Computer-Assisted Synthetic Analysis. A Generalized Procedure for Subgoal Transform Selection Based on a Two-dimensional Pattern Language

Alan K. Long,* John C. Kappos,† Stewart D. Rubenstein, and Gary E. Walker

Department of Chemistry, Harvard University, Cambridge, Massachusetts 02138

Received October 12, 1993•

A new capability for subgoal transform selection has been implemented in the LHASA program for computer-assisted synthetic analysis. The new subgoal system employs a transform cross-referencing mechanism which is based on representation of retron and precursor substructures in a two-dimensional (2-D) pattern language. 2-D patterns in the LHASA transform knowledge base are parsed by an offline compiler to perceive the differences between retron and precursor substructures. These differences are canonicalized to a "change code" which uniquely describes the conversion effected by each transform. During run-time processing, a structurally simplifying tactical combination is selected and its retron 2-D pattern is compared to the target molecule. In the absence of an exact match, the differences between the pattern and target are perceived and canonicalized to a "difference code". Candidate subgoal transforms are selected by searching the LHASA knowledge base for transforms whose change codes match the difference code of the required subgoal step. Sample antithetic analyses are included, and future extensions to 2-D pattern-keyed subgoal capabilities are discussed.

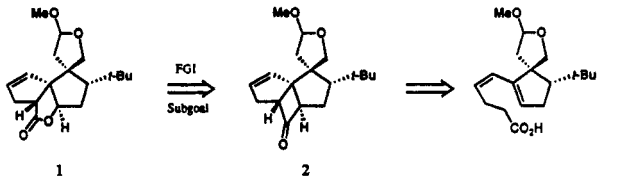
I. INTRODUCTION

The computer program LHASA, which has been under continuous development for more than 20 years, is intended to assist chemists in designing multistep retrosynthetic routes to complex organic molecules.[1] The program accepts as input a target molecule drawn in the language of structural formulas that is common to all organic chemists. A perception of the target is conducted to identify molecular features that influence the development of retrosynthetic routes (e.g. functional groups, stereocenters, rings, etc.). The user is then prompted to specify a strategy and substrategy, or tactic, to guide the retrosynthetic analysis.[2] For instance, the retrosynthetic disconnection of "strategic" bonds[3] or the application of important ring-disconnection transforms (retroreactions) such as the Diels–Alder transform[4] may be designated as goals of the retrosynthetic analysis. Finally, the program selects transforms from its chemical knowledge base in accordance with the processing tactic and applies these transforms to generate antithetic precursors which are structurally simpler than the target. The user selects one of the precursors for further processing, and the analysis proceeds in an interactive fashion until readily available hypothetical starting materials are obtained.

The overarching objective of retrosynthetic simplification is accomplished by applying "goal" transforms to the target. Goal transforms effect simplification by disconnecting carbon–carbon or carbon–heteroatom bonds, reconnecting long appendages (especially those containing stereocenters), or rearranging molecular skeletons. Goal transforms also typically require relatively complex "retrons", or minimum molecular substructures for transform application. Retrons thus correspond to structural elements in the products of synthetic reactions.

In LHASA, goal transforms are grouped according to retron complexity. The category with the least complex keying elements are classified as "one-group" transforms. These transforms require that the target structure contain a particular functional group and a path of carbon–carbon bonds extending from the group origin. For example, the Organometallic Addition to Carbonyl transform is keyed by a hydroxyl and a path of one bond. "Two-group" transforms require more complex retrons for execution, as two functional groups are specified along with the path between them.[5] The Michael Addition transform is a two-group transform keyed by a pair of ketones separated by a path of four bonds. A third group of transforms in LHASA depends on a more general keying mechanism, which permits specification of arbitrarily complex substructures, and is based on a one-dimensional pattern language.[6]

A direct consequence of the complexity of goal transform retrons is that only a few goal transforms will be directly keyed by any given target structure. Many goal transform retrons, however, will partially match a target structure. In such cases of inexact matching, it is frequently possible to apply nonsimplifying transforms to rectify mismatches. These "subgoal" transforms usually manipulate functional groups without disconnecting carbon–carbon bonds. For example, structure **1**, a precursor to Ginkgolide *B*,[7] contains only a



1 2

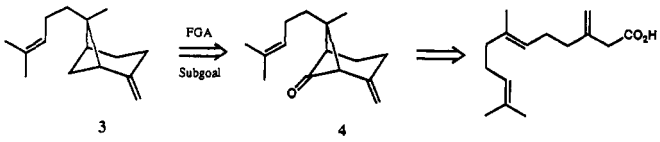
partial retron for the [2 + 2] Ketene Cycloaddition transform, which is keyed by a four-membered-ring ketone. The five-membered-ring lactone **1**, however, can be converted antithetically to the substituted cyclobutanone **2** by the Baeyer–Villiger transform. Structure **2** contains the full retron for the [2 + 2] Ketene Cycloaddition and is greatly simplified by application of this goal transform.

In the LHASA program, subgoal transforms are categorized by the type of functional group manipulation they effect. By far the largest group of subgoal transforms are those that perform functional group interchanges (FGIs), as exemplified by the transformation of lactone to ketone in the step **1** ⟹ **2**.

† Recipient of a Fannie and John Hertz Predoctoral Fellowship.
• Abstract published in *Advance ACS Abstracts*, June 1, 1994.

0095-2338/94/1634-0922$04.50/0

A second group of subgoal transforms effect functional group additions (FGAs) and are typified by the ketone reduction transform, as shown in the step **3** ⇒ **4** from a sequence used in the synthesis of *β-trans*-Bergamotene.[8] A third and less common class of subgoals perform functional group removals (FGRs).

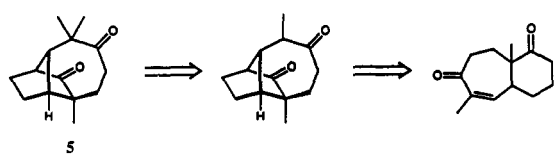


3 4

During the early development of LHASA it was realized that chemists performing retrosynthetic analyses can easily detect full substructures which key powerful goal transforms. Chemists, however, are much less adept at recognizing partial retrons of goal transforms for which execution will require mismatch rectification. Therefore, it was expected that "intelligent" machine selection and application of subgoals would lead to formulation of the most nonobvious and provocative retrosynthetic routes. To this end, LHASA was equipped with the ability to perform single FGIs, two parallel FGIs,[5] sequential FGIs,[9] and single FGAs to permit the application of structurally simplifying one-group and two-group transforms. In addition, the program was empowered to generate long sequences of FGI and FGA subgoals under the explicit guidance of knowledge-base tables which execute important ring-forming processes.[4] These facilities for performing mismatch rectification have since proved critical to the ability of LHASA to generate intelligent multistep retrosynthetic sequences. In fact, the distinction of the LHASA program in the field of computer-assisted synthesis as a true expert system,[10] rather than as a data retrieval program, is due in part to these subgoal capabilities.

Despite considerable gains that have been made in subgoal technology, the rectification of mismatches for goal transform application remains a deficiency in the LHASA program. The difficulty is a consequence of the mechanism LHASA uses to select subgoal transforms. All FGI, FGA, and FGR transforms have a common characteristic: the molecular changes they effect can be accurately described by considering only their retron (subject) and precursor (object) functional group types. For example, in target structure **1** LHASA perceives a lactone. When the [2 + 2] ketene cycloaddition transform is considered by the program, it recognizes that a cyclic ketone retron substructure is required. On the basis of this information, a request is issued for a retrosynthetic conversion of the lactone to a ketone. Since each entry in the LHASA knowledge base of subgoal transforms is cross-referenced by subject and object functional group types, identifying the Baeyer–Villiger transform as a candidate for the required subgoal step is a straightforward task (Figure 1). Similarly, FGA and FGR transforms are cross-referenced by the functional group types they install and remove, respectively.

There are many transforms, however, that would be useful as subgoals but that do not lend themselves readily to classification in this simplified functional group-based format. For example, in analyzing target **5**, a precursor of Longifolene,[11] the Michael Addition transform, which is keyed by a 1,5-dicarbonyl, may be identified as an appropriate goal transform. In order to employ the Michael addition as a goal it would first be necessary to execute the Alkylation of Carbanion transform as a subgoal to remove a methyl group alpha to the large-ring ketone. This subgoal step, however, is not representable in terms of simple functional group changes, since it alters only a carbon–carbon single bond, and is therefore not available to LHASA for this purpose.



5

```
        TRANSFORM 879
        NAME Baeyer-Villiger Oxidation

...March (2nd ed) 990
...
        TYPICAL*YIELD                    GOOD
        RELIABILITY                      FAIR
        REPUTATION                       EXCELLENT
        HOMOSELECTIVITY                  FAIR
        HETEROSELECTIVITY                FAIR
        ORIENTATIONAL*SELECTIVITY        FAIR
        CONDITION*FLEXIBILITY            FAIR
        THERMODYNAMICS                   EXCELLENT
...
...    O          O
...    "          "
... C-C-O-C => C-C-C
...
        ...PATH CHANGE 0
(1)     SUBJECT GROUP IS ESTER
        OBJECT GROUP IS KETONE
        STUDENT
        BROKEN*BONDS BOND1*1 BOND1*3
        ...
        KILL IF ATOM*1 IS IN A RING OF SIZE 4
        KILL IF THERE IS A FUNCTIONAL GROUP ON  &
           ALPHA TO ATOM*1 OFFPATH
        KILL IF THERE IS ANOTHER WITHDRAWING BOND ON  &
           ALPHA TO ATOM*1 OFFPATH
        KILL IF THERE IS ANOTHER HETERO ATOM ALPHA TO CARBON1*2
        KILL IF CARBON1*2 IS MULTIPLY BONDED
        SAVE AS 1 ALPHA TO ATOM*1 OFFPATH
        SAVE AS 2 CARBON1*2
        ....
           DELETE HETERO1*1
           JOIN CARBON1*1 AND SAVED*ATOM 2
           RETAIN AT SAVED*ATOM 2
        ....
        REDRAW THE PRECURSOR
        CONDITIONS Peracid/50
        CONDITIONS Peroxide/alk
        KILL IF PLUS CHARGE BETTER ON SAVED*ATOM 1 THAN ON  &
            SAVED*ATOM 2  ...Undesired direction of migration likely
```

Figure 1. Sample entry from the LHASA subgoal knowledge base. (1) Traditional subgoal cross-referencing information. The "SUBJECT GROUP" is the retron functional group on which the transform operates. The "OBJECT GROUP" is the precursor functionality generated by the transform. For functional group interchange transforms (FGIs), such as the Baeyer–Villiger transform, subject and object functional group information is sufficient to characterize the changes effected.

This deficiency in the subgoal capabilities of LHASA underscores the inadequacy of the original transform cross-referencing mechanism. The transforms of synthetic organic chemistry effect molecular changes which are far too diverse to be subclassified on the basis of functional group changes alone. A more general cross-referencing mechanism, which would permit utilization as a subgoal of any transform known to LHASA, is required. The new system must be flexible enough to permit characterization of the molecular changes from any transform. This generality mandates that the new cross-referencing mechanism be based, at its lowest level, on fundamental atom and bond changes.

II. THE TWO-DIMENSIONAL PATTERN LANGUAGE

To accomplish generalized transform cross-referencing, a system of two-dimensional (2-D) patterns has been developed. Each LHASA transform that is useful as a subgoal contains a pair of 2-D patterns which depict its retron and precursor substructures. A sample header from a transform which employs 2-D pattern cross-referencing is shown in Figure 2. The retron and precursor patterns appear in the transform header after the substrate-independent utility rating[12] and are separated by a retrosynthetic arrow (⇒). The patterns are written in a language of "superatoms", which are connected in two dimensions by bonds. Superatoms are classified as being either "primitive" or "complex". Primitive superatoms

```
        TRANSFORM 37
        NAME Aldol Condensation

...March 692; House 629; B+P 235
...Chem. Lett., 1195 (1983)
...
        TYPICAL*YIELD                       VERY*GOOD
        RELIABILITY                         GOOD
        REPUTATION                          EXCELLENT
        HOMOSELECTIVITY                     FAIR
(1)     HETEROSELECTIVITY                   GOOD
        ORIENTATIONAL*SELECTIVITY           NOT*APPLICABLE
        CONDITION*FLEXIBILITY               GOOD
        THERMODYNAMICS                      EXCELLENT
...
...                          O  H
...                          "  |
... C=C-COR,CHO,COOR    =>   C  C-COR,CHO,COOR
...                          |
...                          H
...
           (2)                      (3)

        ...PATH 1 BOND
(4)     GROUP*1 MUST BE OLEFIN
        GROUP*2 MUST BE KETONE OR ALDEHYDE OR ESTER
        DISCONNECTIVE
        STUDENT
        BROKEN*BONDS BOND1*1
        ...
```

Figure 2. Sample transform header for the aldol condensation. (1) The substrate-independent utility rating. ((2) and (3)) The retron and precursor 2-D patterns, respectively. (4) Transform-keying information. The CHMTRN language qualifiers,[16] mechanism commands, condition statements, etc., appear after the keying information but are omitted from this sample for brevity.

depict a single atom type and are usually denoted by the atomic symbol of the element. Complex superatoms correspond to a grouping or cluster of atoms of specified connectivity. For example, the complex superatom "-COOR" used in the pattern of Figure 2 is a common shorthand notation for the ester functional group and can be expanded to "-C(=O)-O-C". The superatom types available for 2-D patterns are defined for LHASA in the data file SUPERATOM.DAT (see Figure 3). The symbols which join the superatoms in a 2-D pattern represent bonds of specific orders. Single, double, and triple bonds are denoted by the characters "-", "=", and "#", respectively.

Since 2-D patterns contain such explicit atom and bond information for retron and precursor substructures, perception of the fundamental molecular changes effected by any transform is possible. Furthermore, by comparing the retron 2-D pattern of a structurally simplifying transform or tactical combination (TC)[13] to a target structure being analyzed, the atom and bond differences requiring rectification are discernable. Appropriate subgoal transforms may then be identified by matching the description of atom and bond changes for transforms to the molecular differences requiring rectification.

Since tactical combinations are encoded in the 2-D pattern language,[14] the TC environment seemed ideal for developing a prototype pattern-keyed subgoal capability. The implementation of a single-step 2-D-pattern-keyed subgoal procedure for TCs is described in this article.

III. IMPLEMENTATION

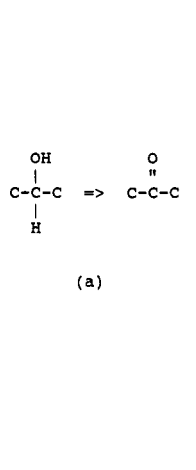
The selection of subgoal transforms based on 2-D pattern information is a relatively complex process which required much new coding in the LHASA program.[15] The highest-level executive subroutine which guides the performance of subgoals for TCs is TCEXEC.[14] There are numerous lower-level executives, however, which are instrumental in the selection of subgoal transforms. Some of these executives are employed offline, during compilation of transform tables, while others are executed at run time when the target structure is known. In a general sense, the process of subgoal transform

```
-H       -H
-HELIUM  -HELIUM
-LI      -LI
-BE      -BE
-B       -B
-C       -C
-N       -N
-O       -O
 .        .
 .        .
 .        .
-I       -I
-XE      -XE
-Z       -Z
-PLUS    -PLUS
-X       -F,-CL,-BR,-I
=N       =N
=NH      =N-H
=NR      =N-C
#N       #N
/N       -N
%N       %N
%NN      %N%N
%NO      %N%O
%NR      %N%C
-NCO     -N=C=O
-NR2     -N(-C)-C
-NHR     -N(-H)-C
-NH2     -N(-H)-H
-NH      -N-H
-NRN     -N-N
=N2      =N=N
-N3      -N=N2
-NO      -N=O
-NX      -N-X
=NNH2    =N-N
=NOH     =N-O
-NCR2    -N=C
-NC      -N#C
-N2R     -N=N
-NHOH    -N(-H)-O
-NROH    -N(-C)-O
-NOH     -N-O
-NO2     -N(=O)-O
-NCONH   -N-C(=O)-N-H
-NHCON   -N(-H)-C(=O)-N
-NRCON   -N(-C)-C(=O)-NR2
-OCON    -O-C(=O)-N
-SIR3    -SI(-C)(-C)-C
#O       #O
=O       =O
/O       -O
%O       %O
%OR      %O%C
%ON      %O%N

-OSIR3   -O-SIR3
-NCOR    -N-C(=O)-C,
         -N-C(=O)-H,
         -N(-C)-C=O,-N-COR
-COOCOR  -C(=O)-OCOR
-W       -COR,-CHO,-COOR,
         -CONH2,-CONHR,
         -CONR2,-COX,-COSR,
         -C=N,-C=NOH,-CN,
         -NO2,-SOR,-SO2R,
         -SO3R,-CX3,-C=C-W,
         -COOCOR
-CO2     -C(=O)=O

         -OSIR3  -O-SIR3
         -OP     -O-P
(2)      -OR     -O-C
         -OH     -O-H
         -OOH    -O-OH
         -OOR    -O-OR
         -ONR2   -O-N
         -NCOOR  -N-C(=O)-O-C,
                 -N-C(=O)-OR,
                 -N(-C)-C(=O)-OR
         -OCOOR  -O-C(=O)-O-C,
                 -O-C(=O)-OR
         -OSO2R  -O-S(=O)(=O)-C
         =C      =C
         =CH2    =C(-H)-H
         #C      #C
         %C      %C
         -COR    -C(=O)-C
         -CHO    -C(=O)-H
         -COOH   -C(=O)-OH
(1)      -COOR   -C(=O)-OR
         -CONH2  -C(=O)-NH2
         -CONHR  -C(=O)-NHR
         -CONR2  -C(=O)-NR2
         -COX    -C(=O)-X
         -CNR    -C=N
         -CN     -C#N
         -CX2    -C(-X)-X
         -CX3    -C(-X)(-X)-X
         -COHOR  -C(-OH)-OR
         -COROR  -C(-OR)-OR
         -CH3    -C(-H)(-H)-H
         -CH2OH  -C(-H)(-H)-OH
         -PH     -C%C
         -AR     -C%C,-C%N,-C%S,-C%O
         =S      =S
         /S      -S
         %S      %S
         %SR     %S%C
         -SH     -S-H
         -SR     -S-C
         -SSR    -S-SR
         -SO     -S=O
         -SO2    -S(=O)=O
         -SO3    -S(=O)(=O)-O
         -SOR    -S(=O)-C
         -SO2R   -S(=O)(=O)-C
         -SO3R   -S(=O)(=O)-OR
         -SCN    -S-CN
         -COSR   -C(=O)-SR
         -SER    -SE-C
         -OCOR   -O-C(=O)-C,
                 -O-C(=O)-H,
                 -O-C=O,-O-COR
```

Figure 3. Definitions of superatoms which can be used in 2-D pattern writing. Each superatom type is followed by an expansion to a line notation which delineates precisely the connectivity of the atoms which comprise the superatom. In certain cases, a second superatom is used within the definition of a superatom. For instance, the "-COOR" superatom (1) is expanded to "-C(=O)-OR". The "-OR" superatom used within this definition can be expanded to "-O-C" (2), so that the full definition of "-COOR" is "-C(=O)-O-C".

selection involves (1) offline perception of changes effected by knowledge-base transforms, (2) online formulation of a description of the molecular differences requiring rectification, and (3) utilization of the transform changes and molecular difference description to select subgoal transforms. The algorithms used to accomplish these steps will now be described in detail.

A. Perception of Changes Effected by Transforms. Transform knowledge-base tables are compiled offline to generate binary files read by LHASA during processing. When the CHMTRN compiler[16] encounters a 2-D pattern, a call is issued to the routine PARSE_PATTERNS. This executive routine guides the textual parsing of 2-D patterns and the identification of significant substructural features (e.g. functional groups). The parsing of pattern text results in the filling of numerous data structures with pattern information. For example, the number of path atoms is determined, and binary sets are filled with positions of carbon path atoms and with positions which bear off-path superatoms. The types of all the superatoms in the pattern and information about the atoms adjacent to each



(a)

```
OTHERATOM[RETRON]                [1] 1
                                 [2] 2
                                 [3] 3

OTHERATOM[PRECURSOR]             [1] 1
                                 [2] 2
                                 [3] 3

OFFP_ATOM[ABOVE][RETRON]         [1] 0
                                 [2] 1
                                 [3] 0

OFFP_ATOM[BELOW][RETRON]         [1] 0
                                 [2] 2
                                 [3] 0

OFFP_ATOM[ABOVE][PRECURSOR]      [1] 0
                                 [2] 1
                                 [3] 0

OFFP_ATOM[BELOW][PRECURSOR]      [1] 0
                                 [2] 0
                                 [3] 0
```

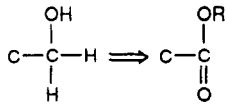
(b)

Figure 4. (a) Retron and precursor 2-D pattern substructures for the Ketone Reduction transform. (b) Data structures filled by module ID_MAP which indicate the correspondence between retron and precursor atoms. The value of an element in the two-dimensional array OTHERATOM[RETRON][n] is the number of the precursor path atom which is mapped to the nth path atom in the retron. The corresponding precursor array gives the number of the retron path atom which is mapped to the *n*th path atom in the precursor. The three-dimensional array OFFP_ATOM[ABOVE/BELOW][RETRON][n] is filled with a different nonzero number for each off-path substituent in the retron. The corresponding precursor array is filled with nonzero numbers for each precursor substituent as well. If a precursor substituent maps to a retron substituent, then they share the same number in corresponding elements of their OFFP_ATOM arrays. If the precursor substituent does not map, then it is assigned a number which is different from all other retron substituents.

path position are stored in arrays. These data are critical to all subsequent use of patterns.

After rudimentary information has been gleaned from the 2-D patterns, PARSE_PATTERNS undertakes perception of changes effected by the transform by analyzing the differences between the retron and precursor substructures. This phase begins with a call to function ID_MAP to establish a correspondence between retron and precursor superatoms. Such a mapping is critical to the identification of the atoms and bonds altered by a transform. For example, to properly characterize the changes effected by the Ketone Reduction transform, which is depicted by the patterns in Figure 4a, the correlation between the path carbon atoms and the off-path oxygen atoms in the retron and precursor patterns must be recognized. Then, it is easy to perceive that this transform increases the carbon–oxygen bond order by one while removing hydrogen atoms from both the carbon and oxygen atoms. The data structures that ID_MAP fills to represent this atom mapping are shown in Figure 4b.

The mapping process is often more complicated than for the Ketone Reduction transform. The 2-D patterns for the reduction of an ester to a primary alcohol illustrate a less straightforward case:



In this example, it is uncertain which of the two oxygen atoms in the precursor should be mapped to the lone retron oxygen. Furthermore, the two mappings lead to very different perceptions of transform changes. Thus, all possible mappings are generated, and each is used in turn to perceive changes. To achieve this level of generality, ID_MAP employs a recursive algorithm which is depicted by the simplified flow chart in Figure 5. ID_MAP returns a linked list[17] of all possible mappings. Each data cell of the list contains the OFFP_ATOM arrays, as described in Figure 4b, for one mapping.

PARSE_PATTERNS unpacks the list returned from ID_MAP and, for each mapping, calls the executive routine ID_CHANGES. ID_CHANGES guides the translation into numerical codes of the atom and bond differences between retron and precursor substructures. This executive routine first issues a call to the function CALC_MATRIX. CALC_MATRIX assigns unique numerical values to each retron and precursor superatom and records in a two-dimensional matrix the orders of the bonds joining all atoms. A sample bond order matrix for the Aldol Condensation transform is shown in Figure 6a. A second array is filled with the primitive atom types of all superatoms which appear in the structure. Generating the bond matrix and the atom type array represents the first step toward obtaining a numerical expression of molecular differences.

ID_CHANGES then calls the function CALC_CHANGE_MATRIX to translate the bond order matrix into an expression of the changes effected by the transform. It is during this stage that bonds which are unchanged between the retron and precursor are disregarded. The translation involves two operations. First, the precursor bond order entries are subtracted from the corresponding elements in the retron section of the matrix. The resulting array contains numbers which represent the differences in bond orders between the retron and precursor. These numbers are then converted into bond alteration codes. The result is a second array, the bond order change matrix, which focuses only on bonds altered by the transform. The change matrix describing the bond alterations effected by the Aldol Condensation transform is shown in Figure 6b.

Finally, the change matrix is submitted to the canonicalization procedure in the routine CANONICALIZE_CHANGEMATRIX. This routine uses the values in the change matrix and the atomic numbers of the atoms that undergo bonding modifications to arrive at a single integer which uniquely expresses the changes effected by the transform. The integer, or "change code", is written to the binary output file which results from compilation of each transform table and is used during run-time processing as the cross-referencing identification number for the transform.

B. Describing Molecular Differences Which Require Rectification. During a LHASA processing session, when the target structure is known and a TC has been selected, a quantitative description of molecular differences is again required. In this case, however, the program is not perceiving the differences between a pair of 2-D patterns. Instead, the program seeks to quantify the differences between the target structure entered by the chemist and the retron 2-D pattern of a powerfully simplifying TC. To accomplish this perception, the program must first establish a mapping between the atoms of the retron 2-D pattern and the target structure.

The executive routine MAP_PATTERN is employed to direct the target-to-pattern mapping process. A simplified flow chart of this routine is shown in Figure 7. One of the calling arguments to MAP_PATTERN signals that the routine should allow mismatches, and thus return partial, or inexact, mappings as well as exact ones. MAP_PATTERN, however, will only formulate mappings for which all path carbon atoms are correlated to structure carbon atoms. In other words, on-path carbon atom mismatches are forbidden.

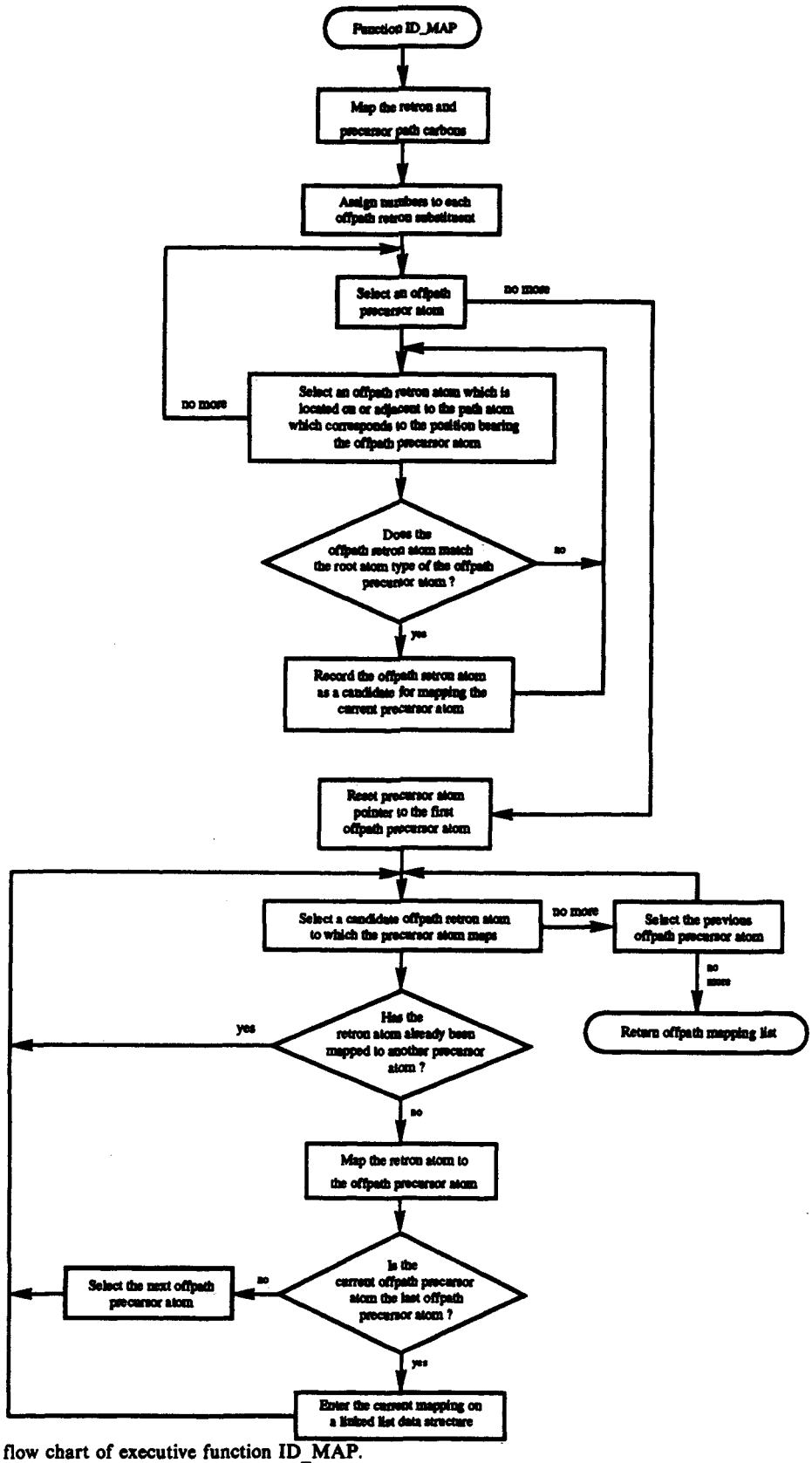
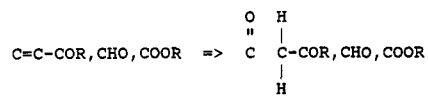


Figure 5. Simplified flow chart of executive function ID_MAP.

The mapping phase begins with the selection of a single structure atom to map to one of the path atoms. Although the atomic number of this "anchor" atom must match that of the path atom to which it maps, the anchor atom can be selected from any atom in the target. Thus, each appropriate structure atom must be considered in turn for the anchor position. The number of the anchor atom is placed in the MAP_ATOM array, which records the correspondence between pattern path atoms and structure atoms. Then, the mapping is grown out from the anchor position by function GROWOUT_MAP. The path positions adjacent to the mapped position are compared to the atoms adjacent to the mapped structure atom. If the atom types match, MAP_ATOM array entries are made to extend the mapping. When there are multiple target structure atoms of the correct type and connectivity to map a pattern atom, one atom is chosen for entry in the




```
Path atoms (P1, P2, and P3) are numbered 1, 2, and 3
Above precursor path atom 1 (AP1) = 4
Above precursor path atom 2 (AP2) = 5
Below precursor path atom 2 (BP2) = 6

(a) Bond order matrix:

   1    P1    0    0    2    0    0
   2     2   P2    1    0    1    1
   3     0    0   P3    0    0    0
   4     0    1    0  AP1    0    0
   5     0    0    0    0  AP2    0
   6     0    0    0    0    0  BP2

(b) Bond order change matrix:

   1    P1
   2     4   P2
   3     0    0   P3
   4    -4    0    0  AP1
   5     0   -1    0    0  AP2
   6     0   -1    0    0    0  BP2

Bond alteration codes:

Single bond disconnect:   1
Single bond reconnect:   -1
Double bond disconnect:   4
Double bond reconnect:   -4
```

Figure 6. Retron and precursor 2-D patterns for the aldol condensation transform. (a) The corresponding bond order matrix with numerical entries for each bond in the retron and precursor patterns. The diagonal entries represent the superatoms in the substructures. The orders of the bonds joining the atoms in the retron are recorded below the diagonal, while precursor bonds are listed above the diagonal. This matrix shows, for example, that atoms 1 and 4 (path atom 1 and the atom above it) are doubly bonded in the precursor pattern, since a "2" is entered in their precursor matrix element. (b) The change matrix, which focuses on the bonds altered by the transform. This matrix is generated from the bond order matrix by subtracting each precursor bond matrix entry from the corresponding retron entry and translating the resulting numbers into bond alteration codes. For example, retron entry (2,1) – precursor entry (1,2) = (2 – 0) = 2, which indicates retrosynthetic disconnection of a double bond. The bond alteration code for this disconnection is 4, which appears in element (2,1) of the change matrix. Note that positive change matrix entries correspond to retrosynthetic disconnections, while negative entries represent reconnections.

MAP_ATOM array and the others are saved on a stack for later consideration. This process is repeated until all path atoms have been correlated to structure atoms.

After completing a path-atom mapping, MAP_PATTERN calls MAP_ONE_ATOM, for each atom in turn, to analyze bond orders and off-path superatom mappings. MAP_ONE_ATOM looks at superatoms adjacent to a given path origin to ensure that all neighbors map to the target and that the bond orders to the origin atom are correct. To map complex superatoms (e.g., "-COOR") to the target, the superatom must be expanded to the explicit atom and bond notation which is given in the definition file SUPERATOM.DAT (Figure 3). Superatom expansion and the mapping of constituent atoms are complex processes. Therefore, a separate module of functions, CHECKTYPE, is employed by MAP_ONE_ATOM to execute superatom mapping.

When MAP_ONE_ATOM encounters an inconsistency between structure atoms and path atoms, it records the mismatch in data structures to be stored with the mapping. The location, nature, and number of mismatches are among the information retained. In the case of a bond order mismatch, it may be necessary for a subgoal step to disconnect an unmapped carbon atom or heteroatom from the position bearing the mismatch. For example, with the mapping shown between the retron pattern and the target in Figure 8, the required subgoal must transform the tertiary alcohol to a ketone. This transformation is possible only if the unmapped carbon appendage is disconnected from the target. MAP_ONE_ATOM must identify those appendages suitable for disconnection and designate them in the appropriate data structures.

Upon return to MAP_PATTERN, the completely scrutinized current mapping is stored, along with mismatch information, in a linked list data structure. MAP_PATTERN then enters a backtracking phase, during which it considers alternative mappings that were accumulated on the atom stack in the growing-out process. The backtracking procedure is illustrated in Figure 9. Alternative mappings for the last mapped path atom are considered first. After these are exhausted, the second-to-last mapped position is permuted. Each new mapping generated during this phase is sent to GROWOUT_MAP to ensure completeness and to MAP_ONE_ATOM to assess mismatches before addition to the mapping list structure. By repeating this backtracking process, every mapping between the target structure and retron pattern is identified.

For each mapping returned from MAP_PATTERN, the executive routine PARTIAL_MAP is executed. PARTIAL_MAP guides the formulation of a bond order matrix for the target structure and the retron 2-D pattern bonds. A simplified flow chart for PARTIAL_MAP is shown in Figure 10. This procedure is analogous to the bond matrix determination conducted by CALC_MATRIX during transform compilation (*vide supra*). Run-time bond matrix generation, however, is complicated by the necessity of accounting for bonds that are buried within complex superatoms. Hidden superatom bonds must be entered into the matrix when the mismatch to be rectified involves these buried superatom bonds. For example, all path atoms and bonds map precisely to the target for the sample mapping shown in Figure 11a. Careful examination, however, reveals that the off-path ester superatom does not map to the structure. In such cases, the superatom must be expanded to explicit notation to facilitate accounting in the matrix for superatom bonds which do not correctly map to the target structure. For the example in Figure 11a, the "-COOR" superatom is expanded to "-C(=O)-O-C". From this definition it is obvious that precursor matrix entries are required for both the doubly and singly bonded oxygen atoms of the superatom, and that retron matrix entries are needed for the target structure alcohol bond and methyl bonds which will be disconnected by the transform. The resulting bond order matrix is given in Figure 11b.

After the bond order matrix is complete, numerical expression and canonicalization of molecular differences proceed as for pattern compilation. In fact, change matrix determination and canonicalization to an integer code are performed by the same routines, CALC_CHANGE_MATRIX and CANONICALIZE_CHANGEMATRIX, that the offline procedure employs (*vide supra*). The result is a single integer, or "difference code", which uniquely describes the atom and bond changes required of the subgoal transform.

C. Subgoal Transform Selection. With numerical descriptions of the molecular difference (difference code) and the transform changes (change codes) in hand, candidate subgoal transforms can be efficiently selected from a large knowledge base. The selection is accomplished by simply comparing the difference code to the change codes for transforms available as subgoals. The high-level executive subroutine TCEXEC coordinates run-time access to knowledge-base change codes,

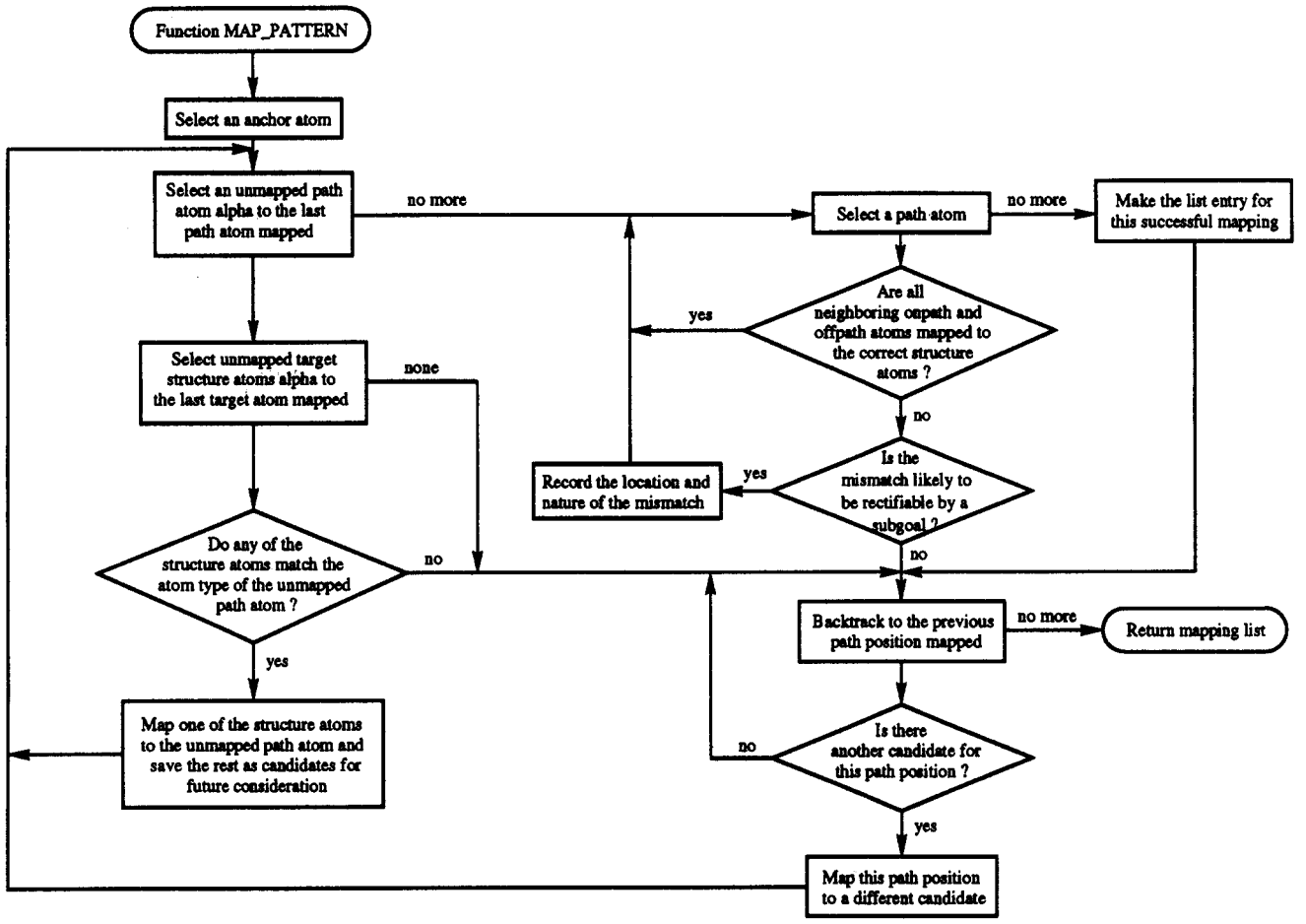


Figure 7. Simplified flow chart of executive function MAP_PATTERN.

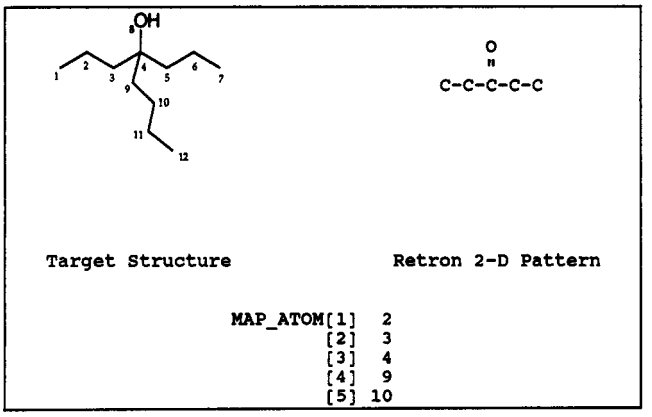


Figure 8. Mapping between a target structure and a retron 2-D pattern which requires a disconnective subgoal for rectification. The carbon–oxygen bond order must increase by one in this step. This alteration necessitates disconnection of a carbon appendage from structure atom 4. The unmapped carbon appendage beginning at structure atom 5 is appropriate for this disconnection.

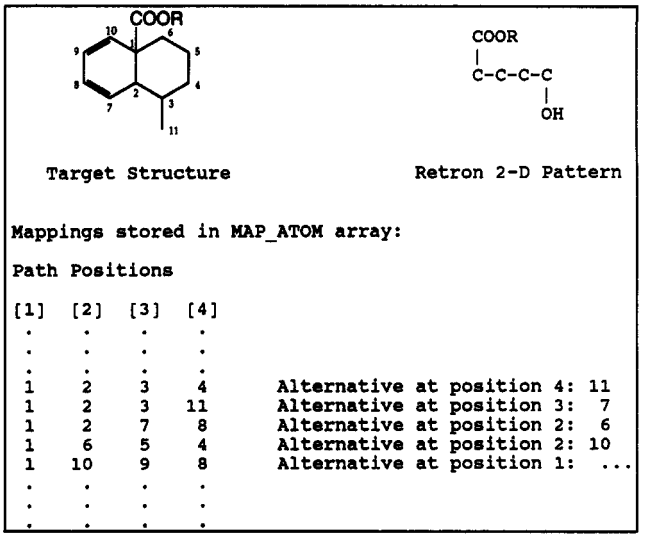


Figure 9. Backtracking phase of the mapping process. If structure atom 1 is mapped to the anchor atom of the 2-D pattern (the first path atom), then five inexact mappings are possible. After the first mapping (1, 2, 3, 4) is grown, alternatives are considered for the last position mapped (i.e., [4]). When alternatives for this position are exhausted, the next-to-last position is permuted. This procedure considers all correlations between pattern path atoms and target structure atoms in order to discover all mappings.

the generation of difference codes, and the selection of candidate subgoal transforms. A simplified flow chart for TCEXEC appears in Figure 12.

Upon entering TCEXEC, a call is issued to the routine FILL_CODE_ARRAY. FILL_CODE_ARRAY opens the binary output file from each transform table in turn and transcribes the change codes into an array. The array is sorted into ascending order[18] and used as an efficient means of accessing the knowledge base for all subsequent transform identification.

TCEXEC then opens a TC table and selects a TC for processing. This selection is subject to the tactical constraints specified by the user at the outset of the analysis.[14] The subordinate executive MAP_PATTERN (*vide supra*) is called to obtain a list of mapping arrays and data structures describing mismatches between the target structure and the retron 2-D pattern of the TC. The arrays are unpacked from the list by TCEXEC, and the number of mismatches for each mapping

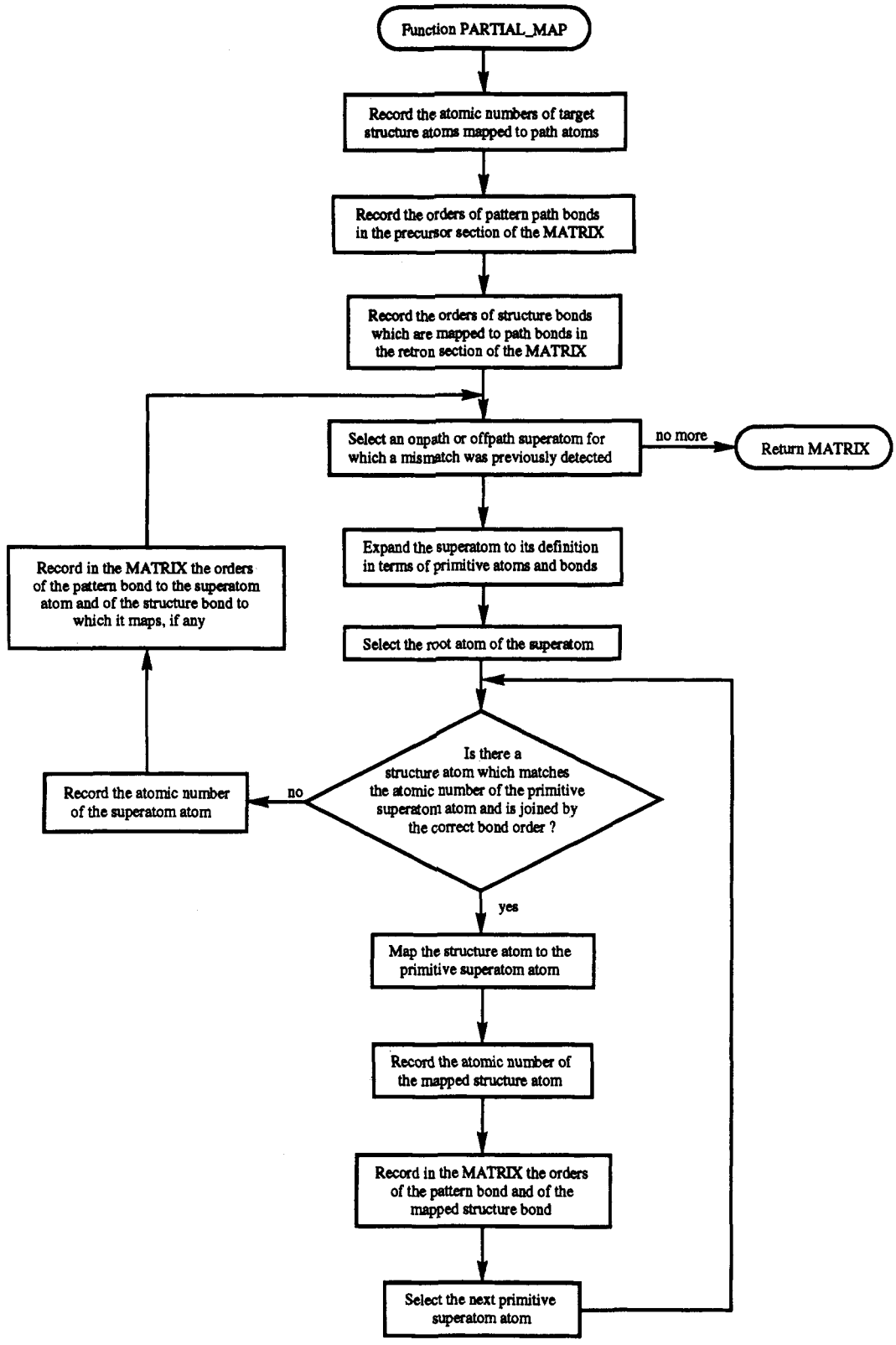
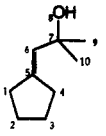


Figure 10. Simplified flow chart of executive function PARTIAL_MAP.

is examined. Mappings with only a single mismatch are considered before those with multiple mismatches. This hierarchy stems from the recognition that a single mismatch is more easily rectifiable by a one-step subgoal procedure than are several mismatches.

For each mapping appropriate for the subgoal hierarchy level, PARTIAL_MAP is called to set the bond order matrix, and CALC_CHANGE_MATRIX and CANONICALIZE_CHANGEMATRIX are called to generate the difference code. At this point, candidate subgoal transforms can be identified by comparing the difference code to the change codes in the transform code array. Before searching for transforms, however, a check is made to ensure that the subgoal being requested has not been attempted previously. This screening of duplicate subgoal requests[19] is conducted by the subroutine TCSGSCREEN. The purpose of duplicate screening is to avoid spending time generating subgoal precursors that were previously created or attempting subgoals that previously failed. The results of every subgoal request are saved on a list, and the list is checked before each new subgoal

(a) Mapping data:



Target Structure Retron 2-D Pattern

MAP_ATOM[1] 4
[2] 5
[3] 6

(b) Bond order matrix:

1	P1,4	1	0	0	0	0	0	0	0	0
2	1	P2,5	2	0	0	0	0	0	0	0
3	0	2	P3,6	1	0	0	0	0	0	0
4	0	0	1	A3,7	2	1	0	0	0	0
5	0	0	0	1	SA3	0	0	0	0	0
6	0	0	0	0	0	SA3	0	0	0	0
7	0	0	0	1	0	0	EA3	0	1	0
8	0	0	0	1	0	0	0	EA3	0	1
9	0	0	0	0	0	0	0	0	H8	0
10	0	0	0	0	0	0	0	0	0	H9

Figure 11. (a) Sample target structure and retron 2-D pattern mapping information. A mismatch is buried within the superatom "-COOR". By expanding this superatom to its definition in terms of primitive atoms and bonds, "-C(=O)-O-C", bond mismatches are noted for both the doubly and singly bonded oxygen atoms. Therefore, these bonds, as well as the hydroxyl and methyl bonds in the target structure, must appear in the bond order matrix. (b) Bond order matrix generated during run-time processing for the above mapping. P1,4 corresponds to the first path atom, which is mapped to target structure atom 4. A3,7 corresponds to the superatom above path atom 3, which is mapped to structure atom 7. All entries after A3,7 are a consequence of expansion of the -COOR superatom. The SA3 entries correspond to each of the oxygen atoms in the superatom. The EA3 entries correspond to the methyl groups to be disconnected from the target. H8 and H9 are precursor valency place holders for the methyl groups, which must have four attachments in the precursor to balance the four in the retron.

is attempted to see whether a former request matches the current one. If a match is found, and the former request generated precursors, then the node numbers of these precursors are returned to TCEXEC. On the other hand, if the former request failed to generate precursors, then the current subgoal and mapping are not attempted.

If the subgoal request has not been attempted previously, TCEXEC calls FIND_TFS to identify candidate subgoal transforms. FIND_TFS performs a binary search in the transform code array for entries whose change codes match the difference code. The set of matching transforms and the binary tables in which they reside are returned to TCEXEC.

TCEXEC then issues a call, based on the name of the table in which a candidate transform is located, to the appropriate goal or subgoal transform executive to perform the transform. Goal transform executives GPAIR, GSING, and PEXEC[20] are called with instructions to execute only specified transforms. Subgoal transform executives SNGFGI and SNGFGA, on the other hand, cannot be called to execute particular transforms. Instead, the interface routines TCFGI and TCFGA are employed to formulate calling arguments for SNGFGI and SNGFGA.[14] The subgoal offspring nodes which result are returned to TCEXEC in a binary set. The subroutine TWOD_DUP_LIST_ENTRY is called to make the duplicate screening list entry, since the outcome of the request is now known.

Finally, each subgoal precursor node is selected in turn for TC execution. MAP_PATTERN is called a second time to perform an exact mapping between the TC retron 2-D pattern and the subgoal precursor. This mapping ensures that the full TC retron was generated by the subgoal. If the mapping succeeds, the TC is executed as previously described.[14]

IV. RESULTS

To assess the utility of the new LHASA subgoal capabilities, it is useful to consider the output from some sample analyses. To this end, a number of simple targets were processed to execute specific TCs which were not fully keyed, but which required mismatch rectification. Equations 1–3 in Figure 13 are representative examples. The transforms employed to rectify mismatches in each case were previously unavailable to LHASA for use as subgoals. The Michael Addition and Aldol Condensation transforms used as subgoals in equations 1 and 2, respectively, are carbon–carbon disconnective "goal" transforms which entail molecular changes not readily cross-referenced by functional group. The Deconjugative Alkylation of Vinylogous Enolate transform employed as a subgoal in eq 3 effects in one step a pair of molecular changes: olefin isomerization and carbon–carbon bond disconnection. This ability to rectify multiple mismatches in one step, which is common to many structurally simplifying transforms, clearly demonstrates the power of using goal transforms as subgoals.

The program was also given a complex natural product on which to perform an exhaustive analysis using its entire TC knowledge base.[21] Several of the resulting retrosynthetic routes which rely on the pattern-keyed subgoal capabilities are included in Figure 14. The target, magellanine (**6**),[22] is a member of the *lycopodium* family of alkaloids which has not yet been prepared by total synthesis,[23] and, as such, permits an unbiased assessment of LHASA output. Without the subgoal facility enabled, only three TCs were keyed directly by substructures present in the target and executed.[24] The use of single-step pattern-keyed subgoals, however, resulted in performance of 18 more TC sequences. From the examples given in Figure 14, it is clear that TC application aided by the pattern-keyed subgoal procedure can produce retrosynthetic routes which are nonobvious and, on a theoretical plane, of high merit.

V. CONCLUSION

With a prototype 2-D pattern-keyed subgoal capability in place, it is interesting to speculate on future extensions and applications of the current technology. A first, obvious extension will be to employ this subgoal capability to rectify mismatches for the application of structurally simplifying transforms. Since nearly every transform in the LHASA knowledge base contains 2-D patterns for cross-referencing, the retron patterns of goal transforms can be used, in a fashion analogous to the use of retron TC patterns, to compute difference codes.

A long-term extension to current capabilities will be to execute multiple subgoal steps for rectification of single or multiple mismatches. Although selection of transforms for sequential subgoal operation is considerably more complicated than a simple comparison of a difference code with change codes, multistep pattern-driven subgoals are expected to be very powerful. Techniques for selecting sequences of subgoal transforms based on 2-D pattern cross-referencing information are currently under development.

Utilization of TCs as subgoal sequences is expected to aid in multistep subgoal generation by accomplishing retrosynthetic objectives in a highly directed manner. Implementation of TCs as subgoal sequences is made possible by the existence of a final precursor 2-D pattern in each TC in addition to a

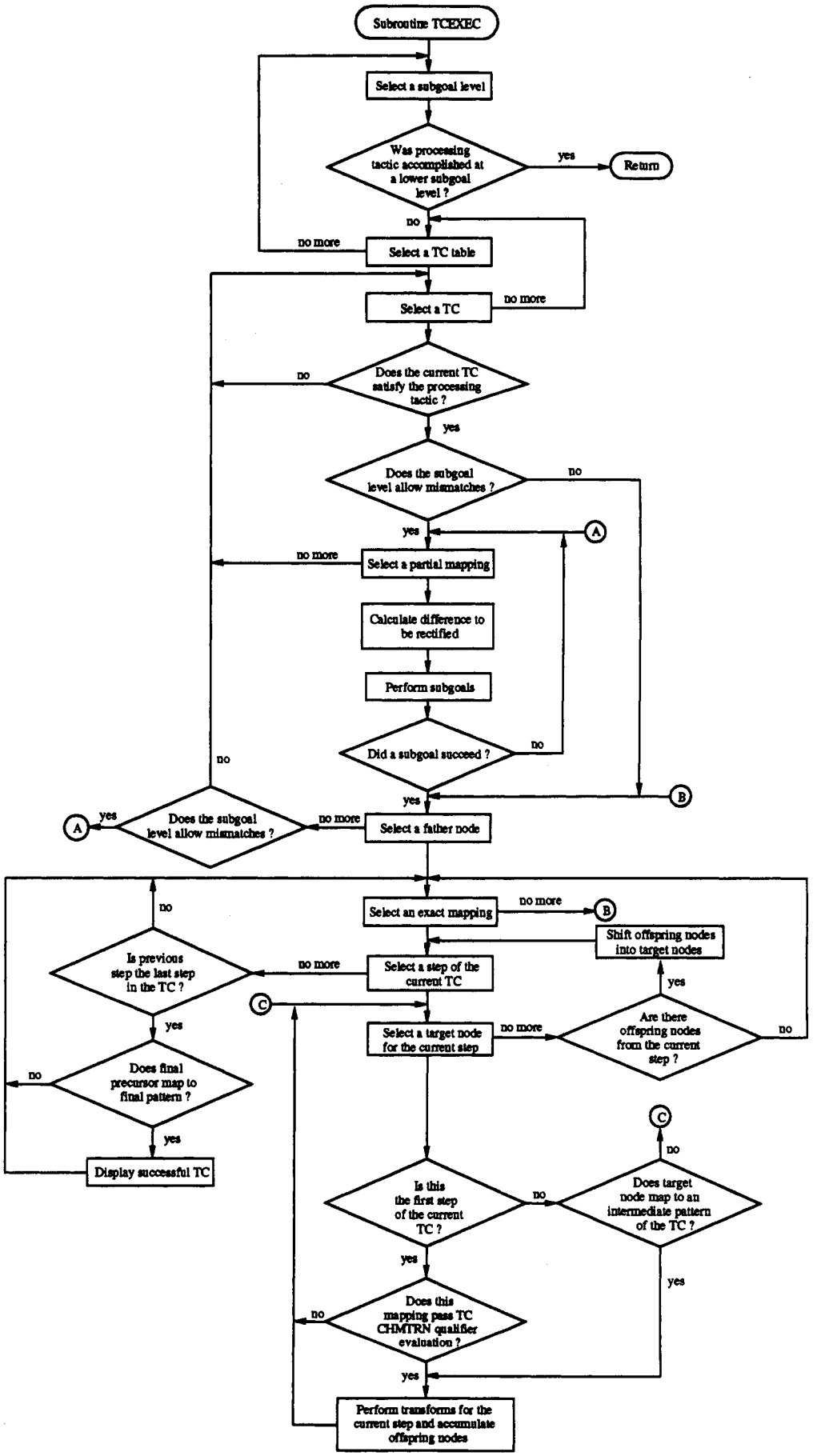


Figure 12. Simplified flow chart of executive subroutine TCEXEC.

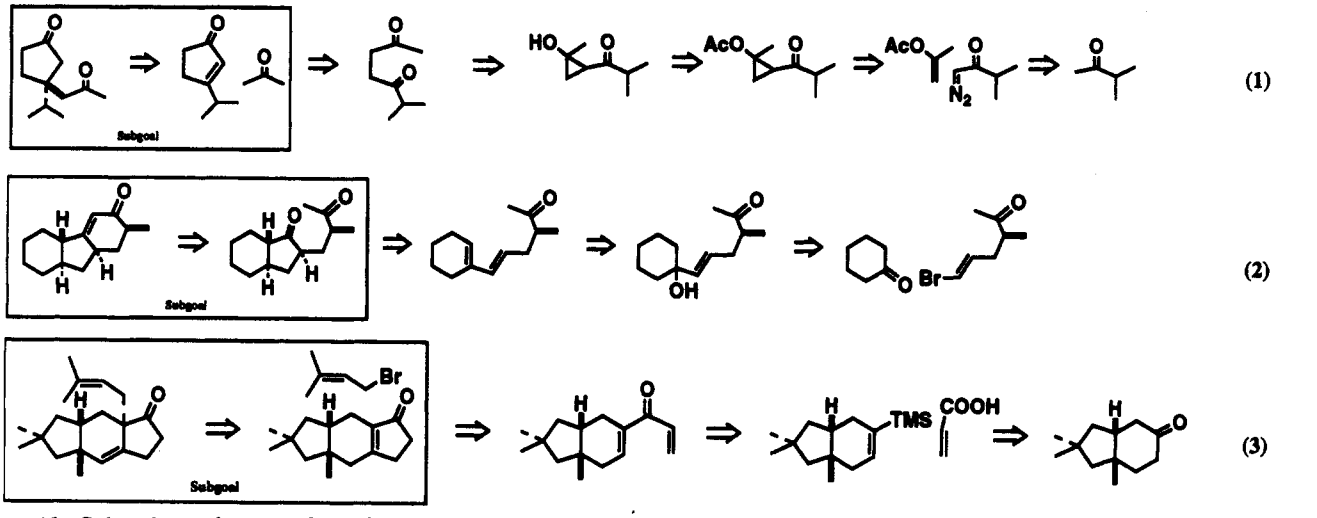


Figure 13. Subgoal transforms performed at the request of TCs that previously failed since the targets possessed incomplete retrons. These transforms were formerly unavailable to LHASA for application as subgoals.

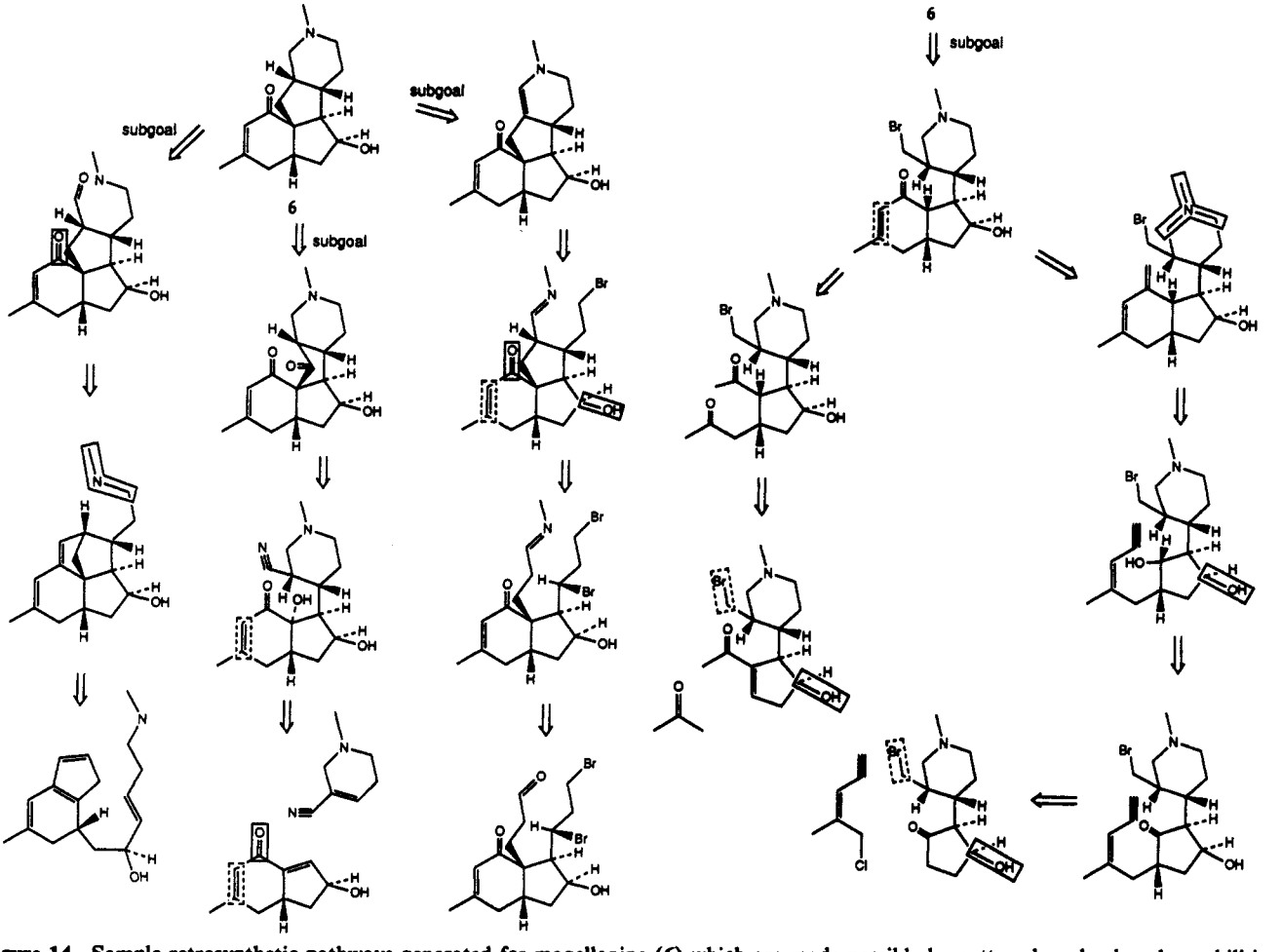


Figure 14. Sample retrosynthetic pathways generated for magellanine (6) which are made possible by pattern-keyed subgoal capabilities. Functional groups which appear in boxes are marked to indicate that protection is necessary to prevent a side reaction.[25]

retron pattern. Perception of the overall changes effected by a TC will be achieved by comparison of the retron pattern to the last precursor pattern.

Finally, the two-dimensional pattern language will gain application as a general transform-keying mechanism. Currently, only TCs are keyed by mapping the target structure to retron 2-D patterns. It will eventually be desirable, however, to standardize the knowledge base on one keying mechanism. Two-dimensional patterns provide a convenient mechanism since they allow specification of arbitrarily complex molecular substructures.

ACKNOWLEDGMENT

We are grateful to the National Institutes of Health for financial support. The assistance of Martin Ott, Lutz Stamp, and Vesa Nevalainen in writing 2-D patterns for the LHASA knowledge base is greatly appreciated.

REFERENCES AND NOTES

(1) (a) LHASA is an acronym for Logic and Heuristics Applied to Synthetic Analysis. (b) The program is implemented on Digital Equipment Corp.'s VAX computers under the VMS operating system and on a variety of computers running Unix. (c) Long, A. K.; Rubenstein, S. D.; Joncas, L. J. A Computer Program for Organic Synthesis. *Chem. Eng. News* **1983**, *6*(19), 22. Corey, E. J.; Long, A. K.; Rubenstein, S. D. Computer-Assisted Analysis in Organic Synthesis. *Science* **1985**, *228*, 408, and references cited therein.

(2) LHASA also has a new mode of operation in which the program uses information gained during its perception phase to suggest strategies and tactics to the user.

(3) Corey, E. J.; Howe, W. J.; Orf, H. W.; Pensak, D. A.; Petersson, G. General Methods of Synthetic Analysis. Strategic Bond Disconnections for Bridged Polycyclic Structures. *J. Am. Chem. Soc.* **1975**, *97*, 6116.

(4) Corey, E. J., Howe, W. J.; Pensak, D. A. Computer-Assisted Synthetic Analysis. Methods for Machine Generation of Synthetic Intermediates Involving Multistep Look-Ahead. *J. Am. Chem. Soc.* **1974**, *96*, 7724.

(5) Corey, E. J.; Cramer, R. D., III; and Howe, W. J. Computer-Assisted Synthetic Analysis for Complex Molecules. Methods and Procedures for Machine Generation of Synthetic Intermediates. *J. Am. Chem. Soc.* **1972**, *94*, 440.

(6) For a detailed discussion of the one-dimensional pattern language, see: Hoyle, P. L. M. Ph.D. Thesis, University of Leeds, 1986.

(7) Corey, E. J.; Kang, M.-C.; Desai, M. C.; Ghosh, A. K.; Houpis, I. N. Total Synthesis of (±)-Ginkgolide B. *J. Am. Chem. Soc.* **1988**, *110*, 649.

(8) Corey, E. J.; Desai, M. C. Simple Synthesis of (±)-*β-trans*-Bergamotene. *Tetrahedron Lett.* **1985**, *26*, 3535.

(9) Corey, E. J.; Jorgensen, W. L. Computer-Assisted Synthetic Analysis. Generation of Synthetic Sequences Involving Sequential Functional Group Interchanges. *J. Am. Chem. Soc.* **1976**, *98*, 203.

(10) Johnson, A. P. Computer Aids to Synthesis Planning. *Chem. Brit.* **1985**, *21*, 59.

(11) Corey, E. J.; Ohno, M.; Vatakencherry, P. A.; Mitra, R. B. Total Synthesis of *d,l*-Longifolene. *J. Am. Chem. Soc.* **1961**, *83*, 1251. Corey, E. J.; Ohno, M.; Mitra, R. B.; Vatakencherry, P. A. Total Synthesis of Longifolene. *J. Am. Chem. Soc.* **1964**, *86*, 478.

(12) Corey, E. J.; Long, A. K.; Lotto, G. I.; Rubenstein, S. D. Computer-Assisted Synthetic Analysis. Quantitative Assessment of Transform Utilities. *Recl. Trav. Chim. Pays-Bas* **1992**, *111*, 304.

(13) Corey, E. J.; Cheng, X.-M. *The Logic of Chemical Synthesis*; Wiley Interscience: New York, 1989; p 31. Long, A. K.; Kappos, J. C.; Nevalainen, V.; Stamp, L. An Extension to the Theory of Retrosynthetic Analysis: "Tactical Combinations" of Transforms. Submitted for publication.

(14) Long, A. K.; Kappos, J. C. Computer-Assisted Synthetic Analysis. Performance of Tactical Combinations of Transforms. *J. Chem. Inf. Comput. Sci.*, preceding paper in this issue.

(15) Most of the code which handles 2-D pattern-keyed subgoal processing is written in the C language. In all, there are about 4400 executable lines of 2-D-pattern C code, organized into 78 functions. Tactical combinations (TC) code is written in FORTRAN and C. There are approximately 3100 executable lines of TC code organized into 32 subroutines. For comparison, the entire LHASA program consists of about 113 000 lines of FORTRAN, C, MACRO, and PASCAL source code grouped into 1025 subroutines and functions, and a chemical knowledge base of roughly 167 000 lines of CHMTRN code.

(16) For a detailed discussion of the CHMTRN language, see: Orf, H. W. Ph.D. Thesis, Harvard University, 1976.

(17) For a discussion of usage of the linked list data structure in the LHASA program, see: Pensak, D. A.Ph.D. Thesis, Harvard University, 1973.

(18) The procedure used for sorting the array of change codes is based on the Shell-Metzner algorithm: A Comparison of Sorts. *Creative Comput.* **1976**, *2*.

(19) The concept of duplicate screening was first developed on the LHASA project for traditional, functional group-based subgoals and is borrowed for the application described herein.

(20) GPAIR, GSING, and PEXEC are the executive subroutines which guide the performance of transforms keyed by pairs of functional groups, single functional groups, and one-dimensional patterns, respectively.

(21) The LHASA knowledge base presently contains more than 450 tactical combinations.

(22) Castillo, M.; Loyola, L. A.; Morales, G.; Singh, I.; Calvo, C.; Holland, M. L.; MacLean, D. B. The Alkaloids of *L. magellanicum* and the structure of magellanine. *Can. J. Chem.* **1976**, *54*, 2893. Loyola, L. A.; Morales, G.; Castillo M. Alkaloids of *Lycopodium Magellanicum*. *Phytochemistry* **1979**, *18*, 1721.

(23) Three approaches to these alkaloids were recently disclosed. See: St. Laurent, D.R.; Paquette, L. A. Cyclopentannulation with a 1,3-Dicarbonyl Dipole Equivalent. Synthesis of Bicyclo[3.3.0]oct-1(5)-ene-2,6-dione. *J. Org. Chem.* **1986**, *51*, 3861. Mehta, G.; Rao, K. S. Model Studies toward Crinipellin Diterpenes and Paniculatine-type Lycopodium Alkaloids from a Common Triquinane Precursor. *J. Chem. Soc., Chem. Commun.* **1987**, 1578. Hirst, G. C.; Howard, P. N.; Overman, L. E. Stereocontrolled Construction of Carbocyclic Rings by Sequential Cationic Cyclization-Pinacol Rearrangement. *J. Am. Chem. Soc.* **1989**, *111*, 1514.

(24) For the magellanine analysis, LHASA was restricted to employing only tactical combinations of transforms. In other words, the goal transform executives GPAIR, GSING, and PEXEC, which are normally active, were bypassed for this analysis. When these goal transform executives are accessed for magellanine, numerous additional retrosynthetic sequences result.

(25) Corey, E. J.; Orf, H. W.; Pensak, D. A. Computer-Assisted Synthetic Analysis. The Identification and Protection of Interfering Functionality in Machine-Generated Synthetic Intermediates. *J. Am. Chem. Soc.* **1976**, *98*, 210. Corey, E. J.; Long, A. K.; Greene, T. W.; Miller, J. W. Computer-Assisted Synthetic Analysis. Selection of Protective Groups for Multistep Organic Syntheses. *J. Org. Chem.* **1985**, *50*, 1920.