Computer Storage and Retrieval of Generic Chemical Structures in Patents. 16. The Refined Search: An Algorithm for Matching Components of Generic Chemical Structures at the Atom-Bond Level

J. D. Holliday and M. F. Lynch*

Department of Information Studies, University of Sheffield, Sheffield, S10 2TN, U.K.

Received July 29, 1994[⊗]

A description is given of an atom-level search strategy for matching components of generic chemical structures, called the refined search. The components used are those represented by nodes of a reduced graph. These nodes describe aggregates of atoms of the original chemical graph which are similar in structure or chemistry. The reduced graph screening search results in a list of pairs of matched nodes, one query structure node and one file structure node in each pair, which is the basis for the refined search. The atoms and bonds of a query structure node are compared with those of a file structure node. Due to the nature of generics, parts of the structure are expressed either in terms of real atoms and bonds or in terms of homologous series identifiers, such as *alkyl*. The fundamental problem concerning an atom-level search is the translation between the two types of representation for these different expressions. An adaptation of the Ullmann algorithm for substructure isomorphism has been devised to solve this problem; this is described together with several example matches.

1. INTRODUCTION

The types of structure variation found in generic chemical structures is well documented[1] and has been summarized by Dethlefsen[2] as four types. These are as follows:

Substituent Variation. This type of variation occurs as a list of alternative substituents on a ring or chain. A ring may be substituted by the variable X, for example, which is later defined as "X is hydroxyl, methyl, or chlorine".

Positional Variation. This type of variation occurs in the positions of attachment through which one component is attached to another, e.g., monochlorophenyl.

Frequency Variation. This is a variation in the frequency of occurrence of a component within the structure, possibly through a choice in the frequency of substitution or by a choice in the frequency of repetition if a group within a chain or ring, e.g., "$-(CH_2)_n-$, n is 0, 1, or 2".

Homology Variation. This is the use of a standard nomenclatural term to represent a series of compounds with common structural features. These compounds are homologous to each other, and the term used in the notation is called a *homologous series identifier*. The term is often further qualified by structural property descriptions defined in terms of numerical values or ranges, e.g., "alkyl 1-6C".

There is an obvious partitioning of generic structures into "partial structures", each of which is related to a single expression in the patent document such as a structure diagram, a line formula, or a radical term. These partial structures are either specific, in which case they can be represented by a structure diagram or they are generic, expressed as homologous series identifiers. In the latter case there are no real atoms, and a representation which describes structural features attributable to the compound is used. In this representation, parameters numerically describe features such as "number of heteroatoms" or "number of ternary branch points".

The size and complexity of generic structures means that an atom level search on the full representation must be preceded by faster but less detailed screening stages. Several representations for generic structures have been developed and are described in previous papers in this series.[3−8] These are the full computer representation (the extended connection table representation[3] or ECTR), the reduced graph representation,[4,5] and the bit screens.[6−8] The diversity in types of representation is due to the need for a more simplistic view of the structure on which the less detailed screening stages may operate.

The reduced graph is a representation of generic chemical structures in which regions of the chemical graph have been replaced by a single graph node. The criterion for such reduction is based on the identification of regions of the chemical graph which are similar in terms of chemistry or structure. The most notable type of reduced graph uses nodes which represent aggregates of ring atoms, noncyclic carbons, and noncyclic heteroatoms. Each node of the reduced graph is mapped onto the original chemical graph by the inclusion of a list of atoms which it represents. However, in the case of components which are expressed by homologous series identifiers, HSIs, there are no real atoms, and the address of the generic partial structure is listed.

The result of such reduction is a tree-structured graph which contains AND and OR branches and optional branches which arise due to the presence of hydrogen as an alternative substituent. As well as a list of respective atoms, each reduced graph node contains a list of parameters, similar to those used to describe generic partial structures, which describe certain structural features such as the number of heteroatoms. An example generic structure is shown in Figure 1 together with its respective reduced graph (optional connections are indicated by a broken line). Note that the partitioning of the full structure representation into partial structures and the partitioning of the reduced graph into graph nodes are not equivalent. As a result a reduced graph node

[⊗] Abstract published in *Advance ACS Abstracts,* December 1, 1994.

0095-2338/95/1635-0001$09.00/0

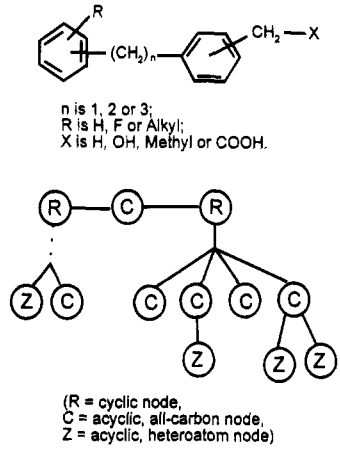


Figure 1. Example generic structure and its reduced graph.

may span any number of partial structures, and a partial structure may reduce to any number of reduced graph nodes.

Previous papers in this series have reported on a type of reduced graph called the "composite reduced graph". In the composite reduced graph, nodes of the same type which are alternatives to each other are collapsed into a single node representation. The parameters of the nodes are conflated into ranges which cover all the alternative parameter values. This treatment reduces the size and complexity of the reduced graphs making storage and searching less computationally demanding. There is a loss of topological information due to the conflation, however, as well as a loss of mapping of the reduced graph nodes onto the chemical graph. This mapping is essential to the refined search which matches the atom-level representations (i.e., atoms and bonds and/or generic partial structures) associated with the reduced graph nodes. This has meant the development of a new type of reduced graph, called the "nonalternant reduced graph", in which no collapsing of nodes is carried out. The result is a reduced graph with a greater number of nodes, which retains the mapping to the original structure as well as the topological information. Screening searches have been carried out using files of nonalternant reduced graphs; the results were presented at the Third International Conference on Chemical Structures at Noordwijkerhout, Holland, 1993.[9]

The result of the reduced graph screening stage, in the case of a successful match, is a mapping of the nodes of the graph representing the query structure to those of the file structure. This mapping is expressed in the form of a list of matched node pairs, one query structure node and one file structure node in each pair. These nodes have been matched at the level of node labels and parameter descriptions within the global node-by-node matching procedure. The refined search compares the two nodes at the more detailed level of atoms and bonds in the case of specific components and parameters in the case of generic components. The process is then to match the regions of the two ECTRs which correspond to the two graph nodes in the pair. The regions of the ECTR corresponding to reduced graph nodes may be expressed in terms of real atoms and bonds, represented as a partial connection table, or in terms of homologous series identifiers, represented as a list of parameters. More importantly, the respective regions of the ECTR may be expressed as a combination of specific and generic components since a reduced graph may span more than one partial structure. The fundamental problem concerning the refined search is the translation between specific and generic components, i.e., components which are expressed in terms of real atoms and bonds and represented by connection tables, and those which are expressed as homologous series identifiers and represented by parameters. This means mapping atoms not just one-to-one but one to N and even N to N.

The solution to this problem, which uses an adaptation of Ullmann's algorithm for subgraph isomorphism, is described in section 2. Of the four types of structure variation described above, the algorithm solves the problem of homology variation only. Subsequent variation is an internodal problem rather than a intranodal one since each alternative substituent, or combination of substituents, produces a separate alternative branch in the reduced graph. A unique representation is therefore held for each alternative, a feature which would not be seen of the composite reduced graph. Figure 1 shows how the partitioning of the nonalternant reduced graph explicitly reflects the logic between alternative substituents. As a result, variation is cured during the node-by-node mapping stage of the reduced graph screening search. Frequency variation continues to be a problem for all search strategies and will be dealt with in the future. Positional variation is found within reduced graph nodes as it occurs between partial structures which may be part of the same reduced graph node. The solution to this problem is described in section 3. Section 4 describes some example matches using the combined algorithm which results from sections 2 and 3.

Matching the regions of the ECTRs must go beyond the boundary of the reduced graph nodes which they represent. The reason for this is described in section 5 together with the solution to this problem.

2. SEARCH ALGORITHM

Choice of Algorithm. Many algorithms have been devised to solve the problems of graph isomorphism and subgraph isomorphism. The ideas put forward for chemical graph isomorphism are diverse, but all involve real atom representation of the chemical structure. Some of the notable algorithms are described here together with their weaknesses for matching generic chemical graphs.

Sussenguth[10] describes an algorithm to match chemical graph based on set reduction. Atoms and bonds are grouped into sets of structurally similar items. An atom set might contain the atom numbers for the carbon atoms in the structure, for example, or a bond set might contain the double bond labels in the structure. This process is repeated for query and file structure. The sets are then compared to find correspondences between the two structures. Figueras[11] adapted this algorithm by including atom desciptors, or codes, as a prescreening mechanism. This effectively reduces the size of each set by making each atom more unique. The problem with set reduction is that correspondences are made on equivalent atoms and bonds on a one-to-one basis. The N:N relationship required in a generic search can be incorporated into the algorithm, but there would be minimum reduction in set size since generic nodes lack the chemical, structural, and environmental features which are used to uniquely identify nodes in any correspondence.

Von Scholley[12] describes a method of matching generic chemical structures, based on Kitchen and Krishnamurthy's suggestion for using relaxation as a chemical structures

screening mechanism,[13] which is similar to set reduction and attempts to solve some of the problems of generics. In the process real atoms are represented by sets which describe them using various levels of specificity, from the specific level of real atom labels to the generic level of "any atom". The algorithm solves the problems of substituent variation and positional variation but is most effective on real atom representations, i.e., nonhomology variant graphs. The algorithm is not as efficient as that required of an atom-by-atom search and can be regarded as a screening mechanism only.

Tokizane *et al.*[14] have attempted to solve the translation problem by using structure attributes, similar to the parameters used to describe generic partial structures in the Sheffield system, which are represented as bit vectors. The attributes are assigned to all atoms, specific or generic, producing an equivalent representation for both types of expression. The matching algorithm is based on the connectivity stack developed by Kudo and Sasaki.[15] Attributes, or parameters, have proven to be a powerful tool in solving translation problem and are the basis for the reduced graph search. The choice of matching algorithm is not suitable, however, to the refined search as it does not take into account all of the features of generic structures, notably positional variation.

Owolabi[16] describes a method of structure match based on a string representation of the connection table. The string is searched using a standard best-matched *n*-gram method. Again, the translation between generic and specific components would be difficult to superimpose on this algorithm, and the high performance would not be expected.

Ullmann's algorithm[17] has been adapted for chemical substructure isomorphism extensively.[18-22] The algorithm operates on adjacency matrix representations of the chemical graph and is effectively a backtracking algorithm.[23] The adaptation of the algorithm to solve the *N*:*N* problem inherent in generic structure matching is not difficult and is described below. The fact that is a subgraph isomorphism algorithm is advantageous for two reasons. Firstly, the same algorithm can be used for full graph and subgraph matches; a full graph search merely requires the complete set of file atoms to be matched instead of a subset. Secondly, the nature of the homology variant components of generic structures likens the N:N translation problem to a subgraph problem. The mapping of a *n*-propyl, for example, onto a generic component expressed as "alkyl C1-6" is more alkin to a subgraph problem than to a full graph. The original algorithm is explained, followed by a description of the adaptation for generic chemical graphs.

The Ullmann Algorithm. The Ullmann algorithm is a depth-first graph-matching algorithm which operates on the adjacency matrices, *A* and *B*, of the two graphs together with a matching matrix *M*. In the matching matrix rows represent the nodes of the query structure, columns represent nodes of the file structure, and each Boolean element of the matrix indicates whether a match is possible between the two nodes. The algorithm uses a refinement procedure which improves the performance of the depth-first processing. The refinement procedure is called for every node pairing attempted in the match and results in the pruning of the many branches in the search tree that would lead to a mismatch.

The matrix *M* is initially filled by a "first look" at each node pairing to see if they are a possible match. This is based on the node labeling (e.g., atom type) and the degree of each node. Then, for the first node ($i = 1$) of the query, a single match is tested using one of the first nodes (j) which might match the query structure (i.e., $M(i,j)$ = true). Other elements in that row are set to false. The matrix is then qualified using the refinement procedure which examiners all of the matches in the matrix and ensures consistency between their neighbors. If this refinement fails, then the next match is tested for that row; if not, then the next node of the query (i.e., the next row of the matrix) is operated on. The process is repeated recursively until a complete match is found or until an empty row (i.e., all elements are false) is found indicating a mismatch. The procedure then backtracks due to the recursion. During the whole process, an array *F* is used to represent those columns which have already been assigned.

```
program ullmann

var A, B, M, F, isomorphisms;

   procedure depthfirst (d, M);

   var M1, F1, j, mismatch;

   begin
   M1 ← M; F1 ← F;
      repeat
      select new column j such that M(d,j) <> 0 and F(j) = false;
      F(j) ← true;
      set all other elements in row M(d) to zero;
      refine (M, mismatch);
      if not mismatch then
         begin
         if d = query size then
            isomorphisms ← isomorphisms + 1;
         end;
      else
         depthfirst (d+1, M, F);
      M ← M1; F ← F1;
      until (j > file strucure size)
   end;

begin
create_matrices (A, B, M);
set all elements in F to false;
if check_m(M) then
   begin
   isomorphisms ← 0;
   depthfirst (1, M, F);
   end;
end.
```

Figure 2. Psuedocode for the Ullmann algorithm.

Pseudocode for the algorithm is shown in Figure 2. The refinement procedure is based on the principle that for any query atom a_i that can be mapped onto a file structure atom b_j there must exist a neighbor b_y of b_j for every neighbor a_x of a_i, and every such neighbor b_j must be mapped onto the neighbor a_x. If this is not so, then $M(i,j)$ is set to false. The procedure uses the adjacency matrices *A* and *B* to resolve this correspondence.

The algorithm was originally written for general graph theory and can be adapted for use on chemical graphs by labeling the nodes with the chemical symbol and also labeling the adjacency matrix elements with bond types.

An Adaptation of the Ullmann Algorithm. The generic version of the Ullmann algorithm is designed to relax the 1:1 match between graph nodes so that there may be a 1:*N* relationship between real atoms and generic components or an *N*:*N* relationship between generic and generic components. Generic components are treated as single atoms and are therefore represented in the adjacency matrices as a single row/column. One important feature is inclusion of two arrays, *AUsed and BUsed*, which indicate the maximum number of atoms in each row of the adjacency matrices *A* and *B*, respectively. For rows which represent real atoms,

this value is initially one, but for rows representing generic components this value is initially set at the maximum atom count expressed by the homologous series identifier. For "alkyl C1-6" this would be 6, for example, this value being taken from the parameter values which represent the generic partial structures of the ECTR. These arrays replace the single array *F* of the original algorithm.

The two arrays facilitate the repeated use of a row or column of the matching matrix *M*, mimicking the match between two atoms, one of the query and one of the file structure, whether they be from real or generic components. The respective elements of the arrays *AUsed* and *BUsed* are decremented by one accordingly when such a match is made. A row or column of the matrix *M* may then be used repeatedly until the value of the array element is zero.

The algorithm is essentially the same as the original, the main alterations are as follows:

Any column may be used more than once providing that the value of *BUsed(j)* and *M(i,j)* are greater than zero.

A row is used repeatedly by including a second call to the procedure *depthfirst* in the recursion for rows representing generic components (i.e., if *AUsed* > 0).

The elements of the matrix *M* are used to count the matches made between the two respective atoms. Where two generic components overlap, the number of atoms which coincide are indicated by this count.

During the processing, a subset of the parameters is enumerated for each generic component; this is called a "feature record". As a generic component is mapped onto a specific atom, any specific feature which it exhibits, such as ternary branching, decrements the associated feature record parameter accordingly. This allows the constant monitoring of such features so that their maximum value is not exceeded.

Considerable changes have been made to the refinement procedure and to the initial matrix creation procedure *create_matrices* since these must now operate on real neighbor descriptions as well as those inferred by generic component descriptions.

The procedure *create_matrices* initializes the matrix *M* by examining the atom types and neighboring connections in the adjacency matrices *A* and *B*. In the original algorithm, if two atoms, *i* in *A* and *j* in *B*, are of the same type and have the same number and types of connection, then the element *M(i,j)* is set to one; otherwise it is set to zero. For generic nodes, only the features expressed in the generic description can be compared, degree of branching and unsaturations for example. As a result, generic nodes usually match every node in the initial mapping stage.

In the refinement stage generic nodes may match against a group of real atoms, so a correspondence must be found between each neighbor of the generic node and a neighbor of any of the real atoms against which it matches. In other words, for a generic node to map onto a group of real atoms, the neighbors of the generic node must map onto the neighbors of the group of real atoms. In the case of a generic−generic mapping, no such correspondence can be investigated. One important consideration is that this refinement cannot be carried out until a row representing a generic node has been completed, in the case of generic query nodes, and the process has moved on to the next level in the depth-first search (or next row), or until the complete matrix has been processed in the case of generic file nodes.

```
program generic_ullmann;

var A, B, M, AUsed, BUsed, isomorphisms;

   procedure depthfirst (d, M, AUsed, BUsed, First);

   var M1, BUsed1, j, mismatch;

   begin
   AUsed ← AUsed - 1;
   M1 ← M; BUsed1 ← BUsed;
      repeat
      select new column j such that BUsed(j) <> 0 and for generic row: TestGRow
                                                  or for specific row: M(d,j) <> 0;
      BUsed(j) ← BUsed(j) - 1;
      set all other elements in row M(d) to zero;
      M(d,j) ← M(d,j) + 1;
      refine (M, mismatch, First);
      if not mismatch then
         begin
         if d = query size then
            if TestGColumn then
               isomorphisms ← isomorphisms + 1;
         end
      else
         begin
         depthfirst (d+1, M, AUsed, BUsed, true);
         if generic row then
            depthfirst (d, M, AUsed, BUsed, false);
         end;
      M ← M1; BUsed ← BUsed1;
      until (j > file strucure size)
   end;

begin
create_matrices (A, B, M);
set all elements in AUsed, BUsed;
if check_m(M) then
   begin
   isomorphisms ← 0;
   depthfirst (1, M, AUsed, BUsed, true);
   end;
end.
```

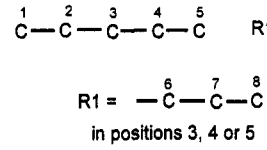
Figure 3. Psuedocode for the generic version of the Ullmann algorithm.

Pseudocode for the adapted algorithm is given in Figure 3. The procedure *TestGRow* makes sure that all new rows in *B* which match a generic node in *A* are neighbors to nodes previously matched with that generic node. The procedure *TestGColumn* makes sure that all columns in the matrix *M* which match a generic node in *B* are neighbors to nodes previously matched with that generic node.

3. POSITION VARIATION

Position variation can occur within reduced graph nodes since a node may span more than one partial structure and position variation may exist between partial structures. Such variation occurs in two instances, single-connection variation or double-connection variation. In single-connection variation an atom may be connected to one of several other atoms in the structure. Figure 4 shows an example of this type of variation. The adjacency matrix is incorrect since it describes all three connections between atom 6 and the parent partial structure where only one may be used. A separate array, *V1*, is used to indicate which connections are variable, shown by the letter *a* in the figure. In the refinement stage, there is then a choice of corresponding neighbors either in *A*, in *B*, or even in both *A* and *B* due to this variability. Processing requires that only one of the atoms may be used for a neighbor correspondence. The recursive backtracking nature of the algorithm allows the testing of all of the alternative connections until one of the choices results in a match, or not, as the case may be.

In doubly-connected position variation, there are two sets of paths between partial structures. Figure 5 is an example of such variation in which there are three possible pairs of attachment positions in the parent PS. The processing is



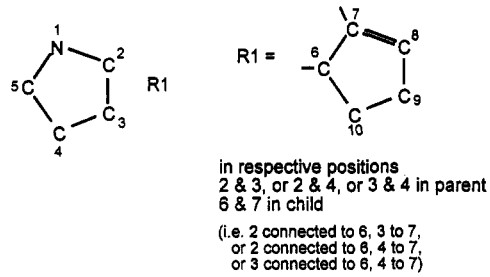
ADJACENCY MATRIX

	1	2	3	4	5	6	7	8
1	0	1	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0
3	0	1	0	1	0	1	0	0
4	0	0	1	0	1	1	0	0
5	0	0	0	1	0	1	0	0
6	0	0	1	1	1	0	1	0
7	0	0	0	0	0	1	0	1
8	0	0	0	0	0	0	1	0

V1

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	a	0	0
4	0	0	0	0	0	a	0	0
5	0	0	0	0	0	a	0	0
6	0	0	a	a	a	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Figure 4. Singly-connected position variation.



ADJACENCY MATRIX

	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	1	0	0	0	0	0
2	1	0	1	0	0	1	0	0	0	0
3	0	1	0	1	0	1	1	0	0	0
4	0	0	1	0	1	0	1	0	0	0
5	1	0	0	1	0	0	0	0	0	0
6	0	1	1	0	0	0	1	0	0	1
7	0	0	1	1	0	1	0	2	0	0
8	0	0	0	0	0	0	2	0	1	0
9	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	1	0	0	1	0

V2

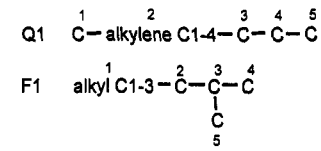
V2[2,6] ⟶ 3,7 ⟶ 4,7
V2[3,6] ⟶ 4,7
V2[3,7] ⟶ 2,6
V2[4,7] ⟶ 2,6 ⟶ 3,6

Figure 5. Doubly-connected position variation.

carried out as for singly-connected variation, and again any one of the possible atoms in the choice of neighboring atoms may be used for a neighbor correspondence. This time this situation occurs twice as there are now two paths between the PSs. What is important in double-connecting variation is that the two paths chosen are consistent with the information described in the structure. In Figure 5, if one path uses atoms 3 and 6, then the second path must be through atoms 4 and 7. For this reason, an array, *V2*, is set up and is used to check this consistency. The array contains the choices of second path which may be used for the first path alternatives. Element *V2[2,6]* in the example indicates that, if the first path chosen was through atoms 2 and 6, then the second path may be through atoms 3 and 7 or through atoms 4 and 7.

4. SEARCH EXAMPLES

Figures 6, 7, and 8 show the results of several searches on test structures which exhibit homology variation and posititional variation. The solutions are given in the form of the resultant matching matrices. These are further illustrated by graphs in which dotted lines are used to join the matching atoms. In the diagrams, the term A3 denotes



ADJACENCY MATRICES

Q1

	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	0	0
3	0	1	0	1	0
4	0	0	1	0	1
5	0	0	0	1	0

F1

	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	0	0
3	0	1	0	1	1
4	0	0	1	0	0
5	0	0	1	0	0

SOLUTIONS

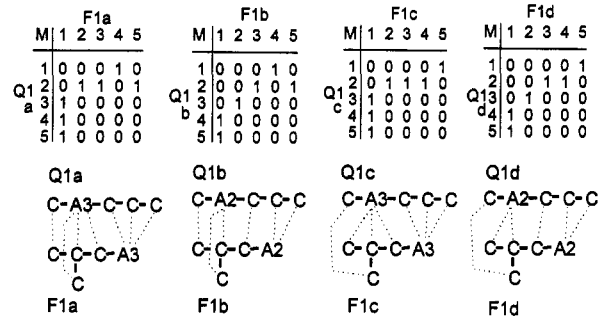
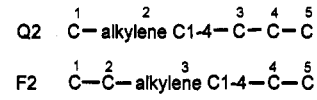


Figure 6. Example match Q1 against F1.



SOLUTION EXAMPLES

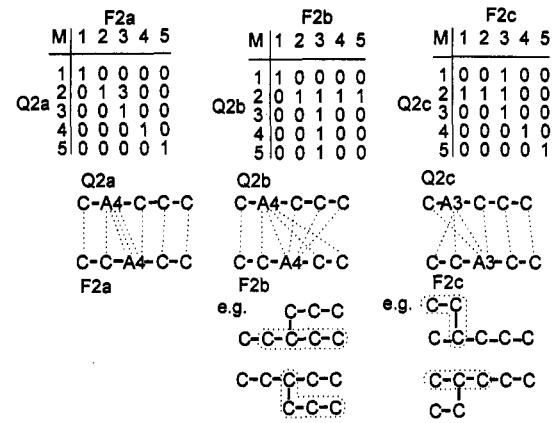


Figure 7. Example match Q2 against F2.

the respective alkyl or alkylene in which an atom count of 3 is exemplified, A4 denotes an atom count of 4, etc.

In Figure 6, the adjacency matrices for the query structure Q1 and the file structure F1 are shown. Note the 1:N correspondence between the first column of F1a, denoted F1a(1), and the three rows of Q1a, (Q1a(3), Q1a(4), and Q1a-(5)). This represents the match between real atoms of the query structure with the generic component of the file structure. In this example, three connected carbon atoms of the query have been matched with the HSI "alkyl C 1-3". Similarly, row Q1a(2) of the query, representing "alkyl C 1-4", has been matched against three columns of the file structure F1a, representing a group of connected carbon atoms. The 1:N correspondence is seen again in the other three solutions.

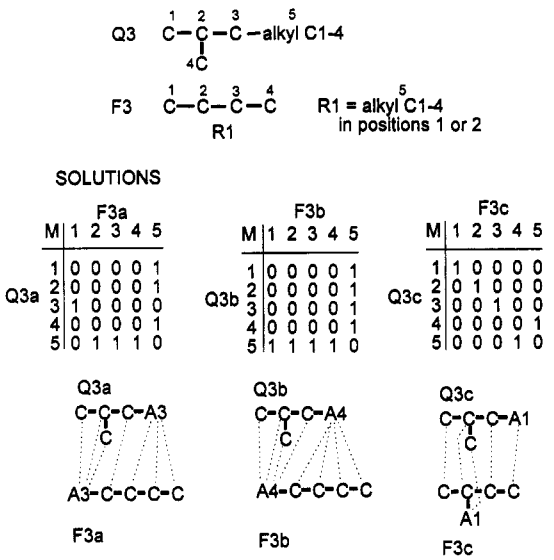


Figure 8. Example match Q3 against F3.

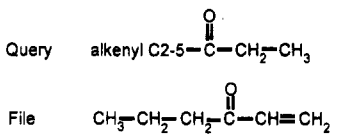


Figure 9. Internodal matching.

In Figure 7, there are many solutions, 14 in all, partly due to the symmetry of the file structure F2. Only some of these are given, the first of which illustrates the N:N overlapping relationship between two generic components. In this example three atoms of the query row Q2a(2) have been mapped on to three of the film column F2a(3), as shown by the figure "3" in the matrix, the fourth atom being mapped to a real atom F2a(2) of the file structure. Similarly, a fourth atom of the file column F2a(3) has been mapped on to a real atom, Q2a(3), of the query structure. The other two examples show 1:1 and 1:N relationships in isomorphoisms which are not immediately apparent and are therefore illustrated using specific examples of the HSIs. (Alkylene corresponds to the encircled carbons in the examples.)

Figure 8 illustrates the combination of homology variation and positional variation. The R1 substituent in the file structure F3 shows a choice of attachment to the carbon chain at atoms 1 and 2. In this example, both of the alternative positions for the alkyl in F3 produce solutions which are illustrated.

5. INTERNODAL MATCHING

So far, the examples given have dealt with matches involving atoms, specific or generic, contained within the reduced graph nodes and do not take into account the internodal mappings. The gross internodal mappings are given by the list of node pairs which result from the reduced graph match. It is important, however, to include a more detailed level of internodal mapping, at the atom level, in order to complete the matching process. Figure 9 illustrates two structures, one query and one file, which match at the reduced graph level of search and whose mapped nodes match at the intranodal level of refined search described above. These two structures represent a structure mismatch, however, due to the different positions of the atoms through which the nodes are connected. It is clear, then, that the connections between reduced graph nodes must be examined at the atom level.

This is achieved by a simple comparison of the atoms through which neighboring pairs if nodes in the query structure and the file structure are connected. There may be a choice of such atoms due to position variation.

6. CONCLUSIONS

A description has been given for a detailed search strategy, called the refined search, for matching components of generic chemical structures. This search strategy is the final and most discriminating stage of search used by the Sheffield system and follows the two faster screening stages, bit screening and reduced graph screening. The algorithm operates on components of the generic structure which have been isolated during the reduced graph search stage. Example matches have been shown for full structure search only, in which every row and column of the matching matrix must be used. Substructure search is carried out using the same algorithm with minor changes, since the Ullmann algorithm on which it is based is a subgraph isomorphism algorithm.

A Pascal computer program has been developed from the algorithm and has been tested on several structures. The results of a more detailed evaluation using several test databases of generic structures will be published at a later date.

ACKNOWLEDGMENT

We gratefully acknowledge funding from International Documentation in Chemistry mbH, Derwent Publications Ltd., Questel SA, and the Department for Education in support of the research described here.

REFERENCES AND NOTES

(1) Barnard, J. M. *Computer Handling of Generic Chemical Structures*; Gower: Aldershot, 1984.

(2) Dethlefsen, W.; Lynch, M. F.; Gillet, V. J.; Downs, G. M.; Holliday, J. D.; Barnard, J. M. Computer Storage of Generic Chemical Structures in Patemts. 11. Theoretical Aspects of the Use of Structure Languages in a Retrieval System. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 233–253.

(3) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 4. An Extended Connection Table Representation for Generic Structures. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 160–164.

(4) Gillet, V. J.; Downs, G. M.; Ling, A. I.; Lynch, M. F.; Venkataram, P.; Wood, J. V.; Dethlefsen, W. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 8. Reduced Chemical Graphs and Their Applications in Generic Chemical Structure Retrieval. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 126–137.

(5) Gillet, V. J.; Downs, G. M.; Holliday, J. D.; Lynch, M. F.; Dethlefsen, W. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 13. Reduced Graph Generation. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 260–270.

(6) Welford, S. M.; Barnard, J. M.; Lynch, M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 5. Algorithmic Generation of Fragment Descriptors for Generic Structure Screening. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 57–66.

(7) Holliday, J. D.; Downs, G. M.; Gillet, V. J.; Lynch M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 14. Fragment Generation from Generic Structures. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 453–462.

(8) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 10. Assignment and Logical Bubble-Up of Ring Screens for Structurally Explicit Generics. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 215–224.

(9) Holliday, J. D.; Downs, G. M.; Gillet, V. J.; Lynch M. F.; Dethlefsen, W. An Evaluation of the Screening Stages of the Sheffield Research Project on Computer Storage and Retrieval of Generic Chemical Structures in Patents. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 39−46.
(10) Sussenguth, E. H. A Graph-Theoretical Algorithm for Matching Chemical Structures. *J. Chem. Doc.* **1965**, *5*, 36−43.
(11) Figueras, J. Substructure Search by Set Reduction. *J. Chem. Doc.* **1972**, *12:4*, 237−244.
(12) Von Scholley, A. A relaxation Algorithm for Generic Chemical Structure Screening. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 235−241.
(13) Kitchen, L.; Krishnamurthy, E. V. Fast Parallel Relaxation Screening for Chemical Patent Data-Base Search. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 44−48.
(14) Tokizane, S.; Monjoh, T.; Chihara, H. Computer Storage and Retrieval of Generic Chemical Structures Using Structure Attributes. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 177−187.
(15) Kudo, Y.; Sasaki, S. The Connectivity Stack, a New Format for Representation of Organic Chemical Structures. *J. Chem. Doc.* **1974**, *14*, 200−202.
(16) Owolabi, O. An Efficient Graph Approach to Matching Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 221−226.
(17) Ullmann, J. R. An Algorithm for Subgraph Isommorphism. *J. Assoc. Comput. Mach.* **1976**, *23*, 31−42.
(18) Willett, P.; Wilson, T.; Reddaway, S. F. Atom-by-Atom Searching Using Massive Parallelism. Implementation of the UllMann Subgraph Isomorphism Algorithm on the Distributed Array Processor. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 225−233.
(19) Brint, A. T.; Mitchell, E.; Willett, P. Substructure Searching in Files of Three-Dimensional Chemical Structures. In *Chemical Structures. The International Language of Chemistry*; War, W. A.; Ed.; Springer-Verlag: Berlin, 1988; pp 131−144.
(20) Downs, G. M.; Lynch, M. F.; Willett, P.; Manson, G. A.; Wilson, G. A. Transputer Implementation of Chemical Structure Search Algorithms. *Tetrahedron Comput. Methodol.* **1988**, *1*, 207−217.
(21) Mitchell, E. M.; Artymiuk, P. J.; Rice, D. W.; Willett, P. Use of Techniques Derived from Graph Theory to Compare Secondary Motifs in Proteins. *J. Mol. Biol.* **1990**, *212*, 155−166.
(22) Sheridan, R. P.; Nilakantan, R.; Rushinko, A.; Bauman, N.; Haraki, K. S.; Venkataraghavan, R. 3DSEARCH: A System for Three-Dimensional Substructure *Searching*. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 255−260.
(23) Ray, L. C.; Kirsh, R. A. Finding Chemical Records by Digital Computers. *Science* **1957**, *126*, 814−819.

CI9400932