Simulation and Evaluation of Chemical Synthesis—SECS:

An Application of Artificial Intelligence Techniques

W. Todd Wipke, Glenn I. Ouchi, and S. Krishnan

Board of Studies in Chemistry, University of California, Santa Cruz, CA 95064, U.S.A.

Recommended by N. S. Sridharan

ABSTRACT

The problem of designing chemical syntheses of complex organic compounds is a challenging domain for application of artificial intelligence techniques. SECS is an interactive program to assist a chemist in heuristically searching and evaluating the space of good synthetic pathways. The chemist-computer team, linked through computer graphics, develops synthetic plans using a logic-centered backward analysis from the target structure. The reaction knowledge base, written in the ALCHEM language, is separate from the program and control strategies. Performance is demonstrated on the insect pheromone grandisol.

1. Introduction

Organic Synthesis is of premier importance in producing new chemical compounds for use as drugs, fuels, plastics, dyes, superconductors, and a whole host of other materials. Thus, there is great interest in being able to design good efficient syntheses of chemical compounds. The design of an organic synthesis requires ready knowledge of many reactions, chemical principles, and specific facts. Over 250,000 chemical papers appear annually in the chemical literature reporting new facts and principles and according to Chemical Abstracts, the chemical literature is increasing at 8.5% per year compounded. Over 4,000,000 different chemical compounds have been reported in the literature. As in all fields of science it is virtually impossible for a single person to keep track of all the developments taking place, even in this seemingly narrow field of specialization.

Clearly it would seem beneficial to have a computer help the chemist digest the available information, and evaluate, correlate, and extrapolate it to provide ideas for new experiments for advancement of knowledge in this field and for solving important synthetic problems. The creation of a computer program which could

Artificial Intelligence **11** (1978), 173–193

aid an organic chemist in this way thus became the goal of the SECS (Simulation and Evaluation of Chemical Synthesis) Project.

SECS was designed to be a "knowledge-based" "performance" program to assist the chemist in planning syntheses of complex molecules. The knowledge base includes information about structural theory, chemical reactions, chemical reactivity, and principles of chemical synthesis. Since new chemical information is continually being discovered in the laboratory, it was important to be able to easily add new pieces of chemical knowledge to the program's knowledge base, without restrictions on the amount of such knowledge that could be added. Our performance goal for the program was that the program should be able to help a chemist find many more good and innovative syntheses than the chemist could working alone. Because of the complexity of the problem domain, we felt the chemist and computer working together with each assigned tasks for which they are best suited, and with efficient interaction between the two, would be more effective than either working alone. Our goal was not to replace the chemist, but to augment the chemist's problem solving capabilities. A non-interactive approach to this problem has been explored by Gelernter and Sridharan in which the program (SYNCHEM) works alone [1, 2]. A recent paper summarizes the experiences gained from that approach [3].

In this paper we describe the problem of synthesis and our approach to the problem, then we present an overview of the SECS system and its control structure, followed by an exploration of the knowledge base of SECS. We discuss two representational problems arising in this domain—representation of reactions and representation of synthesis plans or goals [4]. We conclude by examining some output from SECS on the grandisol problem, a simple insect pheromone which has been studied and analyzed extensively by chemists. This paper does not discuss machine representation of chemical structures [5] or stereochemistry and other supplementary information, which can be found in previous papers from this project [6, 7].

2. Organic Synthesis

The synthetic chemist is a master builder. He precisely joins together molecules which he cannot see or hold to construct elaborate three dimensional molecular structures. Woodward elegantly summarized the profession: "There is excitement, adventure, and challenge, and there can be great art, in Organic synthesis." The building materials available to the synthetic chemist are the small molecules obtainable directly or indirectly from nature. Most commercially available starting materials contain four or fewer carbon atoms. But a given target molecule may contain thirty or more carbon atoms joined together to form a skeleton to which may be attached several oxygen, sulfur, or nitrogen atoms at specific locations. The tool box of the chemist consists of the many reactions which have to date been discovered (probably 20,000). These are the operators which join, cleave, or modify

molecules in specific ways. Each reaction requires certain features to be present in the building materials and additionally requires a particular set of conditions (temperature, solvent, catalysts and reagents) and a procedure detailing the order of mixing, reaction time, and cleanup or purification operations.

The success of using one of these tools is expressed as the percent of molecules which reacted as expected. Because the chemist cannot hold the molecules in his hand he must rely on the natural reactivities of the reactive groups of atoms in the molecule and the selectivities of his tools, the reactions (chemical operators). Chemists have classified these frequently encountered reactive groups of atoms into entities called functional groups. Reactions are conveniently classified by the structural modifications they make which include

(i) interconversion, removal or introduction of functional groups,
(ii) addition to chains or appendages,
(iii) formation of rings,
(iv) rearrangement of a chain or ring,
(v) and cleavage of a chain or ring.

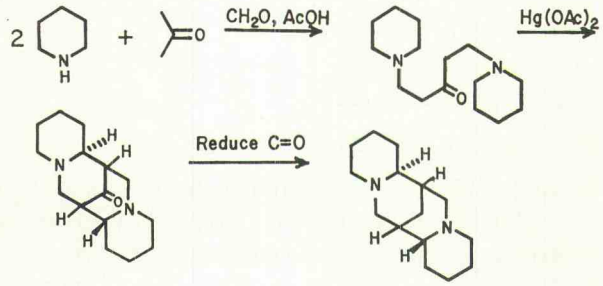
Reactions which simply interconvert functional groups without changing the skeleton are called functional group interconversions (FGI's), and play an important role in steps preparatory to a construction reaction in which a carbon–carbon bond is formed, and are similar to substitution operators in GPS [8]. Just as a builder uses bracing and other temporary construction in order to preserve the integrity of one part of a building while constructing another part, so we find the chemist incorporates certain functional groups temporarily to activate or deactivate a part of the molecule to help control where on the molecule the next reaction will take place, or to protect part of the molecule while reactions occur on another part. These groups are later removed, hence do not appear in the final product.

When the chemist finds there is no reaction that meets his requirements, he frequently tries to discover a new reaction, which is still possible since chemistry is an experimental science. This is part of the "art of synthesis." Another part of the art is the ability to find conditions under which a reaction will succeed when it fails with the conditions reported in the literature. Even small changes in the reactant structure can alter the reactivity causing a reaction to fail. These sensitivities to structural context are called the "scope and limitations" of a reaction. When a reaction is first discovered very little is known about scope and limitations—their elucidation is a continuing process.

Planning an organic synthesis as the reader can imagine from the previous discussion is a formidable intellectual challenge. Sarett [9] in 1964 was the first to realize that "automated planning of synthetic routes could enhance progress by five-fold" and "computer automation will eventually come to synthetic chemistry just as new instrumentation has come to structural problems." He pointed out

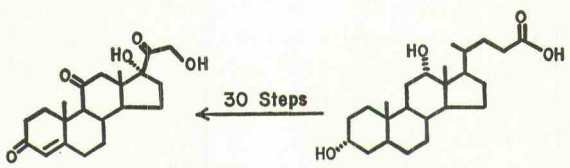
that a good synthetic plan is extremely important, because once a plan is selected, *many person-years may be required to execute the plan* in the laboratory. A more efficient plan may save much labor and time.

Later Corey and Wipke [10, 11] analyzed how a *chemist* plans a synthesis. They described three diverse problem solving techniques utilized by chemists. The first technique is described as *direct associative*, in which the chemist immediately recognizes the required starting materials and sequence of reactions for the synthesis. For example, Van Tamelen and Foltz [12] recognized sparteine (I) could be constructed by successive Mannich condensations, followed by simple removal of the (C=O) group. This type of chemical recognition is exhibited only by the most experienced chemists, and only on the more straightforward problems.



I

The second planning technique, which has been the basis of most syntheses in the literature, involves recognizing a relationship between a critical or major structural unit in the target molecule and a known or potentially available chemical compound. This problem-reduction allows the chemist to use forward analysis from the recognized material and backward analysis from the desired target. This planning method can be both very efficient or inefficient. The first synthesis of cortisone (II) was accomplished from deoxycholic acid (III), a natural product with the same carbon network including spatial orientation [13]. Even with this seemingly similar starting material, over thirty synthetic steps were necessary to complete the synthesis!



II III

The third planning approach, termed *logic-centered*, begins with a careful study of the synthetic objective for structural features which serve as clues in selecting the chemical reactions that might be used in the last step of the synthesis. This

method involves systematically working backward from the target *toward simpler materials*—basically a bottom-up approach [14]. Because this method is not biased by recognition of starting materials within the target molecule, it has greater capability of leading to truly innovative solutions. This approach cannot however discover some syntheses which were designed using a starting material oriented approach, particularly when the starting materials are more complex than the target (see Section 4). The simplicity of the logic-centered approach makes it ideal for teaching students and for systematizing on a computer. Indeed this planning approach was used in the first synthesis program [10], OCSS, and in the SECS program reported here.

Although chemists probably utilized implicitly the logic-centered planning method prior to the implementation of the first synthesis planning program, few if any syntheses in the literature made explicit mention of this method of planning. Today numerous applications of this planning method appear explicitly in the chemical literature [15].

2.1. The problem to be solved

To summarize the problem, the chemist will present the structure of a molecule to be synthesized, and the program is to generate possible synthetic routes and work with the chemist to search for good syntheses. The chemist acts as the evaluation function deciding which syntheses are most interesting for further exploration, and when to terminate the search process. The synthesis problems will be real problems from chemical research.

3. Computational Methods

The basic computational method used by the SECS-chemist team is Heuristic Search [14], but currently most of the heuristics for deciding which node of the search tree to expand next are embedded in the chemist using the program. This might be called "external heuristic search" since the heuristics are external to the program. One reason for taking this approach is that ultimately the goal is to find syntheses which are "good" as judged by the chemist involved. The criteria for what constitutes a "good" synthesis are related to the goals and past experience of the particular chemist user and may include the number of steps, the overall yield, the difficulty of the steps, and the familiarity of the chemist with the steps involved, i.e., chemists trust and prefer to use reactions which they have performed before in their laboratory. It is therefore rather efficient both of computer resources and human resources to have the chemist evaluate the syntheses at the earliest possible stage of exploration.

Another less obvious reason for this approach is that the chemist's own creativity is often stimulated by a particular molecule structure generated by the program even though that particular synthesis might be poor, because the chemist might know a way to make it a good synthesis. If the program were making all

decisions regarding which node in the tree to develop further, the poor route would not be expanded and the chemist might not see the structure.

The search for a synthesis plan proceeds backwards from the goal (target) molecular structure by the program applying reactions in the retro-synthetic direction. This produces a set of precursor structures which the program orders according to the structure's "priority value," the latter being the sum of the plausibility of the implied reaction and a factor representing the degree of strategic simplification achieved by applying this reaction. Structures having the highest priority value appear on the left side of the AND/OR tree (see Fig. 1), in which the nodes are molecular structures, the target is the root node, and the edges are inverse reactions (chemical transforms).

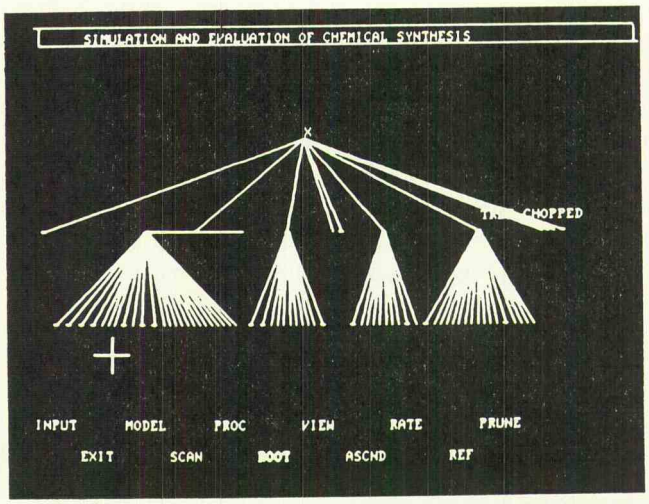


FIG. 1. Synthesis tree

An AND branch, shown as a horizontal line in the tree, arises when the target is cut into two or more fragments by a transform. Such a fragmentation simplifies the problem by creating smaller subproblems and is particularly efficient if the fragments are of nearly equal size.

Plausibility of the transform is determined by evaluating a series of rules which are a part of the definition of the transform (Section 3.2.1). Problem simplification is defined by a goal list which is created by strategy procedures (Section 3.2.3). The goal list however can also be modified interactively by the chemist, thus the chemist can convey to the program some of his preferred strategies for simplifying the problem.

In selecting which transforms to apply, SECS considers first the relevance of the transform to the goal list and second the applicability of the transform to the

target molecule. A transform is relevant if its CHARACTER matches the CHARACTER required for goals on the list, just as GPS used the character of operators for operator selection [8]. The current implementation of SECS expands a single node of the tree (the *current node*) in a breadth-first manner for one level only, and then returns control to the chemist. The exception to this is that when a key transform is found to be relevant, but not applicable, SECS is allowed to apply transforms to make it applicable and then apply the key transform. The chemist evaluates the precursors that were generated and selects one to be the *current node* for further expansion. That is the normal mode of operation. However, the chemist can direct SECS to automatically expand the tree to a certain level exhaustively, or expand only nodes with a priority value above a certain level, or expand only the first n nodes of each set of precursors.

Again, our objective has not been to see how intelligent the program can be made to appear, but rather to delegate to the program those tasks at which the chemist is weak, and to delegate to the chemist those tasks at which the chemist is strong. Thus the program selects and applies reactions, and the chemist guides the program along lines that seem interesting. The challenge here is to develop a program which serves as an intelligent partner to chemists having very different strengths, working styles, and goals. Our aim is to allow the chemist to assign responsibilities, establish preferred goals and strategies, and determine the amount of independence the program is to have. Within those boundaries, the program should take the initiative and be as clever as possible, yet be prepared to relinquish control to the chemist if requested. Success should be judged on the *team's* productivity compared with that of the chemist alone. We shall now give an overview of the system and then examine its knowledge base.

3.1. System overview

SECS is written in FORTRAN with a few assembly language functions for handling sets, lists, and dynamic arrays. It was designed to be relatively machine-independent and currently runs on PDP-10, UNIVAC 1108, Honeywell–Bull, and IBM 370 computers. SECS has its own software virtual memory system for data structures which grow with the search tree, thus large problems can never bring the program down due to insufficient memory. SECS executes in 48 K words (32 or 36 bit) of memory when overlayed, 130 K when not overlayed. A block diagram of SECS is shown in Fig. 2. The chemist communicates with the program either using a DEC GT40 graphics terminal with lightpen and keyboard, or a simple teletype device, but in either case the majority of communication is graphical, the natural language for organic chemists.

The chemist draws in the target structure by moving the lightpen to DRAW, then positioning the cross to the starting point for a bond, depressing the spacebar, moving the pen to the end point of the bond, and again depressing the spacebar. A line appears for the bond. Atoms other than carbon are indicated by selecting

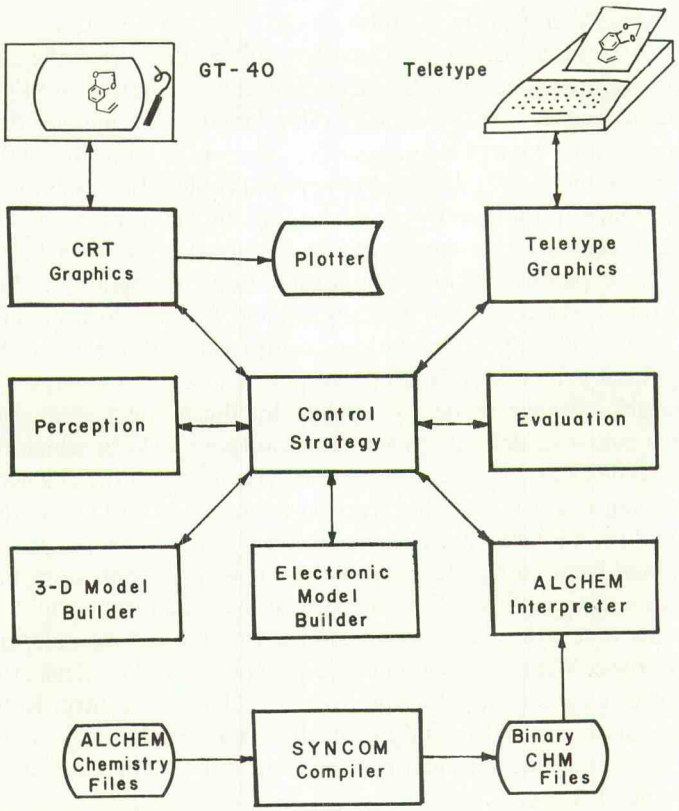


FIG. 2. Block diagram of SECS.

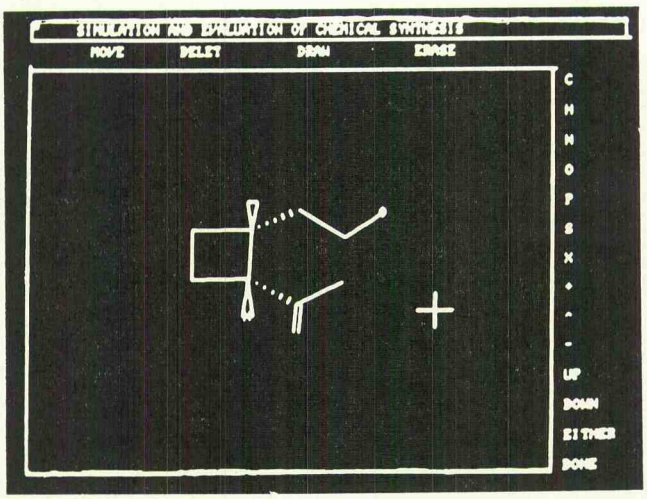


FIG. 3. Graphical input of target.

the atom type from the menu on the right side of the screen (Fig. 3) and then pointing to the atoms to be changed. UP and DOWN are for designating stereochemistry, i.e., the relative three-dimensional orientation of bonds [6]. When the structure input is DONE, the processing phase begins.

First the chemical graphs is analyzed for the presence of cycles of various sizes [7], functional groups [16], stereochemistry [6], symmetry [17], and finally it is canonically named [5]. From the connection table (CT) created by the graphics module, the perception module defines sets of atoms, bonds, rings, and list structures defining functional groups and rings. This information is more readily accessed than the CT.

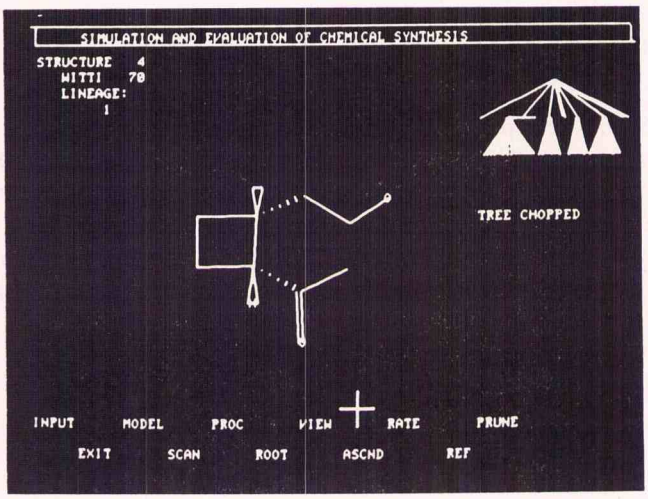


FIG. 4. Display of precursor as generated.

SECS continues this perception on higher levels. Next it estimates the shape of the molecule, because molecular shape influences reactions. Just as Gelernter constructed a model in his geometry theorem prover [18], so SECS constructs a three-dimensional ball and stick model. The model builder uses a steepest-descent minimization technique to optimize the bond lengths, bond angles, and other similar parameters [19]. From this model it then perceives distances between atoms, orientations between bonds, and the amount of crowding around each atom [20]. In addition to the 3-D ball and stick model, SECS also builds a quantum mechanical model from which it derives the electronic properties of the target molecule.

Based on the previous perceptions, SECS develops a default set of goals to be modified if desired by the chemist. These goals control selection of which chemical transforms are considered. The Chemistry module then applies appropriate transforms to the target, generating a set of precursors to the target. These structures are then examined for chemical soundness (well-formed formula) by the

Evaluation module. Surviving structures are displayed to the chemist (Fig. 4), and added to the synthesis tree. The other graphical controls exist to allow the chemist to further prune the tree and select the next structure to be processed.

3.2. Knowledge representation

Knowledge-based AI programs derive their power from the richness and depth of their knowledge base. We learned from the first synthesis program, OCSS-LHASA, in 1967 [10], that representing reactions as subroutines had several disadvantages:

(i) memory limitations,
(ii) difficult for chemist to add new reaction,
(iii) difficult to debug, and
(iv) difficult to know what knowledge was represented.

When SECS was designed in 1969, we established a clear separation between the SECS program and the chemical transforms (see Fig. 2), and devised a format for transforms and a language which would allow representing all the information we know about reactions, in a declarative form that would be readable to a chemist. The language for describing reactions (ALCHEM, A Language for Chemistry) [21] is sufficiently simple that chemists with no programming skills have had no difficulty describing reactions to the system, yet it is sufficiently powerful to be able to describe every reaction we have encountered in the past eight years. Our representation in part resembles a "production system" [22] with "situation–action" rules which are interpreted by the program. Other knowledge-based AI systems have developed similar representations for similar reasons, e.g., the Consultation Systems Project MYCIN [23], and the DENDRAL project [24]. Production rule encoding is a very natural representation for chemical reactions. Currently there are approximately 400 transforms, each representing one or more chemical reaction.

3.2.1. Chemical transforms

A transform consists of the parts shown in Fig. 5: the name is an identifier for reference only; the reference is the literature reference(s) which provided the information in the transform. The substructure can be any chemical pattern, from a single atom to a functional group, a pair of groups, or to a complex substructure including spatial orientation, special atom or bond types, rings, and wild-card atoms. Examples of substructure follow: "KETONE" is a one group substructure, "KETONE ALCOHOL PATH 3" is a substructure consisting of two groups separated by a path of 3 atoms, and "O=CCCOH/" is the exact same substructure expressed in a different way. If the substructure does not "fit" the target structure, then the transform is not applicable, however if it is a "near miss' and it has a high PRIORITY, then a subgoal may be created to apply another transform first to

make this one fit. The PRIORITY is an initial estimate of the merit of this reaction if the transform "fits", but before examining the environment of the reaction center. The CHARACTER of the transform describes the general types of structural changes the transform will perform, e.g., BREAKS RING, ALTERS GROUP, REMOVES GROUP, etc. The scope and limitations part consists of any number of statements which examine the environment of the reaction center and the rest of the molecule for features which facilitate or hinder the reaction, and then raise or lower the PRIORITY accordingly, and may "KILL" the transform if there is serious interference in the molecule. Scope and limitations comprise the bulk of information carried in the transform, sometimes extending for hundreds of statements for just one transform. Reaction conditions are represented as general classes of conditions, rather than specific catalysts or reagents. Any new reagent will necessarily fit into one of these general classes. SECS automatically checks to see if these conditions would adversely affect any other part of the molecule and if so, the PRIORITY is lowered for each adverse interaction. Functional group interchange transforms may be invoked to "protect" a group which otherwise would interfere. Finally, the manipulation statements describe which bonds to make or break in the target to produce the precursor structure, analogous to the ADD and DELETE LIST of STRIPS [25].

Name
Reference
Substructure
Priority
Character
Scope and Limitations
Conditions
Manipulations
"END"

FIG. 5. Components of an ALCHEM transform.

A transform is a completely self-contained statement of scientific facts defining a chemical reaction and all factors which affect the outcome of the reaction. It represents only the reaction center, not a complete molecule, thus describes the "essence" of the reaction directly, rather than by examples. Transforms do not reference other transforms, and are not themselves referenced by name. Consequently a transform can be added to or deleted from the library without changing the SECS program or any other transform. This also allows users to add their own personal transforms to the public ones. The system integrity is protected from user abuse in that transforms can not cause fatal errors and the Evaluation module monitors the output of the transform, deleting any structures which violate the rules of chemical bonding.

In building the knowledge base, the chemist enters transforms into a sequential file in whatever order he desires. The file is then compiled with SYNCOM [26] (see Fig. 2) into the binary pseudo-code which is later interpreted by SECS. SYNCOM also creates several direct-access directories based on the CHARACTER, SUBSTRUCTURE, and other properties of the transforms. This organization allows the program to efficiently select transforms based on the CHARACTER needed to satisfy the goal list (see Section 3.2.3).

3.2.2. ALCHEM Language

ALCHEM was designed to be easy for a chemist to read and write yet permit description of the complex context around a reaction center. A detailed sample transform is shown with comments in Appendix I.

ALCHEM is not a general procedural programming language, but is more declarative in nature. Notably missing are "GO TO"s, loops, and procedure calls. There are several types of statements, but perhaps the most important one is the one describing scope and limitations. The general format is the logical "IF-THEN-ELSE":

IF (query) THEN (statement 1) ELSE (statement 2)

example:

IF ALCOHOL IS ALPHA TO ATOM 2 THEN SUBT 20 ELSE ADD 20

where "query" is a logical expression evaluating to true or false. Either the THEN or ELSE part of the "IF-THEN-ELSE" instruction may be omitted. Some common forms of query expressions are:

⟨set 1⟩ IS (NOT) ⟨set 2⟩

⟨set 1⟩ AND ⟨set 2⟩ ARE (NOT) ⟨relation⟩

PRIORITY IS GREATER THAN 25

where ⟨relation⟩ can be CIS, TRANS, PROXIMATE, EQUIVALENT, IN SAME RING, etc. ALCHEM is based on the concept of set theory. There are three types of sets: sets of atoms, bonds, and functional groups. In comparing set 1 and set 2, ALCHEM provides automatic data type conversion so that set 1 and set 2 are of the same type of set. The two sets are then intersected. If the intersection is not null, the expression is true. In the example above, set 1 is the set of alcohol groups and set 2 is the set of groups one atom away from atom 2, where atom 2 refers to atom 2 in the substructure of the transform and at this time atom 2 is bound to a particular atom of the target molecule. There are sets predefined during perception which can be referenced in ALCHEM for all atom types, bond types, ring sizes, and many other features.

ALCHEM also has value registers for manipulating numerical values as shown in the example above: "SUBT 20" is an abbreviated form of "SUBT 20 FROM PRIORITY". Complete mathematical operations are available, enabling one to

separately adjust many different variables in a transform, e.g., yield, confidence, stereoselectivity, purity, etc. Currently, however, we lump these together into PRIORITY. Relational operations also are available as illustrated above. Compound statements are surrounded by "BEGIN" and "DONE", and may be nested to any depth:

```
IF A SECONDARY NITROGEN IS GAMMA TO ATOM 4 THEN
    BEGIN IF NITROGEN IS ALPHA TO ATOM 2
    THEN SUBT 75 FROM PRIORITY
    DONE
```

Manipulation statements modify the target structure to form the precursor:

```
DELETE ATOM 7
INVERT ATOM 4
IF ATOM 2 IS CARBON THEN ADD S TO ATOM 2
```

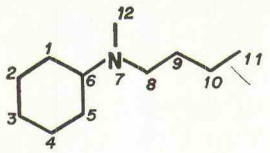
As the example shows, structural manipulation can be conditional, hence the actual structural changes performed by a transform can also be a function of the target structure. The INVERT command changes the relative spatial orientation of bonds attached to the atom. The annoted transform in Appendix I, taken from the current SECS library provides further examples of ALCHEM statements in the context of a real reaction. The SYNCOM compiler (also a FORTRAN program, see Fig. 2) provides the chemist with informative messages for syntax errors, as well as various listings and directories. The majority of the present transform library has been entered by non-programmer chemists over the past eight years ALCHEM has been in use.

3.2.3. Control, Strategies, and Goals

In SECS, strategies are the high level principles of molecular construction and modification, currently represented by program procedures [4]. Strategies are completely independent of the transform library. Many strategies are self-evident while others remain to be discovered. One of the goals of this project is to discover new strategies. Examples of skeleton oriented strategies are, "Try to cleave the target into two nearly equal fragments"; "Simplify rings of eight or more members by joining opposite sides to form two smaller rings"; and "Try to break bonds along elements of symmetry to generate identical fragments." These heuristics are stated in terms of desired skeletal changes, and they usually, but not always result in a simplification of the skeleton. Other strategies are based on functional groups, crowdedness around atoms, and other diverse structural attributes.

Applying a given applicable strategy to a particular target structure generates a list of goals. A *goal* is an instantiation of a strategy in a specific context of the target molecule. A goal may consist of logical connectives (OR, XOR, AND, NOT), functional group modifications, structural modifications, attention foci, and actions to be performed if the goal is or is not satisfied:

goal 1: (KILL IF FAIL, AND, GOAL 2, GOAL 3)
goal 2: (NIL, OR, BREAK 6–7, BREAK 7–8)
goal 3: (NIL, NOT, USE ATOMS 1,2,3,4,5).



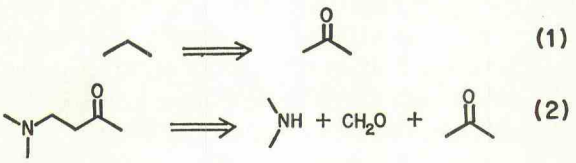
IV

Goal 1 on structure IV will force SECS to select only the transforms which either break bond 6–7 or 7–8 and which do not use atoms 1 through 5. Goal 3 is an example of negative attention focusing.

The Strategy module creates the goal structure by applying the applicable strategies and then allows the chemist to examine and modify the goals. The existing strategies are relatively simple, based primarily on graph theoretic analysis of the skeleton, but work is in progress to construct more sophisticated strategies based on symmetry which SECS perceives [17], and on the three-dimensional model which SECS builds. The best syntheses in the literature were developed from such strategies.

The goal list is then used to select which transforms are to be applied. The connection to the transform is through the CHARACTER descriptor which tells what structural modifications the transform is capable of. In essence, the goal list is the "difference" between the current target and simpler precursors, and the CHARACTER of the transform tells SECS which transforms are likely to satisfy the goal by minimizing the "difference" [8]. Those transforms having the proper CHARACTER are examined further to see if the transform is applicable and plausible. The CHARACTER descriptors are very general (e.g., BREAKS CHAIN), therefore a transform may have the right CHARACTER, but on closer examination of the manipulation instructions, may not break the chain in the desired location. Thus SECS uses the goal list to select transforms, the transforms are evaluated for plausibility, and those which are plausible are checked against the goal list again and may be killed even though plausible because they didn't satisfy the goals.

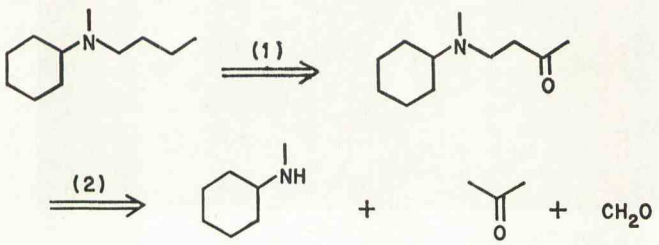
To illustrate some of the points of this section, we need to introduce two transforms:



V

Equation 1 shows reduction of a ketone written in the retro-synthetic direction (product on left, reactant on right; implication arrow is direction of reasoning). This transform has a CHARACTER consisting only of INTRODUCES GROUP. Equation 2 shows the Mannich reaction which has a CHARACTER of BREAKS CHAIN. Let the library consist only of transforms 1 and 2. Now with structure IV as the target and goal 1 as the goal list, the required CHARACTER is BREAKS CHAIN. Transform 2 is *relevant* because its CHARACTER matches, transform 1 is not (even though transform 1 is *applicable* at every secondary vertex of the graph!)

Transform 2 is then found to be inapplicable because a (C═O) is needed 3 atoms a way from the N atom. This "mismatch" is recognized and the "difference" used to spawn a new goal of (INTRODUCE C═O at 10). The CHARACTER for this goal is INTRODUCES GROUP, thus transform 1 is now both relevant (but only when applied at atom 10) and applicable. Transform 1 is applied first and then transform 2 is reevaluated and found to applicable and to satisfy the goal list because it broke bond 7–8 and did not use atoms 1 through 5.



VI

The sparteine synthesis (I) also illustrates this means-end analysis for invoking transform 1. If we had simply tried every "transform that fits," with no goal control, transform 1 would have been applied at atoms 1, 2, 3, 4, 5, 8, 9, and 10, all of which are chemically valid, but only the (C═O) at atom 10 would allow application of transform 2 to break bond 7–8. By using the goal list and this means-end analysis we avoid the generation of the 7 unfruitful structures and avoid having to process each of them further to establish that they are unproductive [27].

Work is currently underway to allow goals to have a duration of many levels in the synthesis tree, giving the program the capability for long range plans. The chemist will still be able to modify these goals and interrupt the processing if he wishes.

4. Example: Synthetic Analysis of Grandisol

As an example of the output provided by the SECS program we have selected an analysis of the insect pheromone grandisol. Grandisol is structurally the most intriguing of a four component active sex attractant for the boll weevil. It is a small molecule, but it contains many synthetically challenging features

(4-membered ring, 2 spatial centers, and a quaternary center) and it has potential economic value, consequently the problem has received the attention of many chemists, giving us a good background for comparing the analysis with SECS.

The chemist-SECS team generated a synthesis tree containing over 300 individual structures during the analysis of the grandisol problem. The chemist processed structures he recognized as being on synthetic paths reported in the literature. The resulting tree was photographed from the GT-40 screen (see Fig. 6).

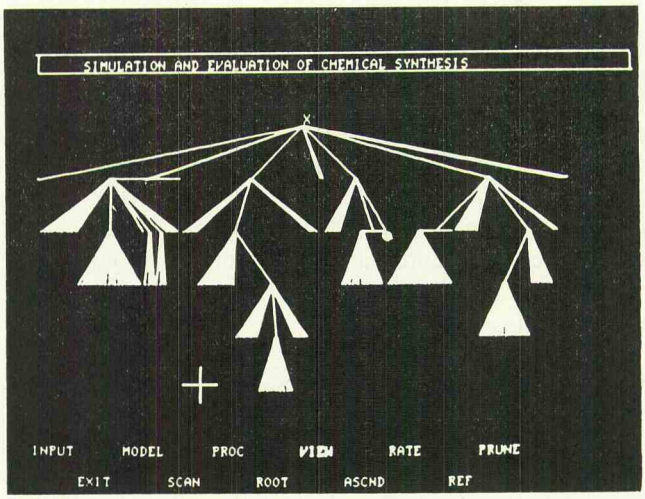


FIG. 6. Analysis tree for grandisol.

Those intermediates derived by the program which are on successful routes reported in the literature are shown in Fig. 7. Letters have been placed below the final intermediates on each path identical or analogous to a known path. Of the twelve routes known to us, the program discovered eight precisely as they appear in the literature. These syntheses include the first synthesis of grandisol by Tumlinson [28] (path A), a synthesis by Gueldner [29] (path B), dimerization based syntheses developed by both Billups [30] (path C) and Corey [31] (path D), the anionic cyclizations accomplished by Stork and Cohen [32] (path E) and Babler [33] (path F), the natural product based synthesis of Ayer and Brown [34] (path G) and the rearrangement synthesis of Golob and Wenkert [35] (path H).

Four syntheses in the literature were not explicitly found: the Zoecon synthesis by Zurfluh et al. [36], and the synthesis by Cargill and Wright [37]. SECS did produce two pathways utilizing five and six membered rings (paths I and J), which led to the same desired intermediates via similar reactions. The program was unable to design the exact original literature paths because both pathways contained reactions in which a large fragment is lost. In the retro-synthetic sense that would mean the precursor would have to be considerably more complex than the target, making it a less logical pathway.

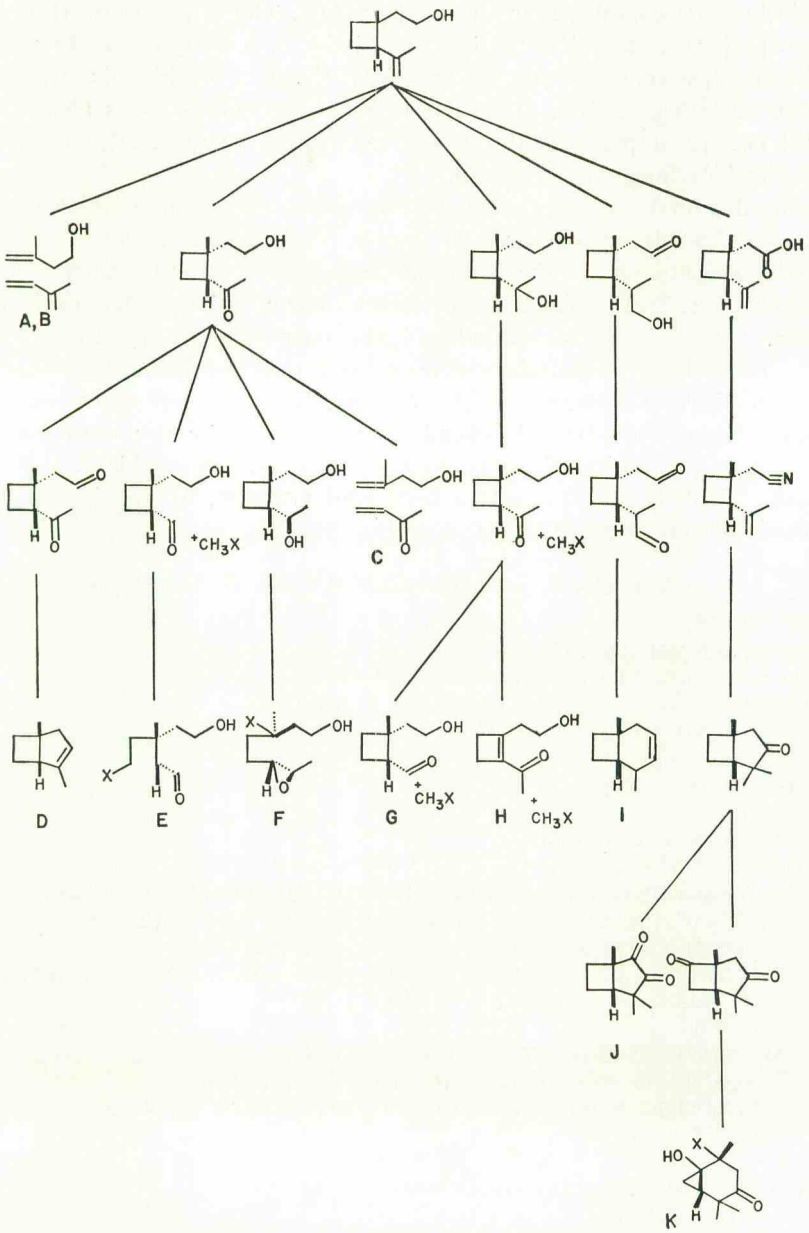


FIG. 7. Synthetic routes discovered by SECS.

The final two undiscovered paths also proceed through intermediates which are extensively more complex than the final product. The synthesis by Hobbs and Magnus [38] is an example of a direct associative planned synthesis. Over fifteen individual steps were necessary to transform the complex bicyclic structure of (–)-β-pinene into grandisol. Trost and Keely's [39] synthesis was also not discovered because it proceeds through a much more complex intermediate, the fused spiro bicycloheptanone system.

During this analysis the program also suggested many routes which have not yet, to our knowledge, appeared in the chemical literature. One of the more interesting suggestions was the stereospecific alkylation of an unsaturated ketone (structure K Fig. 7). This single step provide the correct relative spatial orientation necessary for the target compound and also constructs the quaternary carbon center. The goal of this analysis however was not to search for new syntheses, but to see if the literature syntheses would be found. The search tree generated (Fig. 6) is smaller than it would be if the chemist were searching for new syntheses. These results are representative of the strengths and weaknesses of this logic-centered approach. SECS is by no means a completed program, but versions of it are finding use several places in the chemical industry.

Appendix I. Sample ALCHEM Synthetic Transform

```
TYPE PATTERN
; PROXIMITY GUIDED EPOXIDATION
; ALCOHOL GROUP CIS TO EPOXIDE ON RING
; REF: E. COLVIN, J CHEM SOC PERKIN I 1989 (1973)
; CHEM COMM 858 (1971), HOUSE P. 305
EPOX
O—C—C—@1⟨1, 3, 2⟩/
PRIORITY 0
CHARACTER ALTERS GROUP
;  CHECK IF STEREOCHEMISTRY IS IMPORTANT
   IF STEREOCENTER IS CARBON OFFPATH THEN ; IT IS IMPORTANT
     BEGIN IF ALCOHOL IS WITHIN GAMMA TO ATOM 2 (1) THEN
       BEGIN IF BOND 1 AND (1) ARE CIS THEN ADD 50
       ELSE KILL ;EPOXIDATION WOULD HAVE WRONG STEREOCHEM
       IF (1) IS ONRING OF SIZE 5–6 THEN ADD 50
       DONE
     IF NITRILE IS EPSILON TO ATOM 2 (2) THEN
       BEGIN IF BOND 1 AND (2) ARE TRANS THEN ADD 30
       ELSE SUBT 30 ;EPOXIDE TRANS TO NITRILE IS FAVORED
       DONE
   DONE
   CONDITIONS SLIGHTLY OXIDIZING
   DELETE ATOM 1
   MAKE BOND FROM ATOM 2 TO ATOM 3
   END

COMPLETE
```

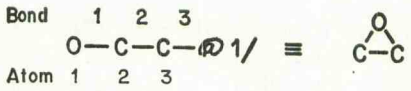
ALCHEM transform explanation

Line 1 This declares to SYNCOM the compiler that this transform is a PATTERN type transform.

Line 2–5 Text following a semicolon is a comment used for documentation. The REF: is the literature reference from which this transform was composed. The investigator can retrieve this reference while running the SECS program by touching the word REF (see Fig.4).

Line 6 This name is displayed in the upper left of the screen when the transform is applied, see Fig. 4.

Line 7 This is the "pattern" which must be present in the target structure for this transform to be *applicable*. In this case the pattern is a simple epoxide.



VII

The atoms are numbered in the pattern from left to right. The −@1 means that bond is connected back to atom 1, the oxygen. Bonds are also numbered from left to right. The ⟨1, 3, 2⟩ indicates an equivalent numbering which is a result of the symmetry involved in the transform. This symmetry is remembered so duplicate structures will not be generated.

Line 8 This is the initial priority; the scale goes from −50 to +100.

Line 9 This transform is a functional group interchange, therefore it has the CHARACTER ALTERS GROUP. SECS uses the CHARACTER to determine if the transform is *relevant* to the goals.

Line 11 This statement asks if there is a carbon stereo-center (i.e., an atom where the spatial orientation is important) present anywhere else in the molecule. If a stereo-center is found then the next statement is queried, else control transfers down to line 22.

Line 12 The structure is now examined to see if there is an alcohol group (OH) which is 3 atoms or less away from atom 2. The pattern (line 7) has been mapped onto the molecule so atom 2 is "bound" to a particular atom in the structure. If the alcohol is present it is labelled (1).

Line 13 If the bond between atoms 1 and 2 in the pattern and the alcohol are cis (on the same side of the plane of a ring to which they are both attached), then the priority of the transform is increased by 50. This is because epoxidations are guided by nearby polar groups and the transform plausibility is raised.

Line 14 If those bonds are not cis then the transform is terminated, since the stereochemistry is wrong.

Line 15 If the alcohol (1) is on a 5 or 6 membered ring then the priority is increased by 50. This reflects the favorability of the reaction when the proper intermediate can be formed.

Line 16 Marks the end of the statement beginning on line 13.

Line 18 A new set of queries are started, with the presence of a nitrile 4 atoms away from atom 2 of the original pattern. If the nitrile is present then it is labelled (2). 19–20 If bond 1 and the nitrile (2) are trans, then we add 30 to the priority, otherwise we subtract 30, indicating that this stereochemistry is the most favorable, but if the epoxide is cis the reaction has still been observed to occur although in a lower yield, thus a lower priority.

Line 21 Matches BEGIN of line 12.

Line 22 If there are any other groups present in the molecule which are sensitive to these reaction conditions, the priority will be lowered for each and the investigator will be notified by way of a message written next to the group on the display with a suggested protecting group. All of the structural queries are now complete and the final priority has been calculated.

Line 23–24 These are the commands which actually manipulate the structure. The oxygen is deleted and a new carbon–carbon bond is formed.

Line 25 The transform is ended.

Line 26 Tells the SYNCOM compiler there are no more transforms.

ACKNOWLEDGMENTS

The work reported here was supported in part by NIH grants RR00578 and RR01059, and by an allocation of the SUMEX–AIM resource at Stanford (RR00785, J. Lederberg, principal investigator). The authors also wish to thank IBM for a postdoctoral fellowship to S. K., and to thank Upjohn, Sandoz, and E. Merck for partial support.

REFERENCES

1. Gelernter, H. L., Sridharan, N. S., Hart, A. J., Yen, S. C., Fowler, F. W. and Shore, H., The discovery of organic synthetic routes by computer, *Top. Curr. Chem.* **41** (1973) 114.
2. Sridharan, N. S., *Advanced papers of the Third International Joint Conference on Artificial Intelligence*, Stanford, 1973.
3. Gelernter, H. L., Sanders, A. F., Larsen, D. L., Agarwal, K. K., Boivie, R. H., Spritzer, G. A. and Searleman, J. E., Empirical explorations of SYNCHEM, *Science* **197** (1977) 1041.
4. Wipke, W. T., Braun, H., Smith, G., Choplin, F. and Sieber, W., SECS—simulation and evaluation of chemical synthesis: strategy and planning, in *Computer-Assisted Organic Synthesis*, Wipke, W. T. and Howe, W. J. (Eds.), *ACS Symposium Series* **61** (1977) 97–127.
5. Wipke, W. T. and Dyott, T. M., Stereochemically unique naming algorithm, *J. Am. Chem. Soc.* **96** (1974) 4834–4842.
6. Wipke, W. T. and Dyott, T. M., Simulation and evaluation of chemical synthesis. Computer representation of stereochemistry, *J. Am. Chem. Soc.* **96** (1974) 4825.
7. Wipke, W. T. and Dyott, T. M., Use of ring assemblies in a ring perception algorithm, *J. Chem. Info. and Computer Sci.* **15** (1975) 140.
8. Ernst, G. and Newell, A., GPS: a case study in generality and problem solving, *ACM Monograph Series*, Academic Press, Inc., New York, 1969.
9. Sarett, L. H., Synthetic organic chemistry: new techniques and targets, presented before Synthetic Manufacturers Association, June 9, 1964.
10. Corey, E. J. and Wipke, W. T., Computer-assisted design of complex organic synthesis, *Science* **166** (1969) 178.
11. Corey, E. J., General methods for the construction of complex molecules, *Pure Appl. Chem.* **14** (1967) 19.
12. van Tamelen, E. E. and Foltz, R. L., Biogenetic type synthesis of dl-Sparteine, *J. Am. Chem. Soc.* **82** (1960) 2400.
13. Fieser, L. F. and Fieser, M., *Steroids*, Reinhold, N.Y., 1959, pp. 644–650.
14. Nilsson, N. J., *Problem-Solving Methods in Artificial Intelligence* (McGraw–Hill, NY, 1971).
15. Lehn, J. M., Design of organic complexing agents, *Bonding and Structure* **16** (1973) 1–69.
16. Wipke, W. T., Computer-assisted three-dimensional synthetic analysis, in *Computer Representation and Manipulation of Chemical Information* (J. Wiley, NY, 1974) pp. 147–174.
17. Wipke, W. T. and Braun, H., Graph-theoretical perception of molecular symmetry, submitted.

18. Gelernter, H., Realization of a geometry theorem-proving machine, in: Feigenbaum, E. and Feldman, J. (Eds.), *Computers and Thought* (McGraw–Hill, NY, 1959) pp. 134–152.
19. Wipke, W. T., Gund, P. H. and Verbalis, J., SYMIN: a general three-dimensional model builder for synthetic analysis, in preparation.
20. Wipke, W. T. and Gund, P. H., Congestion: a conformation dependent measure of steric environment. Derivation and application in stereoselective addition to unsaturated carbon, *J. Am. Chem. Soc.* **96** (1974) 299.
21. Wipke, W. T., Dyott, T. M., Still, C. and Friedland, P., ALCHEM: a language for describing chemical reactions, in preparation.
22. Davis, R. and King, J., An overview of production systems, in: Elcock and Michie (Eds.) *Machine Intelligence* **8**: *Machine Representation of Knowledge* (J. Wiley, NY. 1977).
23. Davis, R., Buchanan, B. G. and Shortliffe, E. H., Production rules as a representation for a knowledge-based consultation Program, *Artificial Intelligence* **8** (1977) 15–45.
24. Buchanan, B. G. and Lederberg, J., The Heuristic DENDRAL program for explaining empirical data, *IFIP* (1971) 179–188.
25. Fikes, R. E. and Nilsson, N. J., STRIPS: a new approach to the application of theorem proving to problem solving, *Artificial Intelligence* **2** no. 3/4 (1971).
26. Wipke, W. T. and Friedland, P., SYNCOM compiler for ALCHEM, in preparation.
27. SYNCHEM [1] applies all transforms that fit when it is expanding a node.
28. Tumlinson, J. H., Gueldner, R. C., Hardee, D. D., Thompson, A. C., Hedin, P. A. and Minyard, J. P., Identification and synthesis of four compounds comprising the boll weevil sex attractants, *J. Org. Chem.* **36** (1971) 2616–2621.
29. Gueldner, R. C., Thompson, A. C. and Hedin, P. A., Stereoselective synthesis of racemic grandisol, *J. Org. Chem.* **37** (1972) 1854–1856.
30. Billups, W. E., Cross, J. H. and Smith, C. V., A synthesis of (+) grandisol, *J. Am. Chem. Soc.* **95** (1973) 3438–3439.
31. Katzenellenbogen, J. A., Insect pheromomone syntheses: new methodology, *Science* **194** (1976) 139–148.
32. Stork, G. and Cohen, J. F., Ring size in epoxynitrile cyclization. A general synthesis of functionally substituted cyclobutanes. Application to (+)grandisol, *J. Am. Chem. Soc.* **96** (1974) 5270–5272.
33. Babler, J. H., Base promoted cyclization of a d-chloroester: application to the total synthesis of (+)grandisol, *Tetrahedron Lett.* (1975) 2045–2047.
34. Ayer, W. A. and Browne, L. M., Transformation of cartisone into racemic grandisol, *Can. J. Chem.* **52** (1974) 1352–1360.
35. Golob, N. F., thesis, Indiana University (1974).
36. Zurfluh, R., Dunham, L. L., Spain, V. L. and Siddall, J. B., Synthetic studies on insect hormones. IX. Stereoselective total synthesis of a racemic boll weevil pheromone, *J. Am. Chem. Soc.* **92** (1970) 425.
37. Cargill, R. L., and Wright, B. W., A new fragmentation reaction and its application to the synthesis of (+)grandisol, *J. Org. Chem.* **40** (1975) 120–122.
38. Hobbs, P. D. and Magnus, P. D., Synthesis of optically active grandisol, *J. Chem. Soc. Chem. Commun.* (1974) 856–858.
39. Trost, B. M. and Keeley, D. E., New synthetic methods. Secoalkylative approach to grandisol, *J. Org. Chem.* **40** (1975) 2013.

Announcement

COMPUTER GRAPHICS
Massachusetts Institute of Technology
August 7–18, 1978

This program will introduce the full repertoire of existing computer graphics facilities and offer a glimpse of the future. Topics include: History of computer graphics, basic display devices, picture generation, raster scan approaches, input devices and techniques, new display technologies, descriptive techniques, turnkey systems and applications. The program is tailored to those interested in starting or improving graphic programs in design, manufacturing, and teaching. The approach will consist of lecture, demonstration and practice. Lectures will be given by MIT faculty and staff, and invited guests. Demonstrations will include hands-on interaction with existing computer programs operational at MIT. Demonstrations will also include examples from industry. Under direction of Professors Nicholas Negroponte and Steven Gregory.

For further information, please contact:

Director of the Summer Session
Room E19–356, M.I.T.
Cambridge, MA 02139, USA

Artificial Intelligence **11** (1978), 194