(9) Lynch, M. F.; Barnard, J. M.; Welford, S. M. "Computer Storage and Retrieval of Generic Chemical Structures in Patents. 1. Introduction and General Stragegy". *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 148-150.
(10) Aho, A. V.; Ullman, J. D. "The Theory of Parsing Translating and Compiling"; Prentice-Hall: Englewood Cliffs, N.J., 1972; 2 Vols.
(11) Knuth, D. E. "Top-Down Syntax Analysis". *Acta Inf.* **1971**, *1*, 79-110.
(12) Wirth, N. "An Assessment of the Programming Language Pascal". *IEEE Trans. Software Eng.* **1975**, *SE*-1, 192-198.
(13) Carruthers, L. M.Sc. Dissertation, University of Sheffield, 1983.
(14) Feldmann, R. J.; Milne, G. W. A.; Heller, S. R.; Fein, A.; Miller, J. A.; Koch, B. "An Interactive Substructure Search System". *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 157-163.
(15) Elder, M. "The Conversion from Wiswesser Line Notation to CIS Connection Table", Proceedings of CNA (UK) Seminar on Interconversion, Loughborough, March 1982; Chemical Structure Association: London, 1983.
(16) The DARING program is marketed by Fraser Williams (Scientific Systems) Ltd., Glendower House, Poynton, Cheshire, U.K.
(17) Garfield, E. "An Algorithm for Translating Chemical Names to Molecular Formulas". *J. Chem. Doc.* **1962**, *2*, 177-179.
(18) Dyson, G. M. "A Cluster of Algorithms Relating the Nomenclature of Organic Compounds to Their Structure Matrices and Ciphers". *Inf. Storage Retr.* **1964**, *2*, 159-199.
(19) Vander Stouw, G. G.; Elliott, P. M.; Isenberg, A. C. "Automated Conversion of Chemical Substance Names to Atom-Bond Connection Tables". *J. Chem. Doc.* **1974**, *14*, 185-193.
(20) Vander Stouw, G. G. "Computer Programs for Editing and Validation of Chemical Names". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 232-236.
(21) Rayner, J. D. Ph.D. Thesis, University of Hull, 1983.
(22) Szczyglowski, W. L. M.Sc. Dissertation, University of Sheffield, 1983.
(23) Figueras, J. "Chemical Symbol String Parser". *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 48-52.
(24) Kinsella, J. E. M.A. Dissertation, University of Sheffield, 1982.

Artificial Intelligence in Organic Synthesis. SST: Starting Material Selection Strategies. An Application of Superstructure Search[†]

W. TODD WIPKE* and DAVID ROGERS

Department of Chemistry, University of California, Santa Cruz, California 95064

Received August 8, 1983

A program for the interactive selection of potential starting materials given a desired target molecule is described. The program uses hierarchical search to rapidly select candidates from a large starting material library and contains a function to evaluate the appropriateness of the functionality of the starting material. The user can specify restrictions on features such as the number of atoms in the starting material, price limitations, chirality, and whether to use superstructure or substructure searching. Several examples of the results of the program are presented.

INTRODUCTION

The importance of selecting good starting materials for an organic synthesis has been known for a long time; in fact, the selection of a starting material can be the major discovery in a synthesis, with the process of converting the starting material to the target minor by comparison. The previous work in computer-assisted design of organic synthesis has focused on reaction-driven analysis backward from the target molecule in an open-ended search.[1-5] Selection of starting materials has been ignored except as an end point for the backward analysis. For example, the SYNCHEM program developed by Gelernter[1] used a catalog of available starting materials to recognize when a precursor on a synthetic sequence was available and thus a termination point.

Empirically, one observes that chemists do not always work systematically backward but sometimes make an "intuitive leap" to a specific starting material from a target without consideration of reactions needed for interconversion. This intuitive leap probably involves a *Gestalt* pattern recognition based on the chemist's knowledge of available starting materials and similarity between the starting material structure and the target structure. By selection of one or more starting materials, the search for synthetic pathways becomes focused on those connecting the starting materials to the target. The starting material acts as a "planning island",[6] an anchor point in the plan to help reduce the planning space and allow forward planning from the material as well as backward planning from the target. Our interest in synthetic strategic planning led us to pursue these starting material oriented strategies as a planning aid.

Our objective was to develop algorithmic techniques to assist the chemist in selecting potential starting materials by "heuristic leap" directly from the target.[7] We chose to treat this heuristic leap as a pure strategy, independent of reaction knowledge.[8] We wished to avoid the costly tedium of reasoning backward from the target with reactions just to hypothesize a starting material. We wished to avoid the biases and limitations that reasoning in reaction space imposes. For example, many of the more creative suggestions for starting materials cannot be discovered by reasoning with reactions either because the reaction library is incomplete or because the crucial reaction in the synthetic sequence has not yet been discovered. Thus, we seek algorithms for this heuristic leap that are free from reaction knowledge.

Although there has been no systematic study of how a chemist makes an intuitive jump to a starting material, clearly the chemist has knowledge of known starting materials and associated information such as approximate price, ease and safety of handling, and chirality. The chemist also has knowledge of synthetic reactions, which is not applied directly but rather used in abstract form to suggest possible substructures in the target and candidate starting material that would be either useful or unwanted. Finally, the chemist has knowledge of other successful syntheses and the starting materials used in them. The approach we are about to describe will use only knowledge of available starting materials.

Because there are so many starting materials available, the computer with its large, fast memory and reliable recall seems like a natural assistant to the chemist in picking starting materials. Assisting in selection of starting materials can be valuable to the chemist even if the chemist fills in the synthesis plan manually, and it can certainly be valuable to computer-assisted planning.

RESULTS AND DISCUSSION

The *Gestalt* pattern recognition that chemists perform is graphically oriented, with probable dominance given to the

[†] Presented at the 182nd National Meeting of the American Chemical Society, New York, Aug 1981.

0095-2338/84/1624-0071$01.50/0

I)	Target = SM	Identical match
II)	Target > SM	Superstructure match
III)	Target < SM	Substructure match
IV)	None of these	Similarity match

Figure 1. Synthetic relationships.

skeletal resemblance of the starting material to the target compound. Other factors such as chirality, nature and location of functionality, and cost are probably considered secondarily. Thus, a key part of the problem depends on recognizing graphical similarity between the target and prospective starting materials.

Relationships between Target and Starting Material. Turning now to the graphical relationship a target can have with the starting material, we note four distinct cases are possible, as shown in Figure 1. Since efficient syntheses minimize the amount of bond making and breaking, the relationships I–IV are, as a general rule, roughly in order of decreasing efficiency of the resulting syntheses. Incidentally, the cases also happen to be in order of increasing difficulty of computer implementation. Therefore, we began with class I and worked toward class IV.

Class I is the case in which the desired target is available for purchase directly; hence, no synthesis is required. This class is found by doing a simple equality check of the SEMA names[9] and was implemented in SECS by a rapid hash table method.[8] The SYNCHEM program only detected class I relationships and did so by matching Wiswesser line notation.[10]

Class II corresponds to a *constructive* synthesis, where the starting material can be incorporated directly into the target compound. This class is found by doing a *superstructure search* over the starting material library, that is, finding all compounds in the library of which the target is a superstructure.[11] Since this is the first mention of superstructure search in the literature and since constructive syntheses are so important, we will discuss this class in considerable detail later in the paper.

Class III corresponds to a *degradative* synthesis, where the starting material is more complex than the target and is degraded down to the target molecule. Degradative synthesis cannot be found by current synthesis programs because the dominant strategy underlying this type of synthesis is starting material oriented. This class is found by doing a *substructure search* over the starting material library to find all starting materials of which the target is a substructure. Substructure search is well understood and has been widely used in many retrieval systems, such as MACCS,[12] COUSIN,[13] and CAS ONLINE.[14]

Class IV is the most difficult case, corresponding to neither a totally degradative nor a totally constructive synthesis but incorporating sections of each. Some of the bonds and atoms of the starting material remain intact in the target, but some do not. Most syntheses appear to belong to class IV, since bonds are both made *and* broken. However, by the technique of abstraction introduced in the next section we can cause most syntheses to fall into classes I–III. (We will report on a direct solution to the class IV problem in a later paper.)

Abstraction of the Starting Materials. The fact that most starting materials are related to the target compounds in a class IV relationship is due to the high level of detail in our problem space. This level of detail obscures the directional nature of most synthetic relationships. By *abstracting* the target and the potential starting materials, we lower the level of detail, making it easier to perceive their synthetic relationship with the target, and reducing the number of Class IV relationships.

Model of the SST Algorithm. Figure 2 shows the general model used in creating the SST program. The top half of the figure is the *abstract domain*, while the bottom half of the figure is the *molecule domain*. Both the target molecule and the starting material library exist in the molecule domain.

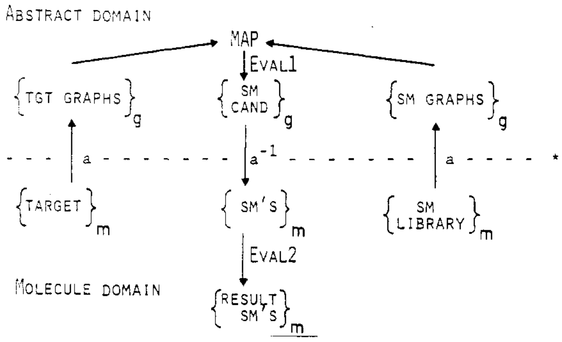


Figure 2. General search model used in designing the SST program. The subscript m indicates the objects in the set {} are molecules; $\{\}_g$ indicates a set of graphs. The operator *a* is the abstraction operator; a^{-1} is its inverse, namely, the retrieval of molecules corresponding to an abstract graph. Eval1 is abstract evaluation; Eval2 is molecule evaluation. SM stands for starting materials.

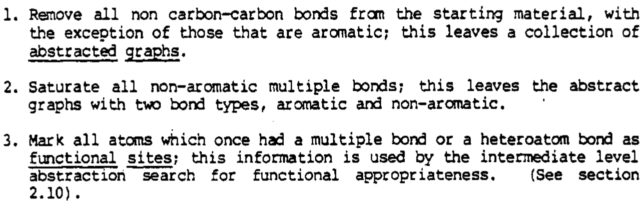
An *abstraction* algorithm derives from the target molecule a set of *target abstract graphs* and from the starting material a set of *starting material abstract graphs*. From the target abstract graphs, a *mapping function* selects a collection of graphs from the starting material abstract graphs. This selection of graphs is then evaluated by an *abstract graph evaluation function* to give a set of *candidate abstract graphs*. The starting materials that had originally given rise to these abstract graphs are then retrieved to give a set of *candidate starting materials*. These candidate starting materials are then evaluated with a *molecule evaluation function*, with the molecules passing this evaluation presented to the user as the *resulting starting materials*.

This general model can be used with a variety of abstraction algorithms, mapping functions, and evaluation functions. We use an abstraction algorithm that helps highlight the major relationships between the compds., so that a mapping function can detect the *substructure* or *superstructure* relationship between the starting materials and the target. The abstract graph evaluation function tests the applicability of the functionality and allows the user to restrict the search by limiting various structural features in the abstract graphs. The molecule evaluation function allows the user to restrict features that appear in the resulting starting materials.

This model actually involves a hierarchy of abstract search spaces similar to Sacerdoti's hierarchical planning first used in the ABSTRIPS program.[15] We used three levels of abstraction to represent the problem space, and at each level there is a process: (1) The *top level* contains abstract graphs of the target and starting materials (process = *mapping*). (2) The *intermediate level* has information about functional sites superimposed onto the abstract graphs (process = *functional evaluation*). (3) The *bottom level* consists of completely specified target and starting materials (process = *molecule evaluation*).

There are two motivations for choosing this hierarchy of abstract spaces. (1) Abstraction eradicates irrelevant differences so we may see the significant similarities, and (2) as we shall show, the abstract domain is considerably smaller than the bottom level molecule domain, so less effort is required to conduct the search for similar starting materials in the abstract domain.

Derivation of Abstraction Rules. To identify which features in starting materials were important, we studied the syntheses contained in "Creativity in Organic Synthesis".[16] We search for patterns in the bonds made and broken during the synthesis, hoping to find patterns reflecting general knowledge rather than knowledge *specific* to one synthesis. Our findings are summarized as follows: (1) Given a sample target structure,



1. Remove all non carbon-carbon bonds from the starting material, with the exception of those that are aromatic; this leaves a collection of abstracted graphs.
2. Saturate all non-aromatic multiple bonds; this leaves the abstract graphs with two bond types, aromatic and non-aromatic.
3. Mark all atoms which once had a multiple bond or a heteroatom bond as functional sites; this information is used by the intermediate level abstraction search for functional appropriateness. (See section 2.10).

Figure 3. Abstraction rules.

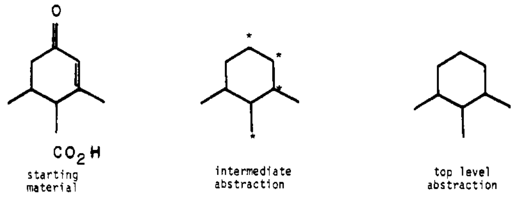


Figure 4. Levels of abstraction. * indicates a functional site.

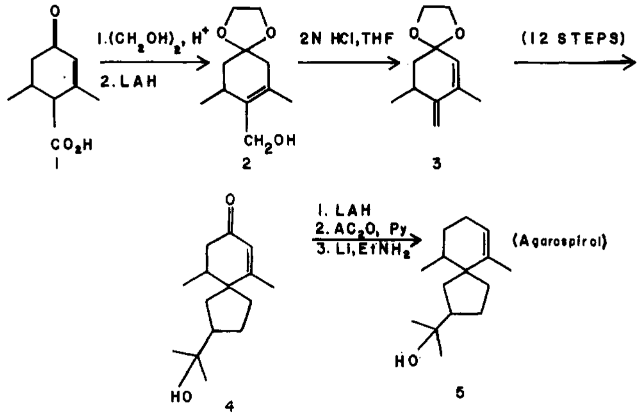


Figure 5. Synthesis of agarospirol.

an average carbon–carbon bond in it was present 80% of the time in a starting material and created during synthesis 20% of the time. On the other hand, carbon–heteroatom bonds were presented only 60% of the time in the starting material and created during synthesis 40% of the time. Thus, a given carbon–heteroatom bond is *twice as likely* to be created during a synthesis and, hence, a less valuable detail for recognizing starting materials. (2) The multiple bonds in the target graphs were often created or moved around the molecule during synthesis; exceptions were aromatic bonds, which tended to originate in the starting material rather than be synthesized. From these findings we developed the rules for abstraction shown in Figure 3.

Application of the first two rules gives the *top level* abstraction of the starting material; addition of the third rule gives the *intermediate* abstraction of the starting material. This process is illustrated in Figure 4 for a potential starting material. These rules of abstraction are applied to each of the compounds in our starting material library, and the resulting abstracted graphs are stored in an *abstracted library*. Each abstracted graph in this library has a list of pointers to the original compounds from which it was generated. This abstraction technique results in a significant savings: the 11 000 compounds in our original starting material library generated an abstracted file representating 1436 top level graphs and 3733 intermediate level graphs.

The synthesis of agarospirol (epihinesol)[17] shown in Figure 5 demonstrates how abstraction can be useful. This synthesis is generally constructive, though some degradation is involved. The authors used a carbonyl group in the starting material (**1**) to assist in constructing the side chain in the intermediate (**4**) and then removed the carbonyl group to give the target, agarospirol (**5**).

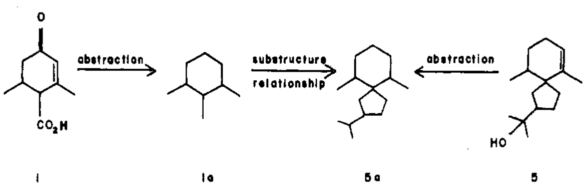


Figure 6. Abstraction with agarospirol.

Table I. Class Membership for Starting Materials

group[a]	no. of members[b]	% of SSM[c]
1	6	3.7
2	27	16.7
3	147	90.7

[a] Group numbers refer to the group definitions given in the text. [b] The number of members is the number of significant starting materials that were contained in this group. [c] The percent of the significant starting materials is the value in column 2 divided by 162, the total number of significant starting materials.

This synthetic pathway would not be found by direct substructure search of the starting material candidate (**1**) onto agarospirol (**5**) because the carbonyl group and the acid group in **1** are not present in **5**. However, in the abstract domain, these minor differences no longer appear. As shown in Figure 6, the top-level abstracted starting material (**1a**) is a direct substructure of the top-level abstracted agarospirol (**5a**).

Judging of the Abstraction Algorithm. Unfortunately, determining if a starting material is good is difficult, for there may exist many potentially good starting materials that have never been used in a published synthesis. This made judging the abstraction algorithm difficult. We decided that the best method for demonstrating its effectiveness would be to compare it to a compendium of known syntheses to see if the starting materials contained in the compendium would be selected by the SST program.

We selected Bindra and Bindra's "Creativity in Organic Synthesis".[16] Many compounds in this book can be considered as starting materials, but we were especially interested in compounds that make a significant contribution to the construction of the target rather than compounds that added only few atoms. For this study, we decided to consider only starting materials that donate five or more carbon atoms to the final target. Such compounds will be called *significant* starting materials. The book contained only one degradative synthesis; while our algorithm would have allowed discovery in this case, there were not enough examples of this class to allow judgement of the abstraction for this class (class III).

Our study consisted of analyzing the significant starting materials used in the book to define the amount of abstraction needed to allow the superstructure search to succeed. A significant starting material from a synthesis was abstracted, and the resulting graph(s) was (were) studied to see if the target graph was now a superstructure of it (them). Abstract graphs smaller than five carbon atoms were ignored.

The book contained 162 significant starting materials for 67 target compounds, with a total of 106 syntheses presented. Three groupings of the starting materials were studied. Group 1 contains the significant starting materials that were substructures of the target compound. Group 2 contains the significant starting materials that became substructures of the target compound if we ignored bond types. Group 3 contains the significant starting materials that became substructures of the target compound if we ignored bond types, removed all heteroatoms, and removed all fragments of less than five carbon atoms. The memberships in the various groupings are not exclusive, for a given starting material may fit in more than one category; that is, group 1 is contained in group 2 is contained in group 3. Table I shows the number of starting materials that belonged to each group and the percent of all significant starting materials that were contained in each.

The results suggest that direct superstructure search of the target onto the candidate starting materials would not be useful, for it was successful only 3.7% of the time. If we ignore bond types, performance improves, but still only about 16% of the starting materials succeed. When both abstraction rules are applied to the starting material and their abstracted graphs are used, a successful superstructure mapping is found over 90% of the time. From this study, we concluded that the graphs generated from the abstraction rules are no longer obscured by fine detail.

The 10% of the cases that fail involved both degradation and construction of carbon–carbon bonds in the starting material. These starting materials have a class IV relationship with the target molecule even after abstraction, which means they will not be discovered with a class II or III search.

Mapping Functions. In Relationships between Target and Starting Material, we developed the four relationships possible between a target molecule and a starting material. Each relationship is the foundation of a mapping function, which can then be used in the general SST algorithm expressed under Model of the SST Algorithm and the MAP in Figure 2. As indicated by Figure 2, all mapping is performed in the highest level abstract domain, regardless of class.

The class I relationship is implemented by an identical look-up algorithm. The use of SEMA name storage and a simple hashing function allows very rapid look up of an abstract target molecule in a library of abstract starting materials.[8]

The class II relationship is implemented by *superstructure search*, while the class III relationship is implemented by *substructure search*. While substructure search is well-known, superstructure search has not been explicitly used in chemistry. We find that superstructure and substructure search can be implemented in an almost identical manner, by using an initial *key search* to eliminate graphs that cannot map and then by using an *atom by atom search* to test for the desired relationship between the target and the starting material. Because of the importance of constructive syntheses in organic chemistry and the newness of superstructure search, we will emphasize this search in our discussion.

The class IV relationship calls for a *common subgraph search* to find abstract graphs that are similar to the abstract target graph. Common subgraph searching over large data files is currently being investigated and will be the subject of a later paper.

Superstructure and Substructure Search. While many algorithms exist for determining the existence of a subgraph (or supergraph) relationship between two graphs, most of them would be far too time consuming for practical use if used alone. For example, with a subgraph search time of 1 s and a library containing 10 000 graphs, the total search time would be about 3 h. To avoid this problem, rapid screening functions known as *key searches* were developed.[18]

A set[19] of features $F = \{f_1, f_2, ..., f_n\}$ can be defined that gives broad characterization of different aspects of graphs such as node type, node degree, presence of cycles of various order, etc. For a given graph g, we can determine the set $K_g = \{f_i \in F \cdot f_i$ is present in graph $g\}$. Then for graphs g_a and g_b having features K_a and K_b, respectively, we can infer that

$$\textit{if } g_a \text{ is a subgraph of } g_b \textit{ then } (K_a \cap K_b) = K_a \qquad (1)$$

i.e., if g_a is a *subgraph* of or identical with g_b, then all features in K_a are also in K_b, but the converse is not necessarily true. The condition $(K_a \cap K_b) = K_a$ is a necessary but not sufficient condition for determining if g_a is a subgraph of g_b. In practice, the set of graphs G is "inverted" on the keys such that S_i is the set of graphs having feature f_i:

$$S_i = \{g \in G\text{: } f_i \in K_g\} \qquad (2)$$

In order to perform a substructure search with target (query) t, we reduce the set of graphs that are candidates for node-by-node comparison from G down to P_t^{sub}, the set of graphs possessing the necessary features:

$$P_t^{\text{sub}} = \bigcap_{i=1}^{n}(S_i\text{: } f_i \in K_t) \qquad (3)$$

Thus, if $K_t = \{f_1, f_4, f_7\}$, $P_t^{\text{sub}} = (S_1 \cap S_4 \cap S_7)$, which is the set of graphs having features f_1 AND f_4 AND f_7. We then proceed to perform node-by-node matching of the target t with each graph $g \in P_t^{\text{sub}}$, giving a final answer A_t^{sub}.

For contrast, let us turn to the supergraph relationship, where g_a is a supergraph of g_b; we can infer

$$\textit{if } g_a \text{ is a supergraph of } g_b \textit{ then } (K_a \cap K_b) = K_b \qquad (4)$$

i.e., if g_a is a supergraph of or identical with g_b, then all features in K_b are also in K_a, but again, the converse is not necessarily true as above. We can again invert on the keys as eq 2 indicates, but the inverted sets S_i appear at first to be less useful since there are features in K_a that need not be in K_b; thus, we cannot use eq 3. All is not lost though, since we can rewrite eq 4 to obtain eq 5 and 6, which now allows productive use

if g_a is a supergraph of g_b *then*

$$(\sim K_a \cap K_b) = \text{null set} \qquad (5)$$

$$\sim P_t^{\text{sup}} = \bigcup_{i=1}^{n}(S_i\text{: } f_i \notin K_t) \qquad (6)$$

of the inverted sets S_i. The union of all sets of graphs S_i having features f_i that are not present in the target gives the *disallowed set of graphs*, $\sim P_t^{\text{sub}}$. Therefore, the set of graphs possessing the necessary features that may potentially satisfy the supergraph search is

$$P_t^{\text{sup}} = G - \bigcup_{i=1}^{n}(S_i\text{: } f_i \notin K_t) \qquad (7)$$

Thus, if $K_t = \{f_1, f_4, f_7\}$ and $n = 7$, then $P_t^{\text{sup}} = G - (S_2 \cup S_3 \cup S_5 \cup S_6)$, which is the set of graphs not having features f_2, f_3, f_5, and f_6 that are absent in the target. We then perform node-by-node matching of each $g \in P_t^{\text{sup}}$ against the target and obtain the answer set of graphs A_t^{sup}.

In summary, we see that we can use the same set of features F in superstructure search as in substructure search and we can also use inverted key sets. The difference is in how the inverted keys sets are combined: In substructure search one takes the *intersection* of key sets for features *present* in the target, which gives the set of *allowed* graphs; in superstructure search, one takes the *union* of key sets for features *absent* in the target, giving the set of *disallowed* graphs, which by difference gives the set of allowed graphs. Following the key search, all candidates must undergo an *atom by atom* search to confirm that the desired relationship exists. We used the same subgraph search algorithm for both substructure and superstructure searching, passing the target and the candidate normally for substructure search and passing them in reversed order for superstructure search.

In our studies, it was common for superstructure search to pass many more structures than the substructure search. This made efficient keys for the superstructure search important to us, and initially, it was unclear whether superstructure and substructure search should have independently optimized selections of keys. We demonstrate under Key Search Efficiency Proof that the same keys are equally efficient for both searches.

The larger number of molecules (graphs) passed by superstructure search is explained as an effect of the distribution of the library in relation to the size of the query molecule. Figures 7–9 show the number of molecules (graphs) in the SM library and the abstract library plotted against number of

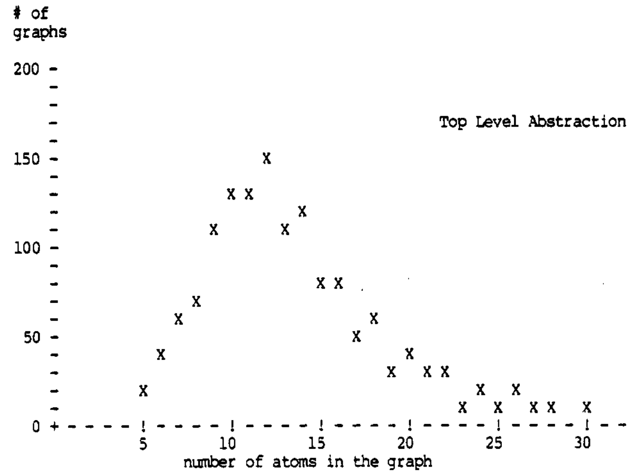


Figure 7. Top-level abstract graphs vs. number of atoms.

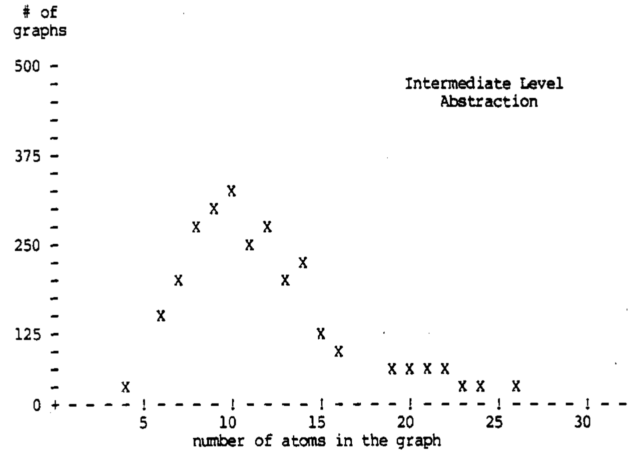


Figure 8. Intermediate-level graphs vs. number of atoms.

atoms. As the number of atoms increases, each shows a maximum around 10 atom molecules (graphs) and then a gradual decline. Most of our queries contained more than 10 atoms; this meant that the pool of candidates would be much greater for superstructure search than for substructure search. It is interesting that the abstracted library retains the same general distribution of sizes as the SM library, though the scale has been greatly reduced.

Key Search Efficiency Proof. We define the set of possible subgraphs S and the set of possible supergraphs P as shown in eq 8. Two types of queries can be made with these sets.

$$S = \bigcup_{i=1}^{m} s_i \qquad P = \bigcup_{i=1}^{n} p_i \tag{8}$$

The *substructure* query involves selecting a graph s from S; an answer set A_{sub} is defined as

$$A_{\text{sub}} = \bigcup_{i=1}^{n} (A_i \in P\text{: } s \text{ is a subgraph of } A_i) \tag{9}$$

Similarly, for a *superstructure* query p from P, the answer set A_{sup} is defined by eq 10.

$$A_{\text{sup}} = \bigcup_{i=1}^{m} (A_i \in S\text{: } p \text{ is a supergraph of } A_i) \tag{10}$$

We can test for both the substructure and superstructure relationships by using the same mapping function M:

$$\begin{array}{lll} \textit{given} & g_{\text{sub}} \in S & \\ & g_{\text{sup}} \in P & \\ \textit{define} & M(g_{\text{sub}}, g_{\text{sup}}) = 1 & \textit{if } g_{\text{sub}} \text{ is a subgraph of } g_{\text{sup}} \\ & M(g_{\text{sub}}, g_{\text{sup}}) = 0 & \textit{otherwise} \end{array} \tag{11}$$

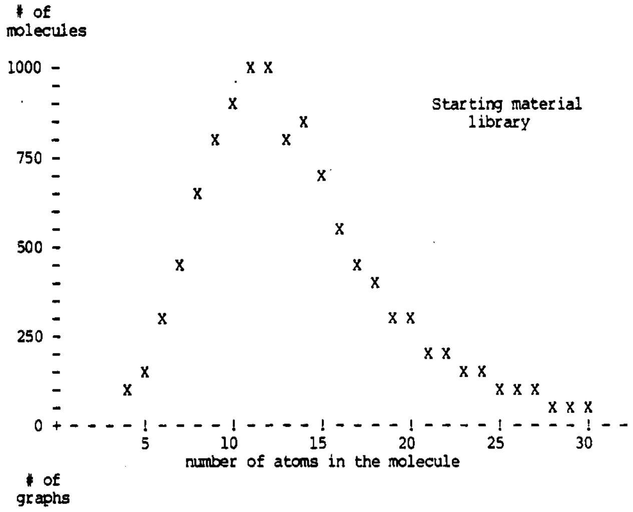


Figure 9. Starting material molecules vs. number of atoms.

As a filter, we approximate M with a *key search* function K, which can be evaluated rapidly relative to M. The key search function is defined so that

$$K(g_{\text{sub}}, g_{\text{sup}}) = 1 \textit{ if } M(g_{\text{sub}}, g_{\text{sup}}) = 1 \tag{12}$$

This definition guarantees that K will pass any graph pair that passes M, though it will also pass some graph pairs that fail M. The graph pairs that pass K but fail M are called *false hits*.

The best key search functions are those that have few false hits relative to the number of total hits; for example, for a given graph ($s \in S$), we define the efficiency of the key search function for this graph as

$$\sum_{i=1}^{n} M(s,P_i) / \sum_{i=1}^{n} K(s,P_i) \tag{13}$$

If the denominator of eq 13 is zero, the efficiency function is defined as 1. Similarly, given a graph ($p \in P$), we define the efficiency of the key search function for this graph as

$$\sum_{i=1}^{m} M(S_i,p) / \sum_{i=1}^{m} K(S_i,p) \tag{14}$$

If the denominator of eq 14 is zero, the efficiency function is defined as 1. These efficiency functions change from query to query; to look at the overall efficiency of the key search function, we need to evaluate this type of function over all possible superstructure or substructures. The global efficiency function is defined to be the sum of the number of total hits over all possible query graphs divided by the sum of the number of total hits plus false hits; eq 15 and 16 show their

$$\sum_{j=1}^{m} \sum_{i=1}^{n} M(S_j,P_i) / \sum_{j=1}^{m} \sum_{i=1}^{n} K(S_j,P_i) \tag{15}$$

$$\sum_{j=1}^{n} \sum_{i=1}^{m} M(S_i P_j) / \sum_{j=1}^{n} \sum_{i=1}^{m} K(S_i,P_j) \tag{16}$$

form for substructure and superstructure search, respectively.

These series, being finite, converge to well-defined values; we also know that any series formed by regrouping terms of a convergent series is also convergent and has the same sum.[20] This means that the order of summation can be changed, proving the expressions are identical. This is expected (though not immediately obvious) since a natural symmetry exists between substructure and superstructure search.

User Constraints on SST. The mapping in the abstract domain is designed to select starting materials from the library depending on their graphical similarity to the target. Since there may be many graphs similar to the target, SST provides many options for the user to control and restrict the search process and to prune a candidate set resulting from a previously conducted search. The following features are currently implemented.

(1) Search Type. The user must specify whether identity, superstructure, or substructure search is to be performed. Superstructure search finds constructive starting materials, and substructure search finds degradative starting materials.

(2) User Abstraction. For degradative synthesis discovery, the user might decide to remove even larger sections of the target molecule than the abstraction algorithm would. This allows the discovery of a greater range of synthetic precursors in the substructure search process. This process is performed after the target molecule is abstracted with the abstraction algorithm and before the mapping is commenced.

(3) Aromatic Construction. Aromatic sections of a target are often incorporated from a starting material directly, rather than synthesized. The user is allowed to force aromatic atoms in the target to map onto aromatic atoms in the starting material candidate. Since the aromatic bonds of a molecule are not saturated during the abstraction process, this test can be done during the top-level abstract graph mapping.

(4) Aromatic Degradation. Degradation of an aromatic ring during an organic synthesis can be difficult; if the user does not want to consider this case, he or she can require aromatic atoms in the starting material to map only onto aromatic atoms in the target. This test is also done during the top-level mapping.

(5) Required Atoms. Certain atoms in the target are often desirable to purchase rather than synthesize. This option allows the specification of atoms in the target that must be used in the starting material mapping for the starting material to succeed. If the atom is a carbon atom, this test can occur during the top-level abstract graph mapping; otherwise, it occurs during the molecule-evaluation stage of the search.

(6) Abstract Graph Atom Count. The user can specify the count of an atom type that the abstract graph must have. Since the abstract graphs are almost entirely composed of carbon atoms, this is the most useful atom type to restrict. This test occurs during the mapping stage of the search.

(7) Evaluation Cut-Off. The SST program contains a function to evaluate the mapping of a starting material onto the target (see the next section for a discussion of the evaluation function). This cut-off is used during the abstract graph evaluation stage.

(8) Chirality. For the synthesis of chiral compounds, chirality is often incorporated through inclusion of a chiral starting material. This option allows the user to select only chiral starting materials for consideration; this test is done during the molecule evaluation stage of the search, because chirality is a property of molecules, not the abstract graphs.

(9) Price. Price can be an important consideration in a synthesis. A price per mole for each compound is kept in the starting material library, and the user is allowed to specify a cutoff price beyond which a compound is eliminated. This test is done during the molecule evaluation stage, again because price is associated with specific compounds.

(10) Atom Count. The user is allowed to specify the count of a given atom type that the starting material must have. For example, the user can restrict candidates to compounds with two to five carbon atoms and zero to two oxygen atoms; unspecified atom types are still allowed in any amount. This occurs during the molecule evaluation stage.

(11) User Evaluation. The user can also request the abstract graph evaluation results to be shown after the score is associated with the starting materials and then judge each starting material individually. This occurs during the molecule evaluation stage of the search since the user prefers to see starting materials rather than intermediate abstract graphs.

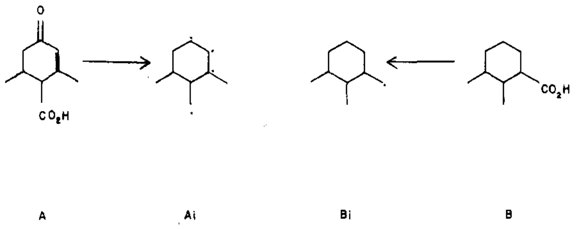


Figure 10. Two compounds with the same top-level abstract graph but different intermediate graphs. Functional sites are marked with *'s.

```
Value = 1 if the atom is a functional site.
Value = 2 if the atom is 1 bond away from a functional site.
Value = 3 if the atom is 2 bonds away from a functional site.
Value = 4 if the atom is 3 or more bonds away from a functional site.
```

Figure 11. Value-rating rules. Atoms get the *minimum* possible value.

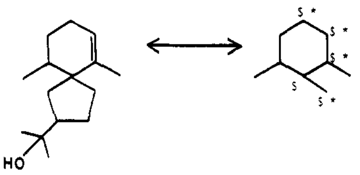


Figure 12. Best mapping of agarospirol onto an intermediate functional graph. * locates a functional site and $ the location of a make/break bond.

```
Score = 1 if the atom has a functional value of 1
Score = 2 if the atom has a functional value of 2
Score = 5 if the atom has a functional value of 3
Score = 10 if the atom has a functional value of 4
```

Figure 13. Scoring of the sites where bonds are made or broken.

Functional Evaluation of Proposed Starting Materials. Independent of the search used on the top-level abstract graphs, we now proceed to examine the intermediate-level abstraction related to functionality. Two starting materials having the same top-level abstract graph may differ greatly in utility owing to the difference in placement of functional groups (Figure 10). The abstraction process eliminates functionality, hence cannot be used in this evaluation stage. Therefore, we created an *intermediate* level abstraction to evaluate functional similarity. This intermediate level abstraction retains information about the location of functionality but not its type.

For agarospirol (Figure 5), the functionality of molecule A (Figure 10) would be much more appropriate to the synthesis than that of B. In particular, molecule A has a functional group at every location in the molecule where we will need to construct or remove bonds.

We use the third abstracting rule in Figure 3 to generate the *functional sites* of the abstracted graph. Given these sites, we then assign a *value* to each of the atoms in the abstracted graph; this value represents the proximity of the atom to a functional site. Each atom gets a numerical rating that is the *minimum* possible value under the classification rules of Figure 11. Thus, a functional site is assigned a value of 1 even though it may also be one bond away from another functional site.

The values are now related to the target that selected the abstract graph. Given a mapping of the target molecule onto the abstract graph, we mark atoms in the abstract graph that need a bond or have an extra bond. For example, Figure 12 shows a potential mapping of an intermediate graph onto agarospirol. Now the marked atoms in the abstracted graph ($ at a location where a bond must be made or lost) are given a score from the scores in Figure 13.

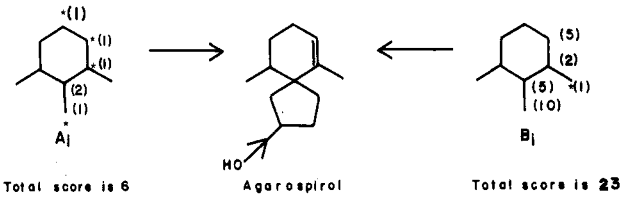


Figure 14. Scoring two abstract graphs for agarospirol.

The overall score for this mapping of the target onto the abstract graph is the sum of the scores at each of its atoms. The smaller the overall score, the better the functionality of the intermediate graph is for the given target. If there exists more than one mapping of the graph onto the target, then each mapping is evaluated, and the lowest (i.e., best) score of all is used as the overall score for that intermediate graph. The two intermediate level abstract graphs A_i and B_i in Figure 10 are scored as shown in Figure 14. They differ greatly in their scores, even though their top-level abstract graphs are identical.

This scoring function is applied to each intermediate graph, and the resulting score is given to all starting materials associated with that graph. If a starting material is associated with more than one abstract graph, then the starting material receives only the best (lowest) score. This is a computationally efficient approach to scoring the starting materials, because many starting materials are scored from calculations on one intermediate abstract graph.

This evaluation function is designed to be a general measure of the utility of the starting material. It is based on a desire to have the functionality of the starting material near the atoms where bond addition or deletion will occur. The function is not designed to measure the interest a chemist might have in the compound but rather to estimate the applicability of the compound's functionality toward the given target. The chemist is allowed to specify a cut-off value for the score to eliminate any candidate that has a rank higher than that cut-off.

EXAMPLES

Superstructure Search Behavior with Agarospirol. We studied the behavior of the SST program in depth with agarospirol. The starting material library we used was a subset of the Aldrich catalog containing about 11 000 molecules. We have not optimized our search algorithm, but at no point did any search shown here take more than a couple of CPU minutes to perform.

To judge the behavior of the hierarchical search, we performed a series of superstructure searches, limiting the size of the abstract graphs to a specific number of carbon atoms (column 1 in Table II). For each search (each row in Table II), we restricted the starting material sets in four ways: no restriction (SM set), starting materials with a score of 30 or less (eval 30–), starting materials with a score of 20 or less (eval 20–), and starting materials with a score of 10 or less (eval 10–). The number of starting materials resulting in each case is tabulated in Table II.

Several features of the superstructure search algorithm were noticed in this study.

(1) The total number of top-level abstract graphs found (last row of column 2) was only 115, and the total number of intermediate level abstract graphs found (last row of column 3) was only 570. The total number of starting materials found (last row of column 4) was 6077. This represents a 60-fold reduction of the search space with the top-level abstract graphs and a 10-fold reduction with the intermediate-level abstract graphs.

(2) The smaller graphs were the least selective for finding starting materials. Of the 115 abstract starting material graphs that were substructure of the agarospirol abstract graph, 15 (13.0%) had two to six atoms. These 15 abstract graphs led to 5175 starting materials out of a total of 6077 (85.2%).

Table II. Superstructure Search for Agarospirol

no. of carbons[a]	top set[b]	int set[c]	SM set[d]	eval 30–[e]	eval 20–[f]	eval 10–[g]
15	1	1	1	1	1	1
14	0	0	0	0	0	0
13	1	1	1	1	0	0
12	8	19	100	9	3	3
11	8	17	40	12	2	1
10	20	50	105	61	29	14
9	28	68	123	83	50	18
8	22	126	279	256	174	67
7	12	99	253	245	225	126
6	7	107	604	591	588	450
5	4	53	527	*	*	*
4	2	21	913	*	*	*
3	1	6	919	*	*	*
2	1	2	2212	*	*	*
2-15	115	570	6077	*	*	*

[a] The number of carbon atoms allowed in the abstract graph. [b] The number of top-level abstract graphs found. [c] The number of intermediate abstract graphs found. [d] The total number of starting materials associated with the abstract set, no cut-off on the functional evaluation. [e] The number of starting materials having a functional evaluation score of 30 or less. [f] The number of starting materials having a score of 20 or less. [g] The number of starting materials having a score of 10 or less.

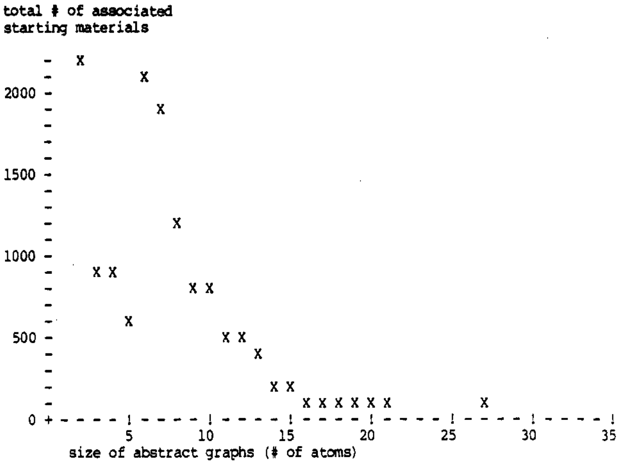


Figure 15. Number of associated starting materials vs. abstract graph size.

(3) Similar behavior is noticed in Figure 15, in which the number of starting materials is plotted against the size of the abstract graphs that they refer to. Unlike the distributions in Figures 7–9, this distribution is skewed toward the lower end, with the smaller abstract graphs representing large numbers of starting materials. That a few abstract graphs select such a large number of candidates makes them less interesting to us, for they are not efficiently filtering the candidates before passing them.

(4) The starting materials found from the smaller abstract graphs were also less distinguishable with our scoring function; for example, while the abstract graph sets for seven and eight carbons led to starting material sets almost identically sized (253 and 279, respectively), the percentage of compounds that had an evaluation score of 10 or less (column 6) was much larger for the smaller graph (48.9% vs. 24.0% for the eight-carbon graph).

Agarospirol Starting Materials Selected by Superstructure Search. To illustrate the use of SST, we show a superstructure search (Chart I) in which we require a carbon range of 9–15 for the abstract graphs and further restricted the final evaluation to compounds that scored 6 or less. (The search was strongly restricted to allow only a small subset to pass, so that

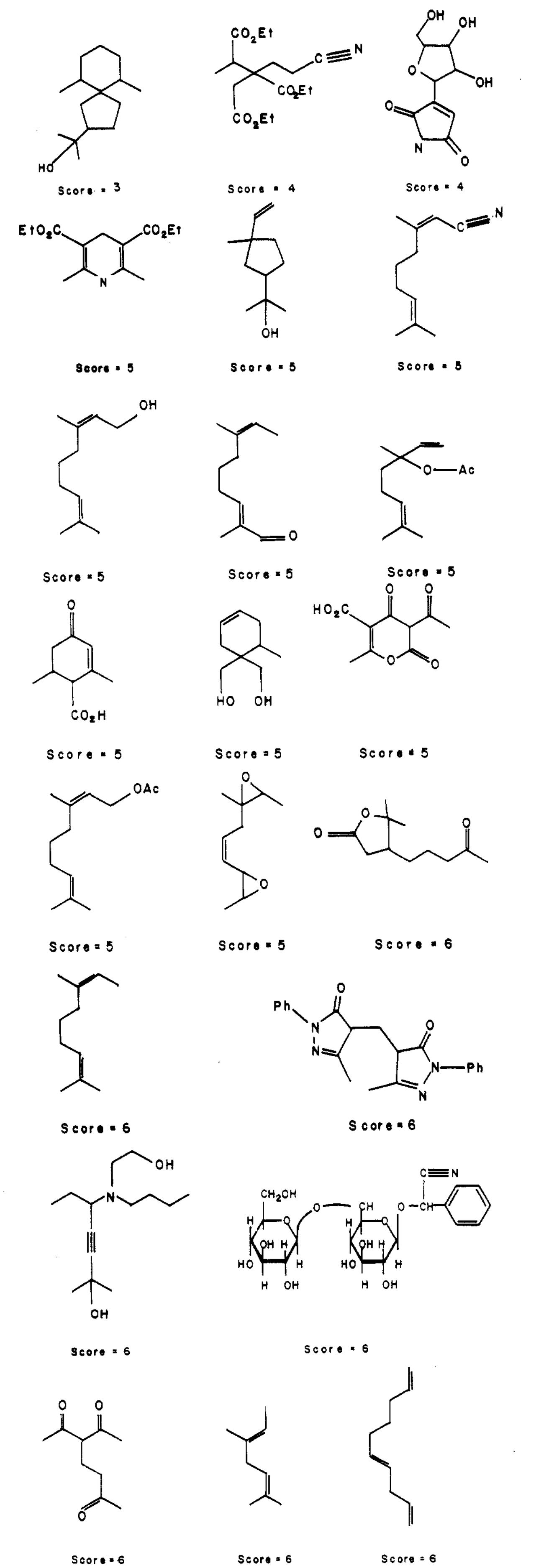


Figure 16. Possible starting materials for agarospirol found by superstructure search.

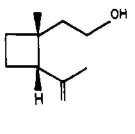


Figure 17. Grandisol.

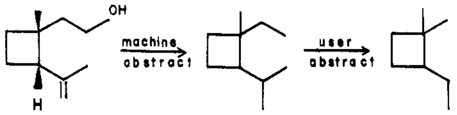


Figure 18. Machine and user abstraction of grandisol.

all results could be included in this paper. Generally, a chemist would wish to view more of the better starting materials.) The script of the trial run is given in Chart I; the user unput is indicated by wavy underline, and explanations are underlined. Default responses are in parentheses following a machine request; the user confirms the default with a carriage return or types in an alternate value.

Figure 16 shows the 22 molecules that resulted from this search. Agarospirol itself was found as an available starting material, and was rated as the best choice. The known significant starting material for agarospirol was one of the top 15 molecules by rating. It is interesting to note that a "class" of isoprenoid compounds with minimal differences was found.

Substructure Search with Grandisol. In a previously published study of the synthesis of grandisol, the SECS program discovered 10 of the 12 published synthesis.[21] We were interested to see if the application of the SST program could aid in the discovery of the final two routes.

The molecule grandisol (Figure 17) has been synthesized at least 12 times by various routes. The small size of the molecule and the four-membered ring structure have lent itself well to reaction-based synthetic schemes. These reaction-based schemes were successfully discovered by the SECS program.[21]

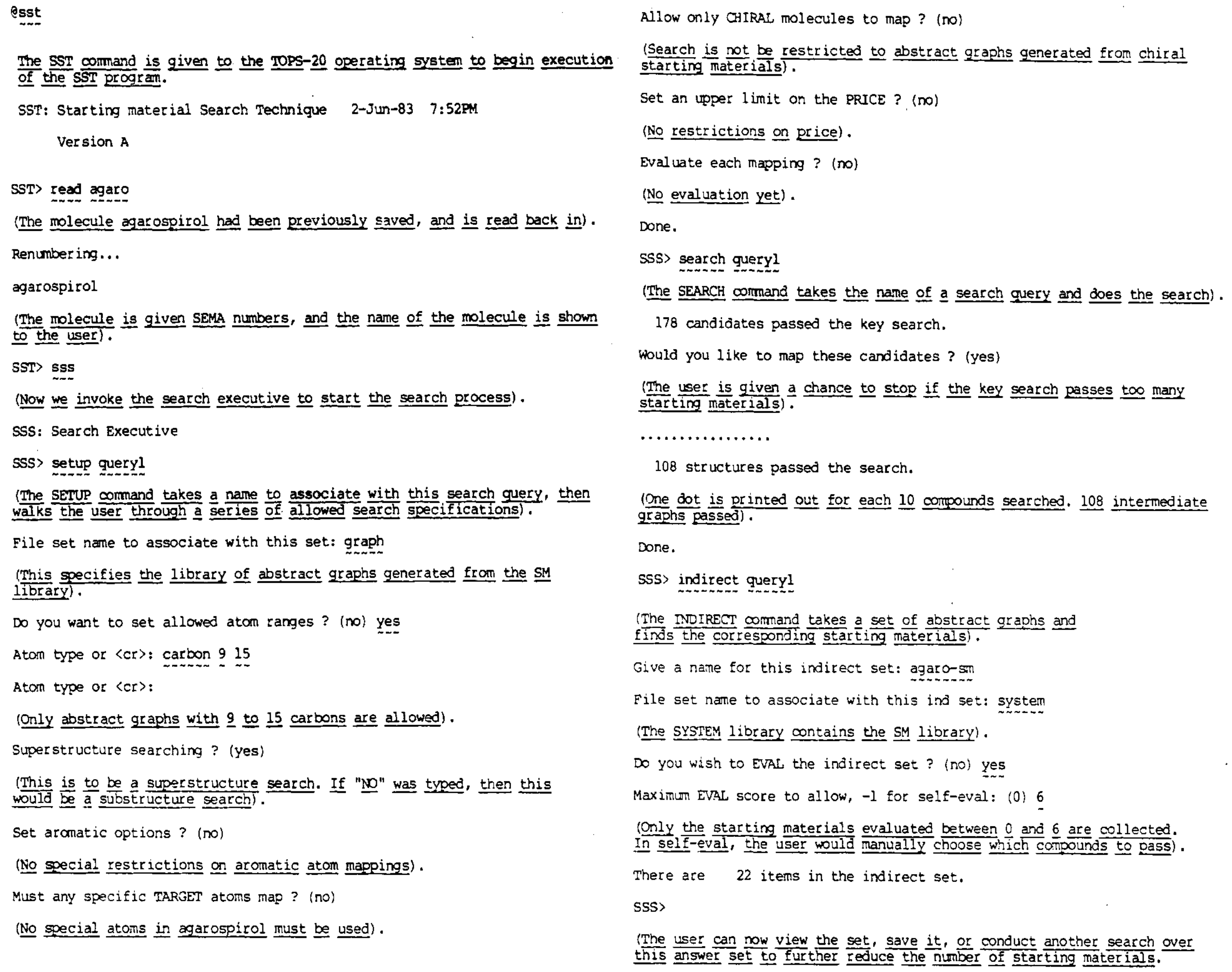
However, the SECS program was unable to discover two syntheses that involved degradative sequences in the reaction pathways. The synthesis of Hobbs and Magnus[22] uses an optically active starting material as the basis of the synthesis. The synthesis of Trost and Keeley[23] constructs a fused spiro bicyclohexanone system from simple starting materials and then degrades the intermediate to the desired grandisol. The Hobbs and Magnus scheme is starting material based, while the Trost and Keeley scheme was not. We were interested in seeing if the starting material used by Hobbs would also be suggested by the SST program.

The top-level abstraction rules applied to grandisol give the abstract graph shown in Figure 18; when a substructure search of this graph was performed, no graphs in the abstract library were found to be superstructures. Since no compounds were discovered with the original abstraction, the graph was abstracted further by removing the former carbon-carbon double bond and the carbon-carbon bond initially next to the hydroxyl group. This discovered eight intermediate-level abstract graphs and led to 14 possible starting materials.

The starting material used in the synthesis of Hobbs was β-pinene. Though that compound was not in our starting material file, α-pinene was equivalent for the purposes of the synthesis; this was found and rated one of the best candidates. In fact, half of the found starting material candidates were pinenes. Figure 19 shows the results of the search and their evaluation scores.

While many features of the substructure search for degradative starting materials could be conducted with any substructure search system, the use of abstraction to reduce the initial search and the evaluation of the candidates are unique to the approach of the SST program. The addition of these

Chart I



```
@sst

 The SST command is given to the TOPS-20 operating system to begin execution
 of the SST program.

 SST: Starting material Search Technique   2-Jun-83  7:52PM

      Version A


SST> read agaro

(The molecule agarospirol had been previously saved, and is read back in).

Renumbering...

agarospirol

(The molecule is given SEMA numbers, and the name of the molecule is shown
to the user).

SST> sss

(Now we invoke the search executive to start the search process).

SSS: Search Executive

SSS> setup queryl

(The SETUP command takes a name to associate with this search query, then
walks the user through a series of allowed search specifications).

File set name to associate with this set: graph

(This specifies the library of abstract graphs generated from the SM
library).

Do you want to set allowed atom ranges ? (no) yes

Atom type or <cr>: carbon 9 15

Atom type or <cr>:

(Only abstract graphs with 9 to 15 carbons are allowed).

Superstructure searching ? (yes)

(This is to be a superstructure search. If "NO" was typed, then this
would be a substructure search).

Set aromatic options ? (no)

(No special restrictions on aromatic atom mappings).

Must any specific TARGET atoms map ? (no)

(No special atoms in agarospirol must be used).

Allow only CHIRAL molecules to map ? (no)

(Search is not be restricted to abstract graphs generated from chiral
starting materials).

Set an upper limit on the PRICE ? (no)

(No restrictions on price).

Evaluate each mapping ? (no)

(No evaluation yet).

Done.

SSS> search queryl

(The SEARCH command takes the name of a search query and does the search).

  178 candidates passed the key search.

Would you like to map these candidates ? (yes)

(The user is given a chance to stop if the key search passes too many
starting materials).

.................

  108 structures passed the search.

(One dot is printed out for each 10 compounds searched. 108 intermediate
graphs passed).

Done.

SSS> indirect queryl

(The INDIRECT command takes a set of abstract graphs and
finds the corresponding starting materials).

Give a name for this indirect set: agaro-sm

File set name to associate with this ind set: system

(The SYSTEM library contains the SM library).

Do you wish to EVAL the indirect set ? (no) yes

Maximum EVAL score to allow, -1 for self-eval: (0) 6

(Only the starting materials evaluated between 0 and 6 are collected.
In self-eval, the user would manually choose which compounds to pass).

There are    22 items in the indirect set.

SSS>

(The user can now view the set, save it, or conduct another search over
this answer set to further reduce the number of starting materials.
```

features may be of interest to designers of chemical structure search systems.

CONCLUSIONS

The SST program is operational and has been tested on a variety of target compounds. It has proven itself to be a capable assistant toward aiding the chemist in selecting starting materials that comprise a significant amount of a target molecule. It can rapidly scan a large data base, presenting the chemist with a range of selections both useful and stimulating.

The technique of hierarchical search proved to be an efficient manner to represent the problem space and, to our knowledge, is the first application of this methodology to chemistry. Specifically, the top-level abstraction of the starting material was shown to allow for the discovery of over 90% of a published collection of starting materials, and studies further showed that the abstracting significantly reduced the search space. The evaluation algorithm allowed molecules with the same top-level abstract graph to be distinguished through differences in functionality. The application of the evaluation algorithm on the intermediate abstract graphs rather than the starting materials significantly reduced the computational effort in the evaluation.

SST stresses interaction with the user and allows the user to play a major role in determining both the range of the search and the features used during the selection process. We have added a graphical input/output section that allows the user to interact with the program by using a DEC GT-40 display system; the display package we used is derived from the SECS[24] program.

One limitation is that the evaluation function for the intermediate abstracted graph misses cases where the similarity between the starting material and the target depends on heteroatom bonds. For example, in the agarospirol search, we found the target compound itself, but the score was not zero. The evaluation function assumes that target bonds must be made at all places where the mapping fails to continue; this does not take into account the fact that if we mapped onto the starting material itself, we may be able to find these bonds already in our starting material. An additional level of evaluation is needed to handle such cases.

A second limitation is that 10% (Judging of the Abstraction Algorithm) of the syntheses involve combinations of construction and degradation or ring expansion/contraction and fall into class IV. We are exploring methods to aid discovery of starting materials in these classes as well.

We introduced the term "superstructure search" and showed that it can be applied efficiently to large files of compounds with essentially the same algorithm as substructure searching and that inverted key lists can be used. We further proved that the efficiency of key screening is the same in superstructure search as substructure search and that the larger number of starting materials obtained from superstructure search was a result of the molecular size distribution of the starting material file. We believe that superstructure search and abstraction may have other interesting applications in chemistry.

Finally, we have shown that by concentrating on graphical similarity and functionality of the starting material candidate, we have achieved our goal of making selection of starting materials independent of reaction knowledge and have in a

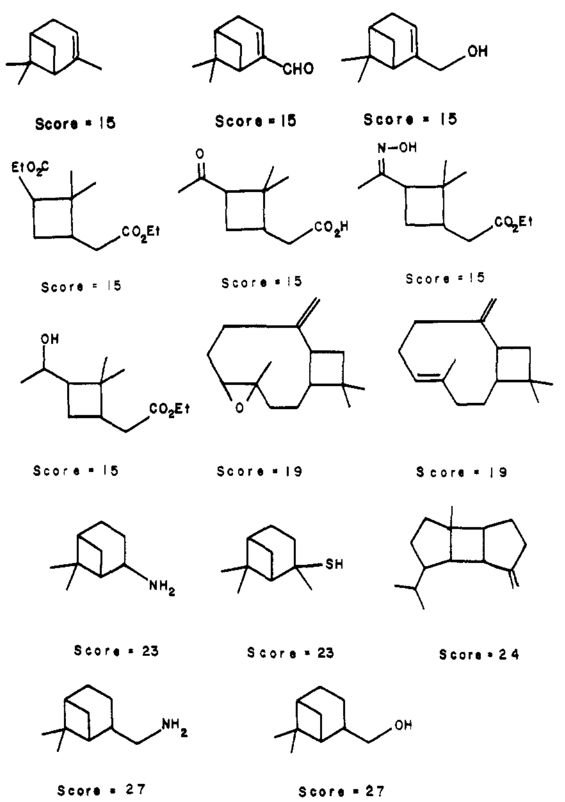


Figure 19. Starting materials for degradative synthesis of grandisol proposed by the SST program.

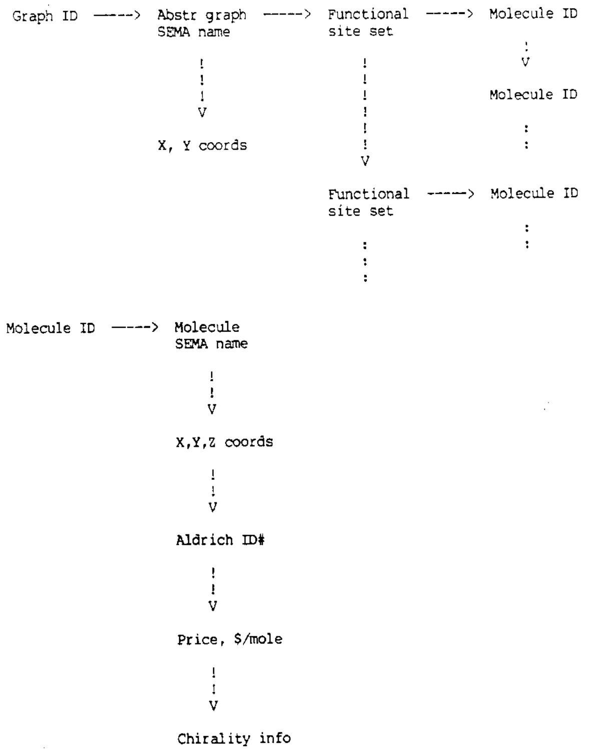


Figure 20. General data structure used in the SST program.

sense modeled the chemist's "intuitive leap" with an algorithmic "heuristic leap" through *Gestalt* pattern matching.

EXPERIMENTAL

The 11 000-compound starting material library is a condensation of a major portion of the Aldrich catalog supplied by Molecular Design, Ltd. As originally supplied, the catalog contained most of the main Aldrich catalog and parts of the specialty catalog. Our processing involved discarding molecules that had atom types other than C, H, N, O, P, S, or Cl, removing molecules that involved fragments, and removing molecules having more than 72 atoms.

The SST program contains ~26 000 lines of FORTRAN code, with small amounts of DEC-20 MACRO code. The program was designed to facilitate eventual conversion to both IBM and DEC-VAX systems. The program is currently running on a DEC-20 system with the SUMEX-AIM computer located as Stanford. Access to an experimental version of the program can be arranged for interested readers.

The data utilized by the program is kept in a structured disk file. Figure 20 shows the general data structured.

ACKNOWLEDGMENT

This work was supported in part by the Biotechnology Resources Program, Division of Research Resources, National Institutes of Health, through Grant RR01059, through an allocation of the SUMEX Resource, Grant RR00785, and through grants from the Stauffer Chemical Co. Special thanks go to the staff and administration of SUMX, who provided a quality environment for our research. We acknowledge Peter Sheperd for his assistance in transforming the database for this work and for his support by the NSF-URP program. We also thank Molecular Design, Ltd., for generously providing the Aldrich catalog in connection table format and Doug Hounshell and Steve Peacock for assisting in its preparation.

Registry No. Agarospirol, 1460-73-7; grandisol, 26532-22-9.

REFERENCES AND NOTES

(1) Gelernter, H. L.; Sanders, A. F.; Larsen, D. L.; Agarwal, K. K.; Bovie, R. H.; Spritzer, G. A.; Searlemen, J. E. *Science (Washington, D.C.)* **1977**, *197*, 1041.

(2) Corey, E. J.; Wipke, W. T. *Science (Washington, D.C.)* **1969**, *166*, 178.

(3) Corey, E. J.; Wipke, W. T.; Cramer, R. D.; Howe, W. J. *J. Am. Chem. Soc.* **1972**, *94*, 421.

(4) Hendrickson, J. B. *J. Am. Chem. Soc.* **1971**, *93*, 6847.

(5) Moreau, G. *Nouv. J. Chim.* **1978**, *2*, 187–193.

(6) Woods, W. In "Readings in Artificial Intelligence"; Nilsson, N.; Webber, B., Eds.; Tioga Publishing distributed by William Kaufman: Los Altos, CA, 1981.

(7) The name of this program suggests an analogy with the SST airplane in its ability to raidly traverse large distances in one jump, nonstop.

(8) Wipke, W. T.; Krishnan, S.; Ouchi, G. *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 32.

(9) Wipke, W. T.; Dyott, T. M. *J. Am. Chem. Soc.* **1974**, *96*, 4834.

(10) Sridharan, N. S. Ph.D. Dissertation, State University of New York at Stonybrook, 1971.

(11) Naming of search methods has been by the *relationship of the query to the library*; i.e., substructure search looks for all library compounds such that the query is a substructure of the library compund. Superstructure search thus finds library compounds such that the query is a superstructure of the file compounds.

(12) Wipke, W. T.; Dill, J. D.; Peacock, S.; Hounshell, D. "Search and Retrieval Using an Automated Molecular Access System"; 182nd National Meeting of the American Chemical Society, New York, Aug 1981.

(13) Howe, W. J.; Hagadone, T. J. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 8–15.

(14) Dittmar, P. G.; Stobaugh, R. E.; Watson, C. E. *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 111–121.

(15) Sacerdoti, E. D. *Artif. Intelligence* **1974**, *5*, 115–135.

(16) Bindra, J. S.; Bindra, R. "Creativity in Organic Synthesis"; Academic Press: New York, 1975; Vol. 1.

(17) Mongrain, M.; Lafontaine, J.; Belanger, A.; Deslongchamps, P. *Can. J. Chem.* **1970**, *48*, 3273.

(18) McNulty, P. J.; Smith, R. B.; "Substructure Searching of Large Chemical Files"; Proceedings of the Manufacturing Chemists Association; Manufacturing Chemists Association, Washington, D.C., Aug 1979.

(19) Sets are symbolized by capital letters and individual entities by lowercase letters. The complement of a set is indicated by the "~" symbol, e.g., ~*A*.

(20) Olmsted, J. M. "Advanced Calculus"; Appleton: New York, 1956.
(21) Wipke, W. T.; Ouchi, G. I.; Krishnan, S. *Artif. Intelligence* **1978**, *9*, 173–193.
(22) Hobbs, P. D.; Magnus, P. D. *J. Chem. Soc., Chem. Commun.* **1974**, 856–858.
(23) Trost, B. M.; Keeley, D. E. *J. Org. Chem.* **1975**, *40*, 2013.
(24) Wipke, W. T.; Braun, H.; Smith, G.; Choplin, F.; Sieber, W. "Computer-Assisted Organic Synthesis". *ACS Symp. Ser.* **1977**, *61*.

Chemical Inference. 2. Formalization of the Language of Organic Chemistry: Generic Systematic Nomenclature

JOHN E. GORDON[1]

Chemical Abstracts Service, Columbus, Ohio 43210, and Department of Chemistry, Kent State University, Kent, Ohio 44242

Received October 14, 1983

The role and importance of generic nomenclature in the definition and manipulation of structural formula and compound classes in chemistry, chemical information science, and chemical education are discussed. Traditional generic organic nomenclature is surveyed, and a formalization of one traditional language of generic names is presented. Principles of naming generic structural formulas that involve structural variables such as 'R', 'X', etc. are discussed. A formal description of a language for naming and indexing connectivity-variable generic structural formulas of both fixed and variable composition is provided.

MOTIVATIONS TO GENERIC NOMENCLATURE

Indexable Line Notation in 1:1 Correspondence with Generic Structural Formulas. In a recent discussion of languages of generic structural formulas (GSFs), we noted their importance as vehicles for the precise definition and discussion of compound and structural formula (SF) classes.[2] Some of the applications identified were for chemical inference (with and without mechanization), for communication with organic chemistry learners, for discussing substructure search, and for constructing and searching chemical patent claims. Much the same significance attaches to languages of systematic generic names, which stand in the same relation to GSFs as specific systematic names bear to individual structural formulas. And, as with their specific counterparts, generic names (GN) correct one major deficiency of GSFs: GNs are indexable whereas GSFs are not.

Use in Database Searching. Much compound/SF-oriented searching is generic, i.e., directed at compound/SF classes. In addition to much of the patent literature, this includes all searches involving unknown (as yet unisolated) compounds and those involving unknown properties of known compounds. The latter types of searching are carried out as searches for structural analogues of the actual target structures. Chemists routinely employ concepts rich in structural analogies not only in their information-seeking behavior but also in their experimental design and inferential activities. Since structural analogies can be formulated on several different dimensions corresponding to choice of different attributes as analogous, and also in various degrees of strength, a given problem or search may at once involve several different SF classes as analogues for the same unknown compound. Thus, if I wish to find information on the photochemical properties of γ-chloro, α,β-unsaturated amidines, and no such specific information exists, I may wish to search for molecular orbital calculations on unsaturated amidines, for photochemical properties of amidines in general, and so on.

Despite the importance and frequency of such SF class searches, they are easy to carry out only in files possessing either strongly hierarchic organization (e.g., Beilstein) or indexes containing large numbers of SF class enttries.[3] The large bibliographic files most useful for specific SF/compound searching (e.g., *Chemical Abstracts*, *CASearch*) are not so indexed.

Formalization Exposes Incompleteness, Inconsistency, and Ambiguity. As with all intuitive naming schemes, the traditional generic nomenclature (see below) is difficult to use because it is nonuniform. Thus, while I may call a certain SF class the *aryl alkyl ethers*, others may have discussed or indexed it under *aromatic ethers*, *alkoxy aromatics*, *alkoxy arenes*, *aryloxy compounds*, etc. A second difficulty lies in the existence of a generic nomenclature only for heterocomposite SF classes,[2] not for classes framed in terms of variable connectivity at constant composition. Only in sporadic cases do we have reasonably systematic names for sets of isomeric SFs—even sets of closely related isomers. We generally resort either to ambiguous *specific* systematic names that can be interpreted as naming SF classes, for example, "dichlorobenzene", or to natural language descriptions of the class. Examples are "1-phenyl-3,5-hexanedione and its tautomers", "2-amino-5-sulfo-1-naphthoic acid and its betaines", and "the isomeric decanes".

Sharpening the Chemist's Accuracy of Expression. Generic names share, in a visually less immediate but often more concise form, the role of GSFs as conceptual tools for visualizing, designing, and specifying subclasses of compounds in which some features are constant, others variable. As in most situations involving language use, chemists tolerate considerable levels of ambiguity in the description of SF classes, because in many cases local convention or quick (conscious or unconscious) inference makes the meaning clear. In some situations, however, such ambiguity is not tolerable. These include on the one hand the formulation of index entries and search queries. Equally important, in all formal or informal information transfers that involve learners, ambiguity is destructive of learning because the learner lacks just the chemical intuition that the expert uses to resolve ambiguity.

DESIDERATA FOR SYSTEMATIC GENERIC NAMES

Generic names (1) should have good formal continuity with specific systematic names, (2) should exist hierarchically in 1:1 correspondence with GSFs, (3) should make as explicit as possible (a) the structurally known vs. the structurally