9 September 1977, Volume 197, Number 4308 SCIENCE

Empirical Explorations of SYNCHEM

The methods of artificial intelligence are applied to the problem of organic synthesis route discovery.

H. L. Gelernter, A. F. Sanders, D. L. Larsen, K. K. Agarwal, R. H. Boivie, G. A. Spritzer, J. E. Searleman

Fewer than 8 years have elapsed since the first report was published describing a serious and substantial attempt to exploit the digital computer as a tool for the design of complex organic syntheses (*1*). At that time, while the traditional use of the computer as a calculating machine was taken for granted by chemists everywhere, the suggestion that programs in the offing might be capable of assuming some of the intellectual burdens of the practice of organic chemistry was generally regarded with benign skepticism by the intended beneficiaries of such a development. The intervening years have witnessed a remarkable change in the way most organic chemists perceive the role of the computer in their discipline. Today, skepticism has in large part been replaced by lively interest. Although progress in the area of computer-directed organic synthesis route discovery is not alone responsible for this change in attitude (*2*), the surprising rapidity with which initial exploratory efforts have produced results that promise practical application in the foreseeable future has attracted the attention and, in many instances, the active participation of an increasing number of organic chemists (*3*).

We describe here the current status of one such exploration into the application of the techniques and methodology of artificial intelligence in general, and of heuristic programming in particular, to the problem of organic synthesis route discovery by computer. In progress at the State University of New York at Stony Brook since 1968, this research has produced a computer program called SYNCHEM, which is able to discover multistep routes for the synthesis of nontrivial organic structures without on-line guidance or intercession on the part of the user. Among the target molecules with which the program has dealt more or less successfully are polycyclic bridged structures and relatively complex heterocycles of biochemical interest. For reasons discussed below, work on SYNCHEM has been abandoned in favor of a second version of the program, SYNCHEM2. In addition to a substantial number of other improvements, SYNCHEM2 will deal with stereochemistry, an issue that we elected to sidestep in our earlier efforts. Despite our current disaffection with SYNCHEM, however, it has provided results of considerable interest and usefulness, as well as the foundation for our present approach to the problem of synthesis discovery by computer.

H. L. Gelernter is professor of computer science at the State University of New York, Stony Brook 11794. A. F. Sanders was and D. L. Larsen is a Lederle postdoctoral fellow at Stony Brook. K. K. Agarwal and R. H. Boivie were and G. A. Spritzer and J. E. Searleman are graduate students in computer science at Stony Brook. Dr. Sanders is now assistant professor of computer science at Wright State University, Dayton, Ohio 45431. Dr. Agarwal is now assistant professor of computer science at Wayne State University, Detroit, Michigan 48202. Dr. Boivie is now a member of the research staff at Bell Laboratories, Holmdel, New Jersey 07733.

Artificial Intelligence and Chemistry

It was not quite 20 years ago that the first reports of success in the new field of artificial intelligence reached the scientific literature. The Logic Theorist of A. Newell and H. A. Simon could prove theorems in the Russell propositional calculus without using a decision procedure, Gelernter's geometry theorem proving machine had mastered a substantial part of high-school level Euclidean plane geometry, and A. L. Samuel's checker program and A. Bernstein's chess program were beginning to puzzle and delight serious students of board games (*4*). It seemed as if so much had been accomplished by so few in so little time with relatively rudimentary computing resources. Surely spectacular results awaited the application of a little more effort by a few more people in a somewhat more concentrated attack on the problem, once the amount of high-speed core storage available could be, say, doubled.

None of the pioneers in artificial intelligence were innocent of excessive optimism, although Simon and Newell, perhaps, bore the brunt of the attack from those made wise by hindsight, because their predictions were widely circulated in print. The ensuing years, during which artificial intelligence did not, for the most part, fulfill its earlier promise, brought much controversy and occasional disrepute to the discipline, with the result that growth in research and scientific interest in artificial intelligence has failed to keep pace with the development of computer science in general. The trouble was that not only had we underestimated the recalcitrance of the problems we wished to solve by these new techniques, but we had also overestimated the difficulty and generality of the problems we had solved. Nevertheless, progress has been made, mostly by attacking and clarifying small pieces of the larger problems still beyond reach or by direct extension of some of the earlier work, but occasionally by producing major and important results that vindicate some of the early optimism (*5*).

The work described here has been motivated by our conviction that, while

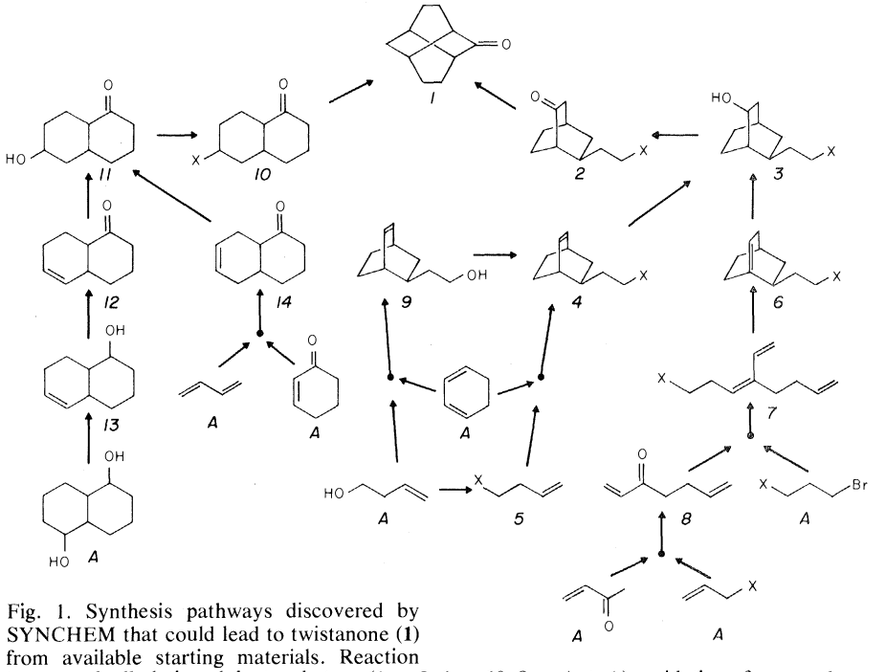

Fig. 1. Synthesis pathways discovered by SYNCHEM that could lead to twistanone (**1**) from available starting materials. Reaction types used: alkylation alpha to a ketone (**1** ← **2**, **1** ← **10**, **8** ← **A** + **A**); oxidation of a secondary alcohol (**2** ← **3**, **12** ← **13**); hydration of an alkene (**3** ← **4**, **3** ← **6**, **11** ← **12**, **13** ←**A**); Diels-Alder reaction (**4** ← **5** + **A**, **6** ← **7**, **9** ← **A** + **A**, **14** ← **A** +**A**); Wittig reaction (**7** ← **8** + **A**); and replacement of an alcohol by a better leaving group (**10** ← **11**, **4** ← **9**, **5** ← **A**). All compounds labeled **A** were found by SYNCHEM on its list of available compounds.

the theoretical underpinnings of artificial intelligence as a scientific discipline are far from satisfactory, the techniques and methodology of heuristic programming have matured to the point where problems that are of substantial interest in themselves rather than mere vehicles for artificial intelligence research ought to be selected in pursuit of further progress. The work of the Stanford group on molecular structure determination from mass spectrometry data (*6*) indicates a growing trend in this direction.

Nine years ago, a small group at Stony Brook began to investigate the feasibility of applying computer-simulated intelligence to the problem of discovering valid and efficient synthesis routes for complex organic chemical structures. Our initial goal was a system that would perform about as well as a first year graduate student of organic chemistry in most respects. In particular, unlike much of the earlier work in problem-solving exemplified by Gelernter's theorem proving program (*7*) or Newell, Simon, and Ernst's general problem solver (*8*), where any formally valid sequence of transformations from premises to goal provided an acceptable solution, we were not to be satisfied by an indicated synthesis route of very low yield, or one requiring difficult or inefficient separations of goal molecules from by-products along the way, at least not before the machine had tried and failed to find a more efficient procedure of higher yield. Nor would the machine be permitted to ignore the technical constraints on a problem. The cost and availability of starting materials and the ease of carrying out a particular step of the synthesis would be weighted in importance according to whether the problem was designated an exploratory laboratory synthesis, an industrial production synthesis, or something between these extremes. It is this question of relative merit of proposed solutions under the constraints of the problem that represents a substantial departure from most of the work reported in the literature of artificial intelligence.

SYNCHEM

Early in 1971, the first version of SYNCHEM began to produce multilevel synthesis-search trees for modest linear organic structures, some of which carried functionality in nontrivial variety. By midyear, the program's domain of competence had expanded to include simple carbocyclic compounds, and during the ensuing 2 years polycyclic structures containing heterocycles became manageable. Among the problems presented to SYNCHEM toward the end of that period were a number of compounds of considerable interest, both intrinsically and because of their relation to biology and medicine. The most exciting and also the most frustrating of the results obtained was the program's proposed synthesis for a naturally occurring antibiotic then under investigation by Rinehart's group at the University of Illinois (*9*). The route discovered by SYNCHEM suggested an approach to the synthesis that was different from those under consideration at the time, one that was deemed sufficiently original and promising to be worth a laboratory investigation of its validity. The procedure in question is discussed later. We must regretfully report here, however, that the synthesis failed to work in the laboratory, hence the frustration mentioned above.

The design of SYNCHEM is described in considerable detail in Gelernter *et al.* (*10*); the following is a brief synopsis of that report. Input to SYNCHEM of the target molecule to be synthesized is most often in the form of Wiswesser linear notation (WLN), although a connection matrix representation, or TSD (for topological structural description), is also accepted by the program. The target compound is analyzed for synthesis-relevant functional groups and structural features ("synthemes"), and some of these are selected for development. Corresponding to each syntheme is a chapter of the reaction library, each chapter comprising an arbitrary number of reaction schemata for the synthesis of that particular syntheme. A syntheme having been selected, the appropriate chapter of the reaction library is brought into the computer. Each schema of the chapter is provided with a set of tests to be performed on the goal molecule. These tests embody many of the chemistry heuristics that guide the program. On the basis of the results of these tests, the program may reject the schema, adjust the ad hoc merit rating for the reaction, modify the reaction procedure, or specify protection procedures for sensitive groups. As an example of merit adjustment, the reaction merit might be raised if a conjugated activating group is present, or lowered if steric hindrance is detected. The specification of a different reagent in the presence of groups sensitive to the usual reagent is an example of procedure modification.

Programmed by the adjusted set of reaction schemata selected for the goal molecule, SYNCHEM generates a set of subgoals for that molecule. For each of these, an ad hoc overall merit is computed, based on both the adjusted reaction merit and an estimate of the complexity of the subgoal molecule. If a synthesis-search terminating condition has not been signaled, the "best" subgoal among all of those generated to that

point in the search is selected for further development, and the procedure is recursively continued. A synthesis-search tree is thereby generated, the structure of which depends on the currently active algorithms for subgoal merit computation and the policies adopted for determining how these ratings will be used to designate the best subgoal for further consideration. A synthesis has been completed when a path has been generated linking the target molecule with the catalog of available compounds.

SYNCHEM's reaction library contained, in varying stages of completeness, chapters for the synthesis of aldehydes, ketones, alcohols, organic acids, esters, halides, acid halides, Grignard and other organometallic reagents, nitriles, the olefin bond, ethers, and a limited selection of structural and multifunctional synthemes. The list of available compounds was a 3000-item subset of the catalog of organic materials obtainable from the Aldrich Chemical Company (*11*), to which several dozen commonly available materials that Aldrich does not list were added (ethylene, for example). The Aldrich catalog was selected because it could be provided on punched cards in WLN representation for direct computer input.

The system was programmed in PL/1 augmented by an extensive library of specially written subroutines to facilitate manipulation of the list-like data structures SYNCHEM uses to represent organic molecules, the search tree, reaction schemata, and much of the program's working storage. We selected PL/1 as the programming language because it provides a number of features designed to make the kind of symbol manipulation that is fundamental to artificial intelligence programming applications relatively convenient—pointer operations, based and offset variables, dynamic storage allocation, and so on. The program ran at Stony Brook on an IBM 370/155, requiring a minimum partition size of 360 kbytes for a multilevel synthesis search.

SYNCHEM's Syntheses

By the time SYNCHEM was retired in 1974, the program had dealt with about a hundred different molecules. Many of these were routine problems designed to check out the behavior of one or another of the complex algorithms and heuristics comprising the program, but among them were also a number of structures of substantial interest in themselves. An early result in the latter category for the molecule twistanone (tricyclo[4.4.0.0] decan-2-one) is discussed in detail in (*10*). The synthesis routes developed by SYNCHEM for twistanone after about 20 minutes of search time are exhibited in Fig. 1. They are of interest because the two major approaches to the problem that may be seen in the search tree, one by way of a decalin structure (**10** in Fig. 1) and the other through a bicyclo[2.2.2]octane (**2**), are essentially similar to those in the published syntheses for the twistane ring system (*12*). Technical weaknesses are evident in many of the individual steps proposed by the program, but it is not difficult to accept the claim that realizable improvements in the synthesis-search algorithm and in the design and scope of the set of chemistry heuristics that guide the search can be expected to provide better pathways.

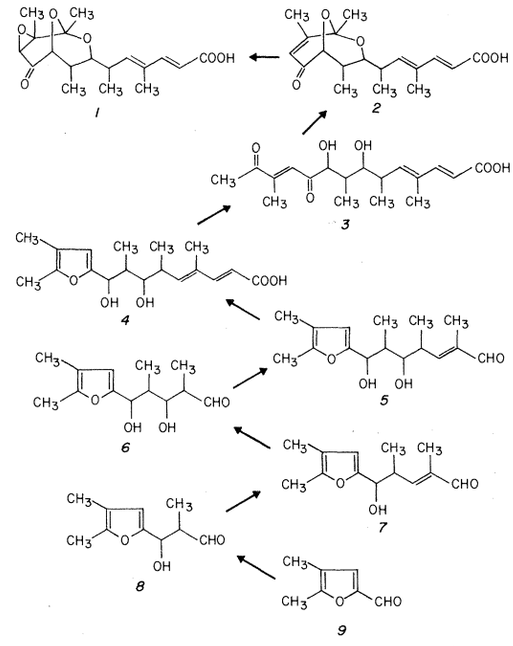

Fig. 2. Pathway proposed by SYNCHEM for tirandamycic acid. Reaction types used: epoxidation (**1** ← **2**); internal ketalization (**2** ← **3**); oxidative cleavage of a furan (**3** ← **4**); Wittig reaction (**4** ← **5** ← **6**, **7** ← **8**); hydration of an α,β-unsaturated aldehyde (**6** ← **7**); and aldol condensation (**8** ← **9**).

Our early results attracted the attention of a number of synthetic organic chemists, who would often challenge the program with problems of special interest to themselves. These were usually structures for which a total synthesis was being sought, or for which they had recently published a synthesis. An example of the first kind is tirandamycic acid, a degradation product of the naturally occurring antibiotic tirandamycin. Figure 2 shows the route developed by SYNCHEM for tirandamycic acid after approximately 15 minutes of computation. The innovative aspects of the procedure reside in the last stages of the synthesis (that is, the first few retrosynthetic steps), wherein the furan ring, which remains relatively inert to the chemistry necessary to build up the side chain, undergoes oxidative cleavage to provide an unsaturated diketone, which can in turn undergo intramolecular ketal formation in a cyclization reaction that results in the required heterocyclic ring formation. The proposed synthesis created a certain amount of interest and curiosity in the Rinehart group, where the problem had originated, and an attempt was made to synthesize the antibiotic by that route in the laboratory. A model study based on an analog of the furan structure (**4** in Fig. 2) indicated, however, that the conditions necessary to open the furan ring could not be tolerated by the remainder of the molecule, and the approach was reluctantly discarded (*13*). To our knowledge, a total synthesis for tirandamycic acid has not yet been reported in the literature.

As a consequence of the interest in SYNCHEM generated by the latter result, V. Lee of the Rinehart group spent a fortnight at Stony Brook during the summer of 1973, where, while learning the details of the program, he contributed a good deal of new chemistry to the reaction library, especially with respect to nitrogen heterocycle transformations. The expanded system was able to produce the synthesis route displayed in Fig. 3 for the compound slafradiol (monomethyl ether), which had been synthesized several months earlier at the University of Illinois by a route similar to that suggested by the program (*14*).

The results discussed thus far all have the following characteristic in common: the input structures were developmental test problems designed to provide information about SYNCHEM's behavior rather than syntheses for the target molecules. Since we wished to learn whether

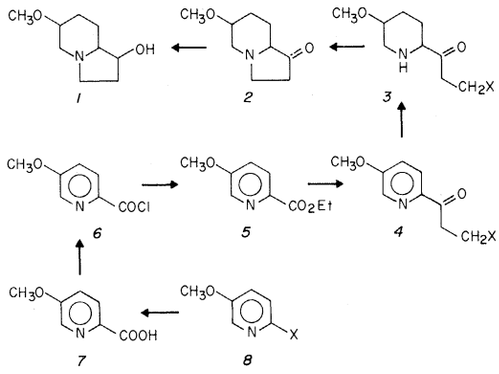

Fig. 3. Pathway proposed by SYNCHEM for the monomethyl ether of slafradiol. Reaction types used: reduction of a ketone (**1** ← **2**); internal alkylation of a secondary amine (**2** ← **3**); hydrogenation of a pyridine (**3** ← **4**); a naive reaction (**4** ← **5**) that could be accomplished by a variety of multistep sequences; esterification of an acyl halide (**5** ← **6**); formation of an acyl halide from a carboxylic acid (**6** ← **7**); and another naive multistep reaction (**7** ← **8**)—for example, oxidation, displacement with cyanide, reduction, and hydrolysis to carboxylic acid.

SYNCHEM possessed the problem-solving ability to generate a reasonable search tree in an acceptable amount of time, given the required level of "knowledge" of synthetic organic chemistry, it was our practice to provide the program with as complete a library of relevant reaction schemata as we could manage before attacking the problem, and to augment and modify the reaction library in response to what we had learned when SYNCHEM's efforts faltered. In the case of tirandamycic acid, for example, the synthesis-search tree was quite pedestrian until F. W. Fowler, of Stony Brook, noticed that the unsaturated diketone intermediate that SYNCHEM had generated by retrosynthetically opening the cyclic ketal could be produced by oxidative cleavage of a furan. The latter reaction was added to the library, whereupon the route displayed in Fig. 2 was produced.

The skeptical reader may question whether such a program is, in fact, behaving intelligently, and not merely regurgitating those mild insights that we might have slyly managed to sneak in beforehand. We offer the doubter a twofold response. On one level, without conceding the validity of his objection, we point out that SYNCHEM can become an extremely useful tool for the organic chemist while serving merely as a massive organic synthesis information system. Returning to the example of tirandamycic acid, one may adopt the point of view that SYNCHEM, through the medium of its reaction library, made it possible for Fowler of Stony Brook to contribute from the base of his particular background and experience to the deliberations of Lee of Illinois. More generally, all who contribute to SYNCHEM's chemical data base become, in a very strong sense, consulting synthetic chemists to all who draw upon it. By this, we mean that information extracted from the reaction library has been highly selected, refined, and tested for relevance to the query by the processes of subgoal generation, synthesis tree pruning, and subgoal evaluation before being offered to the user.

On a second, rather higher plane, we contend that the ability to apply information selectively in a context wholly different from that in which it was gathered is a well-recognized characteristic of intelligent behavior, and that SYNCHEM has this ability. The synthesis developed for the quinuclidine derivative 5-ethenyl-1-azabicyclo[2.2.2]octane-2-carboxylic acid (a cinchona alkaloid precursor), shown in Fig. 4, provides evidence for this assertion. Except for the last few retrosynthetic steps of the route (which we will discuss in some detail below), SYNCHEM produced the result displayed with its then current data base; no external guidance or additional chemistry was provided. In effect, the program was able to use what it had learned in dealing with the slafradiol problem to solve the related quinuclidine problem. Although the quinuclidine result differs from the published total synthesis for that molecule (*15*), we feel that SYNCHEM's proposal, at least in its structure-building aspects, is a good one, with reasonable prospects for realization in the laboratory.

As additional evidence that SYNCHEM is something more than just a very large information retrieval system, we report briefly on the outcome of an exercise completed during the last year of its regular use. The staff of a well-known pharmaceutical company, wishing to evaluate SYNCHEM's potential as an aid to its research and development program, devised the following test. The program was run for approximately 30 minutes on a relatively simple but commercially important compound. At the same time, a number of the company's staff of synthetic chemists spent a few hours writing down every reasonable synthesis route they could think of for the same molecule. SYNCHEM developed about eight different routes; the chemists produced about a dozen. Of special interest, however, is the fact that not only were all of the chemist-generated routes for which the chemistry was available in its reaction library also discovered by the program, but SYNCHEM proposed one additional route which was new and different, and was considered to be of potential value to the company (*16*).

Before concluding the discussion of results obtained with the first version of our synthesis-discovery system, however, it is well to reiterate that even in its final, most highly developed form, SYNCHEM's chemical data base was incomplete at best for some classes of reaction schemata, and rudimentary for most of the rest. From the beginning, SYNCHEM always performed above our reasonable expectations at each stage in its development. Detailed stereochemical transformations, for example, were not included in our initial program design simply because we never expected to be able to deal with structures complex enough to require such considerations. Although we are pleased with the results obtained with SYNCHEM, they are insufficient grounds for jubilation. Extraordinarily rapid progress during the early stages of an attack on a new problem area is a rather common occurrence in artificial intelligence research; it merely signifies that the test cases with which the system has been challenged are below the level of difficulty where combinatorial explosion of the number of pathways in the problem state space sets in. What *is* of significance, however, is the inference that a great many synthesis problems of practical interest lie below the "natural" threshold for combinatorial explosion. It is the goal of artificial intelligence research to move that threshold higher and higher on the scale of problem complexity through the introduction of heuristics—heuristics to reduce the rate of growth of the solution tree, heuristics to guide the development of the tree so that it will be rich in pathways leading to satisfactory problem solutions, and heuristics to direct the search to the "best" of these pathways.

SYNCHEM's Deficiencies

Our first attempt at a synthesis-discovery program provided substantial evidence that our research objectives were feasible and enabled us to make gratifying progress toward those objectives. Nevertheless, we come to bury SYNCHEM, not to praise it. Paradoxically, most of SYNCHEM's greatest weak-

nesses derived from its surprising early show of strength. It was difficult to resist the urge to push forward the program's capability to deal with more complex and structurally diverse classes of organic species after an unexpected success with a problem thought to be beyond its reach. In doing so, our tendency was to make patchwork corrections of program deficiencies that interfered with our progress, and to neglect those that did not. Also, because we are primarily an entity of a computer science department engaged in exploratory research in computer science, we tended to pay inadequate attention to both detail and breadth in considering questions of organic chemistry as opposed to questions of computer science. Building bridges into new terrain leaves little time or energy for filling potholes in the road behind. The upshot was that the system we had to work with soon became unreliable to the extent that, with increasing frequency, SYNCHEM began to strangle on new synthesis problems, which would unearth program bugs that had been glossed over earlier, when they had presented no obstacle to our progress.

A decent burial, however, demands a eulogy. There are some things to be said for SYNCHEM beyond a recitation of interesting multistep synthesis routes for interesting target structures discovered without the help of on-line guidance. We enumerate the features that distinguish our program from others described in the literature. The reader is referred to our earlier report (*10*) for elaboration.

First, SYNCHEM had at its disposal a large library (the augmented Aldrich catalog mentioned above) of compounds generally regarded as reasonable starting materials for a total synthesis. This listing provided more than the pathway-search-terminating criterion necessary for a noninteractive program. Because every newly generated precursor was checked against this list of compounds before being entered onto the search tree, unusual or complex but commercially available starting materials with which the chemist using the system might not be acquainted were often discovered by the program. This would sometimes lead to an unexpected turn in the search strategy, and thence to a set of proposed routes using chemistry unanticipated by the chemist.

Second, a reaction compiler was supplied which made it quite easy to introduce new chemistry into the reaction library. To add a new transformation or to modify an old one, programming was not necessary. Instead, the reaction would be described in tabular format, to be converted into an internal data structure by the compiler. The transformation itself was specified in the same schematic format customarily used by the chemist in communicating such information to other chemists. SYNCHEM's subgoal generator interpreted the data structure to create a set of possible precursors, and then pruned that set according to the qualifying information provided in that same data structure (*17*).

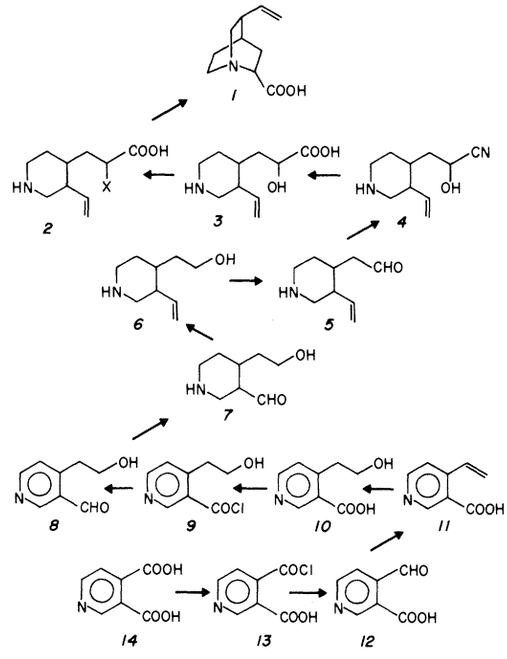

Fig. 4. Pathway proposed by SYNCHEM for a cinchona alkaloid precursor. Reaction types used: alkylation of a secondary amine (**1** ← **2**); replacement of an alcohol by a better leaving group (**2** ← **3**); hydrolysis of a nitrile (**3** ← **4**); cyanohydrin formation (**4** ← **5**); oxidation (Moffat) of a primary alcohol (**5** ← **6**); Wittig reaction (**6** ← **7**); hydrogenation of a pyridine (**7** ← **8**); partial reduction of an acyl halide (**8** ← **9**); formation of an acyl halide from a carboxylic acid (**9** ← **10**); and hydration of an alkene (**10** ← **11**). The remaining steps in the sequence are discussed in the text.

Third, SYNCHEM's search strategy was directed by an algorithm which, for each subgoal precursor that was a candidate for further development, took into account both the "cost" of reaching the target from that intermediate and the estimated difficulty of synthesizing the intermediate from available starting materials. That is, the subgoal selection criterion was a function of the accumulated heuristic estimates of reaction merit and yield along the path from subgoal to goal, and of a prediction of the probable reaction merit and yield along the best path from starting materials to the subgoal. The predictor was based on a heuristic evaluation of the complexity of the intermediate with respect to the standard manipulations of synthetic organic chemistry.

Finally, SYNCHEM contained a quite sophisticated heuristic device for continuously updating the problem-solving tree throughout the search phase of program execution. Information about the problem state space provided by the reaction path merit and compound complexity evaluation functions after each cycle of subgoal generation was backed up and distributed throughout the search tree. The selection of a new subgoal for development after the completion of a generating cycle always began with a scan of the tree from the goal down. If, as was often the case, newly acquired information had modified heuristic merit ratings for subgoals off the path last pursued by the search algorithm, the next subgoal selected for development might lie on a completely different branch of the tree. Thus, the program was not constrained to follow an initially poorly chosen branch to the bitter end before exploring possible alternatives.

Having paid all due respect to SYNCHEM, we return to our reasons for discarding it. We have already mentioned that an accumulation of bugs and patches made the program painfully unreliable. Compounding the problem, its eclectic construction made parts of the program inefficient and obscure. Because we neglected to keep our documentation complete and up to date, all serious attempts to repair the spreading decay ended in failure. Even our successful effort to improve execution efficiency by replacing most of the PL/1-programmed data-manipulation primitives with subroutines written in IBM System/360 assembly language proved, on balance, to be unwise, because it severely limited the portability of the program to other computer installations.

In addition to the general problems described above, SYNCHEM suffered from a number of specific weaknesses. One was a consequence of our selection of WLN to provide a canonical and indexable name for each organic structure accessed and manipulated by the sys-

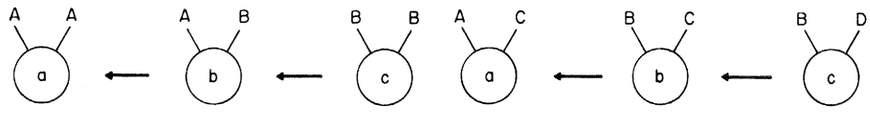

Fig. 5 (left). Abstract representation of the schema validation test problem. Fig. 6 (right). Abstract representation of the generalized form of the schema validation test problem.

tem. The conversion programs that were necessary to translate freely between linear and graphical representations of the same compound often bogged down in the complex morass of the WLN rules. Despite a number of attractive features of WLN (relative readability by chemists, for example), we have since determined that it was a poor choice for use in a program that must continually convert between graphical and canonical linear formats for compounds of arbitrary and unpredictable complexity throughout execution (*18*, p. 148); not only do the WLN rules contain pockets of ambiguity, but they are so structured as to frustrate any attempt to design a reasonably efficient algorithm to realize them.

The fixed input format that made it so convenient for chemists to modify and expand the reaction library also placed restrictions on the kinds of modifications that could be made and on the variety of transformations and heuristic tests that could be specified. It was not difficult to enlarge the scope of the reaction library compiler to accommodate new requirements as we became aware of them, and we often did this, but we found it impossible to keep up with the aspirations of our chemist colleagues, which always managed to remain one step ahead of the compiler then operational. It soon became clear that a fixed format language was not a suitable medium for expressing the wide range of information that chemists wished to communicate to the program.

Finally, as we accumulated experience in using SYNCHEM, evidence began to mount that there were defects in the organization of the subgoal-generating algorithm that would require major redesigning to correct, and, as we mentioned above, the program was too shaky to sustain substantial reworking. The last few retrosynthetic steps of the quinuclidine synthesis provide an example of the kind of situation that SYNCHEM could not manage (although in this specific case, heuristic tests to identify the difference in reactivity at the carbon atoms beta and gamma to the nitrogen in the pyridine ring would have enabled SYNCHEM to complete the pathway unaided). In general terms, the problem is a consequence of the recursive structure of the subgoal generator. Each time a new compound is selected for development, it is treated as if it were the target molecule for a new synthesis problem until that phase of the algorithm is reached where newly created precursors are to be entered onto the search tree. Only then are relationships between new precursors and the previously explored problem space examined. At the particular point of the subgoal-generating cycle where the construction and heuristic screening of new precursors takes place under the control of the reaction schemata, the retrosynthetic history of the molecule undergoing development is not readily accessible to the heuristic test procedures. This restriction on freedom of internal communication affected SYNCHEM's tactics in two ways.

First, the program was unable to take advantage of the fact that the first two steps of the synthesis (from $-$COOH to $-$CHO at the beta position) were identical to the fifth and sixth steps (from $-$COOH to $-$CHO at the gamma position), and could therefore be performed simultaneously. (The question of the suitability of the chemistry is another matter, which is not at issue here.) More generally, SYNCHEM had no way of knowing when a series of reactions performed sequentially could not better be executed simultaneously because, say, each was an oxidation of a functional group (not necessarily identical for each step) with the same category of oxidizing reagent. Second, SYNCHEM was unable to carry the retrosynthesis beyond compound **11** (Fig. 4) because the validation tests for the Wittig reaction excluded precursors carrying a carboxylic acid functionality. This is an example of a more general effect that would sometimes occur when the goal carried more than one instance of an unprotectable functional group for which no multi-instance synthetic schemata were available in the library. Referring to Fig. 5, if functional group A may be derived from functional group B by, say, oxidation, then unless a procedure is known whereby B may be protected from the reagent, a second instance of B in the goal precludes the use of that transformation, for that second B would occur in the subgoal as well and would be oxidized to A along with the B under development. Thus, despite the fact that, in the case under consideration, oxidation of both B's is the desired result, the retrosynthetic step from b to c in Fig. 5 will be excluded by the validation tests.

An even more general and less tractable form of the same problem is illustrated in Fig. 6, where A and B are as above, and C is a functional group different from A which may be derived by the oxidation of functional group D, which is different from B. Again we assume that neither B nor D is protectable, and that suitable multifunctional schemata are not available. And again the retrosynthetic step from b to c will be excluded by the validation tests which discard precursors carrying unprotectable functionalities (in this case, B) that would be attacked by the oxidizing agent, even though that might be exactly what we seek.

SYNCHEM2 was designed to deal with these problems and with the representation and manipulation of stereoisomers. The program was written to rigidly controlled and maintained standards of debugability and reliability, while retaining the features of its predecessor that established the feasibility of noninteractive synthetic route discovery by computer.

SYNCHEM'S Successor

SYNCHEM2 first reached the stage in its development where interesting chemistry could be produced during the summer of 1976. Since that time, our continued progress has been seriously impeded by the replacement of the computer at Stony Brook with a new system lacking a satisfactory PL/1 compiler, so that much of our recent effort has been directed toward the nonproductive activity of moving our operations elsewhere. Nevertheless, by running OS/360, the IBM System/360 operating system, on a temporarily available Spectra 70 emulating IBM architecture, we were able to achieve some results with SYNCHEM2 before the hiatus forced on us by the loss of our computer.

A full description of SYNCHEM's successor is beyond the scope of this article and, in any event, should await the accumulation of a larger body of experience with the new system. For one thing, many of SYNCHEM2's important new procedures that have been fully tested in isolation cannot be integrated into the complete system until it is once again running in a satisfactory environment. We can, however, provide some idea of the current state of the program by exhibiting one of its most recent results.

The target molecule, a 5,7 carbocyclic ketone unsaturated at the fusion junction, was introduced as a discussion exercise in a graduate organic synthesis class at Stony Brook. Because we were running in a highly inefficient emulation mode on a computer with inadequate fast-access storage, four cumulative runs, each about 45 minutes long, were required to collect the search tree displayed in Fig. 7. It is difficult to make a reliable estimate of what the running time would have been on the IBM System 370/155 for which the program was designed and on which our earlier SYNCHEM data had been gathered, but a reasonable guess is that performance degradation approached an order of magnitude. If this is the case, the result exhibited represents about 20 minutes of time on the IBM 370/155, an interval roughly equivalent to that required for the results in Figs. 1 through 4.

The routes developed by SYNCHEM2 for 2,3-cyclopentenocyclophept-2-enone have the following points of interest. First, the branches explored in depth by the program are different from that pursued by the class in committee. The class's selection, compound **3**, would soon have been pursued by SYNCHEM2 had the run continued a while longer, and might in fact have been chosen earlier had the search strategy currently in effect been operational at the time of the run. Second, the branch developed by the program through compound **2** contains an alpha-bromination step of questionable yield. Below that point, however, are a number of novel and quite reasonable synthetic steps—namely, the synthesis of 1,2-cyclopentenocycloheptene through Wagner-Meerwein rearrangement of 5,6 and 4,7 spiro systems, and the synthesis of the 4,7 spiro system by 2+2 cycloaddition. Third, the retrosynthetic step to compound **4**, a favorable gamma alkylation of an α,β-unsaturated ketone, is not immediately obvious on casual consideration, and would be expected to lead to another simple synthesis route. Finally, we mention that the double alkylation synthesis of the 5,6 spiro system could not be found by the program, because at that time SYNCHEM2's multiple-transform algorithm was not yet available. The procedure has since been written and checked out; there is no doubt that when it has been incorporated into the program that path on the tree will be completed without intervention.

We have already indicated that the reprogramming of SYNCHEM was undertaken to correct at the outset a number of serious problems that had developed with our original design. In so doing, several features, some quite innovative, were introduced into the system which we expect will provide a longer useful life for SYNCHEM2 than was enjoyed by its predecessor. For example, we found it almost impossible to modify the data structures representing organic molecules in SYNCHEM to include the additional information necessary to represent stereochemistry because these same structures were accessed at so many different points in so many distinct program modules. Each small change in the data structure neces-

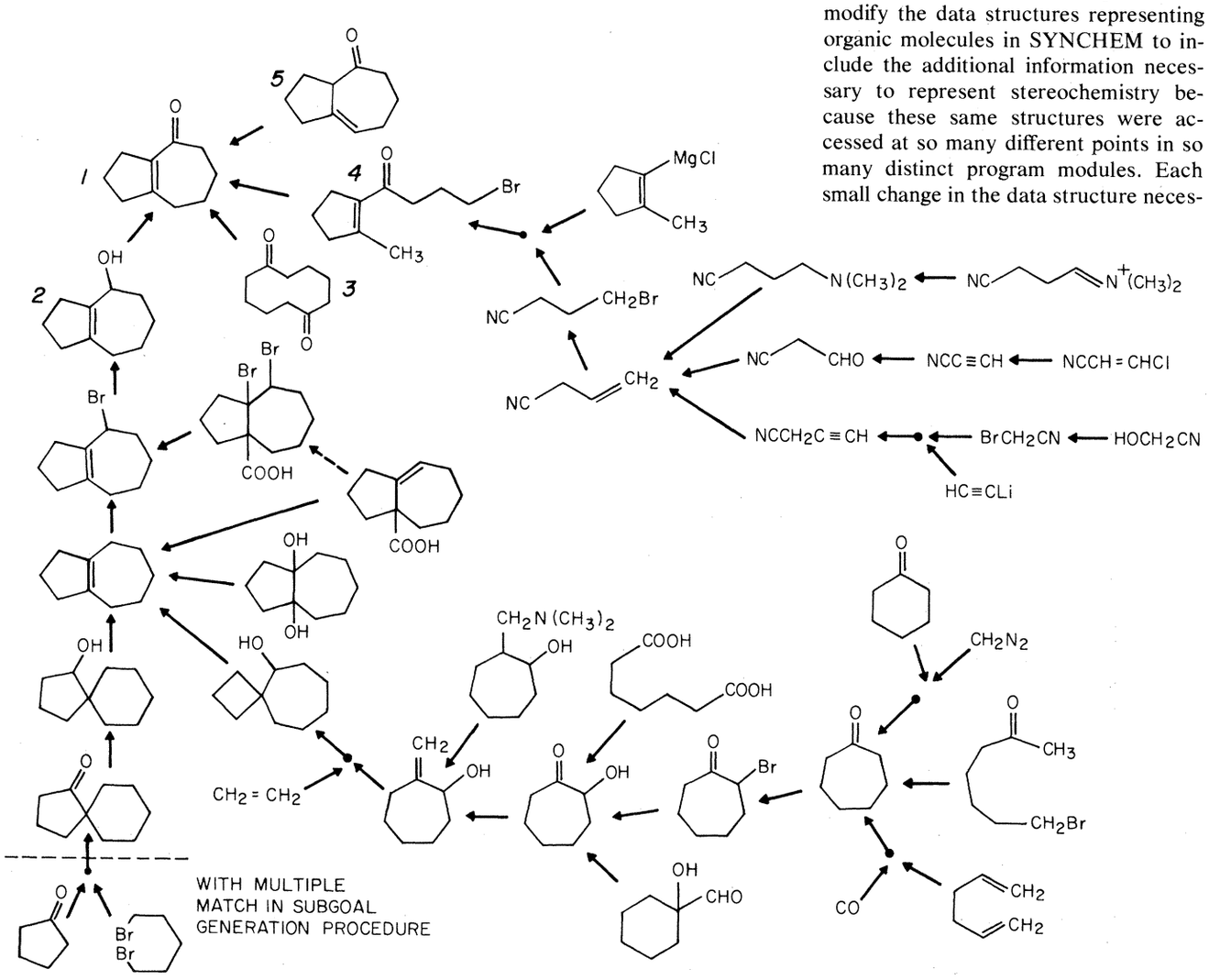

Fig. 7. A partial synthesis tree developed by SYNCHEM2. The more extensive reaction library may be inferred from the greater variety of transformations evident in this search tree. The current lack of a list of available compounds comparable to that of SYNCHEM2's predecessor is also evident in the program's persistence in expanding very simple subgoals.

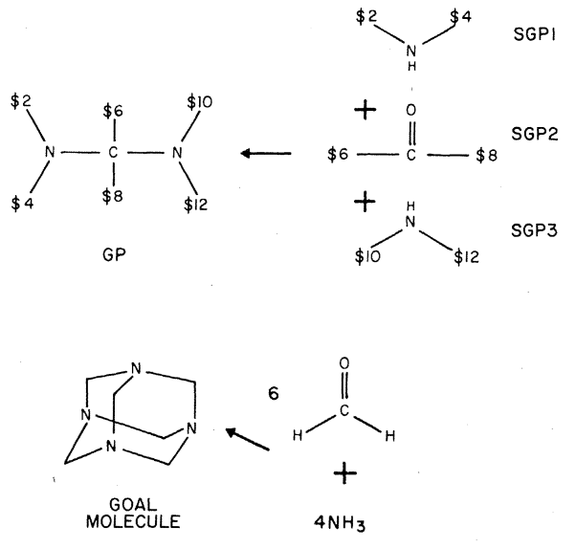

Fig. 8. The nodes $(integer) in the reaction transformation schema (top) stand for molecular fragment variables. The subgoal $6H_2CO + 4NH_3$ is one of the complete set of MULTIPLE MATCH precursors generated by the algorithm for the illustrated goal molecule.

sitated reworking of every module that accessed it, even if the reference had nothing to do with that particular change. In SYNCHEM2, all such globally accessed data are isolated from the program in an entity which we called a metastructure (*18*, p. 154), but which has since appeared in the computer science literature under the name "encapsulated data structure" (*19*). Information in the metastructure is accessible only through an interface of primitives that are conceptually part of the data structure itself. Modifications of information content or format within the capsule require modification only of the primitives that service the capsule; program modules that refer to the metastructure need no alteration to remain valid.

Wiswesser linear notation has been discarded as the internal canonical linear representation for organic molecules in favor of a new nomenclature system called canonical SLING (for SYNCHEM linear input graph) (*20*). Because its transformation rules are relatively simple, canonical SLING is far more suitable for computer manipulation than WLN, which we retain only for the final output listing. Our new notation, unlike WLN, can express the stereochemistry as well as the connectivity of a chemical structure. The algorithm produces the canonical name in two parts, the first representing the constitution of the molecule and the second its stereochemistry. Diastereomers have identical constitutional parts but different stereochemical descriptors. In generating the name, constitutionally equivalent atoms are labeled, as are those that are stereochemically equivalent. Generalized Hückel and resonant substructures of the molecule are recognized, so that different resonant forms of the same molecule may be given the same name. A good deal of useful stereochemical information is readily extracted from the descriptors. For example, diastereomers are easily identified, and a trivial computation yields the smallest number of asymmetric carbons at which two diastereomers differ. Although canonical SLING was developed to provide a complete and independent canonical stereochemical description of a molecular structure for SYNCHEM2, the stereochemical descriptors alone may be used in conjunction with other standard systems of nomenclature (WLN, for example) to extend their range to include stereochemistry.

As might be expected, the graph transformation procedures used by SYNCHEM2 in its subgoal generator routinely manipulate the stereochemistry of a molecule along with its structural connectivity (*18*, p. 113; *21*, p. 118). The algorithms are sufficiently general to permit a given transformation schema to be applied in either direction. Thus, the reaction schemata in SYNCHEM2's library may be used not only to generate all precursor stereoisomers for a target stereoisomer but, applied inversely (that is, synthetically, rather than retrosynthetically), also to generate all stereoisomer products for a given set of reactants (*21*, p. 150). This feature of the system will be used in the path evaluation phase of the synthesis search procedure to enable SYNCHEM2 to predict possible by-products and estimate yields for each proposed step of the synthesis.

Strategies and tactics for guiding the development and traversal of the search tree have undergone substantial modification and expansion in being rewritten for SYNCHEM2. This topic in particular requires a good deal more experience with the fully integrated program before it can be usefully discussed. One tactical device, however, is worth mentioning here. The problem of generating precursors for target molecules that carry multiple instances of the same functionality (illustrated schematically in Fig. 5) is dealt with by a procedure called MULTIPLE MATCH, which incorporates the graph transformation algorithm in a heuristic control program. The heuristic enables MULTIPLE MATCH to generate with some degree of efficiency every possible precursor for every combinatorially possible set of multiple matches of the goal pattern to the multiplicity of instances of the target functionality (*21*, p. 167). Not only does this device solve the problem above, but it also makes it possible for SYNCHEM2 to execute an important class of retrosynthetic transformations that was beyond the capability of SYNCHEM's subgoal generator. An example of this kind of precursor, one of a series of test cases for MULTIPLE MATCH, is exhibited in Fig. 8. Together with the bidirectional graph transform capability, MULTIPLE MATCH makes SYNCHEM2 a potentially valuable tool for the investigation of biosynthesis in general, and metabolic processes in particular—the former because biosynthesis often involves simultaneous transformation of many chemically equivalent sites on large molecules, and the latter because of the important role of degradative processes in metabolic chemistry.

SYNCHEM's fixed format reaction compiler has been replaced in SYNCHEM2 by a PL/1-based reaction input language. While we have thus sacrificed a measure of simplicity and convenience in modifying and updating the reaction library, we have gained substantially in range, versatility, and ease in maintaining the language. In an attempt to recover some of those lost virtues that make a complex system such as ours accessible to the nonprogrammer chemists for whom it was intended, a conversational monitor is being prepared to guide the user who wishes to interact with the reaction library.

Compared with its predecessor, SYNCHEM2 has been relatively easy to maintain and expand. This was no accident; the entire system is designed within the discipline of structured programming insofar as this is possible in PL/1. In addition, debugging trace and central processor unit (CPU) clock access routines are built into each program module. An input parameter array enables the programmer to specify for each run which internal procedure blocks, if any,

he wishes to time, for which program modules, if any, he wishes to have debugging output produced, and the level of detail required for that output. The easy availability of CPU clock information enabled us to determine that the single most time-consuming process in the subgoal-generating cycle was that of computing canonical names for the ensemble of possible precursors created during that cycle. Our response to this finding has been to delay the naming of generated subgoals until the last possible moment in the hope that unsuitable precursors may be rejected by a heuristic test before naming is necessary. For example, many of the subgoal validation tests that had originally operated on the canonical name of a proposed precursor were modified to take the structure graph as input instead. Since a substantial fraction of the validation tests result in discarded subgoals, the change produced a noticeable improvement in the efficiency of the subgoal generator.

Conclusion

We conclude this article with some observations concerning the prospects for continued progress in extending and expanding applications of artificial intelligence to the practical solution of real problems of interest to people in the real world. Years of experience with the programs described here and with others similar in scope and intent make us reluctant to predict that routinely useful programs of this kind lie just one step beyond the current state of the art. From the beginning, the computing resources demanded by this research effort have taxed the limits of the facilities available to us. Each increase in the problem-solving power, search-guidance sophistication, and domain of applicability of our synthesis-discovery system has increased our requirements for computing power, operating software sophistication, and running time. Problem-solving heuristics, some quite clever, others rather pedestrian, have enabled us to extract more and more useful computing from our limited resources, but heuristics, too, demand their share of those resources and levy their toll in computing time.

So much for the bad news. The good news is that as our computing resource requirements have increased over the years, the performance/price ratio for contemporary computer systems has increased more or less in pace. The upshot has been that we have always been able to meet our needs, if occasionally just barely, within the confines of available facilities. It is an interesting although quite useless fact that the computing time required by the antique vacuum tube IBM 704 to prove a theorem of average difficulty in Euclidean plane geometry in 1959—20 minutes or so—was about the same as that required by SYNCHEM for a synthesis of average difficulty, although the latter problem is orders of magnitude more complex than the former. It would be surprising indeed if progress by the manufacturers of computer hardware and software systems failed to continue to keep pace with progress in the design of intelligent programs. In any event, complex developmental programs running on systems intended for the development of complex developmental programs always require more in the way of computing resources and execution time than the same programs running in production on a dedicated system. If SYNCHEM2 were shorn of the many debugging and program development aids built into it, and running on a facility supplied with just the internal and peripheral features necessary to support the program, its computing costs would drop to a small fraction of what they are now. On balance, we would predict that if routine use of computer programs like SYNCHEM2 is not just around the corner, rewarding investigational applications certainly are.

Summary

During the past several years, a substantial body of experience has accumulated in the use of SYNCHEM, a large-scale program which is able to discover synthesis routes for relatively complex organic structures without on-line guidance on the part of its chemist user. These results indicate that the approach to computer-directed organic synthesis route discovery embodied in the program has been valid and reasonable, and that SYNCHEM is likely to be fruitful from the point of view of its intended users as well as for our research objectives in artificial intelligence. The experiments have revealed a number of insufficiencies in the program as well. Most of these are rectified in SYNCHEM2, a revised version of the program which includes, among other improvements, a more highly developed synthesis search algorithm and the routine consideration of stereochemistry.

References and Notes

1. E. J. Corey and W. T. Wipke, *Science* **166**, 178 (1969).
2. S. R. Heller, G. W. A. Milne, R. J. Feldman, *ibid.* **195**, 253 (1977).
3. M. Bersohn and A. Esack, *Chem. Rev.* **76**, 269 (1976).
4. The papers reporting these results are reprinted in E. Feigenbaum and J. Feldman, Eds., *Computers and Thought* (McGraw-Hill, New York, 1963).
5. These issues are discussed briefly (with references) by H. A. Simon [*Science* **195**, 1186 (1977)]. An excellent overview of the area of artificial intelligence research into which the SYNCHEM research project falls will be found in N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence* (McGraw-Hill, New York, 1971).
6. R. E. Carhart, D. H. Smith, H. Brown, C. Djerassi, *J. Am. Chem. Soc.* **97**, 5755 (1975); J. Lederberg, G. L. Sutherland, B. G. Buchanan, E. A. Feigenbaum, in *Theoretical Approaches to Non-numerical Problem Solving*, R. Banerji and M. D. Mesarovic, Eds. (Springer-Verlag, New York, 1970), pp. 401–409.
7. H. Gelernter, *Polytech. Inst. Brooklyn Symp. Ser.* **12**, 179 (1963).
8. See G. Ernst and A. Newell, *GPS: A Case Study in Generality and Problem-Solving* (Academic Press, New York, 1969).
9. D. J. Duchamps, A. R. Branfman, A. C. Button, K. L. Rinehart, Jr., *J. Am. Chem. Soc.* **95**, 4077 (1973).
10. H. L. Gelernter, N. S. Sridharan, A. J. Hart, S. C. Yen, F. W. Fowler, H. Shou, *Top. Curr. Chem.* **41**, 114 (1973).
11. *Aldrich Chemical Catalog 15* (Aldrich Chemical Co., Milwaukee, Wis., 1970).
12. H. W. Whitlock and M. W. Siefken, *J. Am. Chem. Soc.* **90**, 4929 (1968).
13. V. J. Lee, thesis, University of Illinois, Urbana (1975).
14. W. C. Christophel, thesis, University of Illinois, Urbana (1976).
15. G. Grethe, H. L. Lee, T. Mitt, M. R. Uskokovic, *J. Am. Chem. Soc.* **93**, 5904 (1971).
16. S. A. Lang, Jr., private communication.
17. S. C. Yen, thesis, State University of New York, Stony Brook (1974), p. 102.
18. A. F. Sanders, thesis, State University of New York, Stony Brook (1976).
19. Encapsulated data structures, of which the data metastructure is a particular realization, are concrete program representations of the concept of "abstract data type." See B. Liskov and S. Zilles, *SIGPLAN Not.* **9**, 50 (1974); O. Dahl, E. W. Dijkstra, C. A. R. Hoare, *Structured Programming* (Academic Press, New York, 1972).
20. K. K. Agarwal, R. H. Boivie, H. W. Davis, H. L. Gelernter, in preparation; H. W. Davis, *Computer Representation of the Stereochemistry of Organic Molecules* (Birkhauser, Basel, 1976).
21. K. K. Agarwal, thesis, State University of New York, at Stony Brook (1976), p. 118.

22. We are deeply indebted to F. W. Fowler of the Department of Chemistry at Stony Brook for his active collaboration with this research group during the critical early years. We thank I. Ugi for discussions leading to our system for dealing with stereochemistry in SYNCHEM2, and V. Lee for his contributions to SYNCHEM's reaction library. P. Helquist has been most helpful in providing continued interaction with the Stony Brook Chemistry Department. This work has been supported in part by a series of research grants from Lederle Laboratories and by U.S. Environmental Protection Agency contract 68-01-1734.