match no. 6 pp. 201 - 211 1979

SUBSTRUCTURE SEARCHING AND STRUCTURE PROPERTY LOCATING BY MEANS OF SUBGRAPH GENERATION

J. Friedrich[+] and I. Ugi

Organisch-Chemisches Institut, Technische Universität München, Lichtenbergstr. 4, D-8046 Garching

1. Substructure Searching and Graph Theory

The main feature of this contribution will deal with substructure searching in large data bases of chemical compounds and with structure-property selection. The resulting program system was developed since 1975 in our labratory in Munich.

Assume, there is a data base of molecules. For simplicity reasons they shall be made small. E.g. hydrogencyanide, formaldehyde, and carbondioxide would be appropriate. In substructure search, when locking for the O=C-group in the data base, then O=C must first be matched against H-C≡N, and then against the next data base molecule and so on.

Some algorithms for matching substructures in chemical compounds are based upon the classifying of atoms and their valencies, followed by refinement of this classification [2a]. One has to live with exponential time-consuming searches [2c,4]. Other algorithmus in this field make use of fragment codes or screens: Matching is speeded up, but nevertheless, each molecule must be looked at tediously [5,6] (Fig. 1). Furthermore, they have another deficiency.

It is possible that molecules in the data base do not enclose the given substructure but succeed in screen-matching. The reason is that it is not possible to define any graph with a delimited number of (actual) subgraphs or graph invariants; the screen length is delimited and the number of corresponding structure patterns also. Therefore, in succeeding cases a time-consuming atom-by-atom comparison is needed. The same holds true for fragment codes even more restrictedly.

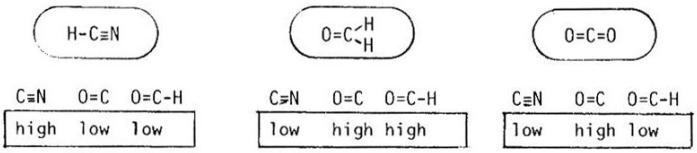

Fig. 1: Molecules with bitscreens

2. Substructure Generation and Retrieval

To overcome the described deficiencies, our proposal is to create all possible substructures beginning from the greatest molecules, level by level in a breadth-first manner, instead of generating the bitscreens.

If the greatest molecules have n bonds, the system first generates all substructures with n-1 bonds and puts them on level n-1. Then, starting from level n-1, including all data base molecules with n-1 bonds, the next level n-2 will be generated, and so on, until at last only single bond fragments are obtained in level 1 (Fig. 2).

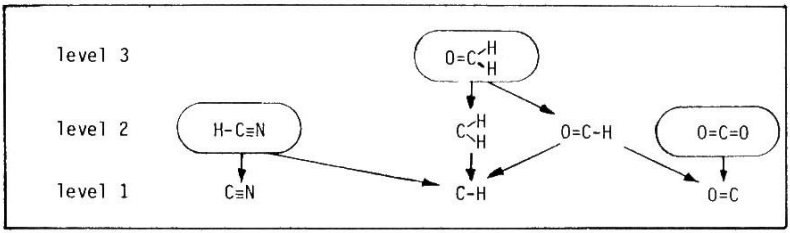

Fig. 2: Hierarchical generation of substructures

A query for O=C now would enter into the generated network at the point O=C and then pursue the arrows backwards until data base molecules are reached. Exactly all the molecules obtained contain the query substructure (Fig. 2).

Two substantial problems had to be solved. First, there are more and more generated duplicates with increasing molecule size and, secondly, the number of different substructures becomes larger than disk storage capacity.

The first problem is solved by a level-by-level generation process. Not any arrow reaches over more then two levels (Note, that the data base molecules are also pushed into their proper level). A graph analysis of each (sub-) structure prohibits duplicate generation. It consists of two parts. First, the constitutional symmetry scanning algorithmof W. Schubert[7] yields the constitutional different atoms and secondly, an algorithm which basis on a linear depth-first search presented by R.E. Tarjan[2b] looks for margin nodes of the molecule graph. We call a node a margin node, if by taking away one margin node, the remaining graph is still connected. The intersection of these constitutional different atoms and margin atoms gives a special set of atoms with the property that by taking away these atoms one after the other from the considered (sub-) structure (under replacing the preceding atoms) no duplicates are generated and that these new fragments are all possible fragments of size n-1. Furthermore, because the substructure relation is transitive, it is superfluous to generate substructures over more than one level. Premises is that each level contains all possible sub-structures, which can be generated from the next higher level. Thus, the redundant duplicate generation is restricted to a minimum.

The second problem, which is the disk storage overflow, is attacked by heuristics. Nobody is interested in the substructure C-H. This is due to two reasons. First, it is chemically meaningless. Secondly, C-H occurs in almost all organic compounds and gives, therefore, no special information. What one needs now is a heuristic function, which says to the generation system, whether a generated substructure is chemically meaningful. Particles like CH_3 should get lower priority. Furthermore, one needs another function, which computes the frequency of occurrence in the entire data base. The higher the frequency, the earlier one can drop or eliminate the fragment, when storage becomes full. These two functions are a tool in avoiding the all damaging storage overflow. Furthermore, they speed up the generation process, because dropped substructures must not be stored or searched. Thus, if need be, it would be possible to create only those substructures which are (each for their own) enclosed in a single data set molecule.

The graph analysis algorithm for selecting the margin nodes works upon the connectivity list only. Many atom vectors are belonging to the same connectivity list (or bond skeleton). That is why a data structure is chosen as described in Fig. 3 for purpose of efficiency.

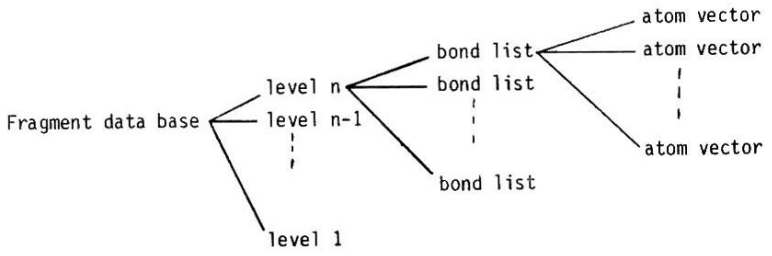

Fig. 3: Data structure of (sub-) structures

This data structure also allows accessing of structures, which are only defined by their connectifity list and some additional atoms building a mask. One has to enter the fragment network at the defined bond list and then, by cancelling the undefined atoms, go down level by level until all substructures with full defined atomvectors are held. By backtracking, all substructures, which have the query substructure mask in common, can be found. From these substructures the search procedure has to go upwards analogous to an exactly defined substructure query.

Algorithms for substructure generation and retrieval with the described features are presented below.

Procedure GEN: Generation of substructures to a given molecule data base.

(1) Determine the number of bonds of the greatest molecules (=:n).

(2) Collect the n-bonded structures (entire molecules and fragments) with same connectivity list to structure families on level n.

(3) Generate (from level n) the level n-1 by eliminating margin nodes and ring bonds of the structure families.

(4) Repeat from step 2 by diminishing n by 1;
if n=1, then stop.

Procedure SEARCH: Search for all data base molecules enclosing the query substructure.

(1) Enter the fragment network at the query connectivity list.

(2) Generate all substructures contained in the query substructure (level-by-level downwards like procedure GEN), until the greatest full defined substructures are obtained.

(3) Pursue the connectivity lists obtained in step 2 by recursiv backtracking, now including all atomvectors which contain the full defined substructures.

(4) Coming back to the connectivity list of step 1, one has to cancel chemical meaningless substructures determined by the last held level of step 3.

If there is no important atomvector left, give a message to the user.

(5) Cheque for each atom vector of step 4, whether it is in the dataset of anticipated fragments already or not.

If not anyone has been anticipated, then the query structure is not in the data base molecules.

(6) Go upwards, beginning from the atom vectors found in step 5, until the desired entire molecules are reached.

In step 3 of GEN and step 4 of SEARCH the above mentioned heuristic function is applied. If a chemical meaningless merit value is returned, then the query substructure is not in the fragment data base: The user has to ask for a more differentiated substructure. Thus, in combination with step 5 of SEARCH exactly is defined, whether a substructure is contained in certain molecules or not.
For a full defined substructure query, step 2 and 3 of SEARCH would be reduced to entering the atom vector of the query substructure. Step 2 and 3 are based upon the Ulam's Conjecture[1], which states that two graphs are isomorphic, if their subgraphs obtained by removing on point after the other (and replacing the preceding) are isomorphic. This theorem is recursively employed.

3. Structure Property Locating

Our object in research about structure/property relationships is to select of all possible substructures those with the highest probability of being responsible for a given property or activity.

Assume there is a data base of molecules being tagged with properties (maybe found in literature). And there is our fragment data base generated as described before. Now, investigating a certain property, from the data base molecules with the desired property the fragment network has to be worked out from the top to the bottom.

Hereby, to each substructure a number is given, which counts all molecules of the desired property lying above (and containing the substructure). Another number stating how many molecules lie above in the whole, was already given in the previous substructure generation. These two numbers, and furthermore the number of the data base molecules with the desired activity, and the number of all data base molecules are the four arguments of a statistical function. The function bases upon a distribution of a binary variable: An entire molecule shows an activity, or it shows none. A more differentiated distribution of a descrete or continuous variable would consider weighted input activities and output intensities.

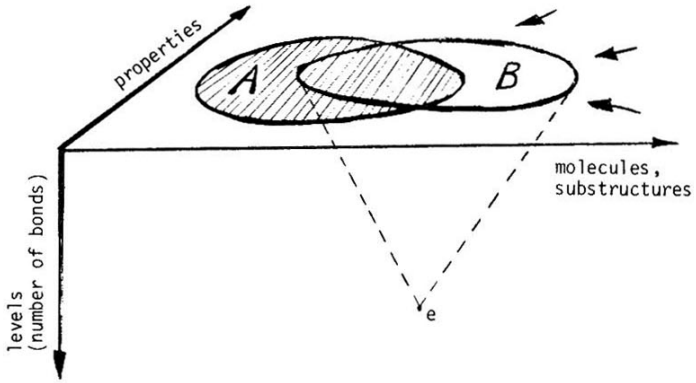

Fig. 4: Locating of an effector substructure e

Fig. 4 shows a more suggestive description. The relationship of molecules, substructures, and properties are described in 3-dimensional model. Splitting up the first two dimensions into a multi-dimensional room, that is, the molecule axis into the n-dimensional room of BE-Matrices[3] and the property axis into a 2-dimensional room (properties and property input weights), would involve the Mathematical Model of Chemical Constitution of J. Dugundji and I. Ugi [3].

The shaded area in Fig.4, which stands for all molecules with the desired property, is connected only for graphical reasons. Normally it would be distributed over all dimensions. The same holds for the area B, which stands for all entire molecules with the common substructure e. Suggestively, the more A and B overlap, the greater is the probability for the fragment e being responsible for the desired property p of A. The degree of overlapping is measured by the mentioned function. In the ideal case A and B would be identical and the returned value would be 1.00 .

An algorithm for giving a merit value to each substructure is:

Procedure VAL: Assignment of values to each substructure obtained from a set of tagged molecules.

(1) Determine the number of tagged molecules with the desired property p containing the substructure.

(2) Determine the number of entire molecules containing the substructure.

(3) Compute the confidence value from the values obtained in step 1 and 2, and from the number of all molecules having the desired p and of all not having it.

An algorithm SEL, which selects the best substructures, reades as follows:

Procedure SEL: Selection of the most confidential substructures.

(1) Determine the rank of all substructures according to the confidential values of algorithm VAL.

(2) Beginning with the highest value, select the most confidential substructures by eliminating from the rank all involving and contained substructures of lower value.

The algorithms VAL and SEL are predicatively described. Thus, in the implementation, step 2 of VAL is contained in GEN, because step 2 is common to all different properties, and SEL is fully integrated in VAL for reasons of efficiency. The counting works from the highest level downwards as in the generation process.

Algorithm SEL provides for selecting those substructures, which have no better substructures above or below (Fig. 5). The effector with a 0.97 confidence value being contained in a better effector is not selected.

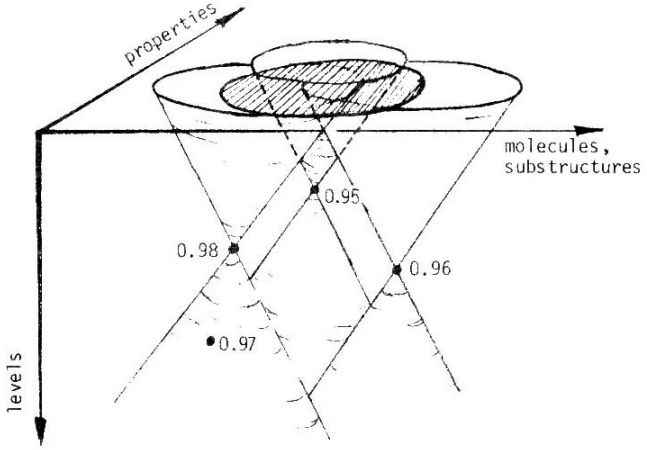

Fig.5: Selection of substructures with the highest confidence values

By the above procedures, substructures are obtained which are responsible for one or several properties. We call them effectors. They yield an effect, an activity, or a property in an independent manner (Fig.4). A special kind of interdependend effectors are named synergistic effectors, because they yield a property only through joint occurrence (shaded area in Fig.6).

Algorithms for detecting effectors and synergistic effectors are:

Procedure EFF: For a given data base of molecules and for a property p, select the best effectors.

(1) Determine all molecules with p.

(2) Weigh and select the substructures of the molecules obtained in step 1 employing procedures VAL and SEL.

Procedure SYN: For a given effector e, select the best synergistic substructures.

(1) Determine all molecules with p and containing e.

(2) Weigh and select the substructures of the molecules obtained in step 1 employing procedures VAL and SEL (see step 2 of EFF).

(3) Eliminate e and father fragments of e from substructures obtained in 2.

(4) Improve the confidence value of s according to the maximal confidence value of 2 and 3.

Step 4 of SYN improves the probability for a substructure being an effector on the condition that it is a synergistic effector. The analog consideration was made for step 4 of ANT described below.

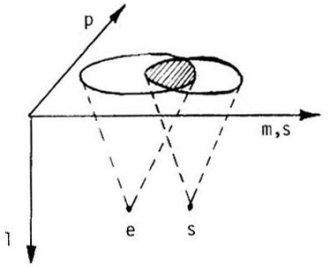

Fig.6: synergistic effector s

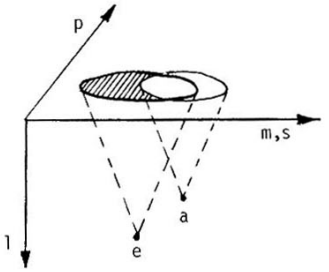

Fig.7: antagonistic effector a

In addition, there are antagonistic effectors, where a substructure (maybe not being affected with a certain property) suppresses a property of an effector (Fig.7). An algorithm for locating these effectors is similar to SYN. It deviates only in the determination of the starting molecules (step 1).

Procedure ANT: For a given effector e, select the best antagonistic substructures.

(1) Determine all molecules without p, but containing e.

(2) - (4) analogous to SYN.

4. Conclusion

An earlier version of the fragmentation and retrieval system[8], being implemented in FORTRAN on a PDP 11/45, has shown that a large file of about 1000 molecules occuring in the environment can be treated. In the interactive system, response times under 1 sec are achieved for OR, AND, or NOT substructure queries. Furthermore, it could be shown that the number of different connectivity lists increases only with the third power of number of atoms in the molecule. Thus, we conclude that out more sophisticated generation system,

which avoids duplicate generation and storage overflow, will be capable of handling molecules with up to 50 atoms. The same holds for the structure/property locating processes, which are of the same time-complexity as the substructure generation, but having much smaller step entities. The current version is being implemented in PL/1 on an IBM 360/91 and AMDAHL 470 V6.

Acknowledgements

We greatfully acknowledge the financial support of this work by the European Communities (Contract Nos. 115-74-10-ENV-D and 310-77-5-ENV-D). We also thank Dr. W. Schubert for supplying us with a version of his canonicalization and symmetry scanning program and Dr. J. Brandt for his inspiring contributions to this work.

References

1) S.M.Ulam: A Collection of Mathematical Problems, Wiley (Interscience), New York 1960

2) a) E.H.Sussenguth, J. Chem. Doc. 5,36 (1965)

   b) R.E.Tarjan, SIAM J. Computing 1,146 (1972)

   c) R.C.Read, D.G.Corneil, J. Graph Theory 1,339 (1977)

3) J.Dugundji, I.Ugi, Top. Curr. Chem. 39,19 (1973)

4) A.V.Aho, J.E.Hopcroft, J.D.Ullman: The Design and Analysis of Computer Algorithms. Addison-Wesley (1974)

5) G.W.Adamson, V.A.Clinch, S.E.Creasey, and M.F.Lynch, J. Chem. Doc. 14, 72 (1974)

6) A.Feldman, L.Hodes, J. Chem. Inf. Comp. Sc. 15,147 (1975)

7) W.Schubert, I.Ugi, J. Am. Chem. Soc. 100,37 (1978)

8) a) J.Brandt, J.Friedrich, J.Gasteiger, C.Jochum, W.Schubert, and I.Ugi in E.V.Ludena, N.H.Sabelli, A.C.Wahl: Computers in Chemical Education and Research. Plenum, New York 1977

   b) J.Brandt, J.Friedrich, J.Gasteiger, C.Jochum, W.Schubert, and I.Ugi: Computer-Assisted Organic Synthesis. ACS Symposium Ser. 61

   c) J.Brandt,J.Friedrich, J.Gasteiger, C.Jochum, P.Lemmen, W.Schubert, and I.Ugi, Pure Appl. Chem. 50, 1303 (1978)

   d) I.Ugi, J.Bauer, J.Brandt, J.Friedrich, J.Gasteiger, C.Jochum, and W.Schubert, Angew. Chem. 91,99 (1979), Angew. Chem. Int. Ed. Engl. 18, 111 (1979)