Computer-Assisted Synthetic Analysis for Complex Molecules. Methods and Procedures for Machine Generation of Synthetic Intermediates

E. J. Corey,* Richard D. Cramer III, and W. Jeffrey Howe

Contribution from the Department of Chemistry, Harvard University, Cambridge, Massachusetts 02138. *Received January 30, 1971*

Abstract: A classification of synthetic reactions is outlined which is suitable for use in a machine program to generate a tree of synthetic intermediates starting from a given target molecule. The generation of a particular intermediate by the program involves the search of appropriate data tables of synthetic processes, the search being driven by the information obtained by machine perception of the parent structure and certain basic strategies. Procedures have been developed for the evaluation of chemical interconversions which allow the effective exclusion of invalid or naive structures. The paper provides a view of the status of computer-assisted synthetic problem solving as of 1970.

The communication of chemical structural information to and from a digital computer by graphical methods has been discussed in detail in a foregoing paper,[1] as has the machine representation and perception of key features within structures,[2] as for example, functional groups and rings. This paper is concerned with the ways in which the structural information made available by the perception process can be utilized to generate a tree of chemical structures[3] which represent possible synthetic intermediates for the construction of a complex target molecule. More specifically, the following topics will be treated: (1) classification of synthetically useful reactions for computerized synthetic analysis; synthetic transforms; (2) programs for chemical manipulation of organic structure; data files; transform selection, evaluation, and application; (3) "chemical packages": two-group, one-group, and functional group interchange transforms.

The problem-solving power and scope of the present computer program (LHASA) are by no means at the level of development which appears to be attainable with presently existing concepts, programming techniques, and hardware. First of all, no stereochemical strategies or manipulations have as yet been included. Further, several major families of synthetic reactions remain to be added to the chemistry program. And, finally, there are hardware limitations imposed by the currently used computing equipment[4] which exclude the possibility of running any program which incorporates all of the important types of chemical reactions and strategies. Therefore, the approach adopted in these studies has been to add to the program major parts of certain families of synthetic processes (as defined immediately below), setting the stage for eventual inclusion of whole families, with the required data base and necessary control strategies, and also for eventual inclusion of a fairly complete collection of families. In the discussion which follows, the degree of implementation of each area of study will be cited.

A variety of rational schemes for creating families of synthetic reactions already exists. However, most of these depend on properties of the reactants,[5] and as such they are irrelevant to a computer program which analyzes the features of a target or product molecule in order to generate appropriate starting materials. One very general treatment of synthetic reactions involves classification on the basis of the structural changes which they induce, for example, (1) interconversion, removal, or addition of functional groups, (2) extension of atomic chains or appendages, (3) generation of atomic rings, (4) rearrangement of chain or ring members, (5) cleavages of chains or rings, (6) creation of stereocenters and stereorelationships, and (7) activation or deactivation of functional groups. These different types of structural change are of varying intrinsic effectiveness. Some are *directly* useful in that they produce certain molecular features extant in the target molecule, while others are *indirectly* fruitful, since they function only to enable the successful operation of reactions which directly contribute to the final architecture. Further, any codification of synthetic reactions based on such generalized structural changes alone is complicated by the fact that many reactions effectively perform *several* fundamental types of structural change in *one step*. For example, a Diels–Alder reaction may produce one or more rings, a number of functional groups (including C=C) in a certain relationship, stereocenters in a specific relationship, and even appendage groups. In contrast, certain other reactions serve only to modify some incorrect or inappropriate unit which was the unavoidable result of a previous step. As a result of this diversity and overlap, the classification of synthetic reactions solely in terms of an associated general structural change is not in itself sufficiently powerful or precise to be of value.[6]

(1) E. J. Corey, W. T. Wipke, R. D. Cramer, III, and W. J. Howe, *J. Amer. Chem. Soc.*, 94, 421 (1972).
(2) E. J. Corey, W. T. Wipke, R. D. Cramer, III, and W. J. Howe, *ibid.*, 94, 431 (1972).
(3) E. J. Corey and W. T. Wipke, *Science*, 166, 178 (1969).
(4) The PDP-1 computer employed so far in this work is a single address machine with a core capacity of 24,576 18-bit words and a basic cycle time of 5 μsec. A magnetic drum (access time 17 msec, 2.25×10^6 bits) which allows swapping extends the usable memory of the PDP-1. This modest facility, vintage *ca.* 1960, though inexpensive to use for program experimentation and development, falls far short of the requirements of a program which would be both chemically complete and sophisticated.
(5) For example, classifications based on type of reagent, reaction mechanism, and structural characteristics of a starting material are in common use.
(6) The consideration of the relationship between structural operations and synthetic reactions is quite valuable as a means of assessing the merit or efficiency of a synthetic scheme.

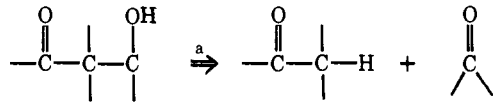
However, by the use of additional classifying elements which relate key structural features of the target molecule (synthons)[7] with specific structural features that might be created by the operation of particular synthetic reactions, a rather precise and useful system of classification can be derived.

The resulting organization of synthetic chemistry is based on the reverse formulation of synthetic reactions. This organization, with its reverse-synthetic or retrosynthetic focus, must be expressed explicitly and consistently in order to avoid intolerable confusion, and this has necessitated the introduction of some new nomenclature. In this article we shall use the term *antithetic* to be the opposite of *synthetic* and synonymous with *reverse-* or *retro-synthetic;* that is, the term *antithetic* is used to designate a direction of analysis or flow opposite to the synthetic direction. Further, we shall denote structural changes in the antithetic direction involving intermediates in the synthetic tree as *transforms* (in contrast to synthetic *reactions* for operations in the synthetic direction). A double-lined arrow will be used to indicate the direction associated with a transform in contrast to a single-lined arrow, which will be used in the conventional way to indicate the direction of a synthetic reaction.

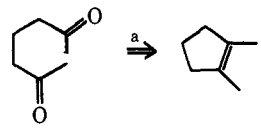
Several important classes of antithetic transforms together with their salient characteristics and an example of each are given in the following outline.

I. Transforms[8] which in the parent require *a pair of functional groups* connected by some atom bond path. These may be further subdivided depending on whether the transform

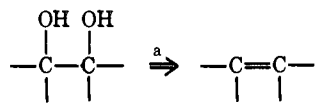
(i) disconnects the path between functional groups



(ii) forms a new path



(iii) modifies functionality without altering path

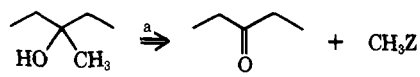


(iv) effects rearrangement, in essence overall disconnection and formation of two or more bonds.

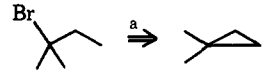
In each instance there is the possibility of further subdivision with reference to stereochemical relationships.

II. Transforms of a *single functional group* resulting

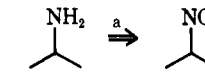
(i) in cleavage along a path to that functional group



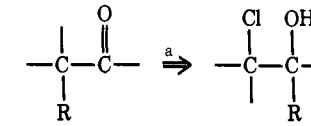
(ii) generation of a ring



(iii) change (interconversion) of functionality only



(iv) rearrangement

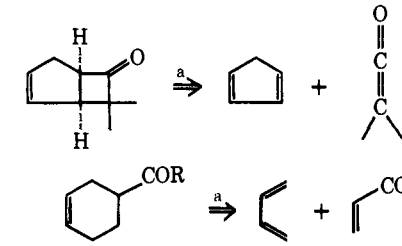


Clearly, subclasses of i–iv with reference to stereorelationships must also be added.

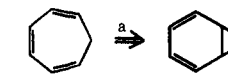
Categories II(i) and II(iii) are probably the most important transforms of the one-group type. In later sections these are referred to as disconnective one-group transforms and as functional group interchange (FGI) (which can be either disconnective or nondisconnective). Although nondisconnective FGI transforms do not reduce molecular complexity, they are important because they set the stage (as subgoals) for the operation of simplifying transforms.

III. Transforms which depend critically on ring size, ring size and functionality, or these features combined with stereochemistry. These may impose the following types of change on the parent structure:

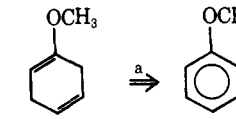
(i) ring scission



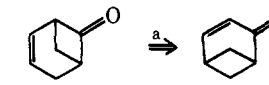
(ii) ring formation



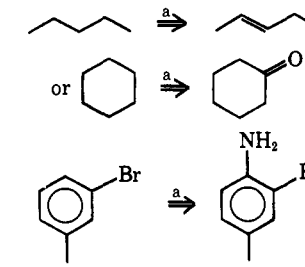
(iii) modification of funtional groups



(iv) rearrangement

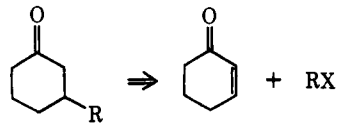


IV. Transforms involving the addition of functionality



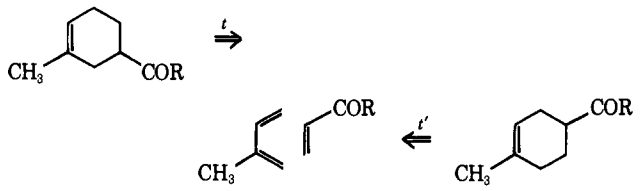
(7) E. J. Corey, *Pure Appl. Chem.*, 14, 19 (1967).
(8) Antithetic (*a*) direction.

Other types of transforms which can be defined clearly and which serve to allow antithetic analysis by correlating some structural feature in the target molecule (even an unconventional feature) with an appropriate synthetic process may be added to the above described categories. For instance, a class which might be termed "generalized pair" transforms can be defined which involves within the target structure (1) a *functional group* and (2) some other structural feature (*e.g.*, appendage, ring fusion) in a particular relationship. The conjugate addition transform is an illustration (functional group + appendage with a two-bond path):



The rings transforms are exceedingly important as a class, and specific cases within this group are among the most powerful synthetic processes known. Such transforms may not be applicable to a particular target structure but may become so after modification of the target by the use of one or more transforms of other types. The intermediates generated by these later transforms may be regarded as subgoals, the accessibility of which allows a "goal" structure which can be disconnected by a rings transform. The search for subgoals in connection with the effective utilization of rings transforms is both more critical and more complex than is the case even for pair transforms.

One obvious difficulty created by using this transform classification scheme is that the generation of a "son" from some "parent" structure by a standard or normal mechanistic transform may not be a legitimate transform in the event that the "son" is capable of reaction by different pathways of the same mechanistic type. For example, the transform *t* is illegitimate, whereas *t'* is acceptable. Complications such as this, while consider-

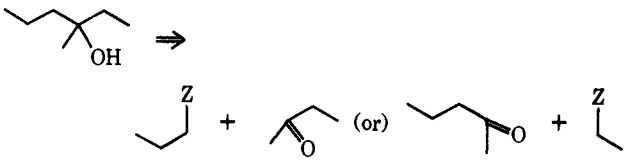


able, do not preclude the application of chemical tables based on antithetic analysis of reaction product rather than the structure of synthetic starting material(s). The data base on which the chemistry part of the LHASA program has been constructed is formulated strictly in terms of "transforms" rather than reaction processes in the synthetic direction.

Presently three of these families of transforms can be applied to a target molecule by the computer: "two-group transforms," comprising the class I just discussed; "one-group transforms," comprising all of class II except transforms in subclass iii; and "functional group interchange" (FGI), which includes normal FGI, subclass II(iii) and disconnective FGI, subclass II(i). These families are applied to a target molecule in quite different ways.

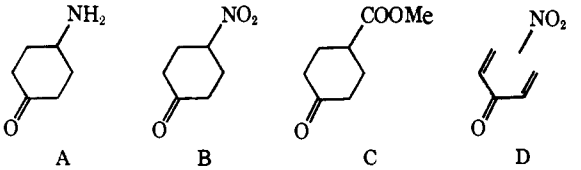
Application of two-group transforms is guided by the simple rule that any two-group transform which corresponds to a chemically reasonable synthesis of the target molecule will be performed. This simple technique is satisfactory, since opportunities for the operation of any given transform with its requirements of specified groups and path (and eventually stereochemistry) are relatively infrequent.

Chemically possible one-group transforms, on the other hand, are much more numerous because there are usually a considerable number of synthetic reactions which can be used to produce a single functional group. In addition, a given transform can usually be applied to a given group in several different ways. For instance, the Grignard synthesis of an alcohol used to illustrate subclass II(i) could have been applied to the target in two other ways:



However, only a small fraction of these chemically possible one-group transforms normally lead to useful synthetic intermediates. In order to screen out uninteresting one-group transforms, input of an additional structural element is required. For example, the collection of "topologically important" bonds is considered by the one-group programs. One particular one-group transform may be required to break an appendage bond, another, a bond exo to one ring and endo to a different ring. Expressed differently, one-group transforms are applied only when there is a specific relationship between an appendage and a group, a ring and a group, a bond designated by the chemist as "strategic" and a group, and so forth.

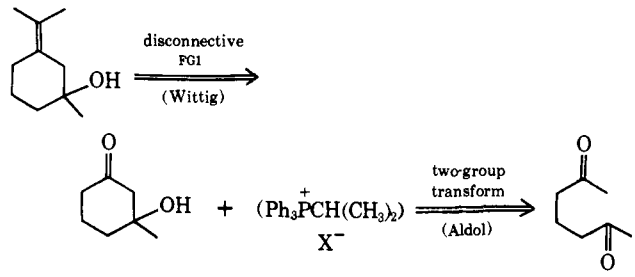
Unlike one- and two-group transforms, functional group interchange (FGI) does not usually lead to intermediates of structure simpler than the target. Consequently, FGI is applied *only* in response to a specific request by some other part of the program, such as "Can this alcohol be transformed into (Could this alcohol have been made from--?) a double bond having a particular orientation?" At present, this kind of request, or "subgoal," is generated by the program responsible for choosing two-group transforms whenever the following situation exists: replacement of a functional group by one of a different type will enable operation of an important but presently inoperable two-group transform. For example, no synthetically useful relationships exist between the groups in structure A.



But if the amine group were a powerful electron-withdrawing group, as in structures B or C, the target molecule could be disconnected using a Michael addition to give a structure such as D. The program which chooses two-group transforms will recognize this situation when examining structure A and generate a request for conversion of the amino group into some kind of electron-withdrawing group.

The operation of FGI is not limited to functionality changes that leave the carbon skeleton intact (those in

class II(iii)) but extends to disconnective processes as well. In fact, disconnective FGI is preferred to simple functionality exchange because it not only satisfies the external request but also simplifies the target in its operation. This is illustrated by the following sequence.



In subsequent versions of the program, chemical situations other than inoperable two-group transforms will generate FGI requests. For example, the existence of a group in the target which because of its chemical sensitivity interferes with the operation of an important transform will invoke a "protective" FGI request, to be satisfied by any new group unreactive toward the required chemical environment. In addition, the goal of breaking a bond designated or perceived as strategic or of creating a certain stereorelationship may require transforms of the FGI type. The selection of transforms of the functional group addition (FGA) class (class IV) requires the highest degree of direction and control. The use of such transforms, as in the case of FGI, will be allowed only if they satisfy subgoals which are created by the various strategies available to the program.

Basic Structure of Chemical Manipulation Programs. The analysis of a synthetic problem can in principle be performed in two ways with regard to the mode of utilization of the available data base. These complementary modes may be described as "data driven" and "target driven," accordingly as (1) the data base, for instance a list of names of transforms, is scanned item by item, each item in turn being compared with the target molecule, or (2) the collection of key structural units in the target, such as functional groups and rings, is examined item by item, each item being compared with the data base.

The relative efficiency of these approaches depends mainly on the size of the data base, the complexity (problem data content) of the target, and the organization of each collection of data. The data-driven approach is more effective when the "problem data" are simple and accessible, and the problem-solving information is much more complex. The program described here makes use of an extensive but highly organized data base which is separate from the program instructions and allows transforms to be chosen by a target-driven process. Detailed evaluation of a particular transform is data-driven. This procedure, developed in detail here, is closely analogous to that actually used by a chemist when confronted with a complex problem.

Each of the three families of transforms, one-group, two-group, and functional group interchange, is carried out by a separate division of the program. Each of these divisions is in turn divided into three functional subdivisions, as follows:

(a) A data table of all possible transforms belonging to that family. Each entry in the table contains all the information necessary to evaluate the merit of some particular transform as applied to *any* target molecule. Included are the functional group requirements and a list of any other structural features that significantly affect the probability that the transform would be successful as a synthetic laboratory reaction. These data tables are written in a new programming language designed to be convenient and simple for a chemist to understand, modify, or write.

(b) A program which compares structural features of the target molecule with a particular data base, looking for opportunities to perform a transform. Promising transforms undergo careful evaluation, and those which receive a sufficiently high rating are listed for subsequent execution. If an otherwise promising transform is blocked by the absence of some structural feature in the target, this program may also generate a request for some other program to remove the obstacle. Such requests are placed on a separate list.

(c) A program which carries out any transform requested by the appropriate program of type b. (There is insufficient computer memory for programs of type b and type c to be in core simultaneously; this is the primary reason for these functions to be separate.[9])

For convenience in subsequent references, each program subdivision will be designated by the name which happens to be used for it within the program.

	Data table	Transform choosing	Transform performing
Two-group	PAIRTB	GPAIR	CHEM6
One-group	SINGTB	GSING	CHEM6B
FGI	FGITAB	FGI	FGICHEM

There is also an executive program which oversees the generation of new intermediates and the transfer of control to the appropriate transform-choosing and transform-performing subdivisions in response to internal (computer) or external (chemist) requests. This part of the program has evolved as a result of the changing chemical capabilities of different versions of the program. We will defer a detailed description of this part, since its final form will not be generated until a complete repertoire containing all of the different types of transforms has been included in the program.

One-Group and Two-Group Transforms—Data Files, Transform Selection, and Structural Manipulation. Data Files. The vocabulary of one- and two-group transforms available as of late 1970 for problem solving by LHASA corresponds to over 250 types of structural change. The number of synthetic reactions represented is considerably greater, since at this stage of development there is only a minimum of input to reflect the fact that any of a number of reagents may exist for effecting a given synthetic reaction (*e.g.*, olefin to 1,2-glycol). Furthermore, even if there is more than a single mechanism for some reaction (*e.g.*, solvolytic displacement of an alkyl halide), the structural change is handled by a single transform entry. Those of the one-group reactions which are disconnective appear in SINGTB, and those which are nondisconnective appear in FGITAB. These tables include a total of about 130 transforms. Approximately 120 two-group trans-

(9) Process b takes from 5 to 60 sec, depending on the complexity of the target structure and whether or not additional requests to remove blocks are being generated. Process c, together with evaluation of each new structure, takes 3–10 sec per structure. Each new structure is displayed for the chemist while evaluation is taking place.

Table I. Entry in PAIRTB for the Aldol Transform

```
1  ...D—C—C—W ⟹ D=C + C—W
2  ...aldol
3  m8:  ..dec 23
4       ..alcohol; ..wgroup; ..at2 tryfgi dec 70
5       ..bond1 broken
6       ..subt dec 100 if grp2; ..is primary amide
7       ..kill if halide; ..within betato cbn2
8       ..addt dec 30 if grp2; ..is nitro
9       ..addt dec 100 if wgroup; ..cbn2
10      ..addt dec 40 if aryl; ..cbn2
11      ..addt dec 50 if nohydrogenon; ..alphato cbn1 offpath
12      ..addt dec 30 if grp2; ..is aldehyde
13        ..and begin
14        ..subt dec 70 if ringwith; ..bond1
15          ..and begin
16          ..kill if quaternary; ..cbn1
17          ..done
18        ..done
19        ..either
20      ..addt dec 80 if ringwith; ..bond1 ofsize; ..ring5 dec 6
21        ..and begin
22        ..if grp2; ..is ketone
23          ..then begin
24            ..either
25            ..if path; ..cbn1 tocbn3 allin; ..ring5 dec 6
26              ..then begin
27              ..subt dec 50 if hydrogenon; ..alphato cbn1 offring offpath
28              ..addt dec 50 if ringwith; ..alphato cnb1 offring offpath
29              ..kill if chminus; ..alphato cbn1 offring offpath betrthan; ..cbn2
30              ..subt dec 40 if wgroup; ..alphato cbn1 offpath onring
31              ..done
32            ..orelse
33              ..subt dec 30 if hydrogenon; ..alphato cbn1 onring offpath
34              ..kill if chminus; ..alphato cbn1 onring offpath betrthan; ..cbn2
35              ..subt dec 30 if wgroup; ..alphato cbn1 offring offpath
36            ..done
37          ..done
38        ..subt dec 20 if ringwith; ..bond1 ofsize; ..ring5 dec 5
39        ..subt dec 40 if wgroup; ..betato hetatm ingrp2 offpath
40        ..done
41      ..orelse
42        ..kill if ringwith; ..bond1 ofsize; ..otherthan ring5 dec 6
43        ..kill if wgroup; ..betato hetatm ingrp2 offpath
44        ..if grp2; ..is ketone
45          ..then begin
46          ..kill if wgroup; ..alphato cbn1 offpath
47          ..done
48        ..done
49      ..subt dec 100 if grp2; ..is cyano
50      ..subt dec 100 if grp2; ..is ester
51      ..condn alk orelse; ..subt dec 20 if condn acidic
52      ..subt dec 60 if wgroup; ..alphato cbn2 offpath
53      ..subt dec 60 if hetero; ..within alphato cbn2
54      ..subt dec 20 if dbond; ..betato cbn2
55      ..subt dec 20 foreach rgp; ..cbn2
56      ..subt dec 30 foreach rgp; ..cbn1
57      ..subt dec 100 if grp2; ..is vinylw
```

forms which are contained in PAIRTB are summarized in Appendix I.

An ideal language for presenting data about chemical structures to a computer should have several important characteristics: (1) be easily comprehensible to a chemist having only an elementary knowledge of computer programming, (2) be very flexible, so that extensive changes can be made to the computer's chemical "knowledge" without requiring major revisions of other parts of the program, and (3) be efficiently handled by the computer. On one hand, memory limitations demand a small and general vocabulary, but on the other hand, time limitations require that the data be in a form readily comparable with whatever structural representations are used.

To illustrate the kind of information that the data tables must contain, the structural requirements for a familiar and important two-group transform corresponding to the generalized "aldol" reaction will be discussed in considerable detail. The actual aldol table entry will then be used to present important elements of the vocabulary and grammar for the present "language." Finally, the representation of the data tables within the computer will be described.

The structural requirements for a successful synthetic reaction can be established by the examination of each intermediate and each mechanistic step in turn for reactions which may cause diversion from the desired path. The mechanism of the generalized aldol transform is well understood.

$$HO-C-C-W \Rightarrow {}^{-}O-C-C-W \quad \text{Step 1}$$
$$^{-}O-C-C-W \Rightarrow O{=}C + {}^{-}C-W \quad \text{Step 2}$$
$$^{-}C-W \Rightarrow H-C-W \quad \text{Step 3}$$

Step 1, which in the synthetic direction is protonation of an oxyanion, is so facile that no interferences are anticipated. However step 2, the addition of a nucleophilically activated carbon (here a carbanion) to some specified carbonyl carbon, will fail whenever the nucleophilic carbon can react more easily in some other fashion. A carbanion could eliminate an α or β positioned anionic leaving group to give, respectively, a carbene or a double bond; displace a leaving group five or six atoms away; shift to a γ position if there is a β,γ double bond; or attack some more accessible or reactive carbonyl group. If too many atoms are attached, the carbanion may be unreactive. On the other hand, this addition will be very favorable whenever the reacting atoms are constrained in a reactive conformation by some other connecting chain, as is usually true whenever the addition step closes a ring of five or six members. Step 3, the activation of a particular carbon atom as a nucleophile, must also be plausible. Many electron-withdrawing groups, such as cyano and ester, are insufficiently activating. Electron-withdrawing groups elsewhere in the molecule activate the formation of competing nucleophilic centers. Further, the reagent necessary for activation may react in other, undesirable ways with the target molecule. On the other hand, attachment of other electron-withdrawing groups to the potentially nucleophilic carbon strongly favors the desired transform.

Once the structural factors which affect the course of a synthetic reaction have been determined, their relative effects must be evaluated. For instance, the activating effect of two electron-withdrawing groups on one carbon may obviate interference by an electron-withdrawing group elsewhere in the molecule, a fact which must be reflected in the evaluation procedure. This kind of reasoning has led to a technique for evaluation in which a transform such as the aldol is given a basic numerical value, or "rating," which is revised upward or downward by varying amounts in the presence of specified structural features. In order to be considered plausible, a transform must attain a rating greater than 0 (usually less than 100) on an arbitrarily chosen scale. However, since time and memory limitations prevent mention in the tables of all the structural features that might make feasible an apparently bad transform, structures resulting from transforms with ratings between 0 and -50 are shown for the chemist to evaluate. Transforms with ratings below -50 are considered no further by the machine. The rating of each transform is displayed along with the resulting new structure to assist the chemist in choosing a new branch of the synthetic tree for analysis.

The data used by the computer for evaluation of an aldol reaction as given in Table I will now be considered. Lines 1 and 2 are "comments" (indicated by the symbol "..."), statements ignored by the computer that give the chemist a schematic representation of the transform and a verbal description of the process (for the synthetic direction). The entry as perceived by the computer begins with the internal transform name, *dec 23*, which addresses the instructions used to change the present target structure by the aldol transform into a new intermediate (*dec* informs the computer that the number following is decimal).

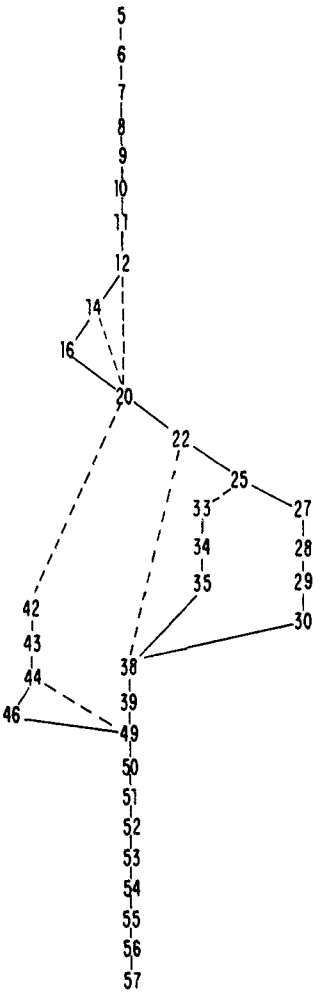


Figure 1. Computer decisions during the evaluation of qualifiers for the aldol transform. The numbers above agree with the line numbers in Table I, and are read from top to bottom. Wherever two lines descend from the same number, the solid line indicates the decision made if the corresponding qualifier is "true" for the current target structure; the dotted line, the decision if the qualifier is false.

On the next line are the basic structural requirements for this transform: the two group names (*alcohol* and *wgroup*), the number of carbon atoms that must separate the groups (*at2*), permission for this entry to request the proper functional group whenever a half-match occurs (*tryfgi*), and the initial transform rating (*dec 70*). Group names appearing in a table may be the familiar specific names, such as *alcohol* here, or general electronic descriptors given to a collection of specific groups having some reactive property in common.[2] In this case *wgroup* includes any group sufficiently electron withdrawing to stabilize an anion on an adjacent carbon. The exact "number of carbon atoms" separating two groups will in general depend on the conventions used to number atoms. Two conventions are used in LHASA data tables. For *specific* groups numbering begins with the first carbon atom encountered when proceeding along the path, starting with the first group named in the table entry. (The schematic representation in the comment statement always reflects this direction.) For *general* groups numbering begins with that carbon atom to which the general group is "attached."

Table II. Some Properties of "Optypes" as Used in Standard Qualifiers

Class of optype	Example of optype class	Illustrative qual. (line no.)	Modifier phrase required by optype
Name of functional group[b]	*halide, dbond*	7, 54	Location-describing[a]
General class of functional group	*wgroup*	52	Location-describing[a]
Substituent types	*aryl, nohydrogenon, hetero*	10, 11 53	Location-describing[a]
Number of atoms attached	*quaternary*	16	Location-describing[a]
Ring including at least one of the bonds on the connecting path	*ringwith*	20	Location-describing[a] and, optionally, an inclusive or exclusive ring size restriction; *e.g., ofsize; ..ring5 dec 7* or *ofsize; ..otherthan ring3 dec 3*
Ring minimally including several path bonds	*path*	25	Special location description naming path atoms (*e.g., cbn1 tocbn6*) and the size of a ring which must or must not include these atoms (*e.g., allin; ..ring5 dec 6* or *notallin; ..ring6 up*)
Entry group(s)	*grp2*	8	A functional group type and the word *is;* (*e.g., is nitro*)
Positional selector[c]	*chminus*	29	Two location-describing[a] phrases connected by *betrthan*

[a] Examples of modifier phrases used for describing locations are given in Table III. [b] The functional group(s) immediately involved in the currently proposed transform are excluded with this class of qualifier optype. For example, line 9 of the aldol transform entry specifies a rating increase when *cbn2*, the desired carbanion, is activated by an *additional* wgroup (Knoevenagel conditions). [c] See text.

The remaining lines of a transform entry, called qualifiers, describe other structural features whose presence would affect the transform rating. The sequence of decisions made by the computer as it passes through these qualifiers is shown in Figure 1. A qualifier line may be either of four elements. The most common element, the "standard qualifier," describes a structural feature in detail. If the feature is present in the current target molecule, the qualifier is said to be "true," and any action specified in the qualifier, usually a rating change, will be taken. Ordinarily each of the qualifiers in an entry is separately applied to a proposed transform until either the end of the entry is reached, a *kill* qualifier (*v.i.*) is found to be true, or the rating dips below a reject value. However, "control phrase" elements can be added to an entry which make the application of some qualifiers dependent on the truth or falsity of a previous qualifier.

The two other elements present specialized chemical information. "Condition statements" describe the laboratory conditions for the corresponding reaction; whenever a functional group is present which is unstable under these conditions, the rating is decreased. "Offspring-describing" statements describe changes to the carbon skeleton that will result if the transform is carried out. This information is needed by some qualifiers and in strategic evaluation of a transform.

Each of these elements will now be described in more detail. Examples are drawn from the aldol transform entry. (At this stage of program development a major purpose of writing qualifiers as extensive as these has been to ensure that the system or "language" being designed would permit description of any chemical situation. Qualifiers are usually added to the program in *ad hoc* fashion; whenever the computer carries out a naive transform, a qualifier is designed to restrict the application of the appropriate entry.)

1. "Standard qualifiers" are constructed from "English" words to give a phrase whose meaning is usually clear to a chemist even on first encounter with the language. A standard qualifier consists of an "optype" (operand type), at least one "modifying phrase," and usually an "action." These parts can be seen by dissecting a simple qualifier, line 8. The qualifier evidently means "add 30 to the transform rating if the second

..addt dec 30 / if / grp2; / .. is nitro
action optype modifier phrase

group (the 'wgroup') is a nitro group" (because the nitro group stabilizes a carbanion under milder conditions than do most wgroups).

Another example, line 7, contains different types of

..kill / if / halide; / ..within betato cbn2
action optype modifier phrase

modifier phrase and action. Here the meaning is "stop considering a proposed transform (*kill*) whenever there is a halogen atom attached to any atom two or fewer atoms away from the proposed nucleophilic carbon (*within betato cbn2*)." The nucleophilic center would cause expulsion of a halide ion, circumventing carbonyl addition.

Finally, line 56 prescribes a third action. In this

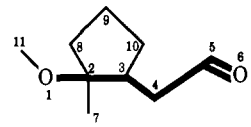
..subt dec 30 foreach / rgp; / ..cbn1
action optype modifier phrase

case 30 is to be subtracted from the rating for each carbon atom (*rgp*) attached to the alcohol carbon atom (*cbn1*) (excluding atoms which lie on the reaction path).

The kinds of "optypes" currently used in qualifiers are shown in Table II. Note that each kind of optype requires a particular type of modifier phrase. The modifier phrase most often required, "location-describing," is a description of the location in the molecule where the optype must be found in order for the qualifier to be true, as in lines 7 and 56 just discussed. Examples illustrating the vocabulary that must be used in "location-describing" modifier phrases are given in Table III. The different modifier phrases required by certain other optypes are shown in the last column of Table II.

The positional selector type of standard qualifier is required for those chemical reactions of a given functional group which can occur at mechanistically but not

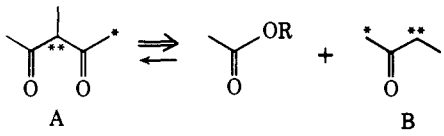
Table III. Illustrations of Location-Describing Modifier Phrases[a] for Standard Qualifiers



Phrases	Atoms or bonds described
Describing general locations:	
. .anywhere	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
. .anywhere onpath[a]	1, 2, 3, 4, 5, 6
. .anywhere offpath[a]	7, 8, 9, 10, 11
Describing particular path atoms or bonds:	
. .cbn1	2
. .cbn3	4
. .hetatm ingrp1	1
. .bond1	(2, 3) bond
Describing groups of atoms or bonds relative to path atoms or bonds:	
. .alphato cbn1	1, 3, 7, 8
. .betato cbn2	1, 7, 8, 9, 5
. .alphato cbn2 offpath[a]	10
. .betato cbn1 offpath[a]	9 (not 10 or 11)
. .within betato cbn1 offpath[a]	7, 8, 9
. .alphato hetatm ingrp2 onpath[a]	5
. .alphato cbn1 onring[b]	3, 8
. .alphato cbn1 offpath offring[a,b]	7
. .alphato bond1 offpath[a]	(2, 7); (2, 8); and (3, 10) bonds

[a] The "path" referred to is a specified sequence of atoms connecting the two groups, in this case the hypothetical reaction path shaded in the structure. [b]The "ring" will be a particular ring, the one mentioned in the most recent qualifier having *ringwith* or *path* as optype.

kinetically equivalent sites; for example, unsymmetrical cases of epoxide CO displacement, C=C addition, allylic substitution, and ketone α substitution *via* an enol or enolate. The evaluation specified by this type of qualifier refers not to the current target structure but to an as yet ungenerated structure which would be the result of the transform in question. For example, for the Claisen condensation shown, success depends upon the preferred formation of an enolate from the ketonic structure B with charge at C** and not at C*. The



A B

relative ease of formation of the various enolates derived from the diketone structure A is irrelevant. Evaluation of positional selector qualifiers depends on the provision of rules for evaluation of steric, electronic, and other kinetically significant factors on competing, mechanistically equivalent, but positionally nonequivalent reactions. The problem of allowing the computer to "effectively visualize" an intermediate in the synthetic tree prior to its actual generation is solved by using "offspring-describing" qualifiers, to be discussed below. This technique depends on the description of crucial structural properties of an offspring node of the synthetic tree (or synthetic starting material) in terms of the parent node (or synthetic product).

The example of a positional selector qualifier in the aldol entry, line 29, can now be understood. The meaning is, "stop considering this transform if, in the intermediate described by previous offspring-describing qualifiers, excess electron density is more easily accommodated on any of the atoms which are *alphato cbn1 offring offpath*[10] than on the atom *cbn2*."

. .kill / if / chminus; /
action optype

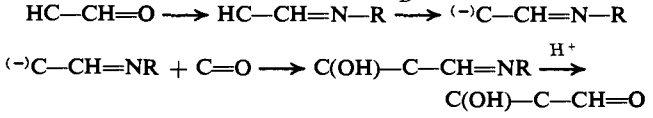
. .alphato cbn1 offring offpath betrthan; /
modifier phrase

. .cbn2
modifier phrase

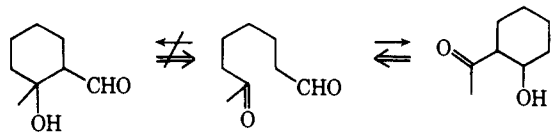
2. "Control phrases" permit qualifiers to interact in logically interdependent ways, allowing a fuller description of chemically complex situations. The two modes of interaction added are the logical "and," in which a group of one or more dependent qualifiers is applied only if some previous qualifier is true, and the logical "exclusive or," in which a group of one or more qualifiers is *not* applied if some previous qualifier is true. (Another mode of logical interaction, the "inclusive or," is the relationship among qualifiers existing when no control phrases are present.)

The control phrases are English phrases equivalent in meaning to logical "and's" and "or's," placed in an entry so as to segregate dependent qualifiers. The phrases *andbegin* or *then begin* introduce dependent qualifiers in a logical "and" relationship to a preceding qualifier. Qualifiers in an "exclusive or" relationship are introduced by *either* and separated by *orelse*. Either type of dependent qualifier group is ended by a *done*. A dependent qualifier may in turn have qualifiers dependent on it, without depth limitations, as will be seen from the aldol transform entry.

In Table I lines 12 through 18 illustrate use of the logical "and" relationship in describing the important chemical possibilities when the second group is aldehyde (line 12). Intermolecularly, an aldehyde "wgroup" allows application of a selective version of the aldol condensation which involves an imine-derived anion intermediate:



But intramolecularly, the only aldol pathway to a β-hydroxyaldehyde is the usual direct aldol addition, which is particularly likely to proceed in the opposite, or wrong, direction if the dicarbonyl precursor is a keto aldehyde rather than a dialdehyde. The various possibilities are handled using an "and" relationship among



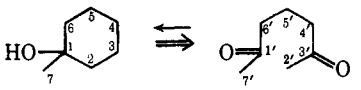
each of three qualifiers. The imine intermediate case corresponds to line 12 being true and line 14 false; line 16 is then not tested, and the rating is raised 30. The dialdehyde cyclization, as described by lines 12 and 14

(10) See Table III for illustrations of the meaning of complex location-describing modifier phrases such as this. The chemical implication of this qualifier for this entry will become clear after the following description of "control phrases."

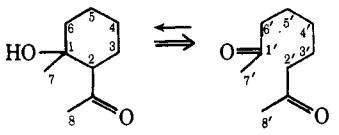
being true and line 16 being false, leads to a rating decrease of 40. When all three qualifiers are true, keto aldehyde cyclization has been encountered, and the transform is killed. Of course, if the wgroup is not an aldehyde, lines 14 and 16 will not be tested.

The "exclusive or" relationship exists at several places within the complex logical structure included in lines 19 through 48, which deals primarily with the value of forming various-sized rings and the difficulties encountered if other "wgroups" exist in positions which might favor formation of a different carbanion and closure of a different ring. First, a distinction between the intramolecular and intermolecular problems must be made. The former is handled in lines 20 through 40, and the latter, in lines 42 through 47. These parts of the entry are in an "exclusive or" relationship. If the bond being formed is in a five- or six-membered ring (line 20), then the subqualifier block lines 21–40 will be tested, and lines 42–47 will be skipped. If the envisioned reaction does not close a ring of favorable size, lines 20–40 are skipped, and lines 42–47 are tested. (Notice the use of indentations in Table I to portray these interrelationships to the chemist.)

A particularly complex situation next arises whenever the "wgroup" involved in the transform being considered is a ketone, since the additional possibilities then exist, first, that addition in the opposite direction can occur (such as that previously shown for an aldehyde "wgroup"), and second, that the entire reaction path, rather than just the bond being formed, may be in a ring. Lines 24–36 treat such possibilities for the intramolecular (line 20) formation of β-hydroxy ketones (line 22), as illustrated below (I and II) for six-membered ring formation. (The following argument would be equivalent for five-membered ring formation.) Another "exclusive or" relationship, this time between the path possibilities I and II, exists within lines 24–36.



I. Entire reaction path in ring



II. Only the bond being formed in ring

When all carbon atoms involved are in the ring (line 25, situation I above), the existence of an atom 7 having attached hydrogen permits addition to form a six-membered ring in the other undesired direction, *i.e.*, 7′ adding to 3′ above (line 27). Atom 7 may then be either part of another ring or part of an appendage. If it is part of a ring, then closure in the wrong direction will be less likely (line 28). Substitution of an anion-stabilizing group at atom 7 renders the required reaction prohibitively unfavorable (line 29). On the other hand, an additional wgroup activating atom 6′ (line 30) favors an enolate which can cyclize only to an inaccessible four-membered ring, so the six-membered closure can still proceed.

The opposite situation with respect to atoms 6 and 7 exists when only one of the path bonds is in a ring (situation II above). Provided atom 6 bears hydrogen, formation of a different cyclic ketone will always be possible (line 33). Superior reactivity of atom 6′ precludes the desired reaction (line 34), whereas activation of 7′ may be inconsequential (line 35).

Whether or not the wgroup is a ketone, lines 38–40 must be applied to an intramolecular aldol transform. The five-membered ring closure is less desirable (line 38), since elimination of water to give an enone will be difficult to avoid. Another wgroup activating a carbon (such as atom 8′ above) other than that which is required for closure might be deleterious (line 39). Closure of a ring of size less than five or greater than six members is unlikely (line 42). Lines 43–47, which refer to the intermolecular reaction, are straightforward in light of the preceding discussion.

3. "Condition statements" describe laboratory reaction conditions. The stability of each functional group in a target structure toward various laboratory conditions is determined, with moderate sophistication, during the perception stage of analysis.[2] Whenever a condition statement is encountered in a transform entry, the rating is lowered for each group unstable under these conditions and not involved in the transform being considered, by an amount dependent on the degree of group instability.

Condition statements may be "exclusive or'd," as illustrated by line 51 of the aldol transform entry. Since condition statements, unlike standard qualifiers, have no innate truth or falsity, the meaning of the *orelse* is simply that those conditions leading to the best overall rating would be used in the laboratory. In line 51 an additional 20 must be subtracted from the rating if the acidic conditions are superior, since elimination of water to yield a different product is then likely. Condition statements may be used within logical structures as subqualifiers only.

4. "Offspring describing" qualifiers, usually found at the start of an entry, name the bonds that will be broken if a transform is carried out. Knowledge of these bonds in advance is useful in evaluating transforms, since it provides a basis for the partial description of the proposed starting material or intermediate to be built up, so that the "position selector" standard qualifiers described above can be used. In addition, the designation of the bonds to be broken circumvents the application of a transform to a target structure having bond multiplicity or a bridgehead atom at an inappropriate location. The identification of the bonds to be broken can also be utilized strategically to limit the application of functional group interchange transforms, according to the rule, "Try a subgoal only if the bond which would be broken by the ensuing disconnective transform both has strategic value and is not broken by a transform which does not require subgoal intervention."

Computer Handling of Data Tables. The computer representation of any program during the time it is actually being run differs considerably from the series of semi-English and algebraic statements of a procedure-oriented language (*e.g.*, FORTRAN). The computer itself consists of a series of memory locations (words), which are "addressed" numerically and which consist of some number of binary digits or "bits" (18 for the

Table IV. Program Which Goes to *Somewhere* if the Third Bit of *Girl* is 1

"Source" program[a] ("English" mnemonics)	"Object" code Address[b]	Contents	Comments
lac girl	0	20 0011	bring contents of *girl* into accumulator (AC)
and tester	1	02 0010	*and*[c] AC with contents of *tester* (logical operation)
sza	2	64 0100[d]	skip over next instruction if AC = 0 (none of bits = 1)
mp somewhere	3	60 0012	does this if AC contains any bits = 1
..	4	..	otherwise program continues
..		..	
..		..	
tester: ..*000004*	10	00 0004	the octal number "4" has only the third bit from the right = 1, all others = 0
girl: ?	11	?	
somewhere: ..	12	..	

[a] Program is written in PDP-1 assembly language. Note that program locations are named, instead of numbered as in FORTRAN. [b] To keep the addresses simple, we assume this piece of program starts in the first memory location of the computer, address 0. [c] The *and* operation makes all bits in the AC zero except for those which are 1's in *both* the AC and the contents of the memory location being referenced. [d] For this PDP-1 command, the instruction field of the word (64) tells the computer that a skip may occur. In this case the address field is used to further specify that the skip should occur only if the AC is zero.

PDP-1). A random PDP-1 word, say the 3621st in memory, might contain:

101110010100000111

For convenience, bit patterns may be converted to octal numbers by grouping the bits by three's and writing the octal value of each of the six resulting binary numbers, as follows

101	110	010	100	000	111
5	6	2	4	0	7

The bits within a word may be used singly or in groups (fields), and consequently, the meaning of 562407 to a PDP-1 depends on the way in which it is encountered by a program. If it is a single datum referenced by some other memory cell, such as one containing the instruction *add 3621* (*i.e.*, add the contents of location 3621 to the contents of the accumulator), the meaning of "562407" is simply "the number" 562407. If 3621 is encountered in a sequence of program instructions, the computer's hardware splits its contents into two "fields" of six and twelve bits

101	110	010	100	000	111
5	6	2	4	0	7

The first six bits have the value "56," which happens to be the code instructing the PDP-1 to divide its accumulator by whatever number happens to be found in the memory location addressed by the number in the last twelve bits, which here is location 2407. It is also possible for an appropriate program to treat each PDP-1 word as consisting of as many as eighteen separate pieces of data, each conveying meaningful information. Such a program must include commands which make the values of individual bits available, for example, "logical" operations and "rotate" and "shift" commands. For instance, one can test the third bit from the right in a PDP-1 word using the sequence of instructions shown in Table IV.

It is the latter sort of meaning that LHASA attributes to a word in the chemistry data tables when the program is running. Using commands similar to the one in the example, the program decodes the eighteen bits in each membory location into three to ten fields, each consisting of one to seven bits. Each field carries a piece of information for the program to use in selecting or evaluating transforms.

Finally, the problem of converting the "English" words that the chemist sees into the data fields "read" by the computer must be considered. The tedious job of translating any "semi-English" program, or "source" program, into a sequence of computer codes or numbers, called an "object" program, is done usually by a computer, following a program called a "compiler." Conversion of chemistry tables into data fields depends on some unusual skills of the DECAL compiler. First, at the beginning of his source program a DECAL programmer may define words (*e.g.*, *rgbond*) not previously recognized by the compiler, by specifying an octal number, or bit pattern, as code that the computer is to place into the object program whenever this word is encountered in the source program. Second, if several words so defined appear in the same line of the source program, the computer packs all of the appropriate octal codes into the same object word. The chemistry tables as seen by the chemist consist solely of lines of words so defined by the programmer. Any numbers encountered by the DECAL compiler within a line of such words are converted into their octal equivalent and packed into the same word of object code. Whenever a semicolon or carriage return is encountered, the packing of a new word is begun.

An illustration of the packing of the words of a typical standard qualifier into fields of numeric data useful to the appropriate program may now be given.

..subt dec 40 if rgbond; ..alphato cbn1 offpath

The programmer has already defined the "English" words at the start of Table V. As mentioned above, *dec* signifies that the following number is decimal. The octal equivalent of *dec 40* is "50" or binary 101000. Therefore, the DECAL compiler would produce the fol-

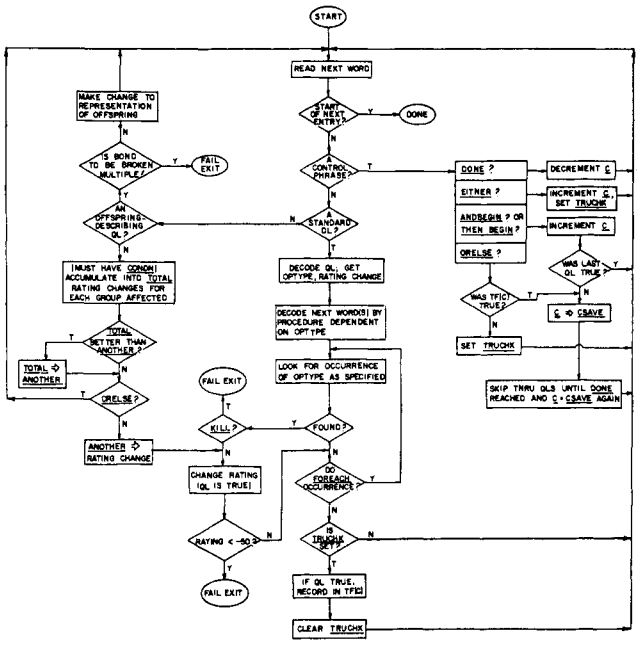


Figure 2. Simplified flow chart of QLTEST, the program which reads and interprets qualifiers. See the text for explanation and discussion.

lowing two words of object code from the source line given

100	101	111	100	101	000
if	*rgbond*		*subt*	*dec 40*	
100	000	000	001	001	001
			offpath	*cbnl*	*alphato*

These words of code can now be interpreted in the same fashion that an appropriate program will later use. The left-hand bit of both words is 1; this signifies that each word is part of a qualifier, not a main entry. The next eight bits of the first word name the qualifier's "optype." The last seven bits of the word give the numerical value of the rating change, and the "1" in the ninth bit from the right means that the rating change is to be subtracted, not added. Similarly the second word, the location-describing modifier phrase, may be decoded field by field.

Table V.

Word	Octal definition	Binary equivalent (bits)
subt	000400	000 000 000 100 000 000
if	400000	100 000 000 000 000 000
rgbond	057000	000 101 111 000 000 000
alphato	000001	000 000 000 000 000 001
cbnl	400010	100 000 000 000 001 000
offpath	000100	000 000 000 001 000 000

The utility of the chemistry table language as outlined here has been extended beyond the PDP-1 system by the creation of a compiler for these tables called TABLETRAN which has been written by Dr. Donald E. Barth in a widely available language (FORTRAN IV). When the entire program is rewritten in FORTRAN, a first step in running LHASA on another computer will be to prepare new table object code by running TABLETRAN with the chemistry tables. At that time the vocabulary of our language will be edited to increase its resemblance to "normal English."

While the program is running, the subroutine QLTEST sequentially interprets each qualifier, looks for the described feature in the current target, and acts as instructed by the qualifier if the feature is found. A simplified flow chart of its operation appears in Figure 2. The "type" of qualifier indicated by the next word of object code in the table is determined. If the word of code introduces a standard qualifier, processing proceeds down the central branch of the flow chart. A branch to the right occurs if the word of code is a control phrase; a branch to the left if a condition statement or offspring-describing qualifier is encountered. All branches which do not terminate in DONE eventually return to the top of the flow chart where a new qualifier is read.

Some symbols used in this flow chart to describe the processing of control phrases will now be defined. "C" is a counter for levels of logical dependence. It is incremented whenever a *begin* or *either* is encountered in the table, and decremented whenever a *done* is seen. "Truchk" is a flag that is set for any qualifier following an *either* or an *orelse*. The truth or falsity of such a qualifier is recorded in the vector tf[c]. Whenever an *orelse* is encountered, tf[c] must be referenced to see if any qualifier "exclusively or'd" with this qualifier has previously been found true.

With these comments the flow chart is a sufficient explanation of the operation of QLTEST. QLTEST returns a failure message whenever a *kill* qualifier is true, or the rating dips below -50, or the bond mentioned in a "bond-broken" offspring-describing qualifier is found to be multiple. When the start of a new entry is reached, a success message is returned to the controlling transform-selection routine, since all qualifiers for the requested entry will have been examined.

Transform-Selecting Programs. GSING and GPAIR, the program units responsible for choosing one- and two-group transforms applicable to a particular target structure, proceed in a fashion that differs only in details of the path-generating and table-matching processes. Since this "table-driven" procedure parallels a highly effective technique for "human" synthetic analysis using these data, it will be described here in some detail. The numbers in the flow chart of this procedure (Figure 3) are keyed to the headings below.

1. Get a new group, or pair of groups. If the molecule has n groups, GSING will be finished only when all n groups have been examined; GPAIR must process the $n(n-1)/2$ possible pairings of groups.

2. For the new group(s), list all the "synonyms," or "general group" names, that might be used in the data tables to describe the group. Transforms which are characteristic of several different types of functional groups are listed under the same "general group" table heading, as was seen for the aldol transform ("*alcohol;.. wgroup*"). The list of all possible synonyms for a group type includes the group type itself.

3. Find a new path involving the group. A path acceptable for a one-group transform will be any string of atoms extending from that group that at least in part is previously untravelled. It need be no longer than

is required for operation of any transform in the tables. (The longest path currently is four carbon atoms, required for 1,4 addition of a copper reagent to a vinyl ketone.)

For two-group transforms, any new path connecting the two groups and less than eight atoms long is acceptable. Clearly, no such path exists when the two groups are in different molecules (as often happens when the current target "structure" is separated from the initial target by at least one disconnective step). If the target contains any rings, there can be several paths between a given pair of groups.

In either case an acceptable path must meet other restrictions. It must consist wholly of nonaromatic carbon atoms,[11] and any multiple bond encountered must be contained wholly within the path. A path involving a functional group which can operate electronically in only one direction must be properly connected to that group. For instance, the path leaving an "ester" group may not traverse a CO single bond; but the path leaving an "esterx"[2] group must traverse the CO single bond.

4. Choose a synonym for each group. Determine the path length appropriate to the particular synonym(s).

It may be worth illustrating the variety of path lengths possible for a given configuration of functional groups. Depending on which group is named first and what synonym is used for "ketone," a vinyl ketone can be named: (A), 3-keto-1-ene, path length of 3; (B), 2-

ene-1-one, path length of 2; (C), 2-wgroup-1-ene, path length of 2; (D), 1-ene-1-wgroup, path length of 1. Three different path lengths for one functional group pair configuration! When all possible combinations of synonyms have been examined, a new path is chosen.

5. Match the synonym(s) and path parameters against an appropriate part of the data table, entry by entry.

At the start of each entry in the data table are the basic transform requirements. If there is a complete match with the present synonym(s), various path parameters are then checked. For two-group transforms, the path lengths must coincide, and for reconnective two-group transforms, such as ozonolysis or the Cope rearrangement, there must not be a serious stereochemical obstacle to the joining of the two groups. An algorithm presently used to screen out many sterically impossible configurations follows. Divide the path connecting the two groups into two nearly equivalent halves (one half must always contain either an extra atom or an extra bond). The two groups will be unable to meet if: there exists a "real" ring having eight or fewer atoms such that: (a) both halves of the path are partially included in the ring, and (b) the number of atoms contained in both path and ring is more than two.

Paths for one-group transforms may have to meet some preliminary chemical and strategic requirements, specified at the beginning of an entry. For example, Grignard addition to a ketone is operable only when the bond being formed does not become part of a ring in the product.

(11) Aromatic chemistry will be handled by a different part of the program still to be written.

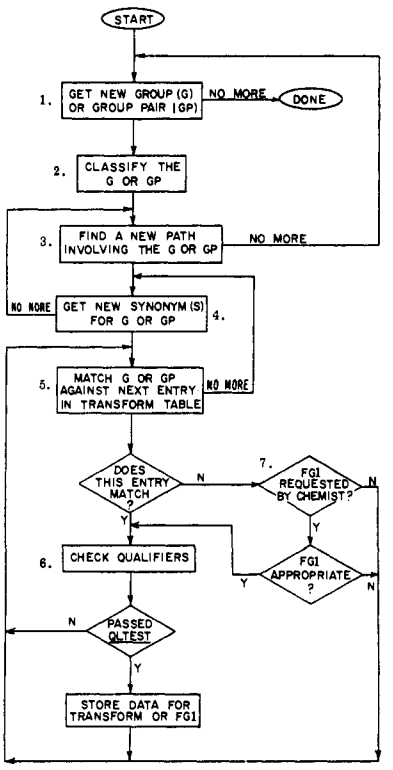
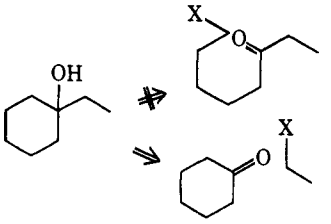


Figure 3. Flow chart of the programs for choosing one- and two-group transforms. Numbered boxes are keyed to descriptions of these processes in the text.



("X" is generalized halide, the precursor to an organo- metallic intermediate.)

Such a requirement is given in the entry for this transform as

..need bond1 nonring

An apparent match will be discarded whenever "bond1" is in a ring.

As discussed in the introduction, most one-group transforms also are required to break a bond having some kind of topological importance, specified in the data table. For example, the table entry for a Wittig reaction (double bond formation) might contain the following lines:

..wdlike bond0 appendage

..wdlike bond0 connective

If a double *bond0* were either attached to a ring but not included in another ring (*appendage*)[2] or centrally located in the structure (*connective*),[2] this transform would be applied. If the chemist wants to break this or any or all bond(s) in the structure, he may designate such bonds as "strategic."[1] This designation overrides any con-

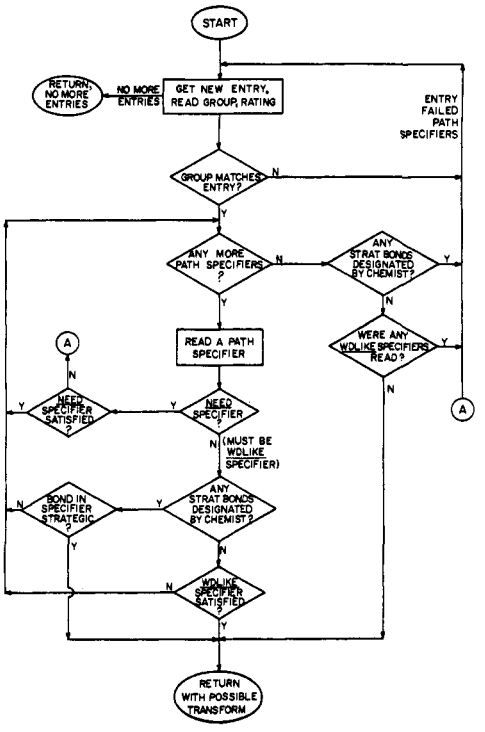


Figure 4. Detailed flow chart of the table-matching procedure during the choosing of one-group transforms (box 5 of Figure 3)

siderations by the computer of topological importance, as can be seen from the flow chart which details the table matching operation of GSING (Figure 4).

6. Use QLTEST to check any qualifiers following a transform entry for which a complete match is found. If the resultant transform rating is greater than the cutoff value, usually −50, store the data required for actual execution of this transform.

7. If the chemist has requested functional group interchange (FGI), consider its possibility whenever a match is not found.

Because there are far more nonmatches than matches when comparing a substructure with the data table, severe restrictions on the kinds of situations that may generate FGI requests will always be necessary. A variety of limiting strategies has been used in various versions of the program. The current (but not necessarily final) approach uses a mix of the following restrictions.

Request an FGI: (1) only if one group matches and the other does not (for two-group transforms); (2) only for selected entries in the table (designated by *tryfgi*); (3) only if the successful FGI appears to allow a transform that either (a) breaks an otherwise untouched bond in the current path, or (b) breaks a topologically important or a strategic bond.

The problem of perceiving all valuable functional group interchanges is complicated further by adjustments to the path length that are often required to make a near match between target and table visible. Considering these possible adjustments effectively doubles the number of entries that must be searched. The problem is illustrated in the following synthetic sequence

$$\mathrm{HO{-}C{-}C{-}C{-}OH} \xrightarrow{\text{FGI}} \mathrm{HO{-}C{-}C{-}C{=}O} \xrightarrow{\text{aldol}} \mathrm{O{=}C + C{-}C{=}O}$$

Examination of the 1,3-diol by GPAIR should generate a request for an FGI to change one of the hydroxyl groups into a ketone, allowing the aldol reaction. It will be recalled that the table entry for the aldol transform is *..alcohol; ..wgroup; ..at2*. But there is no combination of synonyms for the present pair of alcohol groups which gives rise to a path length of 2. GPAIR must be aware that a "1,3 any-group" is comparable, for FGI purposes only, with a "1,2 alcohol, wgroup."

In general, there are two distinct possibilities to be considered by GPAIR in looking for functional group interchanges.

(1) The table entry requires a group having a peculiarity of path numbering which allows fruitful comparison with some target substructure of apparently different path length. The *wgroup* in the aldol entry just discussed has such a peculiarity.

(2) One of the groups in the target substructure being examined has a similar peculiarity, allowing useful examinations of entries requiring a different path length. For example, a 1,2 any-group, dbond is a substructure that should be compared with entries requiring a path length of three atoms:

$$\underset{1\ 2}{\mathrm{G_1{-}C{-}C{=}C}} \xrightarrow{\text{FGI}} \underset{1\ 2\ 3}{\mathrm{G_1{-}C{-}C{-}C{-}G_2}} \xrightarrow{\text{2-group}} ?$$

For execution, the finished list of transforms is passed one by one to the manipulative program by way of a control program that will be described in later publications. The information carried along with each transform includes a list of all the atoms on the path and a pointer to the atom which is the origin of each group involved. With this detail provided, code for individual transforms becomes a trivial series of calls to subroutines. A particularly simple example is shown here.

$$\underset{1\ 2\ 3}{\mathrm{C{=}C{-}C{=}O}} \Longrightarrow \mathrm{C{=}O + C{-}C{=}O}$$

breakb2(at1, at1p1)	...break bond between C_1, C_2 (removes the π bond)
breakb2(at1, at1p1)	...break bond between C_1, C_2 (removes the σ bond)
add1(o, at1)	...attach an oxygen to C_1
makeb2(−1, at1)	...create a second bond between C_1 and the oxygen

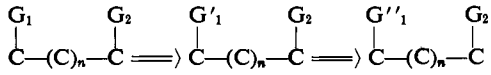
The immediate effect of these commands is to create a new "structure block,"[1] which contains all the data about the resulting intermediate. The coordinates of any new atom added (as by the *add1(o, at2)* command above) are determined by extrapolation of the local geometry of the target structure being operated on. When the structure block is completed, the new intermediate is displayed for the chemist while its evaluation[2] is going on.

Functional Group Interchange—Data, Transform Selection, and Structural Manipulation. In previous sections of this paper, transforms involving disconnective and nondisconnective functional group interchange (FGI) have been defined and their use as subgoals in connection with the application of group-pair disconnections has been mentioned. The disposal of FGI transforms within the program will now be described in more detail. As might be anticipated, FGI transforms (both disconnective and nondisconnective) can logically be divided into subclasses. The categories which re-

sult (neglecting for convenience the disconnective–nondisconnective dichotomy previously mentioned) can be summarized as follows.

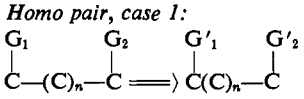
1. Simple FGI. The most commonly occurring group interchange is one in which *one* of a pair of functional groups is modified in one selective synthetic step.

2. Sequential Simple FGI. This type of transform telescopes (corresponds to) *two* consecutive synthetic steps, although as in the previous case only one of the original pair is affected by the change. Only the overall result of the two steps need be displayed.



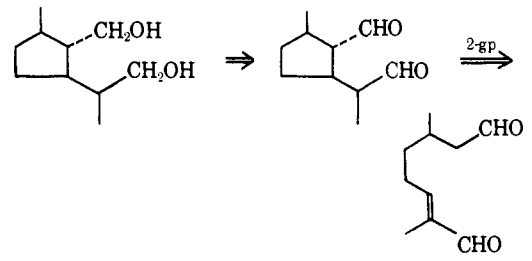
3. Double FGI. Transforms of the double FGI type are those in which each of the groups in the pair undergoes modification. Further classification leads to four subgroups.

Homo pair, case 1:

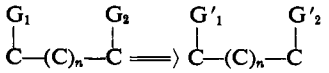


where $G_1 \equiv G_2$, $G'_1 \equiv G'_2$

example:[12]

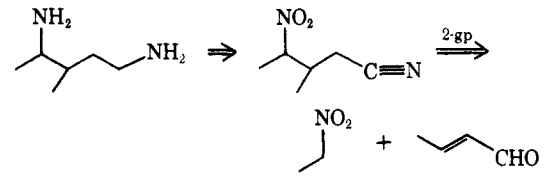


Homo pair, case 2:

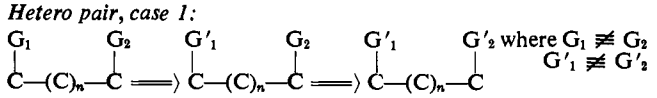


where $G_1 \equiv G_2$, $G'_1 \not\equiv G'_2$

example:

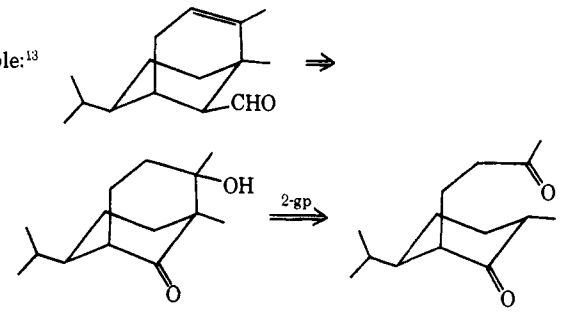


Hetero pair, case 1:

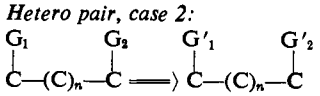


where $G_1 \not\equiv G_2$, $G'_1 \not\equiv G'_2$

example:[13]

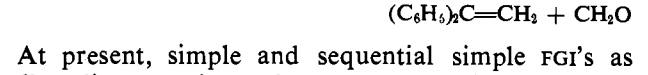


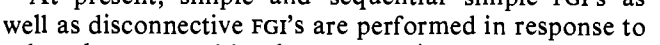
Hetero pair, case 2:



where $G_1 \not\equiv G_2$, $G'_1 \equiv G'_2$

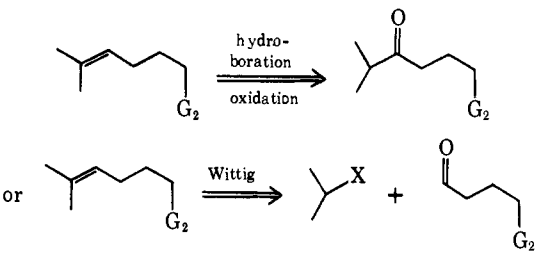
example:



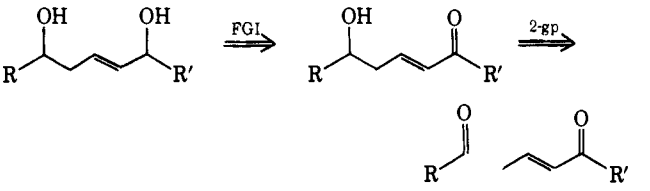


At present, simple and sequential simple FGI's as well as disconnective FGI's are performed in response to subgoals generated by the group pair program, and the implementation of double FGI will soon follow. Eventually FGI requests will be generated as subgoals of other parts of an expanded LHASA including those concerned with single-group transforms, ring- and stereochemically oriented transforms, and control element (*e.g.*, protecting group) application.

The current method of generating FGI subgoals is quite straightforward, since these subgoals originate in only one class of transforms (group pair). A subgoal list is generated during the scan of the group pair table as described above. Each entry on this list is a request to effect the replacement of one functional group (subject group) by another (object group), without specific indication of chemical feasibility. The selection of a specific transform to accomplish the required functional group interchange, if one exists, is the task of the mechanism selection routine (FGI), the associated reaction table (FGITAB), and the manipulation routine (FGICHEM). A given subgoal can frequently be satisfied by more than one transform as illustrated by the following FGI transforms, both of which satisfy a request to replace a double bond (the subject group) by a wgroup (the object group):



Another type of transform which would satisfy the same request (*i.e.*, double bond ⟹ wgroup) is indicated by the sequence



Here the FGI which converts a secondary alcohol function to a carbonyl function also generates a vinylwgroup which here allows a vinylogous aldol group–pair transform. The current program operates to select only one of these three possible transforms for a given subgoal, that with the highest overall rating. A flow chart outlining the sequence of operations involved in this selection is shown in Figure 5. The overall rating depends not only on the result of the operation of FGI or FGITAB, but also on successful manipulation by FGICHEM, as is discussed below.[14]

(12) K. J. Clark, G. I. Fray, R. H. Jaeger, and R. Robinson, *Tetrahedron*, 6, 217 (1959).

(13) E. J. Corey and S. Nozoe, *J. Amer. Chem. Soc.*, 87, 5728 (1965).

(14) The selection of only a single transform for a given subgoal when several are available is a simple but not an optimum strategy. A more effective screening procedure analogous to the cut-off rating for group-pair reactions is projected for later versions of FGI.

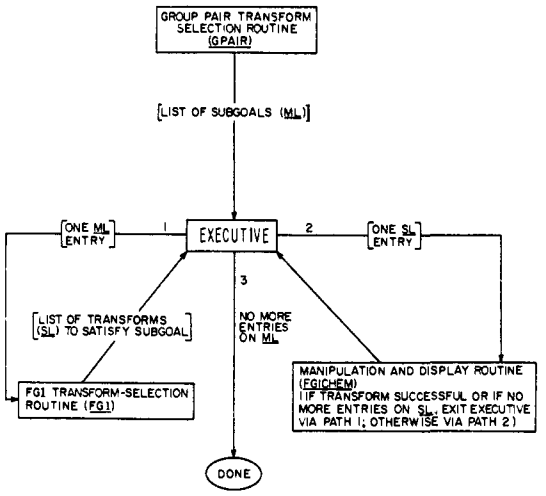


Figure 5. Flow chart showing the relationship among the main programs required for generation and satisfaction of FGI subgoals. Branches leaving the EXECUTIVE are numbered in the order in which they are followed.

FGI **Data Table** (FGITAB). The table of FGI transforms utilizes basically the same qualifier format and vocabulary described for the table of group–pair transforms. FGITAB is actually composed of three subtables, since it has been necessary to separate FGI transforms depending on whether the path between functional groups is to be unchanged, increased, or decreased by the FGI operation. During the search of the GPAIR table, half matches (one functional group only) are recorded for identical paths and also for paths differing by one atom (see preceding section). The subgoal therefore supplies FGI with a modifying path parameter (MODIFY) having possible values of 0, ±1. This modifier is used by the mechanism selection program FGI to reference the appropriate subtable of FGITAB. Each subtable is further divided into "entry blocks," or collections of transforms which apply to the same type of subject group. For instance, the table entries describing transforms 1 and 2 are found in the same entry block, since they both require the presence of a carboxyl group.

$$R{-}COOH \Longrightarrow R{-}CHO \quad (1)$$

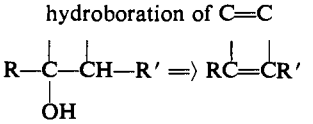
$$R{-}COOH \Longrightarrow R{-}CH_2OH \quad (2)$$

A typical entry in FGITAB includes first the name of the object group and the initial rating of the transform. Next come the qualifiers which, as before, effect an increase or decrease of the rating depending on the presence of specified structural features. Such requirements as the necessity of having a primary alcohol if the conversion carbinol ⟹ carboxyl is to be successful are specified by qualifiers. As with disconnective one-group and two-group transform selection, QLTEST is used to interpret qualifiers.

Disconnective FGI transforms are included in FGITAB as well as nondisconnective transforms.

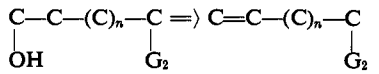
The entry in FGITAB for hydroboration of a carbon–carbon double bond (somewhat oversimplified) will serve to illustrate further the data base used for FGI.

hydroboration of C=C

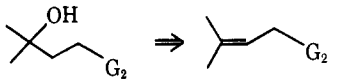


Alcohol: *..dbond dec 85; ..amarkov*
..subt dec 70 if dbond; ..anywhere
..subt dec 70 if tbond; ..anywhere
..subt dec 60 if cyano; ..anywhere
..subt dec 60 if oxo; ..anywhere

FGI transforms which involve a carbon–carbon double bond in an unsymmetrical environment as either the subject group or the object group requires special procedures that check the validity of the transform with regard to orientation of addition and elimination reactions (*i.e.*, Markovnikov *vs.* anti-Markovnikov and Hofmann *vs.* Saytzev). In the entry given above for olefin hydroboration, the term *amarkov* is a pointer to a block of coding in the manipulation module FGICHEM which checks to ensure that the requested transform of alcohol to a given olefin corresponds directionally to that for a hydroboration reaction. For example, if the following FGI is desired,



and the substitution pattern is as below,



then the hydroboration transform would not be allowed. It would also be possible to accomplish this type of screen entirely while processing FGITAB, using "offspring-describing" qualifiers together with QLTEST. The relative merit of these alternative methods will be tested in future work.

Transform Selection (FGI). It is the function of the transform-choosing program FGI to find chemically reasonable ways of satisfying a subgoal. The following description of FGI is keyed to the numbering on the flow chart in Figure 6.

1. Form the set of all acceptable object groups. In cases where the subgoal is to replace a group by another specific group type, the set of object groups (TARGETS) will have only one member. However, if a general object group (*e.g.*, wgroup) is requested, then TARGETS will contain all functional groups of that electronic type (*i.e.*, all electron-withdrawing groups).

2. Find the proper entry block in FGITAB. That is, locate the beginning of the collection of entries which has the proper subject group and also which reflects the current value of the path change parameter (MODIFY), described above.

3. Search the entry block until an entry is found that begins by naming a member of TARGETS. If none are found, then program execution jumps to step 9.

4. If there happens to be another functional group in the molecule of the same type as either the member of TARGETS under consideration ("operand," the specific group found in step 3) or any intermediate group named in this entry (in the case of a "sequential FGI"), then subtract 25 from the rating. This is a crude acknowledgment of the complications which arise from the presence of competing sites in a reaction.

5. FLAG is an indicator which is set the first time a disconnective FGI is stored on the list of chemically possible transforms. If it has not been set, then processing of the present table entry proceeds, and the qualifiers are read by QLTEST. If, however, FLAG is set then only

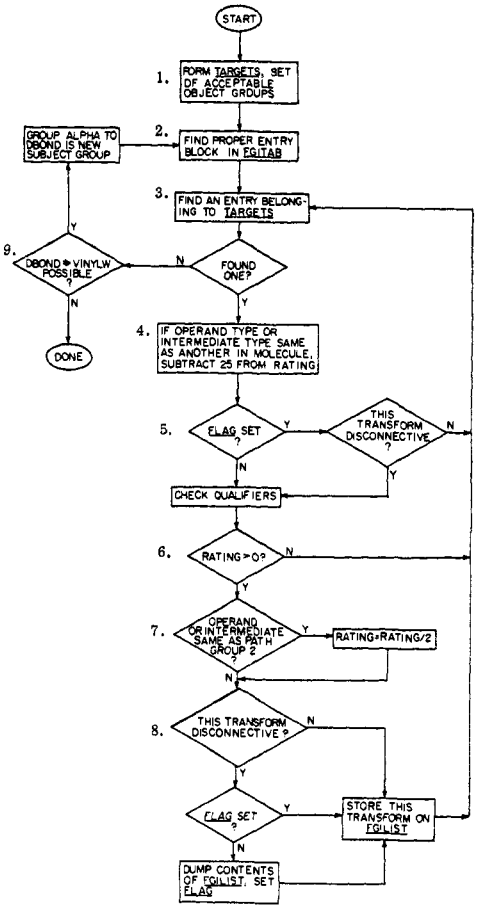
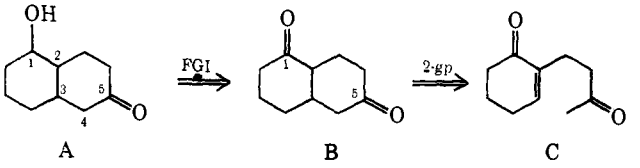


Figure 6. Flow chart of FGI, the program responsible for selecting those interchange transforms capable of satisfying a particular subgoal.

disconnective interchanges are acceptable and so if the entry currently being considered is not disconnective, it is rejected and program execution jumps back to step 3.

6. After qualifiers are read, is the resulting rating greater than zero? If so, continue; otherwise reject the transform and go back to step 3.

7. Another instance where group protection is necessary but usually quite difficult is when the operand or the intermediate group type is the same as the second group of the original pair. The ketone at carbon 5 cannot be protected earlier than C in the synthetic sequence, since its properties as a ketone are required intact for the group–pair reaction. Since selective protection of the 5-carbonyl in B or selective reduction of the 1-carbonyl in B are both problematical, a lowering of the rating for the FGI would seem advisable. The rating in this case for the transform A ⟹ B would in fact be reduced by FGI.



8. If the transform under consideration is not disconnective or if it is disconnective and FLAG is set, then it has passed all necessary tests and is stored on a list with the highest rated reaction at the top (FGILIST).

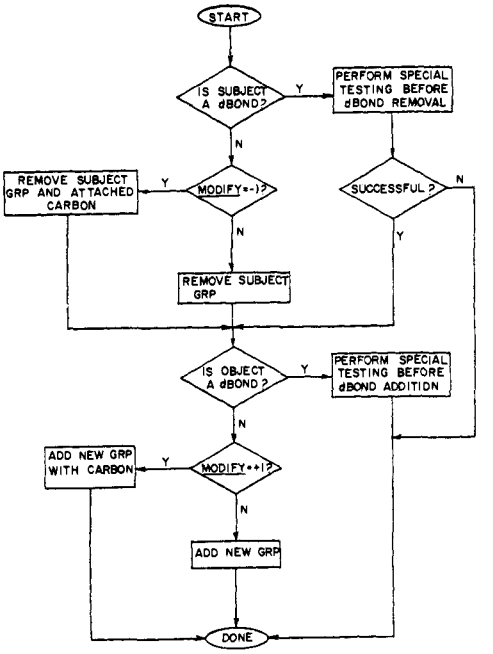
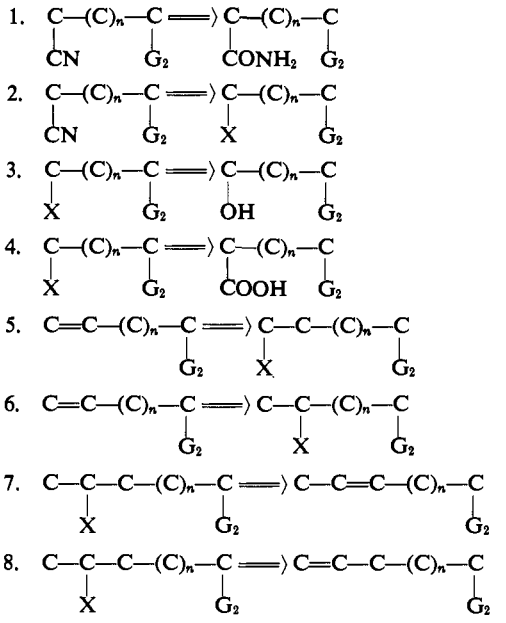


Figure 7. Flow chart of FGICHEM, the program which carries out functional group interchange transforms.

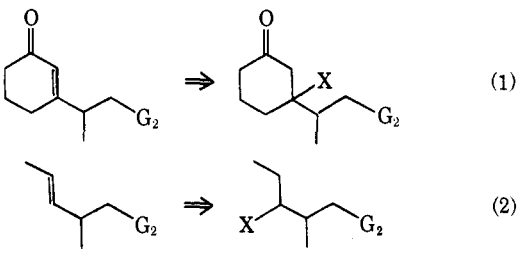
Program execution jumps back to step 3. However, if it is disconnective and FLAG is not set, then the current transform is the first disconnective FGI found for this subgoal. Since in general these are more desirable than normal FGI's, all previous entries on FGILIST are removed and the new entry is stored. To indicate that FGILIST now contains a disconnective FGI, FLAG is set; processing then returns to step 3 in search of another reasonable transform.

9. If the replacement of a double bond by a wgroup is requested by the subgoal, an attempt is made to introduce a vinyl-wgroup, as mentioned above. If a functional group exists α to the double bond, it becomes the "subject group," and the set of electron-withdrawing groups is placed into TARGETS. Program execution jumps back to step 2. FGI's job is done if the above interchange is either not possible or not requested.

FGI-Manipulation Program (FGICHEM). Once the FGI mechanism-choosing program has found all the transforms which satisfy a particular subgoal, the first one is processed by FGICHEM. This routine carries out the same tasks as the manipulation programs for two-group and one-group chemistry, namely, symbolically making or breaking bonds and adding or deleting atoms. There are two steps in the process of replacing one functional group by another. The first, removal of the subject group from the molecular framework, is handled by a general subroutine which does not need to distinguish between different group types. However, for the second step, addition of a new functional group, one of a number of small blocks of coding is used, depending upon the object group type. A chart describing the flow of processing is shown in Figure 7. The structure of the code blocks resembles those used in the other two manipulation programs. In both group removal and group addition steps, care is taken to ensure that the change in path length originally requested by the subgoal is in fact obtained. For example, lines 1 and 2, respectively, show a normal FGI



and one in which a path change is performed during the group deletion stage. In line 4 a change occurs during the group addition stage. Examples 5–8 all involve a double bond, and as mentioned earlier, such transforms require a check for proper orientation. When the subject group is a double bond, the reaction usually corresponds to an elimination of some sort, and as can be seen in 5 and 6, both normal and path-shortening FGI's are possible. Similarly, when the object group is a double bond, both normal and path-lengthening FGI's are possible (lines 7 and 8). When an elimination has been specified by the data table as being Saytzev- or Hofmann-directed, the manipulations are only performed if it is clear that the synthetic elimination would provide the proper double bond. Since electron-withdrawing groups near the reaction site can have considerable effect on the direction of elimination, the controlling effect of such groups on the course of elimination is taken into account by FGICHEM. The following pair of transforms is illustrative.



(1)

(2)

If transform 2 was requested, it would not pass the testing phase of FGICHEM because the synthetic elimination would form the more highly substituted double bond. But transform 1 *would* be allowed because of the dominant influence of the electron-withdrawing carbonyl on the direction of elimination. Transforms which involve a double bond as the object group must also be tested for proper orientation. Substitution checks similar to those described above are performed to ensure that a synthetic addition (specified in the data table as either Markovnikov or anti-Markovnikov, and being either a normal or a path-lengthening FGI) to the double bond will leave the subject group attached to the proper carbon. Again, if the test fails, the next transform on the list is attempted.

Conclusion

The major aim of the previous sections has been to provide a description of one manner in which detailed chemical information, organized according to type of transform, can be made available to and used by a computer performing synthetic analysis. The organization and form of this information have been chosen carefully with regard to both underlying chemical logic and suitability for general chemical problem solving. The qualifiers for a given transform provide an effective means for the selection and rating of synthetic intermediates. They can be written to reflect directly all the information which is available to a chemist regarding the scope of a synthetic chemical reaction. The sample table presented above for the aldol transform demonstrates the way in which the quite complex structural requirements for the applicability of a chemical transform can be included in the chemical data base. More complex situations can be dealt with readily by enlargement of the data base.

Experience with the application of LHASA to synthetic problems has already shown the utility of the techniques previously described for selecting and applying two-group, one-group, and FGI transforms. The use of subgoals, the concept of identifying strategic disconnections (strategic bonds) in advance of (and independent of) the selection of transforms, and the strongly controlled use of one-group transforms are also particularly effective problem-solving procedures. In the last case the use of one-group transforms only in conjunction with a structural feature other than the functional group involved in the transform (*e.g.*, an appendage) provides another effective implementation of the "pairwise" concept of transform selection outlined in previous work.[3]

The methods described herein provide guidelines for the addition of other sections of the data base, including extensions of SINGTB and the one-group transform strategies, functional group addition transforms, ring transforms for carbo- and heterocyclic systems, and transforms dealing with stereorelationships, rearrangements, unsaturated (*e.g.*, aromatic) paths, and control elements (*e.g.*, functional group protection). The construction and utilization of these data packages and the development of relevant strategies for subgoal generation and transform selection constitute a challenge of considerable magnitude which has now been confronted by ongoing research. The crucial test of present and future methods will be the performance of the program as measured against a spectrum of synthetic problems. Subsequent papers in this series will be concerned with the application of the program described above to specific synthetic problems,[15] with other chemical packages and with more sophisticated strategies for subgoal generation and transform selection.[16]

Acknowledgments. We wish to thank Dr. W. Todd Wipke for helpful discussions during the early phases of the development of the two-group transform package. This work was assisted financially by the National Institutes of Health and by support to the Harvard

(15) Illustrating what the program can and cannot do.

(16) For a recent general discussion of the application of strategies in future program development see E. J. Corey, *Quart. Rev., Chem. Soc.*, 25, No. 4 (1971).

Center for Research in Computer Sciences by the Advanced Research Projects Agency.

Appendix I

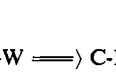
Two-Group Transforms Currently Applied by LHASA

The citation for each two-group transform takes up two lines. The first line shows the part structure required for operation of the transform, followed by a double arrow and the result of its operation. In other words, it illustrates the *antithetic* process. The second line verbally describes the operation in the opposite *synthetic*, or familiar, direction. Those transforms which may invoke FGI (see text) are marked with **.

For presentation the transforms have been organized into two major divisions, according to whether or not their operation affects carbon–carbon bonds. Each major division is subdivided into "path-length" classes, on the basis of the number of atoms separating the two groups, as designated by number and parentheses such as (2). Within a "path-length" class, transforms are ordered by types of group, first according to the left-hand group type and then according to the right-hand group type. (The part structures in the first line of each transform citation are drawn so that the left-hand group type always is equal or higher in order than the right-hand group type.) The ordering of group types from highest to lowest is as follows.

COOH (acid)	C≡C (triple bond)
CO-Hal (acid halide)	C≡N (cyano)
COH (alcohol)	C—NO_2 (nitro)
CHO (aldehyde)	"esterx"[2]
C—N (amine)	epoxide
$CONH_2$ (amide)	C=N (imine)
COOC (ester)	C^+ (carbonium)
C—O—C (ether)	C—D (electron-donating)[2]
C-Hal (halide)	C–W (electron-withdrawing)[2]
C=O (ketone)	C—X (nucleophilic group)[2]
C=C (double bond)	C—OXO[2]

Transforms Which Make, Break, or Rearrange Carbon–Carbon Bonds

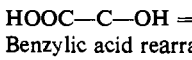


(1) C—C(W)—W ⟹ C-Hal + W—C—W

**Alkylation *via* doubly stabilized anion (*e.g.*, "malonic ester" synthesis)

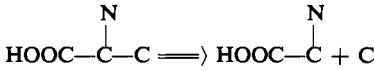
(2) HOOC—C—OH ⟹ C=O

**Cyanohydrin formation and hydrolysis



HOOC—C—OH ⟹ —CO—CO—

Benzylic acid rearrangement

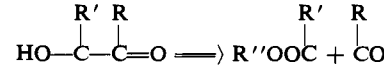


HOOC—C(N)—C ⟹ HOOC—C(N) + C=O

Amino acid synthesis *via* azlactone–carbonyl condensation

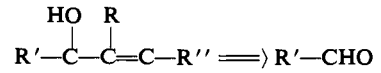
HO—C(R)—CHO ⟹ RCHO

α-Hydroxy aldehyde from (1) dithiane anion addition, (2) dithiane hydrolysis



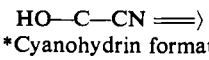
HO—C(R′)—C(R)=O ⟹ R″OOC—C(R′)—C(R)—COOR″

**Acyloin condensation



R′—C(HO)—C(R)=C—R″ ⟹ R′—CHO + RC-Hal + R″CHO

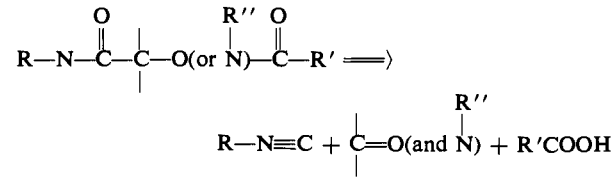
Double Wittig condensation *via* a β-oxidoylide



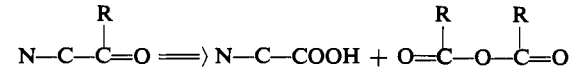
HO—C—CN ⟹ O=C

**Cyanohydrin formation

Aldol condensation with double bond migration



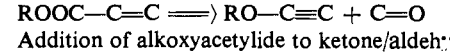
Passerini reaction



Dakin-West α-amino ketone synthesis

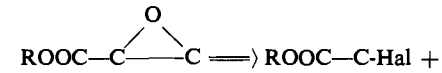
N—C—C—W ⟹ N + C=O + C—W

**Mannich condensation

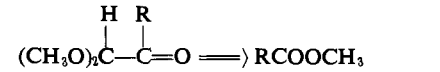


ROOC—C=C ⟹ RO—C≡C + C=O

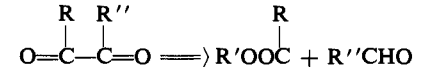
Addition of alkoxyacetylide to ketone/aldehyde



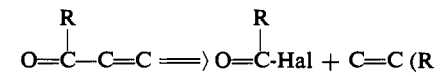
Darzens reaction



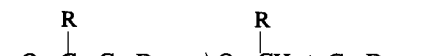
Ester condensation with methylsulfinyl carbanion and iodination in methanol



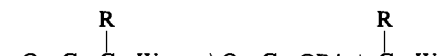
Dithiane anion attack on ester followed by hydrolysis



Friedel–Crafts acylation of olefin



Dithiane anion addition to C=O or C=N and hydrolysis



**C-acylation of a stabilized carbanion

C=C—W ⟹ C=O + C—W

**Aldol condensation

Knoevenagel reaction

D—C—C—W ⟹ D=C + C—W

**Aldol condensation

W—C—C—W ⟹ W—C + C—W

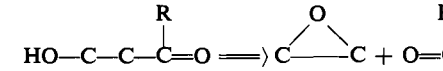
Oxidative coupling

(3) HOOC—C—C=C ⟹ C=C—C + NC—CO—CN

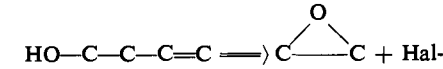
Chain extension by carboxyl ene reaction followed by hydrolysis

HOC—C—COH ⟹ C=C + C=O

Prins reaction



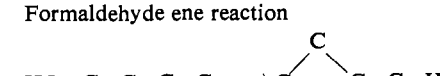
Dithiane (or vinyl amine)/epoxide coupling and hydrolysis



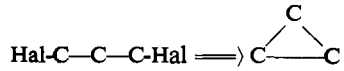
Epoxide opening by vinyl organometallic reagent

HO—C—C—C=C ⟹ O=C + C=C—C

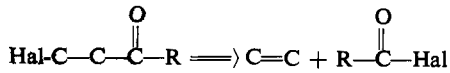
Formaldehyde ene reaction



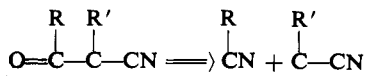
Cyclopropyl carbinyl/allyl carbinyl rearrangement



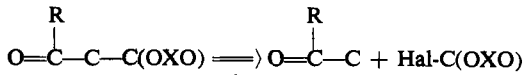
Halogenation of cyclopropane ring



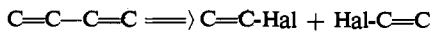
Acylation of double or triple bond



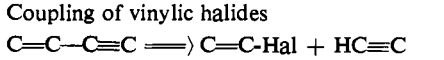
**Thorpe condensation



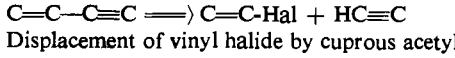
**Enamine-assisted α acylation



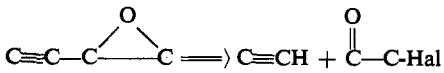
Coupling of vinylic halides



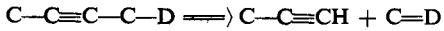
Displacement of vinyl halide by cuprous acetylide



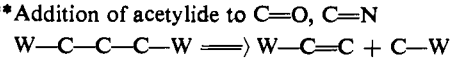
Glaser oxidative coupling



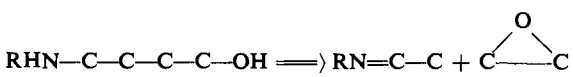
Copper acetylide condensation



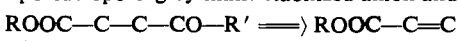
**Addition of acetylide to C=O, C=N



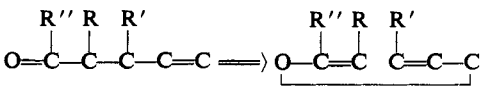
**General Michael addition

(4) 

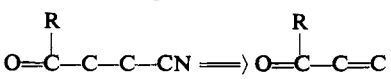
Epoxide opening by imine-stabilized anion and reduction



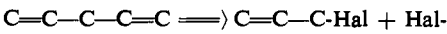
Nickel-promoted conjugate addition of acyl group



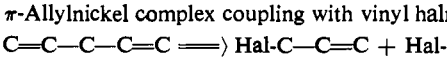
**Claisen rearrangement



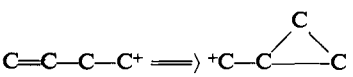
Conjugate addition of cyanide



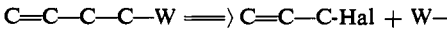
π-Allylnickel complex coupling with vinyl halide



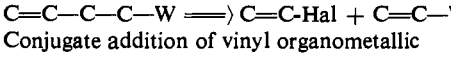
π-Allylnickel complex coupling with vinyl halide



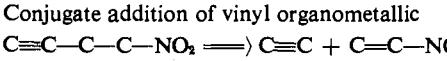
Cyclopropylcarbinyl rearrangement



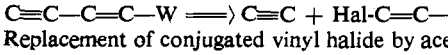
**Displacement of allylic leaving group by stabilized carbanion



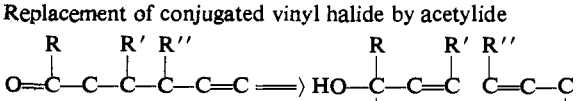
Conjugate addition of vinyl organometallic



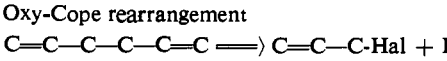
Michael addition of acetylide to nitro olefin



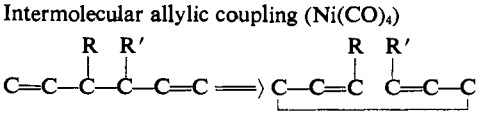
Replacement of conjugated vinyl halide by acetylide

(5) 

Oxy-Cope rearrangement



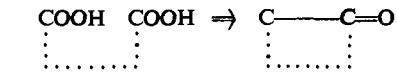
Intermolecular allylic coupling ($Ni(CO)_4$)



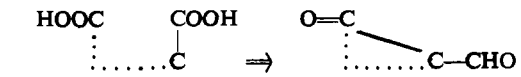
*Cope rearrangement

Ene synthesis

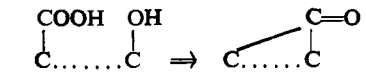
Unspecified Path Length



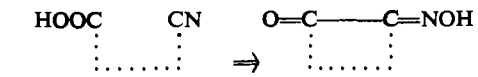
Oxidative cleavage of cyclic ketone



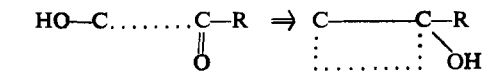
Oxidative cleavage of a 2-formyl ketone



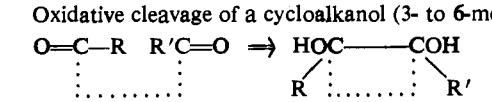
Baeyer-Villiger reaction and hydrolysis



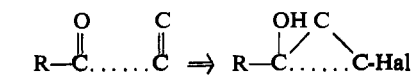
**Fragmentation of an α-ketoximine



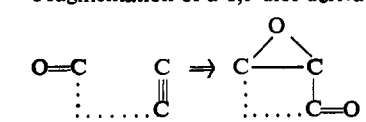
Oxidative cleavage of a cycloalkanol (3- to 6-membered ring)



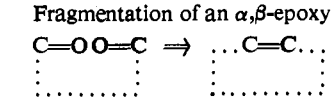
Oxidative cleavage of vic-glycol



**Fragmentation of a 1,3-diol derivative

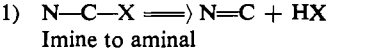


Fragmentation of an α,β-epoxy ketone to an acetylenic ketone

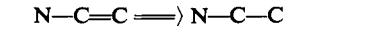


**Ozonolysis of C=C

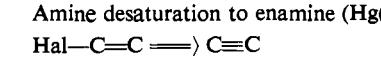
Transforms Which Do Not Affect Carbon–Carbon Bonds

(1) 

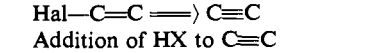
Imine to aminal



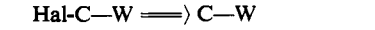
Amine desaturation to enamine ($Hg(OAc)_2$)



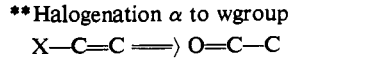
Addition of HX to C≡C



**Halogenation α to wgroup



Carbonyl to vinyl halide



Carbonyl to gem-dihalide, ketal, acetal, etc.

Carbonyl to acetal, ketal

(2)

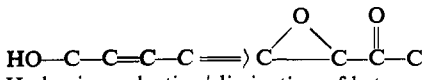
Halogenation α to carboxyl

Double bond hydroxylation

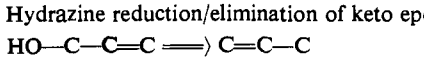
α-Ketoacetal, (1) reduction, (2) hydrolysis

Epoxide opening by nucleophilic attack

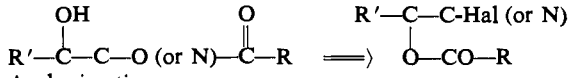
Trans reduction of a propargylic alcohol ($LiAlH_4$)



Hydrazine reduction/elimination of keto epoxide



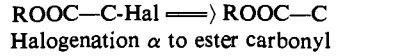
Allylic hydroxylation



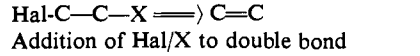
Acyl migration

Condensation with methylsulfinyl carbanion and iodination in methanol

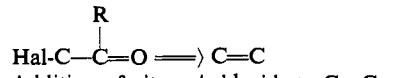
Amination α to ketone *via* oxime rearrangement



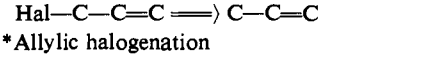
Halogenation α to ester carbonyl



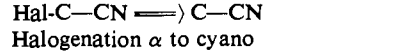
Addition of Hal/X to double bond



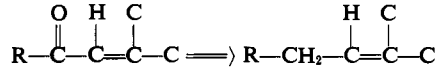
Addition of nitrosyl chloride to C=C



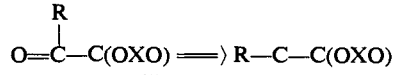
**Allylic halogenation



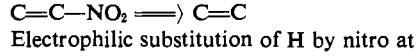
Halogenation α to cyano



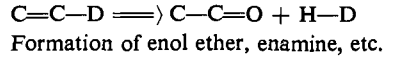
Allylic oxidation



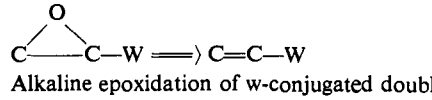
Ketone to α-diketone (SeO_2)



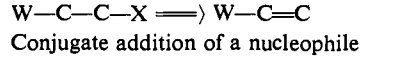
Electrophilic substitution of H by nitro at C=C



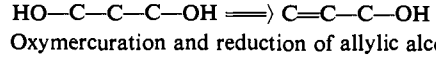
Formation of enol ether, enamine, etc.



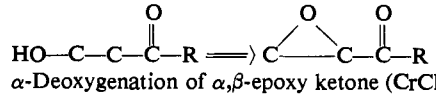
Alkaline epoxidation of w-conjugated double bond



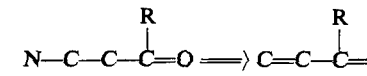
Conjugate addition of a nucleophile

(3) 

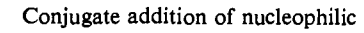
Oxymercuration and reduction of allylic alcohol



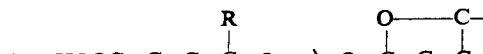
α-Deoxygenation of α,β-epoxy ketone ($CrCl_2$)



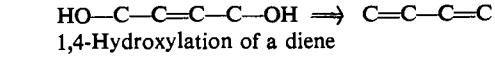
Conjugate addition of nucleophilic nitrogen



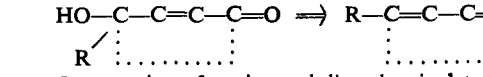
Nickel-promoted alkoxycarbonylation of allylic halide

(4) 

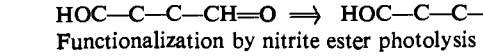
Hydrolysis of an enol γ-lactenone



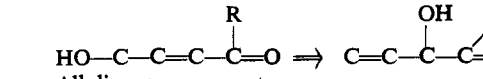
1,4-Hydroxylation of a diene



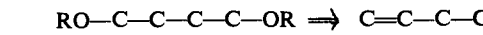
Oxygenation of conjugated diene by singlet oxygen



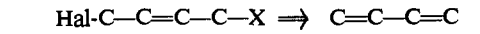
Functionalization by nitrite ester photolysis



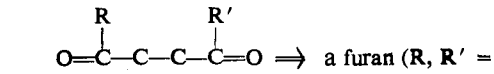
Allylic rearrangement



Alkoxy-directed addition of alcohol to C=C

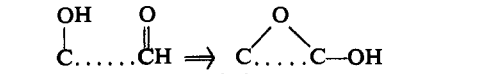


1,4-Halide/xgroup addition to 1,3-diene

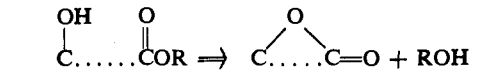


Hydrolysis of a furan to a 1,4-diketone

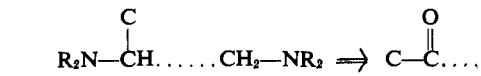
Unspecified Path Length



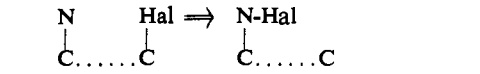
Cyclic hemiacetal cleavage



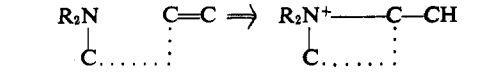
Lactone hydrolysis



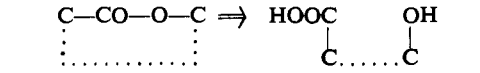
Double reductive amination



Hofmann-Loeffler reaction



Hofmann degradation of heterocycle



Lactone formation