

An Application of Artificial Intelligence to Organic Synthesis

Paul E. Blower, Jr., and Howard W. Whitlock, Jr.*

Contribution from the Department of Chemistry, University of Wisconsin, Madison, Wisconsin 53706. Received August 15, 1974

Abstract: A computer program has been written, employing the heuristic programming approach, which generates synthetic pathways for a class of linear organic molecules. The mechanisms which the program employs for generating and evaluating synthetic sequences are described. Examples of the program's performance are presented.

This paper describes a computer program^{1,2} which has been written to study one method for generating synthetic routes to organic molecules. Although a great amount of effort has been spent on various areas of organic synthesis, only recently^{3,4} has there been any attempt to develop general techniques for planning synthetic schemes. The most precise method for developing, specifying, and testing such techniques is in the form of an algorithm,⁵ and, thus, a computer program is a natural vehicle for studying the general problems of synthetic planning.

The discipline which is concerned with developing computer programs for solving complex, poorly understood, and formulated problems (problems which would require intelligence on the part of human beings) is a branch of computer science known as Artificial Intelligence.^{6,7} Some of the most fruitful efforts in this field have been made using the "heuristic programming" approach. A heuristic is a "rule of thumb" which suggests a method or strategy for searching for solutions to a problem. An example of a heuristic which has wide applicability to organic synthesis has been well-stated by Ireland:⁸ "If there is any key to success in planning a synthesis, it is to work the problem backwards. This is really the cardinal rule of synthesis." Heuristics do not guarantee a solution to a problem; they only provide guidance in searching for such a solution which is more efficient than examining every possibility. The heuristic approach is the one we have taken to the problem of planning a synthesis. Some of the heuristics we have employed are explicitly stated, while others are implicit in our organization of the problem.

Several other programs⁹⁻¹² are being written which address the problem of synthetic planning. Since these programs differ in various aspects from the one described herein, it is worthwhile to indicate its general nature. First, the structure of the synthetic candidate molecules ("target" molecules, to use Corey's terminology) is very restricted in form. Second, the program is not interactive; i.e., the chemist may not interrupt the program to assist in the search for synthetic precursors or in the evaluation of alternative routes. The program must make all decisions for itself. Consequently, it serves no role in assisting a practicing chemist plan a synthesis. It is strictly experimental, designed for the purpose of developing and testing mechanisms for making synthetic decisions.

The form of the molecules accepted by the program is restricted in two ways. First, only certain functional groups are recognized (these are tabulated in Table I), and no molecule may have more than five. Second, the molecules must be acyclic. Because of this latter requirement, the program focuses on the problems of assembling various arrangements of functional groups in the context of a simple carbon skeleton. The major advantage in this approach is that sim-

ple methods can be developed which will provide synthetic solutions to a number of acyclic molecules. However, since it avoids the myriad problems of constructing complex carbon skeletons, one can be certain that these methods will not be sufficient in the general case.³

Overview. Before considering the program's method of operation in some detail, it is helpful to have an overview of the process. The synthetic analysis proceeds in three distinct phases. In the first phase, the program identifies and classifies the molecule's substructures which are potential sites of synthetic interest. In the process, a model of the molecule is constructed which is the representation used in the final two phases.

In the second phase, a synthetic goal is generated. This is the proposal of a specific reaction for the synthesis of some substructure from the model. In general, however, this objective is abstract in the sense that the reaction is not directly applicable to the substructure for which it is proposed.

The final phase of the program involves designing a sequence of reactions to fit the goal to the substructure in the context of the molecule and the creation of the appropriate precursors.

The Model

The first step taken by the program in planning a synthesis is to construct a model of the molecule. This is used instead of the molecular representation in the synthetic analysis. When completed, the model will consist of sets of substructures which are significant, in the sense that they can be related to available synthetic methods. The purpose of creating this model is to obtain a simple, organized description of the molecule.

The smallest functional subunits ever considered independently by the program and the ones from which more complex substructures are built are called "primary functional groups" (PFG). The PFG's are defined in terms of the "primitive."

Definition of a Primitive. Let PR be the set of all pairs of bonded atoms (excluding hydrogen) of a molecule, M. Any $pr_i \in PR$ is a primitive if either: 1. one of the atoms of pr_i is a heteroatom; or 2. both of the atoms of pr_i are carbon and are joined by a multiple bond. The PFG's of M are certain groupings of the primitives.

Definition of a Primary Functional Group. Any nonempty subset P_i of the primitives P is a PFG if it complies with the following two rules.

1. Either P_i is a singleton or every $p_j \in P_i$ shares a common atom with at least one other $p_j \in P_i$.
2. No member of P_i shares a common atom with any member of $P - P_i$.

Every primitive is a member of exactly one PFG. The PFG's include the functional groups commonly recognized

Table I. The Functional Groups Recognized by the Program

Acetal	Aldehyde	Ester
Acetylene	Alkenyl halide	Ketal
Acid	Alkyl halide	Ketone
Acid halide	Alkynyl halide	Nitrile
Alcohol	Enol ether	Olefin

by the organic chemist such as ketones, olefins, esters, etc. For example, the PFG's of ethyl crotonate ($\text{CH}_3\text{CH}=\text{CHCO}_2\text{CH}_2\text{CH}_3$) are $\text{CH}=\text{CH}$ which is also a primitive and CO_2CH_2 which consists of the primitives $\text{C}=\text{O}$, $\text{C}-\text{O}$, and $\text{O}-\text{CH}_2$.

The program employs two classification schemes which provide useful higher order descriptions of the PFG's. First, we recognize three classes of PFG's and every PFG belongs to exactly one of them. The class to which a PFG belongs is determined by the primitives of which it is composed; these are as follows:

1. **Class 1.** Those PFG's composed of at least one primitive defined by rule 1 (above) and at least one primitive defined by rule 2, e.g., an enol ether.

2. **Class 2.** Those PFG's composed only of primitives defined by rule 1, e.g., ketones, esters, etc.

3. **Class 3.** Those PFG's composed only of primitives defined by rule 2, e.g., acetylenes.

It is worthwhile to digress for a moment in order to consider the synthetic significance of this particular classification. A question which arises with great frequency in the course of planning a synthesis is the possibility of interconverting functional groups while maintaining the same carbon skeleton. In principle, any PFG from Class 2 with n carbon neighbors can be converted into any other PFG from Class 2 with n carbon neighbors. It is useful to recognize the potential properties of these PFG's by classifying them together. To a lesser extent, this interconvertibility also applies to the PFG's of Class 3. On the other hand, the PFG's of Class 1 are not good precursors of each other, and the classification only provides negative information in this respect.

The second classification of the PFG's is that each one is given a name. These names are simply common¹³ chemical names such as aldehyde, olefin, etc. The program uses the PFG names¹⁴ in numerous ways. Many chemical properties of a PFG, such as reactivity, methods of synthesis, methods of protection, and so forth, are recognized in connection with the name. These names also provide a somewhat abstract description of a PFG. A great deal of useful chemical information is contained, for example, in the fact that a PFG is a ketone without knowing in particular that it is a hindered ketone or an α,β -unsaturated ketone.

The PFG's are used to construct two types of larger substructures. The purpose of this is to find those substructures which have several^{9a} PFG's as components (e.g., an α,β -unsaturated ketone) and for which synthetic methods are available. The first substructures are those formed by two PFG's connected by a path of zero or more unfunctionalized carbon atoms, i.e., those of the form $\text{PFG}_1(\text{C})_n\text{PFG}_2$. These substructures are called "secondary functional groups" (SFG). (We should point out that, when two smaller substructures form a single larger one, the individual smaller substructures are not part of the model.) All possible SFG's of this type are constructed except those containing PFG's in Class 1. (In the case of Class 1 PFG's, the program is only concerned with a synthesis of the PFG itself.) The number n (of unfunctionalized atoms) in the format above is restricted as follows. If both PFG's are in Class 2, then $0 \leq n \leq 3$; otherwise, $0 \leq n \leq 2$. These values were chosen because they are the only path lengths for which

synthetic methods exist. For example, if both PFG's are of Class 2, then the Michael reaction provides access to the SFG of maximum path length and, for each lesser value, several methods are available. On the other hand, no methods exist for any larger values of the path length unless alicyclic precursors are considered. Two descriptors are associated with each SFG: the class, a simple composition of the classes of the two component PFG's, and the path length. We conclude with two examples. Geranyl acetate [$\text{CH}_3\text{C}(\text{CH}_3)=\text{CHCH}_2\text{CH}_2\text{C}(\text{CH}_3)=\text{CHCH}_2\text{OAc}$] contains two SFG's: $\text{C}=\text{CH}(\text{CH}_2)_2\text{C}=\text{CH}$, of Class 33 and path length 2; and $\text{C}=\text{CHCH}_2\text{OAc}$, of Class 23 and path length 0. The molecule $\text{CH}_3\text{O}_2\text{C}(\text{CH}_2)_5\text{CN}$ contains none.

The model as defined often gives a fairly accurate representation of the molecule, at least, for the purposes of generating a synthetic goal. Situations arise, however, in which the PFG's are so intimately connected that the SFG organization is oversimplified. Suppose, for example, that the molecule contains the functional scheme: $\text{C}=\text{C}(\text{X})-\text{C}-\text{Y}$. This scheme represents a section of a molecule with all other substituents deleted and where X and Y are PFG's of Class 2 (e.g., ketones, alcohols, etc.) and may be identical. This substructure should be recognized as a single entity rather than two SFG's for several reasons. First, the whole array may match the product substructure of a known synthetic method (in this case, the Stobbe condensation). Secondly, and this is more frequently the case, the analysis applicable to either of the SFG's is significantly altered by the context in which it appears. This is illustrated by the following example. Suppose the molecule contains the substructure $\text{C}=\text{C}(\text{CO}_2\text{Me})\text{CH}_2\text{OH}$; all substituents have been deleted. In this case, many of the methods which might provide access to either of the SFG's (namely, the allylic alcohol and the α,β -unsaturated ester) are not applicable. This applies, in particular, to any synthetic method leading to the information of the olefinic bond. Therefore, we define a further structural organization called tertiary functional groups (triples). Evidently, these are most important when the PFG's are arranged in a Y pattern rather than linearly, that is, of the form, $\text{PFG}_1-\text{C}(\text{PFG}_2)-\text{PFG}_3$. The reason for this is that syntheses leading to the formation of any of the SFG's will affect atoms on the connecting path of the other SFG; it is advisable to recognize this situation explicitly.

Definition of a Triple.

1. Two SFG's form a triple if they share an atom P which satisfies the following conditions: a. P is not a member of a Class 2 PFG¹⁵; b. P has one neighbor which is a member only of the first SFG and another neighbor which is a member only of the second SFG.

2. Two SFG's form a triple if they are both of Class 23 and path length 0 and share a Class 3 PFG.

The triples are the largest substructures we will construct. Although we could clearly continue this process to more complex structural arrays, this does not seem to be profitable. For one thing, the more complex the substructures become, the more specialized and less useful they become. Secondly, and most importantly, substructures larger than triples are, in general, so complicated that it is not clear whether synthetic solutions are even available for the arbitrary instance.

In order to distinguish the substructures of the model, they will, henceforth, be called "functional groupings" (FG). At this point, our representation of the molecule consists of various arrays of functionalized atoms and the alkyl appendages, about which we have no information. In fact, except for local information (e.g., is a particular carbon atom quaternary?), alkyl groups are all but ignored throughout. It is, however, useful to know what, if any, rela-

tionship the FG's have with each other; specifically, whether two SFG's share a PFG or not. All pairs of SFG's which have a common PFG are placed on a list of "overlapping" SFG's. (This is distinct from the list of triples.) All other SFG's are placed on a list of "isolated" SFG's. This auxiliary information is simply an organizational device which has the effect of putting the FG's in context without explicitly recognizing any more complex substructures. This is the organizational model of the molecule used to generate a synthetic goal. It consists of the FG's arranged on the following lists: the triples, the overlapping SFG's, the isolated SFG's, and the isolated PFG's.

Before proceeding, we shall consider the models for two examples and, in particular, whether they provide sufficient information to generate a synthetic plan for the molecule. First, consider the following molecule: $\text{HOCH}_2\text{CH}_2\text{CH}_2\text{C}(\text{CH}_3)=\text{CHCH}_2\text{CH}_2\text{C}(\text{CH}_3)\text{C}=\text{CHCO}_2\text{H}$. The model consists of three SFG's: $\text{HOCH}_2(\text{CH}_2)_2\text{C}=\text{CH}$, $\text{C}=\text{CH}(\text{CH}_2)_2\text{C}=\text{CH}$, and $\text{C}=\text{CHCO}_2\text{H}$. As auxiliary information, we have the fact that the first and second SFG's shared an olefinic PFG as do the second and third. Although no direct methods exist for the construction of the 1,5 diene, a number of possibilities are open to either of the other two SFG's. Furthermore, the choice of approach is primarily dependent on the general form of the SFG's (as opposed to their specific structural details). The model certainly seems to provide an adequate description in this case.

As a second example, consider the molecule: $\text{CH}_2=\text{C}(\text{CH}_3)\text{CH}(\text{CH}=\text{CH}_2)\text{CH}=\text{C}(\text{CH}_3)\text{CH}_2\text{OH}$. The model consists of a triple $[\text{CH}_2=\text{CCH}(\text{CH}=\text{CH}_2)\text{CH}=\text{C}]$ and an SFG $(\text{CH}=\text{CCH}_2\text{OH})$ with auxiliary information that the triple and the SFG share an olefinic PFG. Notice, in this case, that it is important to consider the three 1,4 dienes as a triple since many of the approaches to 1,4 dienes are not applicable in this instance. In addition, the fact (as provided by the auxiliary information) that the allylic alcohol is present in this context makes any immediate attack on this substructure unattractive.

Goal Generation

In the second phase, the program generates a synthetic goal for the target molecule. As this goal, the program selects a specific reaction for the synthesis of a particular bond in the target molecule. Goal generation proceeds in two distinct phases. In the first phase, one of the FG's is selected as the target area. In the second, a method is proposed for synthesizing the targeted FG.

The objective of the goal generator is to make a carbon-carbon bond. This objective was chosen because it provides a simple means for ensuring that the program is making progress. Other goals (specifically, goals which only involve functional group modification¹⁶) are sometimes generated, but these are only in support of a carbon bond forming goal.

The input to the first phase of goal generation is the model and the output is the FG¹⁷ toward which further efforts will be directed in the second phase. The objective of this selection process is to locate the FG whose synthesis will afford maximum simplification of the molecule; the effect is simply one of focusing the program on a small, well-defined area of the molecule. The selection process is based on heuristics designed to identify the synthetic problems presented by the FG's. At this stage, however, no methods are considered for synthesizing any of the FG's.

The program recognizes certain priority goals.^{9a} These are functional groupings (e.g., acid chlorides) which the program will attempt to synthesize first. The functional groupings for which a priority goal is generated are listed in Table II, in order of decreasing priority. If the target molecule contains one of these functional groupings, this FG au-

Table II. Functional Groupings for Which a Priority Goal Is Generated

1. A PFG which is an acid halide.
2. An SFG of Class 22 and path length ≤ 1 containing a halide.
3. An SFG of Class 22 and path length 1 and whose PFGs are such that:
 - (a) one of the PFGs is an alcohol derivation.
 - (b) the carbon atom of the other PFG closest to the connecting path is sp^2 or sp hybridized.
 A β -hydroxynitrile is an example of such an SFG.
4. An SFG of Class 22 and path length 0.
5. A PFG which is an alkyl halide and not a member of any SFG defined under condition 2.
6. A PFG of Class 1.

tomatically becomes the target area regardless of, and without considering, any other details of the molecule. For example, in the molecule $\text{CH}_3\text{CH}_2\text{CH}(\text{OAc})\text{CH}_2\text{CH}_2\text{CH}_2\text{COCl}$, the acid chloride immediately becomes the target area. In the molecule, $\text{HOCH}_2\text{CH}(\text{OH})\text{CH}_2\text{C}(\text{CH}_2\text{CH}_3)(\text{CH})\text{OCH}_3)_2\text{CH}_2\text{CH}_2\text{CO}_2\text{CH}_3$, the glycol SFG is also automatically generated as the target area.

If the molecule possesses no priority functional grouping, then the target area is selected by examining the entire model. This process is performed by a group of LISP functions called "selectors". Each selector receives a pair of FG's as arguments. For this purpose, the functional groupings are divided according to their major FG classification (i.e., PFG's, SFG's, and triples) and there is a selector for each pair of these. In addition, for each pair of the larger two FG classes, there are two distinct selectors depending on whether the FG's share a PFG.

The result of applying any selector function to a pair of arguments is simply one of the arguments. If we consider the collection of selector functions to be a single function of two arguments, f_s , and the set of functional groupings of the molecule to be $\{X_1, X_2, \dots, X_n\}$, then the method of applying the selector function to the model can be described symbolically by the expression: $f_s(X_1, f_s(X_2, \dots, f_s(X_{n-1}, X_n) \dots))$. This means that the function is first applied to the functional groupings X_{n-1} and X_n . The value of the function is one of the FG's, say X_n . The second application is then $f_s(X_{n-2}, X_n)$ and so forth. The value of the final application (i.e., $f_s(X_1, X_j)$) is the value of the entire selection process. In practice the selector function is a collection of functions, and the function which is to be applied, in any instance, is itself a function of the types of its arguments (e.g., SFG and PFG). Thus, when two or more functions are being applied sequentially to three or more PG's, e.g., $f_{s_i}(X, f_{s_j}(Y, Z))$; the function f_{s_i} depends on the value of $f_{s_j}(Y, Z)$.


We shall now enumerate the heuristics upon which the selector functions are based.

(1) The functional group selected should be the one of highest reactivity, especially reactivity under nucleophilic conditions. The point of this heuristic is that, if the reactive FG is not the synthetic target, it may be necessary to protect it from the reaction conditions. This may be difficult to achieve or, at least, unnecessarily complicate the synthesis. On the other hand, the synthesis of such FG's will generally eliminate that FG from the precursor.

(2) In choosing between two reactive functional groupings, the one which is most difficult to protect should be selected. An aldehyde, for example, is very reactive toward strong nucleophiles, yet is easily disguised as an acetal. On the other hand, an ester, while less reactive to the same conditions, cannot be conveniently protected from them.

(3) A quaternary center should be introduced as early in the synthesis as possible; this is as late as possible in the antithetic direction, the direction in which the program is

Table III. The FG's for Which There Are Substructure Generators

NIL ^a	XC—C—C—CY	CX—C=C—CY
CX ^b	XC—C—C—C—CY	C=C(CX)CY
C=C ^c	C=C—C=C	C=C(CX)C—CY
		
C—C	C=C—C—C=C	C=C(CX)C—C—CY
C=C—X	C=C—C—C—C=C	C=C(CX)C—C=C
C=C—CX	CXC(CY)CZ	CX—C(CY)—C—C=C
C=C—C—CX	CXC(CY)—C—CZ	CX—C—C(CY)—C—C=C
C=C—C—C—CX	CXC(CY)C—C—CZ	C=C(—C=C)C=C
XC—CY	CX—C—C(CY)C—CZ	C=C—C(—C=C)—C=C
XC—C—CY	CX—C—C(CY)C—C—CZ	C=C(—C=C)—C—C=C

^a This substructure generator takes the central (unfunctionalized) atoms of the molecule as its arguments. ^b CX, CY, and CZ represent any Class 2 PFG. ^c This symbol represents any Class 3 PFG; in particular, it represents an acetylene as well as an olefin.

working. Since the presence of a quaternary center in an FG may greatly restrict the synthetic approaches to that FG, this center should be introduced in a precursor that is as simple as possible and then carried through the remainder of the sequence.

(4) The larger an FG is, in terms of the number of component PFG's, the more important it is as a synthetic target. This is because the size of an FG provides a direct measure of the complexity of the substructure.

(5) The FG selected should be the one lying nearest to the center of the carbon skeleton.

Although these attributes were used as heuristics in writing the selector functions, they do not exist explicitly in the program. Computationally, each selector is a sequence of nested conditional tests.¹⁸ The purpose of this testing is to elucidate just enough of the structural details of the two FG's to make a decision as to which of them represents the more important synthetic problem. In general, this process can be described as follows. A query is made concerning some structural feature of one of the two FG's. Depending on the truth (the nesting continuation) or falsehood (the sequential continuation) of the test, a second query is made. Frequently, the same test is applied to both of the FG's. The testing process continues until enough information has been gathered in order to select one of the FG's. The more similar the structural features of the two FG's, the longer the testing process before a decision can be made.

In the second stage of the goal generation process, an explicit method for synthesizing the target molecule is proposed. The input to this stage is that functional grouping which was selected in the first stage of goal generation, and it is for this FG that a synthetic solution is proposed. From a chemist's viewpoint, the goal generated consists of two items: a substructural array of atoms, some or all of the atoms which make up the FG, and the name of a reaction for synthesizing the substructure. For instance, a goal generated for a tertiary alcohol may consist of CH₂-C-OH as the substructure and the name "Grignard" as the reaction. The reaction name specifies a chemical transformation, and the substructure specifies the atoms to which this transformation is to be applied. The relationship between the two components of the goal is the following. The atoms included in the substructure are those to which bonds will be formed or broken in the reaction or those of PFG's which are required as activating groups.

In general, the objective¹⁹ of the goal generator is to find a method of synthesis for the FG which results in the formation of carbon-carbon bonds. As a result, the reaction proposed for the synthesis of a particular substructure will often not be directly applicable to the existing substructure; i.e., some modification of the functionality may be necessary before the reaction can be applied to the substructure. For example, the Grignard reaction may be proposed for the synthesis of the substructure CH₂-C-OAc.

The input to the second stage of goal generation is the FG selected in the first stage and nothing more. The second stage is only concerned with proposing a method of synthesis for this FG and, in doing so, does not consider the rest of the FG's in the molecule. One important consequence of this is that the reaction proposed may be incompatible with some other FG in the molecule. The point is that the goal generator is only responsible for proposing a synthetic solution for a single FG, not for ensuring that this goal can be realized in the context of the molecule.

The goal is a medium-range objective for two reasons: (1) the synthetic reaction may not be directly applicable to the substructure for which it is proposed; and (2) it may not be compatible with other FG's in the molecule. The processes of (1) finding precursors which can be converted into the target molecule via the proposed synthetic method and (2) establishing a reaction sequence for accomplishing this conversion are the responsibility of the final phase of the program. The goal generator simply proposes a synthetic objective which is abstract in the sense that the details of achieving this objective do not emerge.

The task of proposing a synthetic goal is performed by a collection of LISP functions called "substructure generators". The substructure generators are distinguished from each other by the form of the functional groupings with which they deal. Table III is a list of the FG's for which there are substructure generators, and each of these substructure generators is responsible for proposing a synthetic goal for any representative of the corresponding generalized FG. There are some triples (these are listed in Table IV) for which there is no explicit substructure generator. For these triples, there is a selector function which assigns one of the substructure generators of Table III to one of the component SFG's of the triple.

The operation of the substructure generators is simply one of fitting the synthetic reactions available²⁰ to the particular details of the FG. This process is based on an examination of the internal details of the FG and guided by heuristics which we shall enumerate below. First, it is appropriate to consider the internal form of a goal (i.e., the form generated and used by the program) and indicate its relationship to the rest of the program.

The internal form of a goal is a substructure and reaction pair. The reaction is specified by a name. The names for reactions employed in the program are familiar, descriptive chemical names, such as Grignard and Wittig. Internally, a reaction name corresponds to a table²¹ entry wherein information concerning the named reaction is stored. The most important piece of information cataloged under the name is a symbolic description of the permissible product substructures²² which are accessible via the reaction together with a symbolic description of the corresponding reactant substructures. The goal substructure is specified simply by marking the atoms which are to be included. The atoms

Table IV. The Triples without Explicit Substructure Generators

$C=C-C(-C=C)C-C=C$	$C=C-C-C(CX)C-C-CY$
$C=C-C(C-C=C)C-C=C$	$C=C-C(C-CY)C-CX$
$C=C(-C=C)C-C-C=C$	$CY-C-C(C-CX)C-CZ$
$C=C-C(CX)-C-CY$	$CY-C-C-C(CX)C-C-CZ$
$C=C-C(CX)C-C-CY$	

which are included in the substructure are those to which bonds are formed or broken in the reaction or those of PFG's needed for activation. The program uses the goal substructure (in the goal realization phase) to establish a correspondence between the atoms of the substructure and the product pattern stored in the dictionary under the reaction name.

The product and reactant patterns in the dictionary bear a specific relationship to each other. For example, the product pattern for the Wittig olefin synthesis can be symbolized as $C=C$ and the reactant patterns as CO , $CHBr$. Thus, the olefinic carbon of the molecule which is put into correspondence with the first symbol of the product pattern will become a carbonyl carbon. In order to orient a substructure with respect to the dictionary product pattern, the program specifies a reference atom when the substructure is generated; this is one of the atoms of the substructure and will be the program's entry point to the substructure in the final phase.

We shall now enumerate and discuss the heuristics we have employed in writing the substructure generators.

(1) Eliminate marginally stable functional groupings. These are functional groupings which may tend to decompose due to the positional arrangement of the PFG's, e.g., a β -hydroxy carbonyl compound. Frequently, this will involve selecting as a goal the formation of some bond in the path joining the PFG's.

(2) Emphasize the chemical differences in nonidentical PFG's. Consider the part structure: $R-X-C-C-Y-R'$, where X and Y are two nonidentical PFG's of Class 2 (e.g., ketones, secondary alcohols, ketals, etc.). Both X and Y can in principle be synthesized from ketones, but there is no way to distinguish between the two ketones and that is required. In this case, the formation of a carbon bond outside of the path connecting the PFG's (e.g., the $R'-Y$ bond) would have the desirable effect of emphasizing the differences between them.

(3) Select goals directed toward the elimination of quaternary centers. If suitable functionality exists, this can be accomplished directly. Otherwise, the goal should be the establishment of suitable functionality so that the problem may be solved at a later stage. For example, in the part structure $R_1C(R_2)(R_3)C\equiv CH$, a suitable goal would be to synthesize the acetylene from some precursor which would facilitate the formation of the quaternary center.

(4) Select a goal which minimizes the synthetic difficulty in the corresponding reactant substructures. In a number of instances, one can propose an elegant solution to a synthetic problem if one may presume the availability of the starting materials. The use of various vinyl copper lithium reagents in conjugate addition reactions provides an example of this. While the reaction in question may be specific and effective, the starting alkenyl halides are often difficultly accessible. Thus, the structure of the incipient reactants must be considered in selecting a synthetic goal.

(5) Make maximum use of existing functionality. It is rarely either necessary or desirable to introduce excess functionality to achieve the synthesis of an acyclic molecule. This is largely because the structural problems in an acyclic molecule are never very great. There are exceptions to this, of course. One is the need for extra activating

Table V. The Carbon-Carbon Bond Forming Reactions Used by the Program

Reaction name	No. of patterns	Reaction name	No. of patterns
Acylation	3	Hydrocyanation	5
Activated acylation	1	Knoevenagel	2
Acyloin	1	Mannich	2
Aldol condensation	1	Activated Mannich	1
Alkylation of active methylene compounds	7	Michael	4
Activated alkylation	2	Activated Michael	3
Alkylaluminum reagents	2	Ni(CO) ₄ catalyzed	5
Claisen condensation	1	Organocopper reagents	13
Claisen rearrangement	1	Reformatsky	2
Coupling of BrCH-EW	1	SN2 displacement	6
Directed aldol	1	Stobbe	1
Dithiane reagents	5	Sulfur ylids	2
Fuchs-Corey ^a	1	Wittig	5
Grignard type	4		

^a E. J. Corey and P. Fuchs, *Tetrahedron Lett.*, 3769 (1972).

groups to facilitate some reactions, e.g., alkylations. Another is the introduction of functionality in long saturated carbon chains to allow a convergent synthesis.

(6) Exploit³ the symmetry elements of the molecule. The importance of recognizing various types of symmetry is simply that these considerations may reduce the problem by half. A number of forms of synthetically important symmetry can be recognized in connection with examining the details of a functional grouping. The molecule may contain an actual element of symmetry about an atom or a bond within the FG. In addition, there are several forms of potential symmetry worth considering. Some asymmetrical molecules may be synthesized from two identical molecules, e.g., the self-condensation of an aldehyde. Others may be synthesized from a symmetrical precursor, e.g., the addition of an alkyl aluminum to a symmetrical acetylene.

(7) Attempt to form a bond closest to the center of the molecule. This will generate a converging rather than a linear synthesis.

The heuristics listed above are not explicitly available to the program but rather were used in writing the substructure generators. Each substructure generator is a sequence of nested conditional tests. These tests are designed to elucidate the particular structural details (i.e., the substitution pattern and the component PFG's). When enough information has been gathered, based on these tests and the above listed heuristics, a synthetic goal, in the form of a substructure and reaction, is proposed for the FG.

There is one extension of the basic method of operation of the substructure generators. As mentioned previously, the synthetic goal generated for a substructure may not be directly applicable to the substructure. This may happen, however, only in special cases. Specifically, the functionality in the product substructure may differ from that existing in the target FG only at one Class 2 PFG.

There are two reasons for this particular requirement. First, two Class 2 PFG's with the same number of carbon neighbors can be routinely interconverted so there is no need to generate such conversions explicitly. Within Class 3 PFG's, acetylenes are good precursors of olefins but the converse is not true. The requirement ensures that such PFG interconversions will result only from an explicit request for them. Secondly, if more than one PFG in the FG must be modified before the goal reaction can be applied to the substructure, then a reaction sequence for the several functional group modifications must be determined explicitly because the order in which the functional group changes occur may be important. For example, consider Figure 1.

The overall goal is to synthesize the carbon skeleton via the nickel carbonyl catalyzed conjugate addition of an organolithium reagent.²³ However, no reaction sequence for converting the γ -ketoester to the final acetoxy acid is implied by this goal.

In order to permit the substructure generator to specify a carbon bond forming goal and at the same time request functional group modification in support of that goal^{19b} (i.e., generate a subgoal), the program employs some additional mechanisms. In this situation, the subgoal of functional group modification becomes the goal generated by the substructure generator. The computational form of this goal is essentially the same as before. The atoms of the desired substructure are flagged and one of its atoms is selected as the reference atom (or entry point) to the substructure. The "name reaction" portion of the goal, in this case, is the name of the substructure which is to replace the flagged substructure. In the example above, " γ -ketoester" is the name employed. As with a name reaction, a substructure name corresponds to an entry point in the reaction dictionary. Information is stored under this name (e.g., a symbolic example of the named substructure) which allows the program to calculate a reaction sequence and create the corresponding precursors.

The secondary goal does not apply to the target molecule but to a hypothetical precursor of it which, at the time the goal is generated, does not exist. Referring to Figure 1, for example, the conjugate addition reaction (the secondary goal) applies to molecule B rather than the current target A. In generating a secondary goal, the substructure generator stores sufficient information that the goal can be automatically (i.e., without employing the substructure generator) generated when the proper precursor comes into existence. In this example, information concerning the conjugate addition reaction is stored so that this goal can be generated for molecule B once it is created.

The computation form of a secondary goal also consists of a substructure and reaction pair. The reaction is exactly the same as would be generated for a direct goal. The substructure employed is those atoms of the target molecule which will correspond to the atoms of the goal substructure in the precursor, and these atoms are marked with a special flag (to avoid confusion with the atoms of the subgoal). The atom which will correspond to the reference atom is then stored with the reaction name as the secondary goal.

Once the goal of FG modification is satisfied and the proper precursor has been created, the secondary goal may be generated simply by marking the corresponding substructure of the precursor. To indicate how the "corresponding substructure" is calculated, it is necessary to anticipate the goal realization process slightly. When the reaction precursors (e.g., the internal representation of molecule B in Figure 1) are being created, a mapping is established between the atoms of the precursor and those of the target molecule. Using this mapping, the program can "look-up" the corresponding substructure.

This mechanism for managing immediate and secondary goals is the only extension of the basic goal generation process employed by the program. Incidentally, this extension is an example of how a computer program can create and implement a *plan*.

Goal Realization

In the final phase of the synthetic process, the program attempts to realize the objective proposed by the goal generator. The tasks of this phase are to supply a specific reaction sequence for accomplishing the objective, to ensure that all of the nonreacting functionality is compatible with

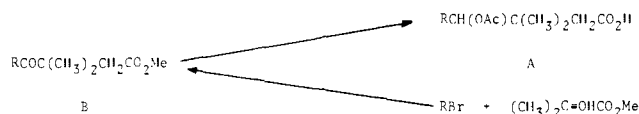


Figure 1.

each step of the sequence and to create the precursors of the reaction sequence. Accordingly, goal realization proceeds in three stages. First, an explicit reaction sequence is generated to fill in the details of applying the goal reaction to the substructure. Second, each step of this sequence is evaluated to determine its feasibility in the context of the molecule. At this stage, certain adjustments may have to be made in the reaction sequence, for example, to allow the introduction of blocking groups. Finally, the precursor structures are created. No precursors are created prior to the satisfactory evaluation of the reaction sequence. In particular, precursors corresponding to the intermediate steps in the reaction sequence are not explicitly created by the program; rather the intermediates are represented by modifications of the target molecule.

The Reaction Dictionary. The information necessary for the generation and evaluation of reaction sequences and the creation of the precursor structures is collected in a table called the "reaction dictionary". Entry points to the dictionary are through the name of the reaction, and these names correspond directly to the reaction names which are part of a goal. For example, information concerning the uses of 1,3-dithianes is cataloged under the name DITHIANE. All reactions have names (with one exception which is described in connection with the generation of a reaction sequence).

Associated with each reaction name are the product and reactant patterns of the reaction. Usually, there are several pairs of patterns with each reaction name. Each pair of patterns (i.e., product and reactant) is one instance of the reaction and is distinct from all others. For example, the monoalkylation of a dithiane reagent and the dialkylation of the reagent are two distinct reactions cataloged under the name DITHIANE.

A reaction product pattern is a list of symbols. The symbols describe the permissible atoms or part structures which may occur in the corresponding substructure and the arrangement of the symbols on the list describe the way in which the substructure atoms are bonded. The bonding conventions used within the product (and reactant) patterns are exactly the same as those commonly employed in writing acyclic organic molecules in line notation. For example, one of the product patterns under the name DITHIANE can be represented as (CO CH) and describes the following situation. A carbonyl group which may be either that of a ketone or an aldehyde bonded to a carbon atom which is unfunctionalized and may not be quaternary.

Associated with each product pattern is a reactant pattern. This is simply another list of symbols which describes the part structures and bonding in the corresponding reactants. The reactant pattern ordinarily consists of several sublists²⁴ to indicate that several (nonbonded) precursors are formed. If the product substructure is (CO CH), as in the DITHIANE example, then the reactant pattern will be (CHO) (CH Br). The parenthesizing indicates that substructural parts enclosed within a set of parentheses are part of the same reactant, and there is no bonding between the part structure in one set of parentheses and those of the other. A numbering system is used in the two pattern lists to establish an atom to atom correspondence.

A second item associated with every reaction is a feasibility function. The purpose of this function is to evaluate the

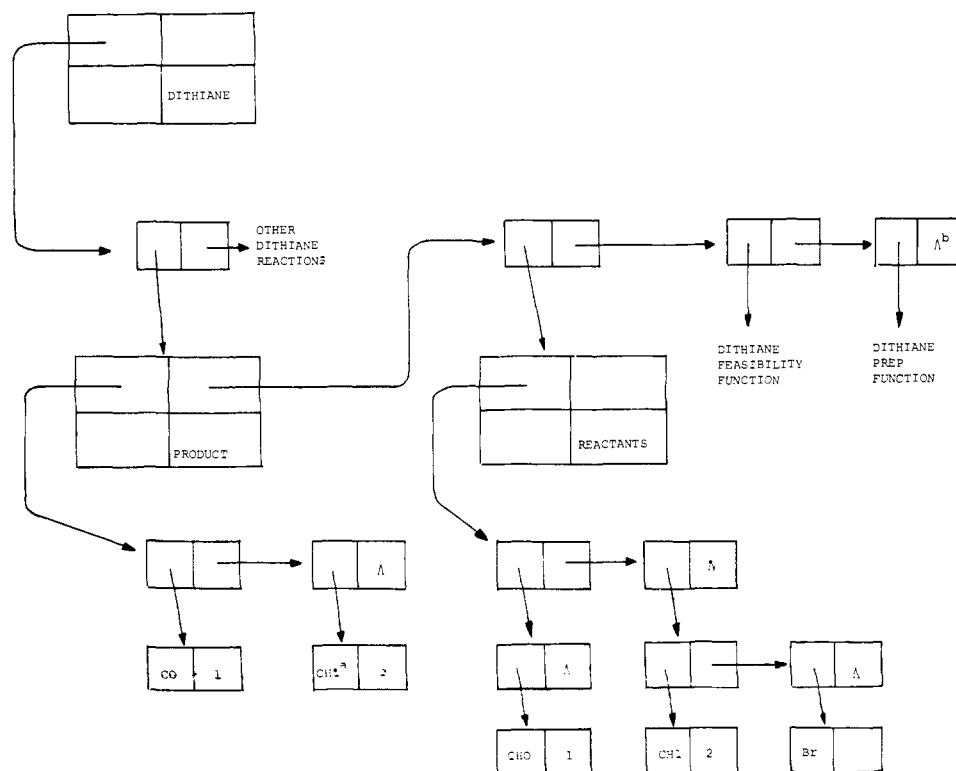


Figure 2. A diagram showing the "dithiane" entry in the reaction dictionary. The symbol CH1 represents any of the following structural units: CH, CH₂, or CH₃. A is the empty list.

feasibility of accomplishing the reaction in the context of a particular molecule. The various aspects of this function are described in the section concerning the evaluation of a reaction sequence.

One final item associated with some reaction names is called a PREP function. Most name reactions involve only a single step in going from the reactants to the product. The dithiane reaction, however, is an example of a multistep reaction; the alkylation of the dithiane and the subsequent hydrolysis must be treated separately during sequence evaluation. Thus, an additional step must be added to the reaction sequence after the alkylation step to indicate that the dithiane is hydrolyzed. The purpose of the PREP function is to adjust the reaction sequence to reflect this fact. This is only done when an isolable intermediate is involved. For example, the formation and reaction of a Grignard reagent is considered to be a single step (Figure 2).

Generation of the Reaction Sequence. The first thing the program does in response to a synthetic goal is to generate a reaction sequence. This reaction sequence provides the details for fitting the reaction part of the goal to the substructure part. In order to distinguish this initial sequence (which is generated to fit the goal reaction to the isolated substructure) from the final sequence for the substructure (which is compatible with all other functional groups in the molecule), the initial reaction sequence will be called the "base" sequence. The base sequence has one of two forms: (1) it is a carbon-carbon bond forming reaction followed by zero or more functional group interconversion applicable to one Class 2 PFG; (2) it is a sequence of functional group interconversions applicable to more than one PFG; this is a sequence generated for a subgoal. These two sequences are not generated in exactly the same way and will be dealt with separately. The process is perhaps most easily explained by examining the action taken for a particular example so we shall do this in both cases.

In the first case, the goal is a carbon-carbon bond form-

ing reaction. Suppose, for example, that the goal substructure is $\text{AcOCH}_2\text{CH}(\text{CH}_2\sim)\text{CH}_2\sim$, and the proposed reaction is the sequential alkylation of an active methylene compound; the actual name generated as the reaction portion of the goal is ACT-ALKYL (activated alkylation). The first task is to correlate the substructure with a product pattern in the dictionary under the name ACT-ALKYL. The product pattern that the substructure will match can be represented as $(\text{EW CH}(\text{CH}')\text{CH}')$; where EW is an electron-withdrawing group which may be either a ketone, acid, or nitrile, the CH is a methine unit, and CH' is any unfunctionalized carbon bearing at least one hydrogen.

The correlation of the substructure with a reaction product pattern is performed by a LISP function called the "pattern matcher". This operates by starting at the reference atom (in this case, the substructure's alcohol carbon) and attempting to make successive matches of the atoms or PFG's of the molecular substructure with symbols of the product pattern. A direct match can be made if the substructural atom or PFG is an instance of one of the permissible atoms or PFG's represented by the pattern symbol. No direct match can be made for the alcohol PFG. Since it is a Class 2 PFG, however, the pattern matcher attempts to make an indirect match via functional group interconversion.

Synthetic routes for converting one such PFG into another are constructed as follows. Associated with the name of each Class 2 PFG is a list of the other Class 2 PFG's to which it may be converted in a single step together with a function for evaluating this conversion (since each step must eventually be evaluated). By searching these lists, the pattern matcher constructs the sequence: acid \rightarrow alcohol \rightarrow ester. This provides an indirect match between the acetate of the substructure and the acid instance of the pattern symbol EW. These two steps then become the last two steps of the base sequence. The remainder of the pattern matching is direct.

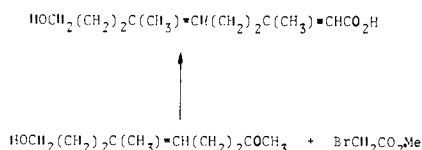


Figure 3.

At this point, the base sequence consists of three steps: (1) alkylation of a malonic ester; (2) reduction of an acid to a primary alcohol; and (3) esterification of the alcohol. There is one thing left to do. In this particular example, the goal was a multistep reaction. Since only one of the ester groups of the malonic ester appears in the product (as the acid), the other must be removed by hydrolysis and decarboxylation. The job of updating the base sequence to include the hydrolysis and decarboxylation steps is performed by the PREP function associated with the reaction ACT-ALKYL in the dictionary. Once this is done, the complete base sequence generated for this example is: (1) alkylation of a malonic ester; (2) hydrolysis of the esters and decarboxylation; (3) reduction of the acid; and (4) esterification of the alcohol.

In the second case where the goal is only a modification of some existing functionality, the process is somewhat different. Suppose the molecule has the part structures, $\text{RCH}(\text{OAc})\text{CH}_2\text{CH}_2\text{CH}_2\text{OH}$, and the goal is to synthesize this part structure from the corresponding γ -ketoester. This is an example of a request for functional group modification in support of a carbon bond forming goal. In this case, two routes (i.e., reaction sequences) are established: one for converting a ketone into a secondary acetate and another for converting an ester into a primary alcohol. The two routes are then merged into the base sequence; i.e., the two sequences are combined to form a single sequence. This merging process is primarily one of determining the order in which the various conversion steps take place. Although the merging of the two routes takes place at this stage of the program, it requires evaluating the feasibility of modifying one functional group in the presence of the other. The reason for performing this evaluation at this point is to ensure that the base sequence which emerges is internally consistent.²⁵ The way in which a reaction sequence is evaluated and the remedies for sequence failure are the subject of the next section.

Once the reaction sequence has been determined, the effect of the sequence on any PFG undergoing functional modification is information stored with the PFG. This is done using a list of pseudonyms which are the names of the reactants and products for that PFG. The first pseudonym of a PFG (i.e., the first name on the pseudonym list of the PFG) is the name of the PFG prior to any functional group modification, in other words, the name of the PFG immediately following the carbon-bond forming reaction in the synthetic direction. The second pseudonym is the name of the product of the first reaction in the sequence designated for that PFG. The last name on the list is the real name of the PFG in the target molecule.

This is the mechanism for representing the intermediate precursors of a target molecule as a modification of that molecule. All transformations that take place after the carbon-bond forming reaction (which is always the first step in the sequence in the synthetic direction) are functional group modifications. As a result, the carbon skeleton of each subsequent intermediate is the same as that of the target molecule, and the position of the PFG's in the skeleton is fixed. Thus, the pseudonyms provide a simple and efficient means for managing intermediates. This is particularly valuable when modifications of the reaction sequence are

required (e.g., the introduction of blocking groups) since the names can simply be changed rather than having to insert and/or delete entire molecular representations in the proposed sequence.

To provide a concrete example of the various aspects of goal realization, we shall trace the action for the product molecule of Figure 3. The goal generated is the synthesis of the acrylic acid substructure ($\text{HO}_2\text{C}-\text{CH}=\text{C}$) via the Wittig reaction. The product pattern with which a match will be made is $\text{EW}-\text{C}=\text{C}$; EW represents an electron-withdrawing group which, in this case, may be an ester, a nitrile, or a ketone. The pattern matcher makes an indirect match between the substructure's acid and the ester of the product pattern. The pattern matching for the remainder of the atoms of the substructure is direct, and the base sequence generated is: (1) Wittig and (2) hydrolyze ester to acid. In order to reflect the fact that the acid exists as an ester immediately following the Wittig reaction, the acid PFG is provided the list of pseudonyms (ester, acid).

Evaluation of the Base Sequence. Up to now, the program has been working within the confines of a particular functional grouping and possibly its local environment. Once a functional grouping was selected, the appropriate substructure generator chose a synthetic goal essentially regardless of the rest of the molecule. Base sequence generation was, likewise, concerned only with modification of functional groups within the goal substructure. We shall now describe the appraisal of this sequence in the context of the molecule and the methods employed for overcoming sequence failure, if it occurs.

The LISP function in charge of coordinating these activities is presented with the base sequence and the molecule. The PFG's of the molecule have been renamed (using pseudonyms) to reflect their status immediately following the carbon-bond forming reaction. If there is no carbon-bond forming reaction (i.e., the goal sequence is only a series of functional group interconversions), then the PFG's are named as they would occur in the precursor. The action taken is to evaluate each individual step of the reaction sequence. This is done by applying the feasibility function, appropriate to the reaction under consideration, to the molecule at the corresponding stage of development. (The stage of development of the molecule is described by the names of its PFG's.)

Associated with each reaction is a feasibility function. The purpose of these functions is to determine, given a target substructure, whether the desired transformation can be carried out in the context of the molecule in which it occurs. For example, suppose the desired transformation is the conversion of a carboxylic acid to a primary alcohol. This is, of course, a matter of reduction and the feasibility of accomplishing it is evaluated by the same function which handles all reductions. The question being asked is: can this carboxylic acid be reduced to the alcohol in this molecule, and not, can it be reduced with LiAlH_4 . Thus, if a number of alternative reagents are available to accomplish a particular transformation, the feasibility function is free to select among them. The evaluation is done by examining the other functional groups of the molecule, searching for any which are reactive under the same conditions (in this case, ketones, aldehydes, etc.). The function will return one of two values: a list of interfering functional groups or the conditions to be employed.

The value returned by the feasibility function determines the subsequent action. If the reaction is feasible, then the reaction conditions²⁶ are built into an output sequence, the names of the PFG's are changed to those of the product, the reaction sequence is moved to the next step, and the exercise continues. In the example of Figure 3, the feasibility

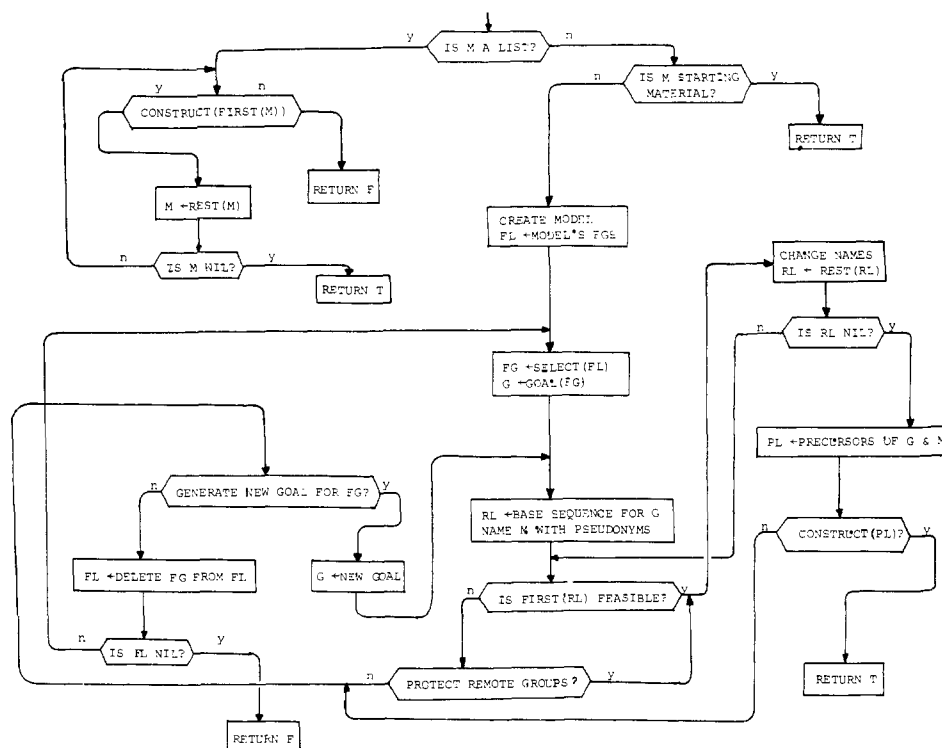


Figure 4. A flow chart for the function CONSTRUCT which makes the high level decisions. The chart uses the following functions: CONSTRUCT is a function of one argument; its action is the subject of the flow chart. Since CONSTRUCT returns a truth value, it may be used as a conditional. Any objects which CONSTRUCT creates in the course of its operation on the molecule M (such as a model of M or a list of precursors for M) are maintained as properties of M. FIRST and REST are both functions of one argument which must be a list. The value of FIRST is the first member of the list; the value of REST is the list minus the first member. For example, if $L = (A B C)$, the $FIRST(L) = A$ and $REST(L) = (B C)$. SELECT represents the selector functions and GOAL a substructure generator. The following symbols are used in the chart: M is either a molecular representation or a list of such; FL is a list of functional groupings of the model of M; FG is a synthetic goal for FG consisting of a reaction name and a substructure; RL is a list of reactions (originally it is the base sequence, but it may be modified during evaluation as described in the text); PL is a list of precursor structures for M; T and F are truth values; NIL represents the empty list.

function for the Wittig reaction determines that the alcohol PFG interferes with the transformation, and this PFG is returned as the value of the feasibility function.

Responses to Sequence Failure. If any step is determined to be unfeasible, then some functional group or grouping, other than one for which the reaction is intended, must be reactive under the conditions. We shall call such functional groups "remote" groups to distinguish them from the functional groups which are intentionally involved in the reaction. A number of recourses are available to overcome the reactivity of a remote group. Most of these involve the use of protective groups. There are several special cases, however. In some cases, a remote group which reacts under the conditions of one step is automatically regenerated later in the sequence. For example, suppose the reaction sequence being considered is: hydrolysis of a malonic ester, decarboxylation, and reesterification. Any remote methyl ester would react in the first step but, since it is regenerated in the last, this is of no consequence. Another special case is that in which the remote group may be regenerated by adding a step to the remainder of the sequence. If the remote group were an acetate, instead of the methyl ester of the previous example, then the alcohol formed in the hydrolysis step could be reconverted by an additional esterification. The program considers these two possibilities first.

Ordinarily, the method for overcoming the inconsistency of a remote group is by carrying it through part of the sequence in a protected form. We recognize two classes of protective groups. Those which are created at some prior stage of the sequence are called "blocking groups". Protective groups which appear as such in one of the precursors (e.g., prior to the carbon-bond forming reaction) are called

"latent functionality".²⁷

There are several minimal conditions which a protective group must satisfy in order to be useful. It must be stable to every step of the sequence in which it appears. At some point in the remainder of the sequence, it must be possible to convert the protective group to the remote group. If it is to be a blocking group, then it must be possible to create it at some previous point in the sequence. Whenever it becomes necessary to utilize a protective group, the program uses the heuristic—a protective group should be created as early in the sequence as possible and removed as late in the sequence as possible. This suggests that latent functionality will be given preferential consideration, and this is the case.

The information made available to the LISP functions responsible for protecting remote groups includes the reaction under consideration and the functional group to be protected. In addition, two partial sequences are provided. The first of these is that part of the reaction sequence which has already been evaluated and the second is that part which is, as yet, unevaluated. Either of these partial sequences may, of course, be empty.

The first task is to find a protective group which is stable to the reaction conditions. Associated with each functional group name is a list of potential protective groups for that functional group. For example, a ketal is associated with a ketone as a potential protective group. Each potential protective group is evaluated in turn until one is found that is stable to the reaction conditions. If a protective group fails to satisfy either this or any of the following requirements, the next candidate is evaluated. If no satisfactory protective group can be found, then the reaction sequence fails. When this happens, the program abandons the current reaction se-

```

EXAMPLE 129
*****
TARGET MOLECULE:
BRCH2C=3CCH2COOCH3
PRECURSORS ARE:
HOCH2C=3CCH2COOCH3
REACTION SEQUENCE IS:
1 HALOGENATE ALCOHOL W/ PH3P-BR2
*****

TARGET MOLECULE:
HOCH2C=3CCH2COOCH3
PRECURSORS ARE:
CH2O
THP-OCH2C=3CH
REACTION SEQUENCE IS:
1 GRIGNARD
2 ESTERIFY ALCOHOL W/ ACID-CHLORIDE
3 HYDROLYZE ACETAL W/ ACID

EXAMPLE 2
*****
TARGET MOLECULE:
CH3CH2CH2CH2C(CH3)2CH2OH
PRECURSORS ARE:
CH3COOCH(CH3)2
BRCH2CH2CH2CH3
REACTION SEQUENCE IS:
1 ALKYLATE W/ LIN(ET)2
2 REDUCE ESTER W/ LiBH4

EXAMPLE 3
*****
TARGET MOLECULE:
HOCHOCH2CH2CH2CH2COOCH2CH3
PRECURSORS ARE:
CH3C(CH3)2OCOCH2COO(CH3)3
BRCH2CH2CH2COOCH2CH3
REACTION SEQUENCE IS:
1 ALKYLATE W/ BASE
2 HYDROLYZE ESTER W/ TSOH/BENZENE

EXAMPLE 4
*****
TARGET MOLECULE:
CH3COCH2CH2CH2C(CH3)2CHO
PRECURSORS ARE:
CH3COOCH2COCH3
CH3OCH(OCH3)C(CH3)2CH2CH2BR
REACTION SEQUENCE IS:
1 ALKYLATE W/ BASE
2 HYDROLYZE ESTER W/ KOH
3 HYDROLYZE ACETAL W/ ACID

*****
TARGET MOLECULE:
BRCH2CH2C(CH3)2CH(OCH3)2
PRECURSORS ARE:
HOCH2CH2C(CH3)2CH(OCH3)2
REACTION SEQUENCE IS:
1 HALOGENATE ALCOHOL W/ PH3P-BR2

*****
TARGET MOLECULE:
HOCH2CH2C(CH3)2CH(OCH3)2
PRECURSORS ARE:
CH3COOCH2C(CH3)2CH(OCH3)2
REACTION SEQUENCE IS:
1 REDUCE ESTER W/ LiBH4

*****
TARGET MOLECULE:
CH3COOCH2C(CH3)2CH(OCH3)2
PRECURSORS ARE:
CH3CH(OH)CH3
BRCH2COOCH3
REACTION SEQUENCE IS:
1 ALKYLATE W/ Mg-ENAMINE
2 HYDROLYZE ENAMINE W/ ACID
3 METALIZE ALDEHYDE

EXAMPLE 5
*****
TARGET MOLECULE:
CH3CH(CH3)CH=CHC(OCH3)-CHCH3
PRECURSORS ARE:
CH3OC(CH3)=CHCH3
BRCH2CH(CH3)2
REACTION SEQUENCE IS:
1 WITTIG
*****

TARGET MOLECULE:
CH3CH=C(CH3)OCH3
PRECURSORS ARE:
CH3CH2COCHO
CH3OH
REACTION SEQUENCE IS:
1 TSCL / O-ALKYLATE W/ BASE

*****
TARGET MOLECULE:
CH3CH2COCHO
PRECURSORS ARE:
BRCH2COCH2CH3
REACTION SEQUENCE IS:
1 OXIDIZE ALKYL-HAL W/ DMSO

*****
TARGET MOLECULE:
BRCH2COCH2CH3
PRECURSORS ARE:
CH2N2
CLCOCH2CH3
REACTION SEQUENCE IS:
1 DISPLACE N2 W/ HBR

EXAMPLE 6
*****
TARGET MOLECULE:
CH3CH2CH2CH2CH2CH2CH2CHO
PRECURSORS ARE:
CH2O
BRCH2CH2CH2CH3
CH3OCH(OCH3)CH2CH2CH2BR
REACTION SEQUENCE IS:
1 ALKYLATE W/ DITHIANE
2 REDUCE DITHIANE W/ RANEY-NI
3 HYDROLYZE ACETAL W/ ACID

EXAMPLE 7
*****
TARGET MOLECULE:
CH3C(=CH2)CH2CH2C(CN)(CH3)CH(CN)CH2CH2CN
PRECURSORS ARE:
CH3COOCH(CN)C(CN)(CH3)CH2CH2C(CH3)=CH2
CH2=CHCN
REACTION SEQUENCE IS:
1 MICHAEL W/ BASE
2 HYDROLYZE ESTER W/ KOH

*****
TARGET MOLECULE:
CH3C(=CH2)CH2CH2C(CH3)(CN)CH(CN)COOCH3
PRECURSORS ARE:
CH3C(=CH2)CH2CH2C(CH3)=C(CN)COOCH3
REACTION SEQUENCE IS:
1 MICHAEL W/ ET3AL-HCN

*****
TARGET MOLECULE:
CH3C(=CH2)CH2CH2C(CH3)=C(CN)COOCH3
PRECURSORS ARE:
CH3COOCH2CN
CH3C(=CH2)CH2CH2COCH3
REACTION SEQUENCE IS:
1 KNOEVENAGLE

EXAMPLE 8
*****
TARGET MOLECULE:
CH3C(CH3)=CHCH2CH2C(CH2OH)-CHCH3
PRECURSORS ARE:
CH3COOCH(COCH3)CH2CH2CH=C(CH3)2
REACTION SEQUENCE IS:
1 REDUCE KETONE W/ NABH4
2 ELIMINATE ALCOHOL W/ SOCL2/PY
3 REDUCE ESTER W/ LiBH4

*****
TARGET MOLECULE:
CH3COOCH(COCH3)CH2CH2CH=C(CH3)2
PRECURSORS ARE:
CH3COOCH2COCH3
CH2(BR)CH2CH=C(CH3)2
REACTION SEQUENCE IS:
1 ALKYLATE W/ BASE

EXAMPLE 9
*****
TARGET MOLECULE:
HOCH2CH(OH)CH2C(CH2CH3)(CH(OCH3)2)CH2CH2COOCH3
PRECURSORS ARE:
CH2=CHCH2C(CH2CH3)(CH(OCH3)2)CH2CH2COOCH3
REACTION SEQUENCE IS:
1 VICINAL ADDITION W/ OSO4

*****
TARGET MOLECULE:
CH2=CHCH2C(CH2CH3)(CH(OCH3)2)CH2CH2COOCH3
PRECURSORS ARE:
CH2=CHCH2CH(OH)CH2CH3
CH2=CHCOOCH3
REACE
REACTION SEQUENCE IS:
1 MICHAEL W/ ENAMINE
2 HYDROLYZE ENAMINE W/ ACID
3 METALIZE ALDEHYDE

EXAMPLE 10
*****
TARGET MOLECULE:
HOCH2CH2CH2C(CH3)=CHCH2CH2C(CH3)=CHCO2H
PRECURSORS ARE:
CH3COOCH2BR
THP-OCH2CH2CH2C(CH3)=CHCH2CH2COOCH3
REACTION SEQUENCE IS:
1 WITTIG-W/E
2 HYDROLYZE ESTER W/ KOH
3 HYDRDLYZE ACETAL W/ H3O+

*****
TARGET MOLECULE:
THP-OCH2CH2CH2C(CH3)=CHCH2CH2COCH3
PRECURSORS ARE:
CH3COOCH2COCH3
BRCH2CH=C(CH3)CH2CH2CH2-OTHP
REACTION SEQUENCE IS:
1 ALKYLATE W/ BASE
2 HYDROLYZE ESTER W/ KOH

```

Figure 5.

quences, returns to the point where this sequence was generated, and attempts to generate a new one.

Assuming there is a protective group stable to the reaction conditions, the next step is to find a synthetic method for converting the protective group into the required group and inserting the conversion step(s) into the second sequence (of the above two). (The program finds synthetic routes for converting protective groups to the remote groups

and vice versa in the same way as was described in the generation of the base sequence.) It is necessary that the protective group be stable to the sequence up to the point that the conversion takes place and that the molecule be stable to this conversion. The later partial sequence is evaluated, accordingly, and if these requirements are met, the appropriate adjustments are made in the sequence; i.e., the conversion steps are added to the sequence at the proper points.

The final step in the process to ascertain whether the protective group can be maintained as latent functionality through the first sequence or whether blocking steps (i.e., synthesis of the blocking group) may be inserted into this partial sequence. Except in certain instances,²⁸ the possibility for latent functionality is considered first. If latent functionality is allowed, it is only necessary to ensure that the protective group is stable to each step of the first partial sequence. On the other hand, if this is to be a blocking group, then a method will have to be found for converting the functional group into its protected form and inserting these steps into the first partial sequence. If all of the requirements for a satisfactory protective group can be met, then the functional group is temporarily reassigned (via pseudonyms) as its protected form and the two partial sequences are adjusted to include any necessary conversion steps.

In the example of Figure 3, the alcohol PFG must be protected under the conditions of the Wittig reaction. Of the two partial sequences, the first (i.e., the steps already evaluated) is empty and the second (i.e., the steps remaining to be evaluated) consists only of the ester hydrolysis. The first protective group cataloged under the alcohol PFG is a tetrahydropyranyl ether. In order to evaluate the stability of the tetrahydropyranyl ether group to the conditions of the Wittig reaction, the alcohol is renamed as a THP ether using pseudonyms. Once this is done, the feasibility function for the Wittig reaction is reapplied. The protective group is determined to be stable, so the THP ether satisfies the first requirement for a protective group.

The second step in the process is to find a method for converting the THP ether into the primary alcohol and inserting the conversion steps into the reaction sequence. The conversion, of course, is simply hydrolysis. According to the heuristic concerning the use of protective groups, the program tries to insert this step as late in the sequence as possible. Although it does not make much difference in this case, the program tries to insert the hydrolysis of the THP ether as the last step of the sequence. Thus, the hydrolysis of the ester and the hydrolysis of the acetal are evaluated, respectively, and the sequence is updated to include the acetal hydrolysis.

The final step in the process is to determine whether the protective group can be maintained as a latent functionality. In this case, the question is trivial since the step for which the protective group was required is the first step. The process of protecting the alcohol is complete, and the new reaction sequence is: (1) Wittig condensation; (2) hydrolyze ester to acid; and (3) hydrolyze acetal to alcohol.

Generation of the Precursor Structures. Once the reaction sequence has been evaluated and determined to be viable, the only remaining task is the creation of the corresponding reactant structures. This is a purely mechanical operation. The precursor structures will differ from the target molecule in only two ways: that area of the target containing the goal substructure and any functional groups designated to be carried into the precursors as latent functionality. The remainder of the target is identical to the precursors. This is very nice since we can then get the nonreacting portions of the precursors simply by copying the appropriate portions of the target molecule.

The reacting portions of the precursors (i.e., the reactant substructures) are calculated using the product and reactant patterns from the reaction dictionary. There will generally be a many to one correspondence between the admissible atomic arrays of the substructure and the symbols of the product pattern. For example, if the reaction being considered is the alkylation of a malonate, then a symbol in the product pattern will indicate the carbon atom being alkyl-

ated by either a CH or a C. Accordingly, a symbol in the reactant pattern indicates that the corresponding carbon in the precursor will be either a CH₂ or a CH, respectively. The appropriate atom is then created (by copying an example) and becomes the corresponding atom in the reactant substructure. The correspondence between the atoms of the newly created reactant substructure and those of the product substructure is maintained using an association list, which is simply a LISP device for indicating a mapping.

As the atoms of the reactant substructure are being created, they are bonded appropriately. Two atoms are bonded if their corresponding symbols are adjacent in the reactant pattern. If they share a bond other than single, their symbols are separated by a bonding symbol (e.g., = indicates a double bond).

When all of the reactant substructures have been created, the nonreacting appendages of the product substructure are copied and bonded to the appropriate atoms of the reactant substructures. Finally, any functional groups designated to be latent in the precursors are replaced by their protective groups (Figure 4).

Examples of Program Performance

In this section, reproductions of the computer output for various synthetic problems are presented in Figure 5.

Acknowledgment. This work was supported in part by the National Science Foundation.

Supplementary Material Available: additional information and examples (16 pages). Ordering information is given on any current masthead page.

References and Notes

- (1) The program is written in the programming language LISP.
- (2) See paragraph at the end of this paper concerning microfilm material.
- (3) E. J. Corey, *Pure Appl. Chem.*, **14**, 19 (1967).
- (4) R. E. Ireland, "Organic Synthesis", Prentice-Hall, Englewood Cliffs, N.J., 1969.
- (5) D. E. Knuth, "Fundamental Algorithms", Addison-Wesley, Reading, Mass., 1969, pp 4-7.
- (6) M. Minsky, Ed., "Semantic Information Processing", MIT Press, Cambridge, Mass., 1968.
- (7) J. R. Slagle, "Artificial Intelligence: The Heuristic Programming Approach", McGraw-Hill, New York, N.Y., 1971.
- (8) R. E. Ireland, ref 7, p 17.
- (9) (a) E. J. Corey and W. T. Wipke, *Science*, **166**, 178 (1969); (b) E. J. Corey, *Q. Rev., Chem. Soc.*, **25**, 455 (1971); (c) E. J. Corey, W. T. Wipke, R. D. Cramer, III, and W. J. Howe, *J. Am. Chem. Soc.*, **94**, 431 (1972); (d) E. J. Corey, R. D. Cramer, III, and W. J. Howe, *ibid.*, **94**, 440 (1972).
- (10) I. Ugi and P. Gillespie, *Angew. Chem., Int. Ed. Engl.*, **10**, 915 (1971).
- (11) H. Gelernter, N. S. Sridharen, A. J. Hart, S. C. Yen, F. W. Fowler, and H. J. Shue, *Top. Curr. Chem.*, **41**, 113 (1973).
- (12) W. T. Wipke and T. M. Dyott, *J. Am. Chem. Soc.*, **96**, 299, 4825, 4834 (1974).
- (13) Esters are exceptional since we need to distinguish between those of the type RCO₂CH₃ and those of the type RCH₂OAc (cf. ref 9c). These PFG's have additional names associated with a particular reference atom.
- (14) In fact, the program cannot work with a PFG unless its name is known. The program, however, contains no trapping mechanism for dealing with molecules containing unnamable PFG's, so its behavior with such molecules is unpredictable.
- (15) This requirement simply ensures that all neighbors of P will be carbon atoms.
- (16) By functional group modification, we mean replacing one functional group by another without altering the carbon skeleton.
- (17) If the molecule has no FG's (i.e., it is an alkane), then the center (graph theoretical; cf. F. Harary, "Graph Theory", Addison-Wesley, Reading, Mass., 1969, p 35) of the carbon skeleton is used as the target area.
- (18) This decision process can be represented as a binary tree; each node is a test and the two branches leading from the node are the alternative paths to be followed based on the results of the test (e.g., the left branch is taken if the test result is a falsehood and the right branch is taken if the result is truth).
- (19) The only exception to this objective occurs with FG's containing functionality defined in Table II. In this case, the objective is simply to find a method for synthesizing the functionality whether this involves the formation of carbon-carbon bonds or not.
- (20) The synthetic carbon-carbon bond forming reactions used by the program are listed in Table V.

- (21) The program maintains a table, called the "reaction dictionary," in which the reactions available to the program are listed. A description of the information stored in this table is given in the section entitled "The Reaction Dictionary".
- (22) We shall henceforth call this the "product pattern".
- (23) E. J. Corey and L. Hegedus, *J. Am. Chem. Soc.*, **91**, 4926 (1969).
- (24) Sublisting is used in reactant patterns to indicate both distinct precursor substructures and branching. The first level of sublisting indicates distinct precursors; any deeper level (i.e., any sublist occurring within the first level) indicates branching.
- (25) It may happen that the two routes are incompatible in that there is no

- ordering of the steps in the two routes such that each PFG is stable to the reaction taking place at the other PFG.
- (26) The conditions for any reaction are embedded in the appropriate feasibility function.
- (27) D. Lednicer, *Adv. Org. Chem.*, **8**, 179 (1972).
- (28) One instance where latent functionality is not permitted is when the functional group which must be protected is one of those for which a functional group interconversion has been generated as a goal. In that case, the form in which this functional group must occur in the precursor is fixed by the goal.
- (29) The symbol = 3 represents a triple bond.

Chemistry of Superoxide Ion. I. Oxidation of 3,5-Di-*tert*-butylcatechol with KO_2

Yoshihiko Moro-oka and Christopher S. Foote*

Contribution No. 3498 from the Department of Chemistry,
University of California, Los Angeles, California 90024.
Received June 21, 1975

Abstract: The reaction between 3,5-di-*tert*-butylcatechol and KO_2 has been studied. Two main products, 3,5-di-*tert*-butyl-5-(carboxymethyl)-2-furanone (**6**) and 3,5-di-*tert*-butyl-5-(carboxyhydroxymethyl)-2-furanone (**7**), were obtained, resulting from oxidative cleavage at the 1,2- and 1,6-positions of the catechol ring. The two oxidative cleavages observed correspond to the enzymatic oxidations of catechol by pyrocatechase and metapyrocatechase, respectively.

In recent years, special attention has been directed to superoxide ion, $\text{O}_2^{\cdot-}$, as a possible active species for certain biological oxidations.¹⁻¹⁵ It has been demonstrated by various methods, especially by using superoxide dismutase¹⁻⁹ that superoxide ion is formed in several biochemical reactions involving molecular oxygen. A number of oxidations are clearly inhibited in the presence of this enzyme. In some cases, the effect of superoxide ion has been examined by adding it directly into the biochemical systems¹⁻⁴ and, in other cases, its presence has been confirmed using ESR.¹¹⁻¹⁵

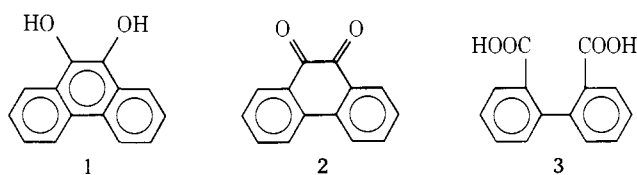
In spite of its biochemical importance, very few studies have been done on the organic chemistry of superoxide ion.¹⁶⁻²² Superoxide ion sometimes acts as oxidant⁷ but more frequently as reductant^{7,21} with organic substances; this behavior seems to be very important to its role as active species in biological systems. Thus further study of the interaction of superoxide ion with organic substances, especially with metabolic intermediates, should aid in understanding the mechanisms of biological oxidations involving molecular oxygen.

Oxidative cleavage of catechol is one of the most important reactions catalyzed by dioxygenases.²³⁻²⁶ Catechol is oxidized to *cis,cis*-muconic acid by pyrocatechase²⁵ and to *o*-hydroxymuconic semialdehyde by metapyrocatechase.²⁶ Recently, Tsuji et al. reported an oxidative cleavage of catechol to *cis,cis*-muconic acid with molecular oxygen activated by cuprous chloride.²⁷ In the present work, a preliminary study of the reaction of potassium superoxide with some catechols is reported. While some reactions were carried out with catechol itself, and some muconic acid is formed, this line of inquiry was abandoned because of extensive polymerization accompanying oxidation. 9,10-Dihydroxyphenanthrene was used as a model substrate because of the simplicity of its reaction. The use of 3,5-di-*tert*-butylcatechol instead of catechol in oxidations has been often reported, because most of the reactive ring sites are blocked by bulky groups.^{28,29} This paper will show the similarity of products obtained in the potassium superoxide oxidation of this sub-

strate with those of enzymatic oxidations and present a proposal for the oxidation mechanism.

Results

9,10-Dihydroxyphenanthrene. 9,10-Dihydroxyphenanthrene (**1**) was oxidized by potassium superoxide suspended in THF in two different ways: in one case, under 1 atm of oxygen pressure in a closed reactor and, in the other, under a nitrogen stream in an open reactor. The results are summarized in Table I. Diphenic acid (**3**) with a minor amount of 9,10-phenanthrenequinone (**2**) was obtained in every run. The reaction was quantitative, and no other products except **2** and **3** were detected. After recrystallization, the proper-



ties of the recovered products were in good agreement with those of authentic samples.

The oxidation of the quinone **2** by KO_2 to produce **3** has been reported by Le Berre and Berguer,¹⁷ and the reaction was repeated and confirmed. Oxidations of both **1** and **2** by KO_2 were accompanied by the evolution of oxygen; the observed amounts were smaller than expected from eq 1 and 2. In addition, the amount of unreacted KO_2 was larger than stoichiometric after the reactions, especially under oxygen. Thus, it is clear that molecular oxygen as well as superoxide ion took part in the oxidations.



Reaction of **2** with KO_2 proceeded more slowly in CH_3CN than in THF. A higher yield of **2** was obtained in the oxidation of **1** in CH_3CN by stopping the reaction in a