOAL is given to the program as a chemical structure description. The description, whether given as a canonical compact linear notation (Wiswesser Notation, Reference 3) or as a topological structure description, gives information about what atoms are present in the molecule and how they are connected.

The structure of the molecule is then examined to identify its SYNTHEMES, such as the presence of certain types of bonds, the occurrence of certain groups of atoms and generally the substructures of given types. Such information is collected automatically into an ATTRIBUTE LIST.

A large set of chemical reactions (over 100) is compiled and each reaction is schematized to be usable as an OPERATOR in developing the search space. In using the reaction schema as an operator the reaction is used in its inverse direction (i.e. from the reaction product to the reactant) analogous to the use of a rule of logical deduction in its inverse direction in a theorem proving task.

The collection of reaction schemata is known as the REACTION LIBRARY. The reaction library is arranged as several CHAPTERS, each containing reaction schemata that are relevant to or affect a syntheme of target molecule -- the theme of the chapter.

Each reaction scheme has detailed TESTS OF RELEVANCE and TESTS OF APPLICABILITY toward the target molecule. The tests are performed before the operator is employed. The application of an operator on a specific attrfhute of a molecule results in one or more subgoals. Each subgoal in turn has one or more CONJOINED molecules to he used together in the reactton. A subgoal thus generated is further subject to TESTS OF VALIDITY. The distinction between the two sets of tests is that one set is conducted on the target molecule, whereas the other set is conducted on the subgoals after subgoal generation.

The successive application of operators on the subgoal compounds and all their subgoals generates the SEARCH SPACE. The strongest eondition for termination of path development is the availabllity of the compounds needed. The availability is checked using a compound catalog of a- chemical manufacturing company, a list of about 4000 compounds,

Figures 2 and 3 describe the schematic flowchart of the algorithm and the five layers of the PROBLEM SOLVING TREE generated In developing subgoals one level.

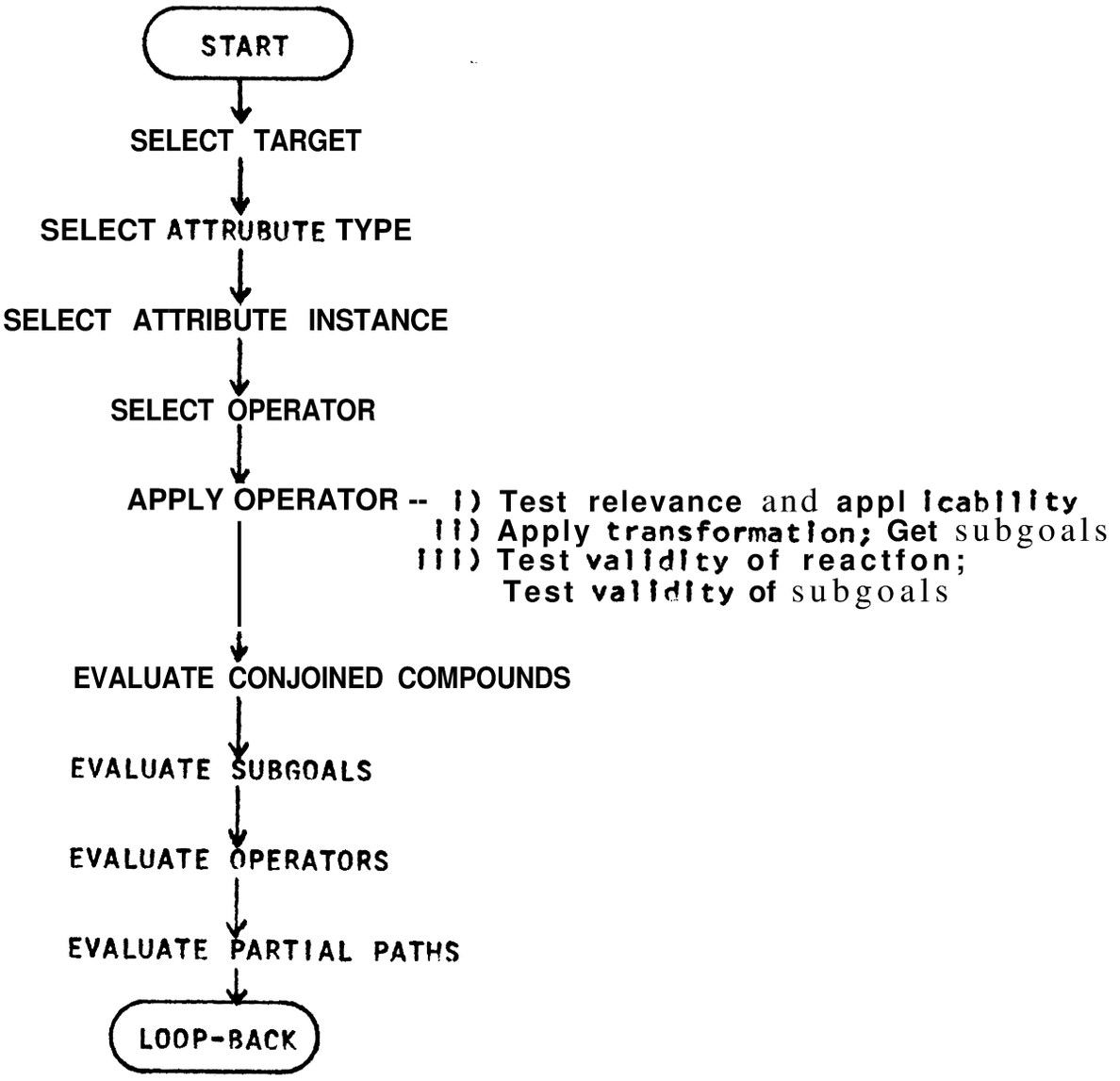
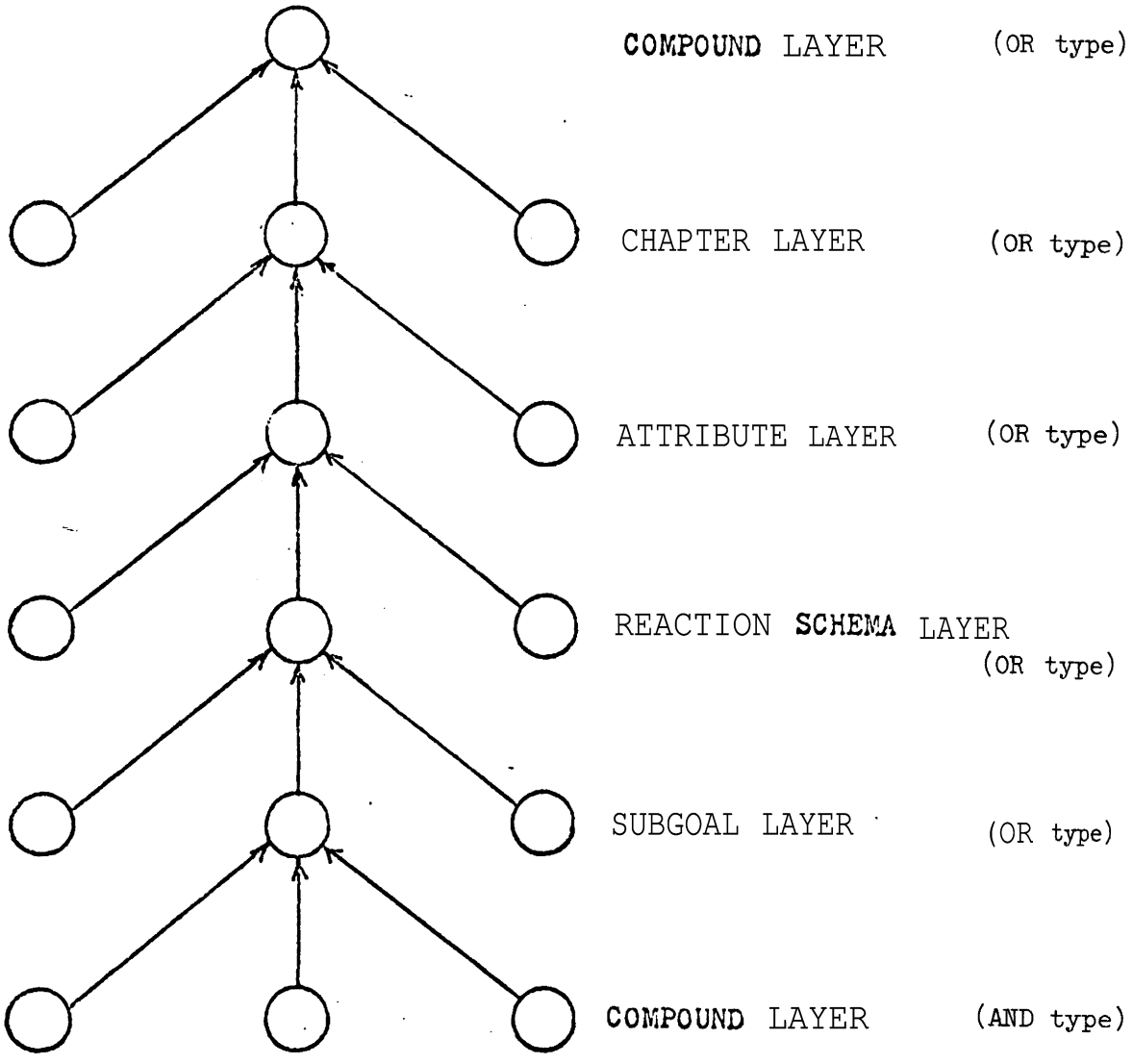



Figure 2. FLOWCHART OF SEARCH ALGORITHM




FIVE-LAYER STRUCTURE OF THE AND/OR PROBLEM SOLVING TREE

Figure 3

## SAMPLE PROBLEM AND EFFORT SPENT

It is a matter of considerable difficulty to estimate the size of search space either in general or for a specific example. An attempt is made here however, to arrive at a figure for the search space of the compound VITAMIN A. This compound bears a complex structure (Figure 4) and has held the attention of synthesis chemists for more than a decade of research effort.

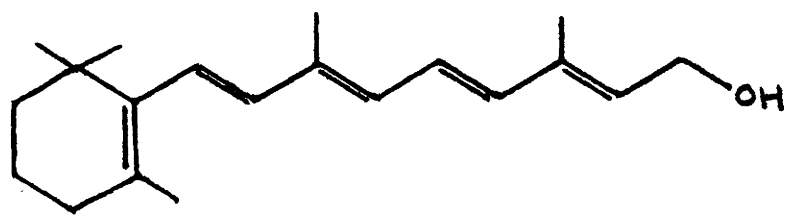



Figure 4. Structure of VITAMIN A

There are two synthemes of the molecule for which the program finds reaction chapters. There are five instances of the syntheme DOUBLEBOND and one instance of the syntheme ALCOHOL, Thus there are six attribute nodes in the first level of subgoal generation (Refer Figure 5). The reaction chapters have five and four reaction schemata in the respective chapters. One schema is invalid according to the tests and one schema fails in matching the goal pattern specified

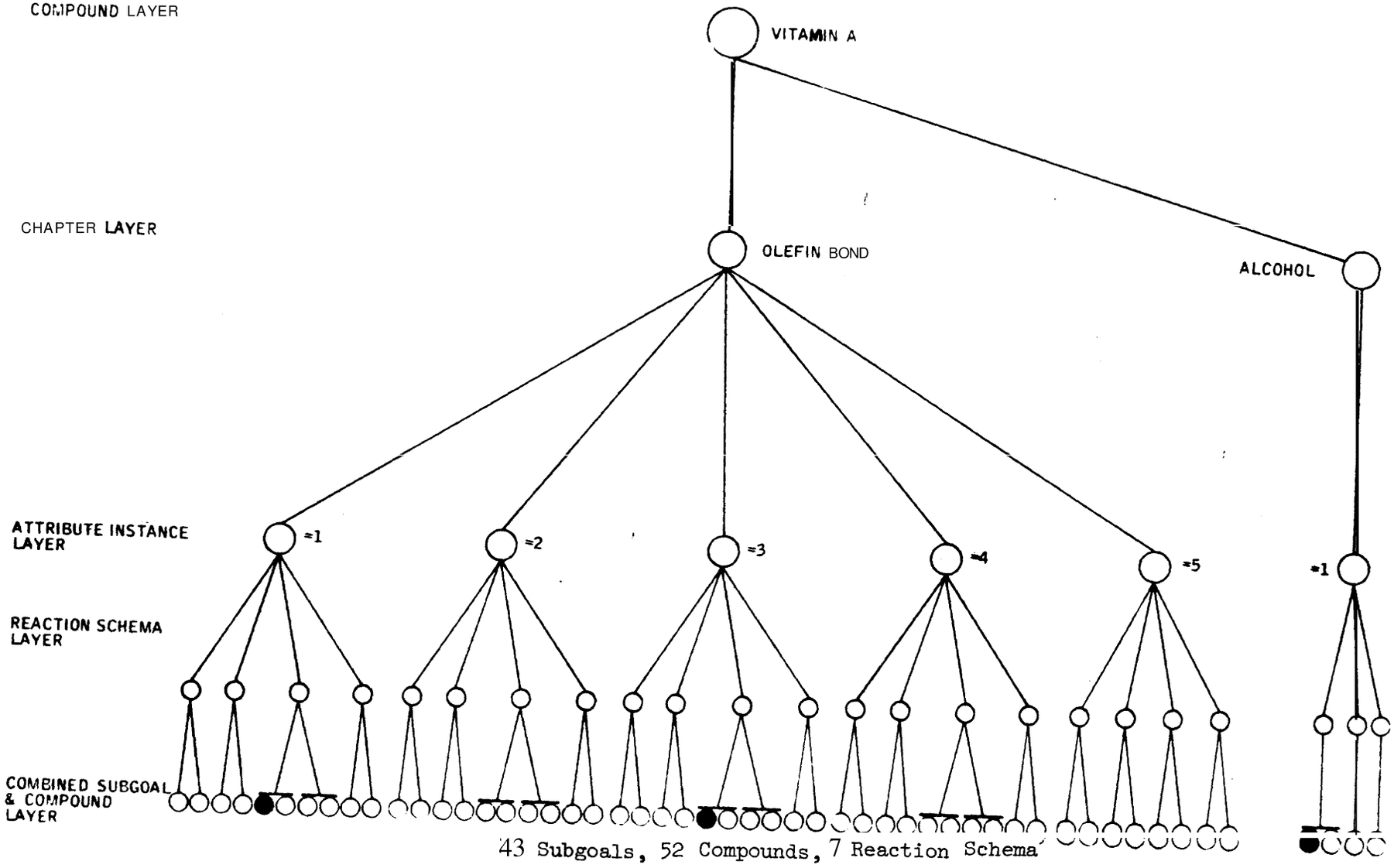



Figure 5.
ONE LEVEL OF PROBLEM SOLVING TREE GENERATED BY THE PROGRAM

Legend: Filled in circles represent compounds found available in the Aldrich Catalog

in the transformation, with the structure of the molecule. After validating and pruning out duplicates, 43 subgoals are entered in the problem solving tree to conclude the first level of subgoal generat ion. None of these subgoals completes a synthesis for Vi tamin A. Some of the subgoals are of single molecules while others are of two. There are 52 distinct compounds in the subgoals and only three of these are found readily available through the compound catalog.

The program developed the space to a maximum depth of nine subgoal level-s, or (9 times 5 plus 1 =) 46 layers of the problem solving tree. if the potential problem solving tree were considered to be branching uniformly at all levels, it would represent a potential search space of (50)**9 or approximately (10)**12 subgoals. However, the growth of the problem solving tree can he attenuated strongly by a variety of factors such as the duplication of subgoal compounds, the completion of syntheses or the reduction of the number of applicable operators at deeper levels of the tree, Allowing such attenuation the search space might then be of the order of (10)**9 subgoals. This estimate is conservative,

The program explored the search space for a time duration of SIX MINUTES (*) and examined about 120 SUBGOALS. These subgoals include only those generated from applicable schema, validated and retained for further perusal. Of these, over 28 suhgoals were expanded and had suhtrees developed for them. At least 6 DIFFERENT COMPLETED SYNTHESES were' extracted from the search tree, and many

more were interesting and near completion. The problem solving tree actually developed by the program is summarized in figure 6.

---

(*) Program written mainly in PL/ONE running on IBM 360/67 under Batch mode.

---

Note on Figure 6.

Synthesis-search tree (schematic) for Vitamin A. Filled-in circles represent reactants of subgoals selected for further development. Order of development is indicated by the circled numerals. Compound nodes connected by a horizontal line segment (as in subgoal 3) are both required for a given reaction. All generated subgoals on the tree that were not selected for exploration are represented by a horizontal bar, with the number of subgoals in the unexplored group indicated under the bar. Subgoals that were selected for exploration that have no progeny on the tree (as in subgoal 8) failed to generate any subgoals that could pass the heuristic tests for admission to the search-tree.

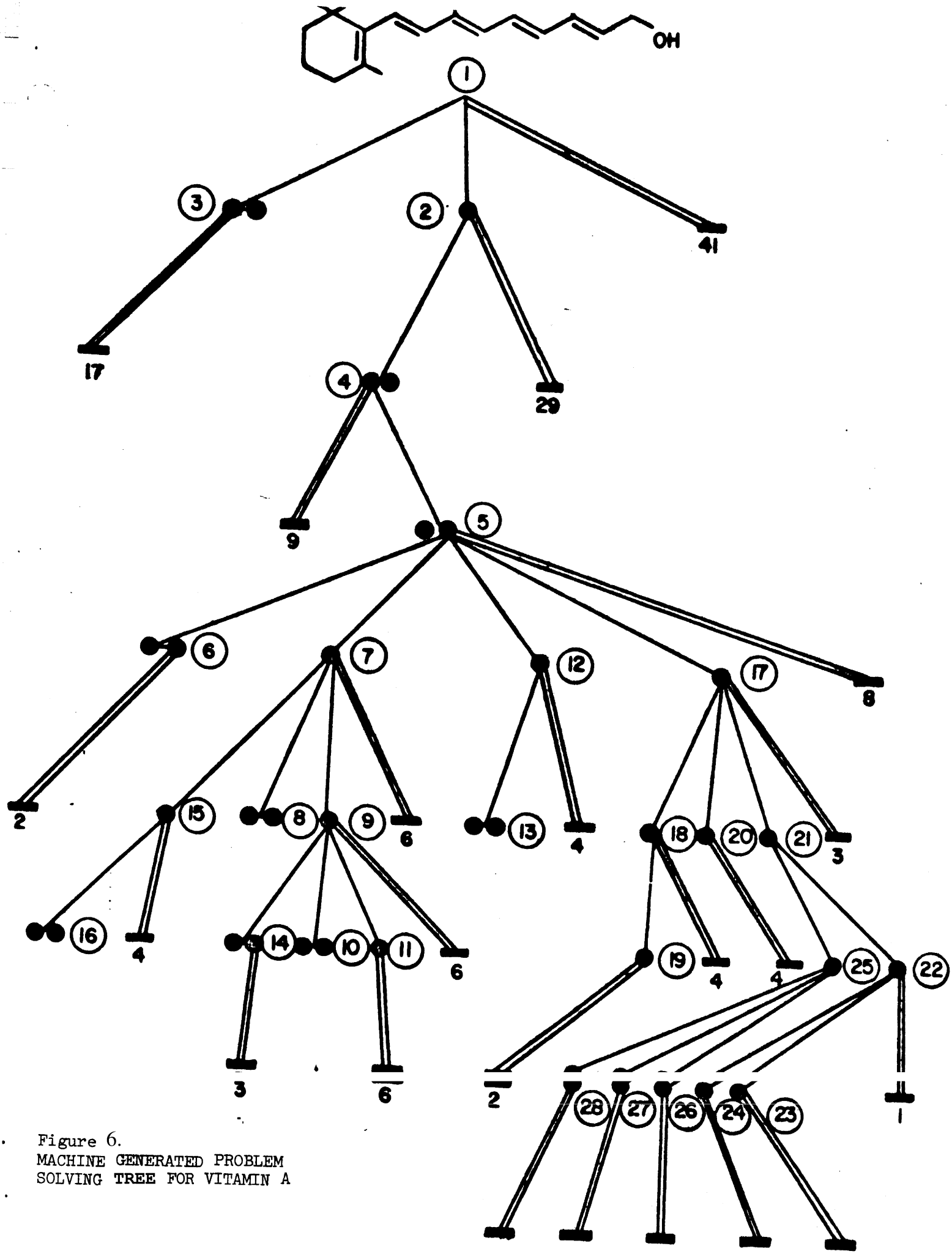



Figure 6.
MACHINE GENERATED PROBLEM SOLVING TREE FOR VITAMIN A

## DESIGN OF SEARCH STRATEGY

The Importance of guiding the search properly through the search space cannot he overemphas i zed. Many a designer of AI programs has wrestled with the questIon of what is the 'best' strategy for guiding heuristic search, taking into account the charactertstfcs of the space and the requirements on the solution. The strategies considered vary in their choice of primitives and their sources of information.

The programmed determination of a search strategy -- an aspect of what may he termed the PARADIGM ISSUE IN ARTIFICIAL INTELLIGENCE -- is worthy of attention. Although we do not have a program to generate tts own strategy as yet, we do have a program that selects a strategy suitable for the sttuatton from among prespeciffed alternatIves. The following strategies can either be observed as program's hehavlour or can be consldered useful for Tncorporatfon.

## FIXED STRATEGY IN CHEMICAL SYNTHESIS

Fixed strategies are useful when one needs to be systematic in generation. The depth-first and one level breadth-first strategies are well known and are quite unsuitable for developing syntheses.

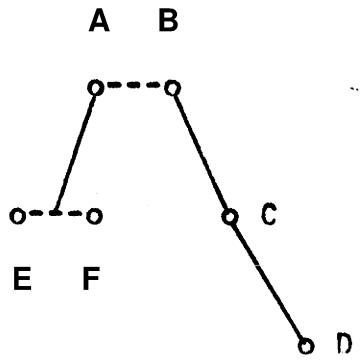
However, under most schemes of evaluation and subgoal selection there are situations when several contenders tie to the highest value. A fixed strategy is usually pursued in those instances. The synthesis program will select the latest subgoal first among those whose priority is not resolved otherwise.

Most organic compounds of 'small' size are either available or can be easily synthesized. When the program encounters small compounds that are readily available, search is terminated along that path after assigning a compound merit determined by the catalog entries like the cost of the substance. Search is terminated for small compounds even when not readily available, with the computation of the estimated difficulty of its synthesis.

## PARTIAL PATH EVALUATION IN CHEMICAL SYNTHESIS

The predominant strategy that the program uses is to evaluate every path in the search tree leading down from the prime target molecule and to choose one that gets the highest value. The compounds that terminate the branched path and the reactions used in every step enter into computing the value for each path. The program has rules on computing compound merits, combining merits of conjoined compounds to get subgoal merits and combining those with reaction merits to obtain values that can be backed up the tree.

Conjoined subgoal compounds A and B




Backup Merit-. for C

= f( Merit of 0, Reaction Merit D --> C )

Backup Merit for B

= f( Merit of C, Reaction Merit C --> B )

Backup Merit for A

= f( Merit of E, Merit of F

Reaction Merit of E + F --> A )

Reaction Merit of E + F --> A )

Backup Merit for Subgoal AB = g( Merit of A, Merit of B )

Presently, the functions f and g simply multiply their arguments and return the product normalized to the scale O-10. The definittons are present 1 y adequate but can be changed easily.

The selection of subgoal proceeds from the top of the tree downward, selecting the subgoal with the highest merit at every level. However, conjoined compounds represent AND-nodes in this AND-OR tree,

and so the compound with the least merit is chosen from among conjuncts. This is in accordance with the general strategy of dealing with AND-OR problem solving graphs.

The evaluation, backup procedure and goal selection are described in fuller details in the thesis (reference 1).

## COMPLEXITY/SIMPLICITY OF SUBGOAL COMPOUNDS

At every stage of evaluation and search continuation, the terminal nodes of the search tree are compounds. A Graph-Traverser-1 like strategy will evaluate the terminal nodes and continue search with one of highest merit. In designing syntheses, the intervening reactions are as important as the subgoal compounds. Thus this strategy in itself is unsuitable. But again, among partial paths that get equal evaluation, it is reasonable to choose those that are terminated by subgoals of higher merit. (If the subgoal is of higher merit this would imply that the reactions are poorer on that path; thus one may actually prefer terminating subgoals with the lowest merit depending upon solution requirements.)

## SIZE OF SEARCH SPACE

It is also reasonable to use an estimated size of search that may ensue on different paths, in order to continue search. It is especially useful when such program resources as time or storage are dwindling or when the evaluation leaves a LARGE NUMBER of subgoals of equal priority.

## APPLICATION OF KEY TRANSFORMS IN CHEMICAL SYNTHESIS

The democratic tenet "All reactions are created equal" has to be cast aside, in order to allow preferential treatment for key transformat fans. The present reaction library contains a priori merit ratings of react ion schemata. The merit of each schema is further adjusted when used, to correspond to the specific application of the transformation. This technique allows preferred pursuit of paths having the key transforms.

This a priori preference system can be overridden by the program under special situations. An example is the technique known to chemists as BLOCKING or PROTECTION. Blocking of certain structural features of molecules is a very useful synthesis technique facilitating solutions to many problems. Sometimes a synthesis without Hocking may not be possible. With reference to Figure 7, the reasoning may proceed as fol lows.

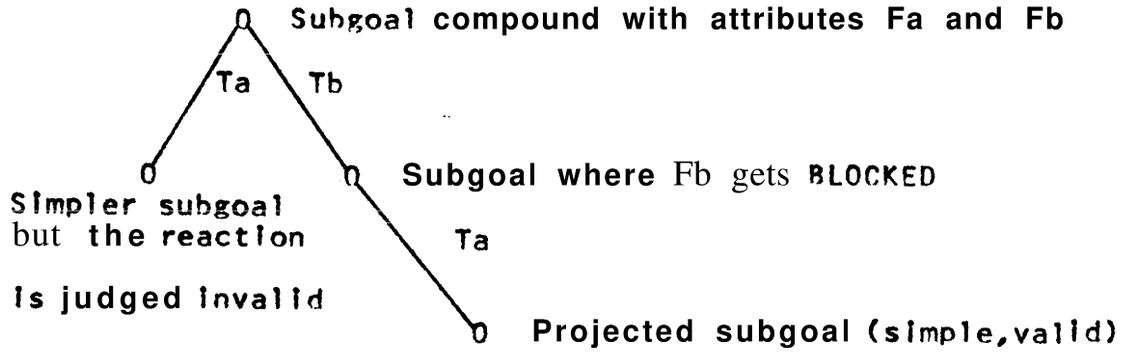



Figure 7. APPLICATION OF KEY TRANSFORM - BLOCKING

The transformation Ta is a preferred transformation but it is made inapplicable as functional group Fb is very sensitive to the reaction, making it invalid. The transformation Tb which does not have a priori high merit, however, removes Fb or changes it to Fb'; and Fb' is not sensitive to Ta. Thus subgoal resulting from Ta can be terminated. The subgoal from Tb is realized to have higher merit in this context, because it can now be subject to Ia to yield a simpler valid subgoal. Such a sophisticated attention refocussing scheme using contextual evaluation produces excellent results, by overruling the standard evaluation and forcing development along lines that are intuitive to the consulting chemist.

## SELECTION AND ORDERING OF ATTRIBUTES

Some attributes of molecules prove to be more sensitive than others toward all or most transformations. Thus, while selecting attributes one may impose an order of preference or one may exclude certain attributes, saving the effort to be spent on whole chapters of the reaction Library. The a priori ordering of attributes with

due consideration to reactivities is another piece of chemical knowledge thus available.

Further, a contextual reordering is possible here. Vitamin A for example, has four instances of the attribute OLEFIN BOND. One of the operators results in a smaller but similar compound with only three OLEFIN BONDs and the reaction itself has high merit. When continuing search with this new subgoal a clear indication now comes from the above observation, to prefer to operate on another OLEFIN BOND. The similarity of the resulting compound also raises the expectation that successive application of the same transformation may solve the problem at hand.

KEY INTERMEDIATE COMPOUNDS IN CHEMICAL SYNTHESIS (suggested)

Some compounds can be changed quickly into a variety of similar but different compounds and are often used as key intermediate compounds In synthesis. When a subgoal compound is similar to a readily available key intermediate, synthesis search may profitably be geared toward the specific intermediate. On the other hand, when a key intermediate subgoal is generated that is not available a synthesis for that intermediate subgoal is to be actively pursued with high priority.

USE OF ANALOGY IN CHEMICAL SYNTHESIS (suggested)

Quite often chemists arrive at syntheses by following the known synthesis of an analogous compound. Situations where solution (or simplification) by analogy can be applied arise profusely:

the goal compound is analogous to a compound whose synthesis is published, a key intermediate can be synthesized by analogy to an available key intermediate, a subgoal generated is similar to one or more intermediate compounds generated and solved by the program during this run alone. However the advantages of overruling normal search by reasoning through analogy in these situations is not clear.

It is needless to emphasize that the synthesis of an intermediate compound solved at one instance in the problem solving tree is available throughout the course of the program run and is reused by direct reference.

## EXTERNAL CONDITIONS GUIDING THE SEARCH

There is need for tempering the selection of syntheses with such considerations as the toxicity of the substances to be manipulated, special apparatus needed to contain and react gases and cost associated with expensive commercial compounds, reagents or catalysts. However the problem at present is seen as being one of filtering out syntheses not desired from the output of the program. this allows a fuller set of prejudices and personal preferences of chemists to be imposed upon the choice of syntheses.

We have consciously avoided developing an interactive system where a chemist supplies guidance on-line to the program. Our interest in the problem is mainly as an A? endeavour and to that extent our attention was given to designing a good blend of search strategies as outlined above that could effectively substitute for the chemists' guidance.

## REMARKS

The strategies discussed above fall roughly into subgoal-dependence, transform-dependence and partial-path-dependence, The criteria to be used in *each* strategy (the limits, thresholds, orderings and merit boosts) can have several sources of information (Figure 8).

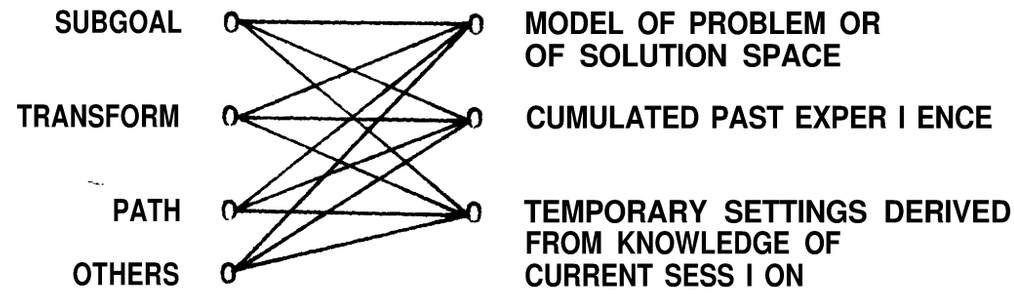



Figure 8. SOURCES OF INFORMATION AND STRATEGIES

Firstly, quite often the criteria derived from models (implicit or explicit) are in the form of absolute limits or fixed orderings, reflecting the static nature of the model one has in mind. In "tuning" these criteria, one is readjusting the model of the problem or solution space. Secondly, in certain cases, the program can be delegated the task of keeping itself tuned with respect to certain criteria, using cumulated past experience, giving rise to an adaptive (and may be learning) characteristic. Thirdly, the contextual evaluations explained in the last section illustrate how the program can, using knowledge acquired from the current session, temporarily overrule a model prescribed to aid it in finding better solutions faster, without leading to adaptation or adjustment of the model.

**Acknowledgement:** Help from Mr. Arthur Hart and Mrs. No-Jane Shue, and guidance from Professors Herbert Gelernter and Frank Fowler is acknowledged with deepest thanks.

I also thank Dee Larson for competent secretarial help.

## REFERENCES


1. Sridharan, N.S., An Application of Artificial Intelligence to Organic Chemical Synthesis, Doctoral Thesis, State University of New York at Stony Brook, New York, July 1971. (available through University Microfilms.)

   Sridharan, N.S., et.al., A Heuristic Program to Discover Syntheses for Complex Organic Molecules. Stanford Artificial Intelligence Laboratory Memo AIM-205 (CS-73-370), June 1973.

2. Buchanan, B.G., and Lederberg, J., "The Heuristic DENDRAL Program for Explaining Empirical Data", Proc.IFIP Congress 71, Ljubljana, Yugoslavia (1971); (also Stanford University AIM 141).

   Nilsson, N., "Searching Problem-Solving and Game-Playing Trees for Minimal Cost Solutions", in A.J.H. Morrel (ed.), Information Processing 68, Volume 2, pp. 1556-1562, North-Holland, Amsterdam, 1969.

3. Smith, E.G., The Wiswesser Line-Formula Chemical Notation, McGraw-Hill: New York, 1968.

4. Corey, E.J. and Wipke, W.T., "Computer-Assisted Design of Complex Organic Syntheses" in SCIENCE, Volume 166, October 1969, p. 178-192.