Computers and Organic Synthesis

MALCOLM BERSOHN* and ASHMEED ESACK

Department of Chemistry, University of Toronto, Toronto, Ontario, Canada, M5S 1A1

Received December 17, 1974 (Revised Manuscript Received February 26, 1975)

Contents

I. Preface

An observer of the achievements of synthetic chemists cannot fail to be amazed at the speed with which they are discovering new useful methods and improving old methods. Compared to this dazzling revision of the ways in which chemical transformations are being effected, the beginning trend to design syntheses by processing the already known information by a computer program is a relatively minor development. The most frequently asked question about such programs is "Can they help me now?". The answer to this question in 1976 is "no, if you have great skill and years of experience." (If you are a novice, even the programs of 1969–1971 would have been of some help to you.)

Logically the next question is "Will such computer programs ever be indispensable for even the great synthetic chemists?". Statements about the future perhaps do not belong in a review article so we content ourselves with pointing out that many crucial breakthroughs have already been made in this area. Chemical reactions have been simulated; stereochemistry has been correctly manipulated; strategies have been implemented; the presence of the chemist at the time of execution of the program has been shown to be dispensable. The programs need to be improved in certain obvious ways such as increasing the number of reactions available and increasing the detail in specifying the conditions under which reactions are applicable. The great development begun by E. J. Corey of exactly defining synthetic strategies needs to be extended.

As the available synthetic reactions become more and more diverse, the management of this complexity in designing efficient syntheses will become a more and more formidable problem, even if the syntheses are to be "standard" (pedestrian?) ones using only known chemistry. We can expect more and more effort by chemists to enunciate precisely the rules for managing the complexity of synthetic possibilities. A computer program is one natural way to implement these rules.

II. Introduction

A. The Computer as a Nonnumerical Chemical Problem Solver

Physicists and physical chemists commonly use a large computer as an arithmetic tool, a giant calculator. Conceived of as merely a big arithmetic machine for solving equations, the computer would seem useless to the synthetic organic chemist; arithmetic enters only trivially into synthetic chemistry. Actually, computers have wider capabilities than arithmetic; they are machines for executing any systematic procedure for manipulating data. The computer processes not only numbers but information in general. This is because information can be represented by symbols; symbols can be represented inside a computer by appropriate numbers. The computer can compare strings of symbols and take further action that depends on the result of the comparison. (Three results are a priori possible; the strings can be the same, or string *x* can have a greater or a lesser numerical equivalent than string *y*. This allows room for three different subsequent actions, depending on the result.)

Suppose, for example, we define in a program a symbol string that represents the carbonyl group of a ketone. A computer under the direction of such a program can take different subsequent action depending on the comparison of the parts of a molecular structure in its memory with the string that represents a ketone carbonyl group.

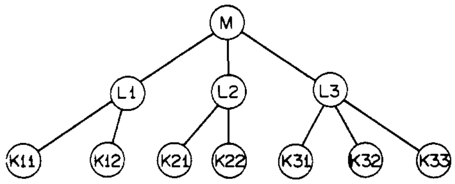


Figure 1. A synthetic tree. M is the goal molecule.

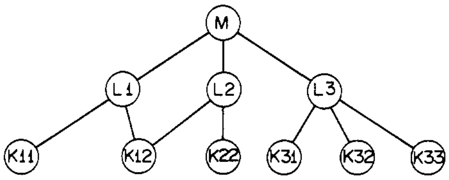


Figure 2. A synthetic graph. There are two routes to the goal from K12.

Another capability of the computer is the manipulation of addresses of the information in its memory. For example, the symbol string representing the ketone carbonyl group is a number, and, using this number, we can find in a table a list of addresses of information describing appropriate reactions for the ketone carbonyl group (either to produce it or using it as reactant). The ability to manipulate addresses leads to the ability to make links between information. For example, with the dossier that describes molecule *z* we can enclose the addresses of the dossiers of molecules *x* and *y* which react together to produce molecule *z*. If the addresses are placed or marked appropriately, a suitably written program will recognize that these are the addresses of the information describing the reactants that gave rise to *z*. It is apparent that if we can reduce the thinking that is required to generate an optimal synthetic route to a systematic procedure, then that procedure can be programmed for execution by a computer.

B. Historical Notes

The idea of obtaining suggestions for good syntheses from a computer was first put forward by Vleduts.[1] The first actual contact of computers and organic chemistry was in the work of Lederberg and coworkers on the elucidation of molecular structure with the aid of mass spectra, entirely carried out by their computer program, named DENDRAL.[2] In this program, series of plausible reactions are carried out for every isomeric possibility to determine whether the daughter ions to be expected actually appear in the spectrum. In this work, there already appeared multistep reaction sequences, with the elementary operations of making and breaking bonds, adding and removing hydrogen atoms, etc.

Synthetic chemistry appeared in a computer program for the first time in the famous work of Corey and Wipke,[3] which presented the computer-assisted approach using a one-step program. Subsequently, the development of another interactive program[4] and two noninteractive, multistep synthesis programs[5,6] were reported. This review is complete only for journal articles through August 1974.

III. The Synthetic Graph: A Map of the Problem of Synthesizing a Compound

A. Synthetic AND/OR Graphs and Synthetic Trees

The information structure of the synthesis problem has been variously treated.[3,5] If we describe it as a tree, we have a diagram like that of Figure 1. (The reader will note that the tree is growing downward. This vagary of convention has been discussed by Knuth.[7]) In Figure 1, M is the goal molecule and L1, L2, and L3 are possible predecessors of it; in other words, L1, L2, or L3 could give rise to the goal molecule M in a single chemical step. Similarly, the molecules K11 and K12 are predecessors of L1, etc. On close examination of this tree concept, we see that Figure 1 is really accurate only in describing isomerizations; e.g., K22 isomerizes to L2 which, in turn, isomerizes to the goal molecule M. However, as is known, the usual reaction is not an isomerization; usually at least some small molecule like CO_2, CH_3OH, H_2O, O_2, H_2, etc., is a coreactant. If we dismiss such trivial molecules from view, Figure 1 looks more promising as a description of the situation. However, often some molecular structures appearing as subgoals in different parts of the tree are, in fact, identical. From experience, we know that there are more than one way to proceed to a goal molecule from some key intermediate. If, for example, K12 and K21 are the same molecule and we decide not to repeat a given molecule anywhere in the diagram of the situation, then we have the description of Figure 2. Figure 2 is not a tree, since it has rings in it, impossible for a tree by observation, from botany, and by definition, from graph theory.

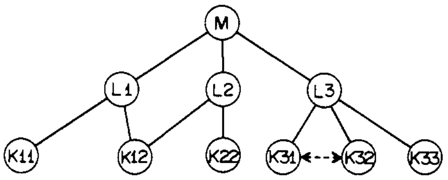


Figure 3. A synthetic AND/OR graph. K31 and K32 are coreactants.

There is still one more important aspect of the picture which is not described by a tree. Many reactions are not describable as A + B → C, where A is nontrivial and B is trivial. Often both A and B are not ordinary reagents and present synthetic problems, so both can be classified as subgoal molecules. This implies a special relation between A and B, i.e., that they are both necessary for the production of C. Suppose, for example, that K31 and K32 react together to form L3 and neither K31 nor K32 is trivial. Then, to be fully descriptive, we write a special sidewise link between the two coreactants, K31 and K32, as in Figure 3. In Figure 3, we can see most clearly that there is a synthetic route available which is K31 + K32 → L3 → M. Other routes are K22 → L2 → M, K11 → L1 → M, etc. A diagram such as Figure 3, with sidewise links between nontrivial coreactants, would be called by the computer scientist an AND/OR graph. It is the easiest way to represent the problem inside the computer; hence, it is popular with computer scientists. A diagram like that of Figure 1 is called a synthetic tree. It is only reasonable to refer to the information structure of the synthesis problem as a synthetic tree if we remember that (1) the coreactant relationships are omitted and (2) the same molecule will often appear at different places in the tree.

B. Effect of Breslow-Type Remote Functionalization Reactions

The average number of predecessors that we can find for molecules depends on the number of applicable reactions. The number of applicable reactions in turn depends on the number of functional groups present in the molecule since most synthetic reactions can be described as the transformation of functional groups into functional groups. In recent years, however, Breslow[8] has been demonstrating that it is possible to introduce functionality into a molecule far away, in fact, an arbitrary distance away from some other functional group. This means that functional groups can appear in a site in a molecule where there previously was no functionality or neighboring functionality. If, as seems possible, this type of remote functionalization reaction becomes widespread and well developed, then the average number of immediate predecessors that we can find for molecules will vastly increase. At first thought, this might seem to imply that systematic, i.e., algorithmic, attack on the synthesis design problem becomes unfeasible because of the unmanageably large number of possibilities. However, Breslow-type remote functionalization reactions should make the lowest cost synthetic routes much shorter. Hence, by a sort of conservation process, the AND/OR graph and the synthetic tree become very much broader and considerably shorter. There are many more paths to consider, but their average length is quite shorter so the number of intermediate compounds necessary to generate may not change appreciably.

IV. Algorithms for Finding Optimal Syntheses

A. Algorithms: Precisely Defined Stepwise Procedures

An algorithm is a precisely defined stepwise procedure for solving a problem which is effective after executing a finite number of such steps. Algorithms may be inefficient but they must be effective. The precise definition of each step means that there must be nothing left "understood" but unstated and there must be no ambiguity or vagueness.

Let us consider what possible strategies for finding best synthetic routes can be rigorously described.

B. Backward Search without Pruning

From the goal molecule M, we derive all possible predecessors L1, L2, . . ., L*n* (cf. Figure 1). From each possible predecessor L*i* we derive all possible predecessors K*i*1, K*i*2, . . ., K*im*, and so on until a predecessor is found which is an available substance. Then the pathway from the available substance to the goal molecule M is a possible synthetic route provided that all coreactants along the route can also be synthesized. After collecting a number of such possible synthetic routes, we choose the one(s) judged by the program to be cheapest for experimental trial. Presumably, we complete the effort of generating synthetic routes when the cost barrier of computer expenses has been reached.

The computer scientist would describe the above algorithm as a "blind" process since all routes are investigated indiscriminately.[9]

The objection to the backward search method without pruning is that we will suffocate in the dense thicket of possibilities. If each molecule in the diagram has an average of n chemically reasonable predecessors, then, after we have found all possible k-step synthetic sequences (which may not begin at available molecules), we will have examined n^k sequences. For important problems where the goal molecule has several functional groups and several chiral centers, then n can be 40 or more. All possible five-step sequences will then involve more than 40^5 successive molecular structures, i.e., more than 100 million. Days of computer time on the largest and fastest computer would be required; backward search without pruning is definitely impractical.

C. Backward Search with Cost Pruning

The history of every computer application is marked by an initial overoptimism. The overoptimism results from the fact that people are successful in getting the computer to consider and solve simple problems in their area, but the transition from simple problems to complicated ones is much more difficult than anticipated. For example, it was a widespread belief in the early 1960's that only a decade would suffice and then a computer program would be produced that could play chess as well as any unassisted human being. However, the situation in 1973 is succinctly described by Mittman[10] as follows: "Some chess programs have earned a rating of Class C but they lack the human ability to screen out uninteresting lines of activity without careful analysis." It is fair, also, to ask if a program that looks for optimal syntheses without human intervention is not inevitably similar to the chess-playing programs in that it could be interesting but never practically appealing, because the size of the problem precludes a mechanical, exhaustive treatment. As to this point, we should consider the possibilities afforded by cost pruning.

If we have some minimum requirement of yield, or equivalently some maximum limitation on the cost, then "branches" of the synthetic "tree" can be eliminated on a large scale. This feature is absent in a chess-playing program since a possible line of play cannot rigorously be excluded from consideration before the player is checkmated in his hypothetical use of this line of play. Suppose that the goal molecule has 40 immediate predecessors, but that these latter have, in turn, an average of only 30 predecessors since the overall yield requirement will rule out some a priori possibilities. The 1200 two-step sequences may give rise only to 12 000 three-step syntheses and, after three steps, the consequences of pruning may be quite drastic. These numbers are somewhat hypothetical and await experimental demonstration by a total synthesis program which is equipped with a relatively full repertory of synthetic reactions. In any case, if we take a 100 000 molecule pruned synthetic "tree" as the maximum practical for present day computers, then it appears likely that the algorithm of backward search with cost pruning is practical in at least some cases.

D. Backward Search with Cost Pruning and Heuristics

It may be practical for the computer to consider 100 000 different molecules; it certainly is not so for a human being. In addition to the trick of pruning away overly costly pathways, the human being must be equipped with a number of additional tricks. Most or all of these tricks may occasionally result in the overlooking of good synthetic routes. But, as a reward, they clear the thicket of possibilities and enable one to rule out lines of approach which, on the basis of experience, are relatively unpromising. All such strategems which simplify the problem on a probabilistic basis, using practical experience as a guide to the "promise" of uncompleted synthetic routes, are referred to as "heuristics" by the computer scientist.[11] Heuristics prune the search tree far more completely than cost limitations. As an example, if the molecule has a labile functional group, such as a β-hydroxy ketone, which easily dehydrates in the presence of either acid or base, then a heuristic is to form the labile group at the end of the synthesis. This single heuristic rules out a large majority of otherwise promising paths.

The "promise" of incomplete synthetic routes is estimated by an "evaluation function". Chess-playing programs do poorly because they lack good evaluation functions.

In the discussions by Corey[12,13] of the heuristics for finding optimal synthetic routes, pruning is not referred to explicitly. Instead, he discusses the ordering of incomplete routes by the evaluation function. If some routes are favored for investigation, the finiteness of investigation resources, e.g., time, will guarantee a de facto pruning.

E. Forward Search

In a forward search for a synthetic path, we start with available materials and hope to arrive by stages at the goal molecule. Presumably, the starting material(s) will be chosen because of some resemblance to the goal molecule. The guiding principle seems to be to make each product resemble the goal molecule as much as possible. We know of no advocacy of this method in the literature; informal presentations of synthetic achievements are often couched in the forward search terminology. "The attempt to convert X to the desired XIV by the blank reaction failed, but we succeeded in transforming X via XI, XII, and XIII to XIV in a good overall yield." In industry, the forward search becomes mandatory when there is available a large quantity of some byproduct for which no use is known. The problem of how to make the byproduct into a goal molecule is necessarily a forward search.

F. Combined Backward and Forward Search with Cost Pruning and Heuristics

Corey believes[12] that the synthesis program of the future will combine backward and forward searches. Key intermediates of the forward search will become available substances for the backward search. Important intermediates of the backward search will become goals of the forward search.

V. Multistep Synthesis Programs and One-Step Programs

One approach is to delegate the whole task of generating optimal synthetic routes to a computer program. We can call this kind of program a multistep synthesis program. The other approach is to carry out the generation of optimal synthetic routes by a system consisting of a chemist plus a program which generates for him all possible intermediate predecessors of any molecule that he is interested in. This is the computer-assisted approach. The program used could be called a one-step program since it does not by itself generate synthetic paths. The ultimate decisions about which incomplete paths to investigate further are made by the chemist, although the one-step program may rank the predecessors according to how promising they appear. The relative merits and demerits of these two approaches are so numerous and so mixed that it is safe to predict that both types of programs will be developed for many decades. Great subtleties of organic chemistry are to be found in the variations in yield away from a standard figure that occur when a standard reaction is applied to a particular molecule. In this regard, the interactive system has only those limitations of the chemist's knowledge. Hence, it is immediately practical. The results will be spectacular or poor depending on the skill of the chemist directing the system. In any case, the one-step program cannot fail to improve most chemists' performance in the design of syntheses.

The multistep synthesis programs have to have built into them the knowledge about variation of yields, depending on the presence of various functional groups, the proximity of these functional groups to the reaction site, steric hindrance, etc. To the extent that the multistep synthesis program is deficient in this knowledge, it will generate poor suggestions for total syntheses. Removing this deficiency involves building in more knowledge; this process is slow, consuming many man decades, but we do not see any practical limit to the educability of such programs. What man knows about chemistry he can state in a rigorous form. Ultimately, the enormous speed of the computer relative to the human brain will probably make the multistep synthesis program the preferred system.

Having mentioned the difference between these two approaches, we point out the similarities. In both cases, the program must be able to examine the molecule and find its objects of synthetic interest (Corey's "Synthons" [12] and Gelernter's "Synthemes" [6]). These are functional groups or rings or combinations of these. In both cases, we organize the reaction list (Gelernter's reaction library) into subdivisions which contain all the reactions that produce a particular synthetic object. Typically, there are chapters for aldehydes, ketones, nitriles, etc., three-membered rings, cyclohexenyl rings, and so forth. Yields can be assigned based on the typical yield reported in monographs dealing with the reaction.[5] Alternatively, some figure of merit is assigned to each generated predecessor, depending on which reaction it participated in to produce the next molecule nearest to the goal molecule.

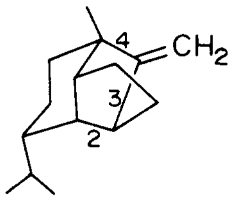


Figure 4. The strategic bonds of sativene.

VI. E. J. Corey's Heuristics for Selecting Preferred Routes[12,13]

We recall that a heuristic is a problem-simplifying strategem. The heuristics for simplifying the synthesis problem, or, in other terms, the heuristics for selecting the optimal strategies of synthesis remained largely unsystematized and even unarticulated until the need for a program to consider synthesis problems was advanced by a synthetic chemist.[13] This aspect of Corey's work, the precise specification of simplifying strategems, is useful to all synthetic chemists, whether or not they use a computer program.

1. Preliminary Scan of the Problem

Three quick questions that can be asked are:

a. Is there symmetry or near-symmetry of two parts of the molecule? If so, one should try to make the molecule by joining two identical or similar fragments.

b. Is the problem much like an already solved problem? For example, if we have to make a new indole, one should presumably use Fischer's indole synthesis.

c. Is the molecule a string of available or at least simple pieces? If so, we should assemble the chain by standard procedures. It would be foolish, for example, to make a polypeptide by some route which requires the making of carbon to carbon bonds. This heuristic is referred to by Corey as a "direct associative" strategy.

2. Strategic C–C Bonds in Ring Systems

Corey's definition of a strategic C–C bond in a ring system is as follows:

A strategic C–C bond must (1) be endo to a five-, six-, or seven-membered ring; (2) be exo to a ring larger than 3; (3) be a perimeter bond (this means it may not lie on the intersection of two rings if the envelope of these two rings has seven or more atoms); (4) be endo to a ring of maximum bridgeing (i.e. ring(s) bridges to maximum number of other rings); (5) not leave stereocenters on side chains after cleavage; (6) minimize the cyclic order of the largest resulting substructure.

To illustrate the definition and its component six rules, Corey gives an example which we repeat here. There are 12 bonds in the three rings of sativene. Of these, only bonds 2, 3, and 4 are strategic (cf. Figure 4).

The heuristic that accompanies the concept of the strategic bond is that we should make the strategic bond last or as late in the synthesis as possible. If the bond cannot be made without introducing functionality that is absent in the goal molecule, then it should still be made and the additional or incorrect functional group(s) can be removed or altered later.

3. *Labile Groups to be Added Last*

If the goal molecule has functional groups which are very sensitive to acid or base, they should be introduced last or as late as possible. Presumably, only a computer program needs to be taught this, but we include this heuristic here for completeness.

4. *Create as Much Functionality in the Product as Possible*

In generating the immediate predecessors of the molecule, we should preferentially select those reactions which create more functional groups in the product. Thus, if other things were equal, we would favor obtaining an allyic bromide from bromination rather than from transformation of an allylic alcohol.

5. *Favor Closely Related Molecules as Reactants*

This is probably a relatively low-priority heuristic. It tells us that when we are generating the predecessors of a given molecule, we should favor those reactions that proceed from reactants closely related to the product. Examples of close relatives are "cyclic ketal and ketone or carboxylic acid and carboxylic acid ester; or lactone and hydroxy acid; or α,β and β,γ-enones."

6. *Insert Interfering Groups after Key Steps*

If a key process, e.g., the making of a strategic bond, is interfered with by the presence of a certain substructure, then we should add that substructure after the key process has been accomplished. In other words, we should give a low priority to the addition of this interfering substructure. The reactant without this interfering group will be easier to produce by the key process.

7. *Favor Breaking Bonds or Bridges between Atoms of the Goal or Subgoal Molecule*

This usually consists of opening a six-membered ring to form the desired molecule. Corey gives four conditions under which this heuristic will be promising.

8. *Obtain the Goal Molecule or Subgoal Molecule by Transformations of the Product of a Particularly Powerful and Useful Reaction*

For example, if a goal or subgoal has a six-membered nonaromatic carbocyclic ring, then one should try to use the Diels–Alder, Robinson annulation, Birch reduction, or cation–olefin cyclization reactions. One should attempt then to convert the product of these reactions to the molecule of interest.

We have stated all these heuristics in a forward looking way, describing the reactions as they are actually performed. Corey has developed a vocabulary to describe the backward search, using which he has presented his heuristics. We present these terms now and close this section by restating the heuristics for the interested reader in Corey's terminology.

There are clearly two ways to program a synthetic reaction. In the product-generating form, we input the reactants and are given the products. In the reactant-generating form, we input the product and are given the reactants. Corey calls the reactant-generating form a "transform". He has not assigned a special name to the product-generating form. The backward direction is called "antithetic"; this is the direction of backward analysis. The forward direction is called "synthetic"; it is the direction of the synthetic chemical processes. Thus, a transform takes us in the antithetic direction to a reactant. The disconnecting of bonds in the transform corresponds to their making in the actual synthetic reaction. Introducing a functional group in a transform is really the removal of this functional group in the actual synthetic reaction. The transform is not to be confused with the backward reaction of the kineticist; it is merely a mental process by which we get back one step.

Corey's heuristics can now be restated in language closer to his original words.

In the retroanalysis, labile groups should be removed first.

Transforms should remove as much functionality as possible and should remove stereochemical centers where possible.

Favor transforms that generate closely related intermediates.

If a key process, e.g., the disconnection of a strategic bond, is interfered with by a certain substructure, then we should use a transform that removes that substructure.

Where possible use transforms to build bridges between the goal atoms.

Convert the molecule you are considering into the product of one of the powerful reactions, such as Diels–Alder, Robinson annulation, Birch reduction, or cation–olefin cyclization reactions.

Disconnect appendages "attached to atoms bearing certain functional groups such as OH or C=O".

Consider all pairs of functional groups in the molecule to ascertain whether known transforms can disconnect any of the intervening bonds.

In a ring system preferentially disconnect the strategic bonds.

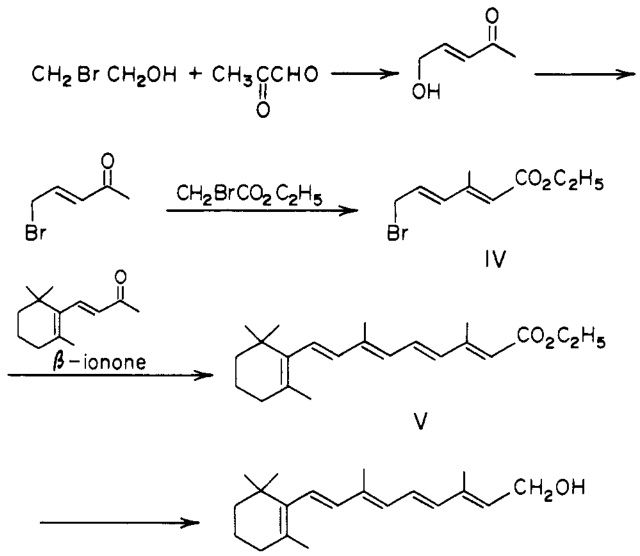
VII. *Chemical Achievements of the Noninteractive Synthesis Programs*

H. Gelernter's program has solved several dozen problems.[6,14] The level of competence required to solve these problems is about that of a beginning graduate student in organic chemistry. The program uses the algorithm of backward search with heuristics but without cost pruning. Since the program is not allowed to run forever, we have a de facto pruning by the heuristics which prefer certain lines of approach rather than others. The program developed a number of syntheses for vitamin A. One of these is shown in Scheme I.

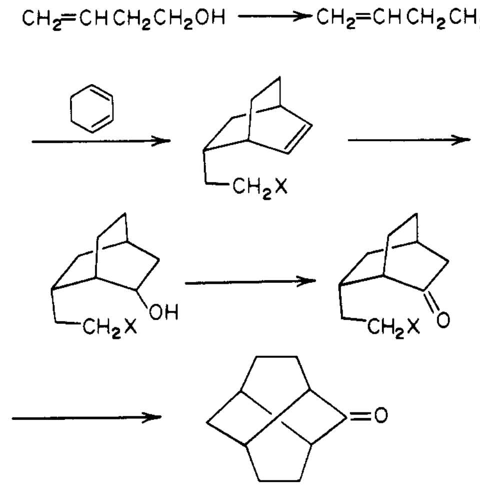
We note that the program did not attempt to produce the goal molecule directly with a Wittig reaction. This is because in the description of the Wittig reaction contained in the chapter of the reaction library containing reactions that produce the carbon to carbon double bond, a hydroxyl group in the product is indicated as making the reaction considerably less preferable. The reaction of reducing the ester subgoal to the goal alcohol thereby received a higher ranking.

With so many double bonds present in the ester subgoal V, we may ask why the program preferred to develop first the middle one. That is because, on checking the shelf library, the program discovers that ionone is immediately available, whereas none of the other possible predecessors of the ester polyene V is available. We see here the operation of the simplifying heuristic which regards the task of making IV as the easiest. Gelernter states that he is programming an available substructure maintenance heuristic which will perform two tasks.On one hand, it will recognize that the available molecule, β-ionone, is essentially contained in the goal

SCHEME I

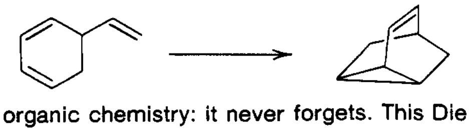


SCHEME II



molecule so the program should give priority to the production of substructures which are not contained in the β-ionone-like moiety. On the other hand, the simplification afforded by the availability of β-ionone need not be taken advantage of at once. In other words, the program could produce, as its final carbon to carbon double bond, the one furthest from the ring or the one next furthest from the ring with as high a figure of merit (ranking or promise) for the corresponding predecessors as was assigned to IV.

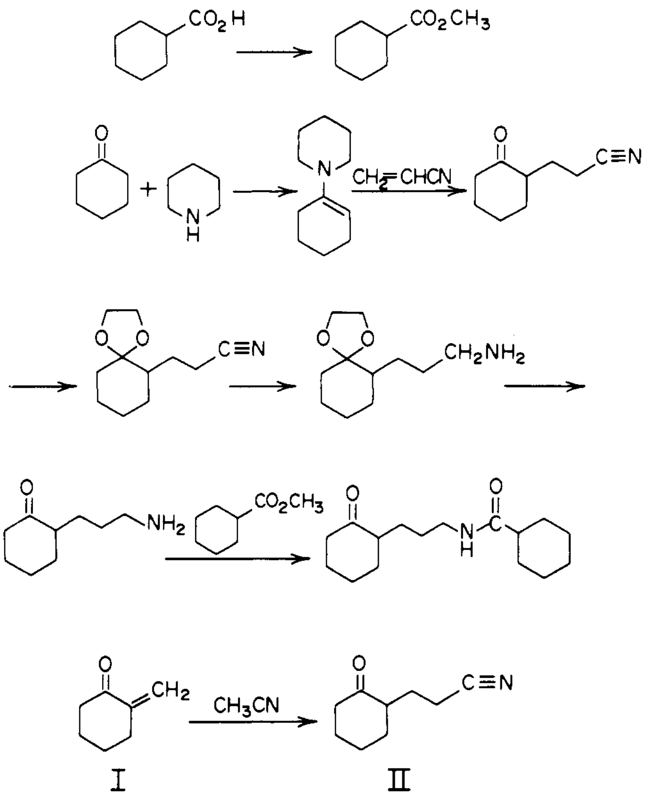
Gelernter illustrates with the internal Diels–Alder reaction below an important aspect of the power of a computer program in organic chemistry: it never forgets. This Diels–Alder reaction is not one which would come quickly and invariable to mind for a beginning graduate student in this field.



The most complicated problem attempted by Gelernter's program to date is the generation of a synthesis for tricyclo-[4.4.0.0^{3,8}]decan-2-one. A solution suggested by the program is shown in Scheme II.

The Diels–Alder of the first step probably proceeds in inadequate yield, if at all. The dienophile of the first step needs to be improved. The program description of the Diels–Alder reaction evidently needs to contain more about the electrophilic requirements for dienophiles when the diene is electron rich. This by itself is a small and easily accomplished change. The evolution of this program will involve the addition of very many such small pieces of information.

SCHEME III



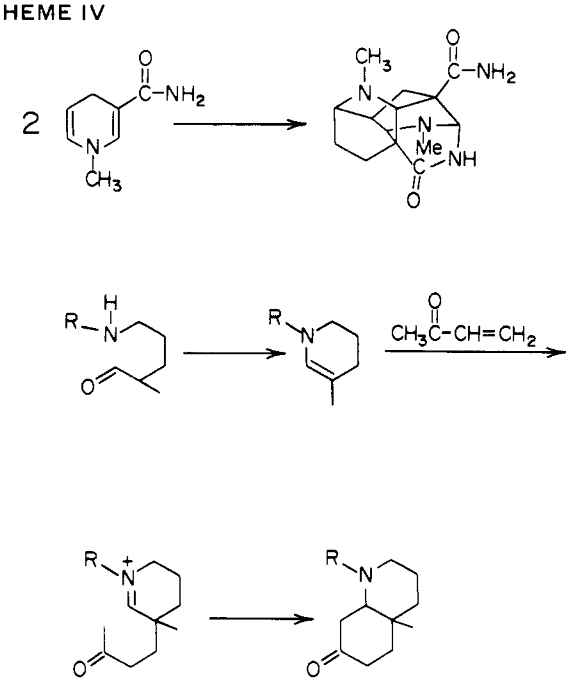
Keeping clearly in mind that, unlike in the computer-assisted approach, no chemist made any of the decisions shov.n in this synthesis, this work should be regarded as the most impressive evidence that multistep programs will be important to the technology of chemistry.

The available list or shelf library of Gelernter's program consists of 8000 compounds from the Aldrich Chemical Company's catalog plus some common compounds not sold by that company. It is apparent from the vitamin A synthesis that the effectiveness of a multistep program increases with its knowledge about the availability of compounds.

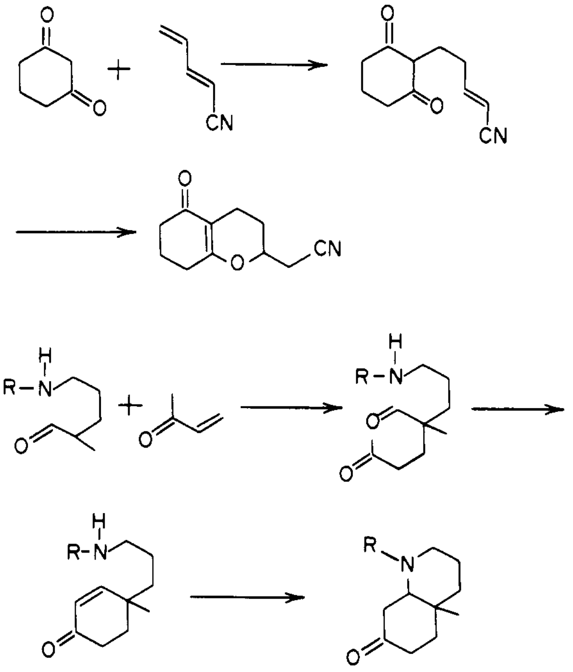
The first multistep synthesis program to be reported on[5] is similar in major respects to the Gelernter program. Pathways of too high a cost were actually erased. Also the program dealt explicitly with reaction yields and overall yields. The program had the repertory only of elementary organic chemistry; hence it only solved problems of that level. A sample-solved problem is shown in Scheme III. The program also tried to synthesize the ketonitrile II by the base-catalyzed reaction of acetonitrile and the unsaturated ketone I. This was judged to be a lower yield reaction than the addition of acrylonitrile to the enamine of cyclohexanone.

The question of cost of running the program was examined, and the conclusion was reached that while the program when equipped with a full repertory of reactions could in principle produce the best syntheses for anything, in practice it would be too expensive to turn it loose on a complex program, using its algorithm, backward search with pruning, and only one simplifying heuristic. It is likely that, with present hardware, no program written in a higher level language can generate the plausible part of a five- or six-level synthetic graph in an economically acceptable time if the molecule has several functional groups and chiral centers. The program required about half a second to generate a subgoal molecule, check its availability, calculate the overall yield from it to the goal molecule, examine it for rings and functional groups, etc.

SCHEME IV



SCHEME V

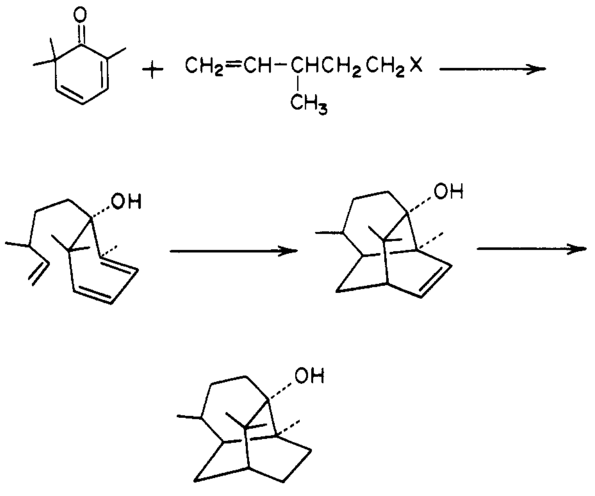


VIII. Chemical Achievements of the Interactive Synthesis Programs

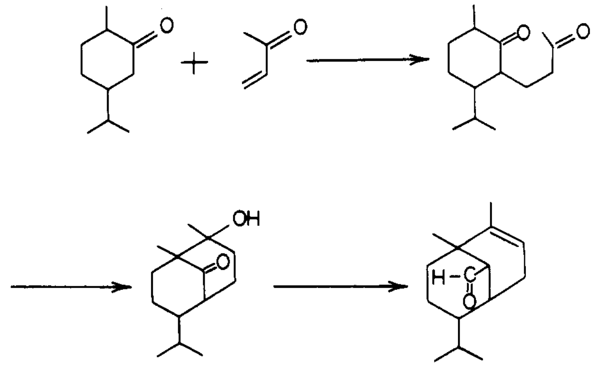
Some of the elegant results of the Corey synthesis program are given in Schemes IV–VII.[3,15] It must be emphasized that, while a chemist made every decision, it was the program that suggested each possible intermediate. These examples illustrate the wide knowledge of chemical reactions now possessed by Corey's program.

Barone, Chanon, and Metzger[16] have developed another interactive program. Their program operates in a manner similar to Corey's and is currently knowledgeable about heterocyclic chemistry. A synthesis of luciferin proposed by their program is reproduced in Scheme VIII. These one-step programs and the multistep programs have in common a poor understanding of stereochemical alternatives. This is a remediable deficiency. The virtue of the interactive systems here is that, unlike in the case of multistep synthesis programs, the chemist is free to use his knowledge of stereochemistry to direct the search for good syntheses.

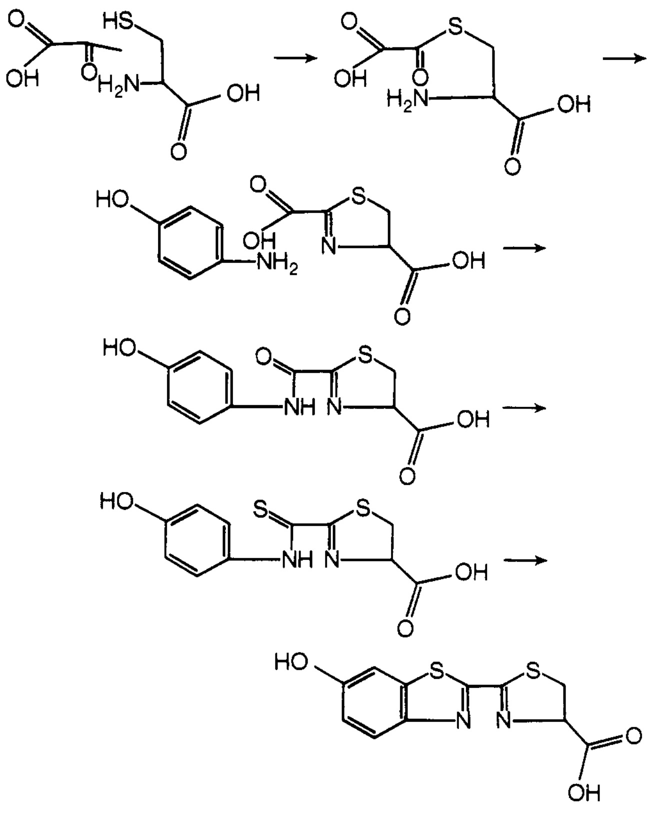
SCHEME VI



SCHEME VII



SCHEME VIII. Luciferin Synthesis



These examples are broad enough to validate the conclusion that these programs can be taught any reaction and reinforces the expectation that the computer-assisted approach will have a major effect on synthetic design.

IX. Implementation Details

A. Programming Languages for Synthetic Chemistry Problems

The ideal programming language for this purpose should (1) be universal, i.e., programs written in this language can run on any large computer in common use; (2) be able to manipulate directly the smallest units of information, i.e., binary digits; (3) make maximum use of the fast instructions of the particular machine on which the program runs; and (4) be easy to program in.

Fortran, in which the interactive programs are written, scores high on points 1 and 4 and fails on points 2 and 3. PL/1, in which Gelernter's multistep synthesis program is written, scores high on points 1, 2, and 4. So also does Algol. Assembly languages fail on points 1 and 4, do well on point 2, and can score highest on point 3. It is apparent that there is no ideal programming language available. Algol and PL/1 are inherently more flexible and versatile than Fortran so they would seem to be the best choice for new investigators entering the field, particularly in writing a multistep synthesis program. A practical production program can always be converted to an assembly language program, which can be most efficient in execution, if well written. This conversion can be done gradually, in stages. Hybrid programs, i.e., part Algol and part assembler, are quite feasible.

B. Representation of Molecular Structure in a Computer

The problem of how to represent three-dimensional molecular structures in the one-dimensional memory of a computer had been solved in a variety of ways before there were any synthetic chemistry programs written. This subject has been well reviewed in the book by Lynch et al.[17] and in certain other compendia.[18] We will not review this topic again here; we will examine only the particular representations of molecular structure used in the various reported synthesis programs.

Gelernter's program is unique in that molecular structure is represented in two different forms.[6,14] The program uses the Wiswesser Line Notation[19] for checking the availability of a compound, because its shelf library (available list) is in Wiswesser notation and is so provided by the Aldrich Chemical Company. For internal use, in manipulating molecular structure the program uses a modified connection table. Briefly, a connection table is a set of rows, one for each nonhydrogenic atom of the molecule. Each such atom in the molecule is numbered and the numbers in certain columns of the ith row indicate which atoms are bonded to the ith atoms. Redundant connection tables show a bond between atoms i and j twice, by putting the number of atom j in the ith row and the number of atom i in the jth row. Nonredundant connection tables show the bond only once. Gelernter refers to his rows as "nodes". Gelernter's connection table differs from the usual type in that saturated hydrocarbon chains, carbonyl groups, dioxo groups, e.g., the dioxo group of a nitro group, and, in general, any group of atoms which is treated as a unit by the Wiswesser notation, are all stored in single rows. These groups might be thought of as "superatoms". Benzene is treated as a single such superatom, with an effective valence of six. The rule for the order of numbering of the nonhydrogenic atoms is that they must follow the order of the Wiswesser notation, reading from left to right.

Other data about atoms can be stored in each row of the connection table. In particular, it is convenient to put in each row the number of hydrogen atoms connected to the atom in question. Gelernter's program also puts into each row the number of unshared electrons available as well as the multiplicity of the bond to each atom not in the row.

The other multistep program[5] puts the atoms in a table in which the multiplicity of bonding is indicated, as well as the number of attached hydrogen atoms, the element name, and the charge or chirality, if any.

In the interactive program, of Corey and Wipke,[3,15] the atoms are numbered in order of their insertion by the chemist or attachment by the program. An important aspect of their representation of molecular structure is that both atoms and bonds are separately represented. In other words, there is both an atom-atom connection table and a bond table. The bond table specifies the multiplicity of the bond and the particular atoms connected by the bond, and also has room to describe the stereochemistry about that bond. The program of Barone, Chanon, and Metzger[16] is capable of describing ionic intermediates and radicals as well. A unique feature of Corey's representation of molecular structure is that only the connection tables of the goal molecule being examined are stored as such. The other structures are stored as lists of changes to be made in deriving the connection table from that of its successor in the "synthesis tree". The latter is a useful device for saving valuable storage space.

C. Representation of Chemical Reactions in a Computer

Subroutines for making and breaking bonds have been described.[4,5] In addition, one must have conditions for avoiding the reaction. In Corey's interactive program, these conditions are stated in semi-English, e.g., "kill if halide".[15] In Gelernter's program, the conditions for avoiding the reaction are given via a bit string. A bit string is a string of yes or no devices. If the ith device says "yes", that means that functional group i rules out the reaction. The former method is more convenient for the chemist; the latter method is more efficient for saving computer running time.

One may wish to require for some reactions that they may not be performed prior to a certain reverse reaction.[5] For example, if an ester has been generated as the reactant of a hydrolysis reaction that produces a carboxylic acid, then we cannot use the esterification reaction to produce this ester. Such "loops" would in any case be deleted eventually on the grounds of their low yield, but it is simpler not to generate them.

Yields are given as such, in per cent,[5] or as figures of merit[6,14] or as ratings.[15] These yields or ratings are subject to variation, of course, by the conditions defined in the reaction description.

D. Recognition of the Functional Groups and Rings

Functional groups can in principle be detected in three ways: (1) by a knowledge driven process, in which we pay no attention to the particular molecule except to ask it questions, "are you a nitrile?" "where, if at all, do you have a carboalkoxy group?", etc.; (2) by a data driven process, in which the search around the molecule automatically produces the functional groups without questions or conditional statements in the program; and (3) by some combination of examination of the data of the molecular structure assisted by some knowledge of organic chemistry. The first way is inept. The reported programs all use some version of the third way. In other words, for example, we examine a molecule for a ketone group only after an unsaturated oxygen atom is found. Corey was the first to elaborate this in detail.[15]

A variety of algorithms for ring recognition are available.[20,21] A major question is the definition, for the program, of what constitutes a synthetically relevant ring. Rings of more than six atoms which are the envelope of smaller rings are not relevant.[20] Ring recognition algorithms should find all the synthetically relevant rings and no other rings.

E. Evaluation of Steric Effects

Wipke[22] has made first excursion into the development of a general and automatic procedure for quantitative evaluation of the effects of stereochemical features on the direction of a reaction. A congestion function is defined, and a three-dimensional model describing the steric environment of one side of a given substructure is constructed and used to assign a numerical value to the function. The side of the substructure which presents the clearer path of approach to the reagent is the one with the smaller value for its congestion function. Using this model, predictions of the direction (exo as opposed to endo) of reduction of some sterically hindered ketones correlated well with the experimentally observed stereoselectivity. Work is currently under way to expand applicability of this procedure by taking into account reagent size and torsional, inductive, and transition-state conformational effects.

X. Ugi's Work on the Special Problem of Optimal Chain Synthesis

The problem of building a linear chain of small units, e.g., a polypeptide, is somewhat special. In this case, we need not be concerned with the internal structure of each small molecule which is incorporated in the chain, provided that the substructures in each such molecule are insensitive to the reactions that link up the units or are protected in a standard manner. For each amino acid and each of the possible roles that the amino acid could play in a standard condensation reaction, there is a definite yield. The choice of which standard reactions to use to link the units is made in advance before the program is executed. The only question for the computer program to handle is the order in which to link the units.

If, for example, we wish to build the chain ABCD, we can do this in the following ways:

1. A + B $\rightarrow$ AB; AB + C $\rightarrow$ ABC; ABC + D $\rightarrow$ ABCD
2. A + B $\rightarrow$ AB; C + D $\rightarrow$ CD; AB + CD $\rightarrow$ ABCD
3. B + C $\rightarrow$ BC; A + BC $\rightarrow$ ABC; ABC + D $\rightarrow$ ABCD
4. B + C $\rightarrow$ BC; BC + D $\rightarrow$ BCD; A + BCD $\rightarrow$ ABCD
5. C + D $\rightarrow$ CD; B + CD $\rightarrow$ BCD; A + BCD $\rightarrow$ ABCD

Similarly, there are 14 different ways to make the chain ABCDE and for a chain of many units the possibilities naturally expand considerably. Each of the possible routes has a different cost. Backward search with cost pruning was elegantly applied by Ugi[23] in his work on the possible syntheses of polypeptides. He calculated that 139 000 000 different routes via peptide linking exist for the A chain of insulin. His program necessarily eliminated most of these after only generating their last few steps, on the grounds of inadequate yield. One synthesis was found to be better than all the others. This conclusion should be reasonable because the yield estimates of the program are those of reactions which are essentially independent of the number of units in the chain. We have only to answer questions such as which two amino acids are involved, which one is at which end of its chain, etc. It should be noted that the particular condensation method considered was Ugi's four-component condensation method. Other condensation procedures would have different optimal sequences for assembling the chain.

XI. Hendrickson's Approach to a Systematic Analysis of Synthetic Decisions

As a preliminary to building a synthesis program, Hendrickson[24] has produced a systematic analysis of synthetic reactions which gives a new view of synthetic organic chemistry. In this view, mechanism is not present; one is concerned only with the net changes involved in synthetic reactions. The conventional basic concepts such as the functional group (e.g., aldehyde) and type reactions (e.g., oxidation to carboxylic acids) are secondary or derived concepts in Hendrickson's scheme. His basic entities are four different types of bonds to carbon atoms. Classes of chemical reactions are defined by the alterations which they produce in the numbers of the four different types of bonds at various carbon sites. The four different bond types are (1) δ bonds to other carbon atoms (the symbol for such a bond is R); (2) bonds to heteroatoms of greater electronegativity than carbon, such as N, O, P, S and halogens (these bonds are indicated by the symbol Z); (3) bonds to hydrogen or to other elements of lesser electronegativity than carbon, such as boron or metals (these bonds are indicated by the symbol H); (4) π bonds to other carbon atoms are indicated by the symbol Π. The symbol F, for functionality, means either Π or Z.

A chemical transformation at a carbon site always involves replacement of a Z, R, H, or π bond by another Z, R, H or π bond. Evidently there are 16 types of reaction at one carbon site. These types are listed in Table I, given by Hendrickson.

The symbol for the reaction at the carbon atom states which kind of bond replaces which kind of bond. Thus, RΠ means that at a certain carbon atom a σ bond to some carbon atom replaces a π bond to (some other) carbon. This occurs, for example, at the β carbon atom in Michael addition. At the adjacent α carbon atom the process that occurs is HΠ, a bond to hydrogen replacing the π bond. All reactions which are describable using the symbols R or Π must occur at at least two carbon sites since these bonds are between two carbon atoms.

Carbon atoms can be characterized by a certain value of an integer quantity, σ, whose possible values range from 0 to 4; 6 specifies the number of carbon to carbon 6 bonds incided to the carbon atom in question. A value of 4 means the atom is a quaternary atom, 3 means a tertiary atom, etc. Each of the values of σ denotes a *skeletal class* of carbon atom. Reactions at a carbon atom that affect its skeletal class are σ_{34}, σ_{23}, σ_{12}, and σ_{01} for constructive reactions and σ_{43}, σ_{32}, σ_{21}, and σ_{10} for cleavage reactions. The first digit of the subscript of σ_{ij} indicates the skeletal class of the atom in the reactant. The second digit shows the skeletal class of the same atom when it is part of the product. Carbon atoms can further be usefully classified according to the number of their π bonds to carbon, i.e., their Π class and the number of their Z bonds. To gain some insight with less complexity we can obliterate the distinction between Z and Π and simply consider the F classes or functionality levels. An integer number *f* can be defined which again runs from 0 (for alkane carbon atoms) to 4 (for CCl_4). The functionality level 3 is for nitriles and carboxyl derivatives. Ketonic, aldehydic, and acetylenic carbon atoms have $f = 2$. The functionality level $f = 1$ is for the appropriate carbon atoms of olefins, alcohols, ethers, and monohalides. The values of *f* then define fairly readily interconvertible classes. Analogously Hendrickson defines a quantity *h*, which is the number of H bonds to the atom in question.

Using the two characteristics, σ, the skeletal class, and *f*,

TABLE I. Types of Reaction at One Carbon Site

	Symbol	$\Delta\sigma$	Δz	$\Delta\pi$	Δh	Δc	Δx
I. Substitution							
Proton exchange	HH	0	0	0	0	0	0
Carbon interchange	RR	0	0	0	0	0	0
π rearrangement	ΠΠ	0	0	0	0	0	0
Nucleophilic substitution	ZZ	0	0	0	0	0	0
II. Oxidation-reduction							
Oxidation	ZH	0	+1	0	−1	+1	+2
Reduction	HZ	0	−1	0	+1	−1	−2
III. Construction-cleavage							
Oxidative construction	RH	+1	0	0	−1	+10	+1
Reductive cleavage	HR	−1	0	0	+1	−10	−1
Reductive construction	RZ	+1	−1	0	0	+9	−1
Oxidative cleavage	ZR	−1	+1	0	0	−9	+1
Constructive addition	RΠ	+1	0	−1	0	+9	0
Fragmentation	ΠR	−1	0	+1	0	−9	0
IV. Elimination-addition							
Oxidative elimination	ΠH	0	0	+1	−1	+1	+1
Reductive addition	HΠ	0	0	−1	+1	−1	−1
Reductive elimination	ΠZ	0	−1	+1	0	0	−1
Oxidative addition	ZΠ	0	+1	−1	0	0	+1

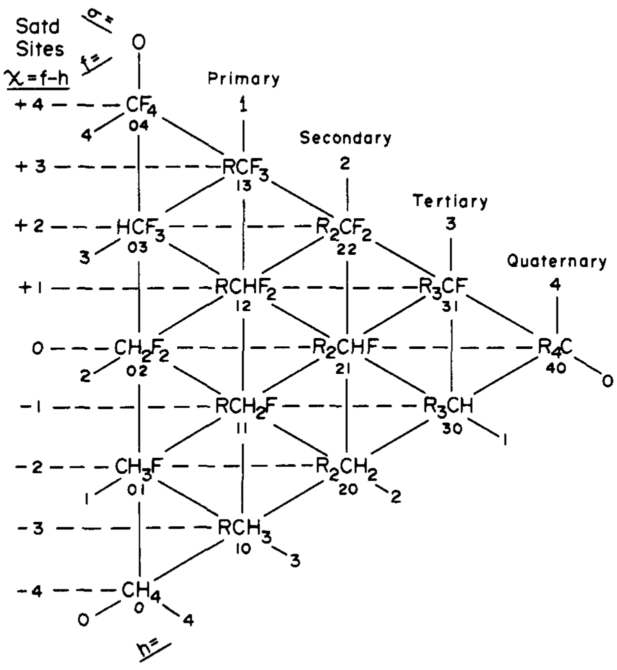


Figure 5. The possible interconversions of the various types of carbon atoms.

the functionality, one can distinguish 15 types of carbon atom. Not all of these can be interconverted in one step. One can categorize all synthetic reactions according to the number of carbon sites involved and their types before and after the reaction. In Figure 5, from Hendrickson, each vertex represents 1 of the 15 possible types of carbon atoms. Each solid line between two vertices indicates a chemical transformation that converts one type of carbon atom into the other. The line represents the effect of the forward reaction or the backward reaction, depending on the direction. A total of 30 links connect these 15 types (Figure 5). They represent the effects of 30 kinds of forward reactions and 30 kinds of backward reactions. In addition there are the FF kind of reactions which leave the type of the atom unchanged. Examples are the oxidation of amines to *N*-oxides or displacement of I by OH. Since there are 10 types of carbon atoms seen in Figure 5 with at least one F bond, it follows that we must distinguish 10 additional basic reaction effects, making a total of 70 "interconversion modes".

In the chart each site is labeled by a number called the *character* of the site, *C*. $C = 10\sigma + f$. This means that the first digit of *C* is the skeletal class and the second is the functionality level. Any transition from the leftmost vertical row to the next vertical row is a σ_{01} reaction. The reader can count eight lines representing different modes of σ_{01} transformations. In contrast there are only two lines representing σ_{34} reactions. One of them is f_{10}, a reduction, and the other is f_{00}, neither an oxidation nor a reduction. Hendrickson points out that a synthetic principle is displayed here: avenues of synthesis of quaternary sites are the most limited and hence in solving a synthetic problem we should give special attention to these sites.

The 70 interconversion modes referred to above can be grouped into seven reaction types as shown in Table II.

When two sites are involved there are 12 possible reaction types as shown in Table III.

Each reaction type can be subdivided into classes according to the values of f_{ij} and σ_{ij} describing the reaction at each site. There are exactly 231 nontrivial categories of reactions at two sites. Hendrickson points out that some of these categories of reactions have not yet been invented. Examples of reactions belonging to these vacant categories are RCHO + R′CHO → RCOCOR′ belonging to the $f_{22}\sigma_{12}\cdot f_{22}\sigma_{12}$ category and $CO_2 + R_2C{=}O + H_2 \rightarrow R_2C(OH)CO_2H$ belonging to the $\sigma_{23}f_{21}\cdot\sigma_{01}f_{43}$ category.

If, in a synthetic sequence, the goal molecule has a carbon atom which has character C_0, then the character of this same atom in the next to the last stage of the synthesis has the value C_1. In the stage before this, it has the character C_2, etc. *If we consider all possible syntheses in which the site is* altered *n* times, then we have a map of the history of transformations at the site. Hendrickson's rules for obtaining all possible predecessors C_{i+1} of C_i are:

1. if $\sigma_i + f_i < 4$, then $\Delta f = 1$ or $\Delta\sigma = 1$ are possibilities
2. if $f_i > 0$, then $\Delta f = -1$ and $\Delta\sigma = 0$ or 1 are possibilities
3. if $\sigma_i > 0$, then $\Delta\sigma = -1$ and $\Delta f = 0$ or 1 are possibilities

If C_i satisfies one, two, or all three of the conditions, there are respectively two, four, and six possible values of C_{i+1}. For each of these values, we can obtain the possible values of C_{i+2} by using the same rules.

Let us apply these rules to a quaternary site $C_0 = 40$. Only condition 3 is satisfied; hence there are only two possibilities for C_1, i.e., 30 ($\Delta f = 0$) or 31 ($\Delta f = 1$). Applying the rules to

TABLE II. Classification of Reaction Types at Single Carbon Sites

Forward reaction types	*f* classes[a]			Reverse reaction types
Substitution ($\Delta c = 0$)	f_{44}	$CO_2 \rightleftarrows CZ_4$		
(FF)	f_{33}	$COOH \rightleftarrows CZ_3$		Same
	f_{22}	$C{=}O \rightleftarrows CZ_2$		
	f_{11}	$COH \rightleftarrows CZ$		
Reduction ($\Delta\sigma = 0$)	f_{43}	$CO_2 \rightleftarrows HCOOH$	f_{34}	Oxidation ($\Delta\sigma = 0$)
(HF)	f_{32}	$COOH \rightleftarrows CHO$	f_{23}	(FH)
$\Delta c = -1$	f_{21}	$C{=}O \rightleftarrows CHZ$	f_{12}	$\Delta c = +1$
($\Delta f = -1$)	f_{10}	$CZ \rightleftarrows CH$	f_{01}	($\Delta f = +1$)
Reductive construction ($\Delta h = 0$)	f_{43}	$CO_2 \rightleftarrows RCOOH$	f_{34}	Oxidative cleavage ($\Delta h = 0$)
(RF)	f_{32}	$COOH \rightleftarrows RC{=}O$	f_{23}	(FR)
$\Delta c = +9$	f_{21}	$C{=}O \rightleftarrows RCZ$	f_{12}	$\Delta c = -9$
($\Delta f = -1$)	f_{10}	$CZ \rightleftarrows CR$	f_{01}	($\Delta f = +1$)
Oxidative construction ($\Delta f = 0$)	f_{33}	$HCN \rightleftarrows R{-}CN$	f_{33}	Reductive cleavage ($\Delta f = 0$)
(RH)	f_{22}	$CHO \rightleftarrows RC{=}O$	f_{22}	(HR)
$\Delta c = +10$	f_{11}	$CHZ \rightleftarrows RCZ$	f_{11}	$\Delta c = -10$
	f_{00}	$CH \rightleftarrows CR$	f_{00}	

[a] Only *f* classes are listed and the sample generalizations shown exemplify common conversions (usually with oxygen groups), but any heteroatom shown may be replaced by another without change in the *f* class or type (e.g., $f_{33} \equiv$ HCN, HCOOR, etc). Unlabeled bonds are to R or H, but not Z. σ classes for each type: substitution or oxidation-reduction: σ_{33}, σ_{22}, σ_{11}, σ_{00} (σ unchanged); construction: σ_{34}, σ_{23}, σ_{12}, σ_{01}; cleavage: σ_{43}, σ_{32}, σ_{21}, σ_{10} ($\Delta\sigma = \pm1$).

TABLE III. Possible Reaction Types at Two Carbon Sites Only

	Construction (+R)	Cleavage (−R)
Oxidative (−H or +Z)	RH·RH	ZR·ZR
Isohypsic	RH·RZ	ZR·HR
Reductive (+H or −Z)	RZ·RZ	HR·HR
	Elimination (+Π)	Addition (−Π)
Oxidative (−H or +Z)	ΠH·ΠH	ZΠ·ZΠ
Isohypsic	ΠH·ΠZ	ZΠ·HΠ
Reductive (+H or −Z)	ΠZ·ΠZ	HΠ·ΠH

30 and 31, we obtain 20, 21, 22, 30, 31 as the five possible nontrivial values for C_2. Building a graph of possible modifications of the character of each atom of the goal molecule can give us an exhaustive list of the sequence of reaction types required for syntheses of the molecule. "Selection criteria" are needed to prune the graph of possibilities. In addition, Hendrickson's approach can improve existing syntheses by detecting wasted motion in which a change in skeletal class and functional level was attained by too circuitous a route.

Hendrickson has also produced a systematic procedure for finding all possible viable routes to polysubstituted benzenes from disubstituted benzenes.[25]

A map of such a problem is shown in Figure 6.

Hendrickson employs these simplifying principles.

1. The introduction of a blocking or activating substituent which is subsequently replaced by hydrogen should be done at most once in a synthesis.

2. All substituents can be grouped into 9 classes, i.e., O, N, S, C (saturated carbon), H, X, C_0 (unsaturated carbon), S_0 (positively charged sulfur, as in $-SO_2$), and N_0 (positively charged nitrogen). Tables of feasibility of substitutions can be drawn up in terms of these nine possible substituent classes instead of the various possible individual substituents, a notable simplification.

3. There is a dominant orienting influence in every substituted benzene ring. The analysis of feasibility can be done in terms of this dominant influence. The dominant influence can often be taken simply as the orienting direction of the substituent which lies first on the list O, N, S, C, H, X, C_0, S_0, N_0.

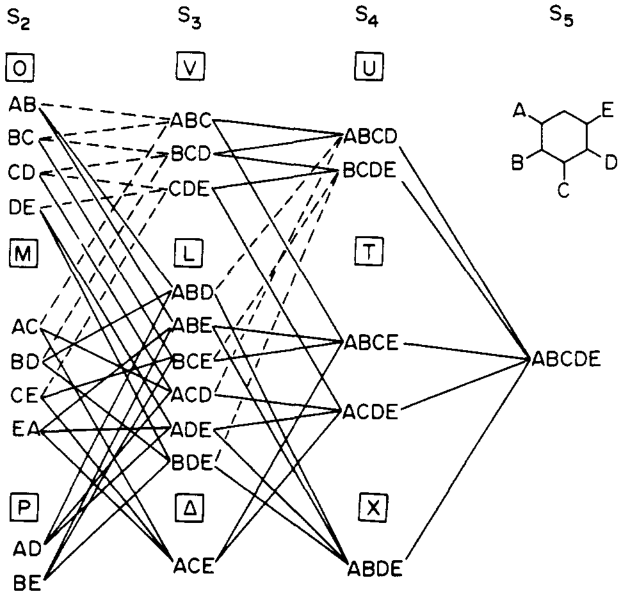


Figure 6. Graph of direct routes to S_5.

Hendrickson has also introduced a simple notation for a substituted benzene type. For example, $CHHHN_0HH$ represents molecules such as 4-nitrotoluene or 4-trimethylammonium 1-ethylbenzene. The string XC_0HHHH represents molecules like 2-iodoacetophenone.

XII. Wipke's Computations of Stereochemical Changes

The first program to take account of stereochemistry in simulating chemical reactions is that of Wipke's group.[26] Wipke's representation of stereochemistry is built around his concept of a *stereocenter*. Stereocenters are atoms whose substituents are arranged in such a way that the interchange of certain pairs of substituents produces a different isomer. Examples are: (1) those doubly bonded atoms which are bonded to two nonequivalent substituents which are doubly bonded to atoms that have two nonequivalent substituents, (2) chiral atoms, (3) the most central atom of allenes and higher cumulenes which have four nonequivalent terminal substituents, and (4) equivalent ring junction atoms such as those of *cis*- or *trans*-decalin.

In case 1, interchange of nonidentical substituents results in cis–trans isomerization. In cases 2 and 3, interchange of any two substituents changes the chirality. In case 4 interchange of any substituents at one of the atoms is equivalent to cis–trans isomerization.

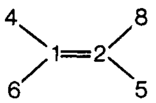
A noninteractive program, trying to synthesize a chiral molecule without proper stereochemistry handling routines, soon begins to make proliferating errors. An interactive program such as that of Wipke could in principle be corrected by the chemist as he chooses the stereochemically proper isomer(s) from among the output of each reaction. However, it saves the chemist considerable time in using the program if the program itself "knows" about stereochemistry and always generates the correct isomers.

Wipke has written input–output programs such that the familiar wedged and dotted lines can be entered by the chemist on a cathode ray tube terminal, translated to a machine representation, and as desired retranslated back to a wedged and dotted line diagram for display to the chemist.

Petrarca, Lynch, and Rush[27] first showed how to represent chirality unambiguously by a linear string. Thus if the substituents of a chiral atom are listed as W X Y Z and this is *R* chirality, then W Y X Z represents the *S* chiral arrangement. But, one may ask, since any interchange of two substituents which are adjacent in the linear list inverts the chirality, X W Y Z and W X Z Y must also represent *S* chirality of these four ligands, how are we to know on examination, without prior information, whether, for example, Z W X Y represents *R* or *S* chirality? The answer lies in having sequence rules for canonically numbering the atoms. If W X Y Z is the canonical order, i.e., W outranks X which outranks Y, etc., then if we agree that W X Y Z has *R* chirality, all arrangements obtained from W X Y Z by even permutations represent *R* chirality also. All arrangements obtained by W X Y Z by an odd permutation represent, then, *S* chirality. Even and odd permutations are equivalent, respectively, to an even or an odd number of interchanges of substituents adjacent in the list. Since one inversion produces opposite chirality, two successive inversions produce no net effect on chirality, three inversions produce a chirality opposite to the original one, etc. Hence, on the above assumption, i.e., that W X Y Z is *R* chirality, then we confidently state that Z W X Y represents *S* chirality.

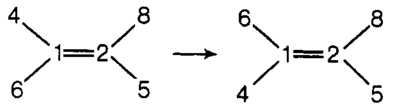
In Wipke's representation of chiral stereocenters, the neighbors of each chiral atom are presented in an ordered list, arranged so that the first three neighbors are clockwise when viewed from the side opposite the fourth neighbor. Wipke observes that this is the same as the clockwise arrangement of the *last* three neighboring atoms when viewed along the direction from the first neighboring atom toward the chiral atom. This means that, for example, if the neighboring atoms are listed as W X Y Z and the chemical reaction involves a replacement of Q by Y with retention of configuration, then the reactant must have the configuration W X Q Z. If there was replacement of Y by Q with inversion of configuration, then the reactant ligands must have the configuration W Q X Z or some other listing obtainable from W X Q Z by an odd number of exchanges of adjacent ligands.

The pair of stereocenters which describe the configuration of a double bond is represented as in the following example.



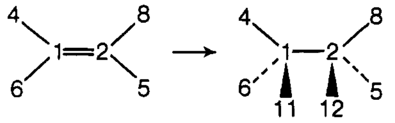
$$\begin{array}{l}1\!:\ 2\ 6\ 4\\2\!:\ 1\ 8\ 5\end{array}$$

The rule is that if the neighbors of atom 1 are listed in clockwise order when viewed from a certain direction, then the neighbors of atom 2 must also be listed in an order that is clockwise when viewed from the same direction. Cis–trans isomerization of the double bond is represented as follows:



$$\begin{array}{l}1\!:\ 2\ 6\ 4\\2\!:\ 1\ 8\ 5\end{array}\longrightarrow\begin{array}{l}1\!:\ 2\ 4\ 6\\2\!:\ 1\ 8\ 5\end{array}$$

Cis addition is indicated as follows:



$$\begin{array}{l}1\!:\ 2\ 6\ 4\\2\!:\ 1\ 8\ 5\end{array}\longrightarrow\begin{array}{l}1\!:\ 11\ 2\ 6\ 4\\2\!:\ 12\ 1\ 8\ 5\end{array}$$

If the cis addition is on the side of the plane of the double bond where the substituents 2 6 4 appear to be counterclockwise, then the new atoms are placed at the end of the list, i.e.

$$\begin{array}{l}1\!:\ \ 2\ 6\ 4\\2\!:\ \ 1\ 8\ 5\end{array}\longrightarrow\begin{array}{l}1\!:\ \ 2\ 6\ 4\ 11\\2\!:\ \ 1\ 8\ 5\ 12\end{array}$$

The reader will note that using these ingenious rules the chiral configuration of the resulting stereocenter is automatically correctly represented in all the cases where addition to the double bond is only possible from one side. Trans addition is represented by a cis addition followed by the inversion of a particular one (not either one) of the resulting stereocenters.

XIII. A Projected Use of Ensemble Matrices in the Computer Generation of Synthetic Pathways

Ugi and his group[28] have outlined a scheme for a noninteractive, multistep synthesis program. In this scheme, molecules are represented as square matrices. If there are *n* atoms being considered, the matrix is $n \times n$. The atoms to be considered are those of the goal molecule, plus a number of common byproducts such as NH_3, H_2, O_2, CO_2, H_2O, NaCl, $(C_6H_5)_3PO$, etc. The ensemble of these molecules is successively converted by bond-breaking and bond-making processes from available starting materials to an ensemble in which the goal molecule is present as a complete entity along with other, less important molecules.

In the "ensemble matrix" representing an ensemble of molecules the *ij* and *ji*th elements are equal to 0, 1, 2, or 3 depending on whether atoms *i* and *j* are not bonded to each other, singly bonded, doubly bonded, or triply bonded to each other, respectively. By adding a reaction matrix, also $n \times n$, to the ensemble matrix, we convert it to a new ensemble matrix describing chemically transformed molecules. If the *ij*th and *ji*th elements of the reaction matrix are -1, then adding the reaction matrix has the effect of breaking a bond between atoms *i* and *j*. Similarly, the corresponding addition of $+1$ has the effect of making a bond between atoms *i* and *j*.

The suitable reaction matrices can be generated from any ensemble matrix by considering which bonds can reasonably be broken, e.g., multiple bonds, bonds to heteroatoms, and bonds one or two atoms away from these. Since as many bonds must be made as are broken and we cannot make bonds to saturated atoms, then it develops that there are only a manageably finite number of suitable reaction matrices which can be generated from each ensemble.

Ugi's group proposes to generate the intermediate molecules in a breadth first fashion. This means (1) generate the set of reactants $\{F_i\}$ which can produce the goal molecule G; (2) generate the set of reactants, $\{E_i\}$, which can produce the members of the set $\{F_i\}$; (3) continue in this manner until a certain number of available molecule starting points have been found.

The projected program of Ugi's group is set apart from all of the other synthesis programs being developed by its use of the connection *matrix* of an ensemble of molecules as the basic entity to be transformed. The other programs all manipulate some form of connection *table* of individual molecules.

XIV. Some Unresolved Problems

There is a need to determine the limits of effectiveness for practical use of the algorithm of backward search with cost pruning and without heuristics. To restate the question, to what degree can the exhaustive consideration of every possibility that meets the cost (yield) requirements remove the need for evaluating the promise of incomplete paths? We are referring to practical problems. This algorithm must surely fail if the shortest possible synthesis of a goal molecule really requires, let us say, 20 steps. But, in any event, 20-step syntheses have no practical, i.e., economic, use. Let us anticipate the likely conclusion that there is a wide range of practical problems which are too difficult to be solved by such a brute force method as the investigation of every possibility on the synthetic graph which is not pruned by cost considerations. They are too difficult because there are too many possibilities. If this is the conclusion, then the crucial task becomes the development of problem-simplifying heuristics. Corey has made the major beginning advances in this area. Another way of stating this problem is that one needs to find the rules by which expert synthetic chemists make their exclusion decisions and incorporate them into the evaluation function, the part of the program that evaluates the promise of incomplete paths. We are not referring to the special knowledge that a chemist might have about details which would make a standard reaction give poor results for a given molecule. This sort of knowledge about what functional groups decrease the yield of a reaction, how sensitive the reactions are to steric hindrance, etc., can be built into the description of each reaction in the reaction library. We are referring to strategic decisions of a higher level, e.g., which part of the molecule to build last, etc.

An important problem for the development of the noninteractive programs is efficiency. Accelerations of 100 times or more are possible if the right fast procedures are used to examine a molecule, find its objects of synthetic interest, store the molecular structure, search the list of available molecules, etc. Problems of efficiency might rightly be dismissed as engineering details were it not for the cost problem which overhangs the noninteractive programs. The cost reduction of the running of any key part of such programs must be elevated to the status of a high priority subject of investigation.

Brilliant feats of stereoselective syntheses at present require the imagination of the great synthetic chemists. A trend has begun for this "imagination" to be dissected, analyzed, and made automatic. A flourishing science of the algorithms and heuristics of optimal synthesis design can be expected to arise, from the foundation laid by Corey.

XV. Addendum

An extensive discussion of the ring-finding problem has been presented by Esack.[29] The methods described in his paper, and implemented in a noninteractive synthesis program, appear to be the fastest yet devised.

The question of the discovery of functional groups was mentioned in section IX.D. Esack and Bersohn[30] have a subprogram which looks at an input structure and finds its functional groups with almost no logical queries. The usual if-then-else procedures are replaced by extensive table look up. The program describes sites where there are heteroatoms and/or unsaturation in terms of the atomic numbers of the atoms concerned and their immediate neighbors, the number of hydrogen atoms bonded to them, and their degree of unsaturation. For the latter, values of 0 are given to saturated atoms: 1 to aromatic atoms, 2 to carbon atoms doubly bonded to carbon atoms, etc. These data result in a number which then characterizes the functional group. This number is used as is or can be converted to an index to reference tables which suggest the relative importance of the group or to retrieve synthetic reactions relevant to this group. In functional groups involving heteroatoms, one atom is designated as the "central atom" and its row in the connection table is labeled with the number of the functional group.

Synthetic experience shows the importance of certain key reactions such as the Birch reduction, the Robinson annulation reaction, the Diels-Alder reaction, and ring formation by carbonium ion addition to a carbon-carbon double bond. Instead of simply having these reactions available, an interactive program can specifically give these an especially high recommendation to the chemist decision maker. This latter feature was present in Corey's earlier program.[31] More recently Corey has advanced this idea still further. His program now, on request from the chemist, actively seeks for the product of a relevant Diels-Alder reaction from which the molecule at hand can be derived. As many as 15 steps may be necessary to convert the Diels-Alder product into the molecule being considered. His program can now handle such complexity.[32] Corey's group is developing packages which exploit the power of various other favored reactions and exhaustively search for ways to obtain the molecule being considered from products of the favored reactions. This notable extension of Corey's program gives it a degree of multistep character and gives the chemist a somewhat greater role at programming time and a correspondingly lesser role at the time the program is being run by the man-computer interactive system.

In the course of a synthesis chiral centers may be destroyed, e.g., by making a double bond. A computer program which, knowing the product, generates the reactant, may not know what chirality to assign to the chiral atoms of the reactant. (Even though it knows which side of the ring the substituents are on, it does not know the definition of the term "clockwise.") Lacking this knowledge it is difficult for the program to determine if the reactant is the same as an available substance or if the reactant has been previously generated by the program. A program can assign the chirality by using another nearby center of known chirality as a "compass" to determine what is meant by clockwise and anticlockwise. The implementation of this solution to the problem as well as other chirality change procedures are discussed in Bersohn and Esack.[33]

XVI. References and Notes

(1) G. E. Vleduts, *Inf. Storage Retr.*, **1**, 101 (1963).

(2) J. Lederberg, G. L. Sutherland, B. G. Buchanan, E. A. Feigenbaum, A. V. Robertson, A. M. Duffield, and C. Djerassi, *J. Am. Chem. Soc.*, **91**, 2973 (1969); A. M. Duffield, A. V. Robertson, C. Djerassi, B. G. Buchanan, G. L. Sutherland, E. A. Feigenbaum, and J. Lederberg, *ibid.*, **91**, 2977 (1969).

(3) E. J. Corey and W. T. Wipke, *Science*, **166**, 178, (1969).

(4) R. Barone, M. Chanon, and J. Metzger, *Rev. Inst. Fr. Pet. Ann. Combust. Liq.*, **28** (5), 771 (1973).

(5) M. Bersohn, *Bull. Jpn. Chem. Soc.*, **45**, 1897 (1972).

(6) H. Gelernter, N. S. Sridharan, A. J. Hart, S. C. Yen, F. Fowler, and H. Shue, *Top. Curr. Chem.*, **41**, 113 (1973).

(7) D. E. Knuth in "Fundamental Algorithms", Addison-Wesley, Reading, Mass., 1969, pp 306-307.

(8) R. Breslow, *Chem. Soc. Rev.*, **1**, 553, (1972).

(9) N. J. Nilsson, "Problem Solving Methods in Artificial Intelligence", McGraw-Hill, New York, N.Y., 1971.

(10) B. Mittman, *Datamation*, 84 (June 1973).

(11) J. R. Slagle, "Artificial Intelligence: The Heuristic Programming Approach", McGraw-Hill, New York, N.Y., 1972.
(12) E. J. Corey, *Q. Rev., Chem. Soc.*, **25,** 455 (1971).
(13) E. J. Corey, *Pure Appl. Chem.*, **14,** 19 (1967).
(14) N. S. Sridharan, Ph.D. Thesis, State University of New York at Stony Brook, 1971; University Microfilms, Ann Arbor, Mich.
(15) E. J. Corey, W. T. Wipke, R. D. Cramer III, and W. J. Howe, *J. Am. Chem. Soc.*, **94,** 421, 431, 440 (1972).
(16) R. Barone, M. Chanon, and J. Metzger, *Rev. Inst. Fr. Pet.*, **5,** 771 (1973).
(17) M. F. Lynch, J. M. Harrison, W. G. Town, and J. E. Ash, "Computer Handling of Chemical Structure Information", Macdonald, London, 1971.
(18) Survey of Chemical Notation Systems, National Academy of Science–National Research Council Publication 1150, 1964; Survey of European Non-Conventional Chemical Notation Systems, National Academy of Sciences–National Research Council Publication 1278, 1965; Chemical Structure Information Handling, National Academy of Sciences–National Research Council Publication 1733, 1969.
(19) E. G. Smith, "The Wiswesser Line-Formula Chemical Notation", McGraw-Hill, New York, N.Y., 1968.
(20) E. J. Corey and G. A. Petersson, *J. Am. Chem. Soc.*, **94,** 460 (1972).
(21) M. Bersohn, *J. Chem. Soc., Perkin Trans. 1*, 1239 (1973).
(22) W. T. Wipke and P. Gund, *J. Am. Chem. Soc.*, **96,** 299 (1974).
(23) I. Ugi, *Rec. Chem. Prog.*, **30,** 289 (1969); *Intra-Sci. Chem. Rep.*, **5,** 229 (1971); G. Gokel, P. Hoffmann, H. Kleimann, H. Klusacek, G. Ludke, D. Marguarding, and I. Ugi in "Isonitrile Chemistry", I. Ugi, Ed., Academic Press, New York, N.Y., 1971, p 201.
(24) J. B. Hendrickson, *J. Am. Chem. Soc.*, **93,** 6847 (1971).
(25) J. B. Hendrickson, *J. Am. Chem. Soc.*, **93,** 6854 (1971).
(26) W. T. Wipke and T. M. Dyott, *J. Am. Chem. Soc.*, **96,** 4825, 4834 (1974).
(27) A. E. Petrarca, M. F. Lynch, and J. E. Rush, *J. Chem. Doc.*, **7,** 154 (1967); J. E. Blackwood, C. L. Gladys, A. E. Petrarca, W. H. Powell, and J. E. Rush, *ibid.*, **8,** 30 (1968); A. E. Petrarca and J. E. Rush, *ibid.*, **9,** 32 (1969).
(28) J. Blair, J. Gasteiger, C. Gillespie, P. D. Gillespie, and I. Ugi, "Computer Representation and Manipulation of Chemical Information", W. T. Wipke, S. R. Heller, R. J. Feldmann, and E. Hyde, Ed., Wiley, New York, N.Y., 1974, Chapter 6.
(29) A. Esack and M. Bersohn, *J. Chem. Soc., Perkin Trans. 1*, 2463 (1974).
(30) A. Esack, *J. Chem. Soc., Perkin Trans. 1*, 1120 (1975).
(31) W. J. Howe, Ph.D. Thesis, Harvard University, 1972.
(32) E. J. Corey, W. J. Howe, and D. A. Pensak, *J. Am. Chem. Soc.*, **96,** 7724 (1974).
(33) M. Bersohn and A. Esack, *J. Chem. Soc., Perkin Trans. 1*, 1124 (1975).