

struction of (sub)structures on a graphics screen are lost.

REFERENCES AND NOTES

- (1) Allen, F. H.; Kennard, O. Crystallographic Databases. In *Cambridge Structural Database*; Data Commission of the International Union of Crystallography: Utrecht, 1987; Chapter 2.1, pp 32-76. See also references cited therein. Bellard, S. *Ibid.*
- (2) Wood, G. H.; Rodgers, J. R.; Gough, S. R. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 118-123.
- (3) CAOS/CAMM Center. Dutch National Facility for Computer Aided Chemistry.
- (4) Machin, P. A. *Crystallographic Computing 3*; Sheldrick, C. G. M., Kruger, C., Goddard, R., Eds.; Oxford University Press: Oxford, U.K., 1985; p 106.
- (5) *Parallel Computers 2*; Hockney, Jesshope, Eds.; 1988. ISBN 0-85274-811-6.
- (6) Cambridge Structural Database System, Users Manual; Version 3.4, Part III, 1989.
- (7) VAX/VMS Software Product Description, Release 5.0; Utility SYSSCRMPSC.
- (8) Requests for the CONVEX optimized version of CSD/QUEST should be directed to the Cambridge Crystallographic Datacentre, University Chemical Laboratory, Lensfield Road, Cambridge CB2 1EW, U.K.
- (9) CONVEX UNIX Programmers Manual, Part II, Section 2.

Prolog-Based Functional Group Perception and Calculation of 1-Octanol/Water Partition Coefficients Using Rekker's Fragment Method

KEI TAKEUCHI, CHIAKI KURODA, and MASARU ISHIDA*

Research Laboratory of Resources Utilization, Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama, Japan 227

Received June 15, 1989

A new algorithm is developed to perceive functional groups in a chemical structure from the method that generates simple pairs of atoms comprising functional groups. Description of the program to perceive functional groups is given. The program is applied to calculate log *P*, logarithmic 1-octanol/water partition coefficients, on the basis of the method developed by Rekker in which log *P* is calculated from fragment constants for various functional groups.

INTRODUCTION

Perception of functional groups is essential for the computerized chemical structure-handling techniques¹ in structure-activity correlation, physicochemical property estimation, and computer-assisted organic synthesis design.

There are four ways to automatically perceive functional groups. The first one, the most time-consuming one, is to perceive a list of functional groups of interest by atom-by-atom search. The second one, which is more sophisticated in searching a list of functional groups, is the "table-driven" method,² which involves logical manipulation of the list. An algorithmic definition of functional groups is the third approach. Chu et al.^{3,4} used augmented atom fragment⁵ to satisfactorily represent functional groups in substructure analysis. Although this method is much faster than the atom-by-atom search, problems of redundancy caused by overlap of structures arise.¹ The fourth is an algorithmic perception method using an encoding scheme that uniquely represents each functional group. This method has been developed in the organic synthesis design^{6,7} and the mechanistic evaluation of organic reactions.⁸ A canonical connection table is employed in the former to compile the machine formulas for functional groups from the viewpoint of a central atom of the group, whereas, in the latter, information on where functionality begins and ends or how paths grow from the origin is important for compiling codes for each functional group.

The present paper is concerned with the program to perceive functional groups with logic programming language Prolog.⁹ Prolog has been developed as an artificial intelligence language and widely applied to the field of chemical structure handling, such as molecular property estimation,¹⁰ calculation of topological index,^{11,12} organic synthesis design,^{13,14} and structure elucidation.¹⁵⁻¹⁸ These applications demonstrate that Prolog is useful not only for a knowledge-based system but also for manipulation of chemical structures. Prolog programming has several advantages. Substructure search becomes quite simple

by using the inference engine installed in Prolog.^{12,17} The algorithm is so clear that the program becomes very compact and is easily developed.

When the routine to perceive functional groups becomes available, one can apply it to calculation of log *P*, the logarithm of the 1-octanol/water partition coefficient. log *P* is a well-known hydrophobic parameter that is closely correlated with various biochemical activities. This value is hence widely used for a descriptor in structure-activity studies, and its estimation is proposed by Rekker.^{19,20} Rekker's method is based on the assumption that this value possesses additive-constitutive character. Rekker assigned "hydrophobic fragment constants" to each substructure, and the log *P* value can be calculated by appropriate fragment constants and correction factors. The fragments consist of functional groups and carbon atoms with attached hydrogen. In that calculation, Rekker used several correction factors that are also related to the interactions between the functional groups. Hence, a powerful routine to perceive functional groups must be developed to calculate log *P* by Rekker's method. Recently, Darvas et al.¹⁰ developed a calculation system for Rekker's method programming in Prolog, called PRO-LOGP. However, the algorithm to perceive functional groups in it is not reported.

PERCEPTION OF FUNCTIONAL GROUPS

Functional groups as well as ring fragments are chemical and problem-dependent substructures. Their constituent atoms are generally heteroatoms, such as S, N, O, and P, and halogens or unsaturated carbon. In some cases, hydrogen atoms are attached to them. The size and shape of their skeletal structures are not fixed. This causes difficulty in perceiving functional groups by a computer.

An efficient algorithm without encoding scheme, which we call "simple pair generation method", is outlined here. A simple pair X-Y comprises a bond by which two atoms X and Y are linked. Suppose that functional groups are classified into elementary and complex ones. Functional groups such

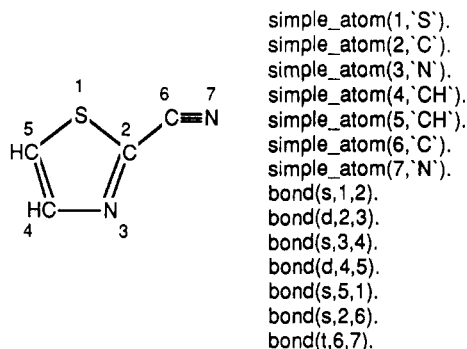
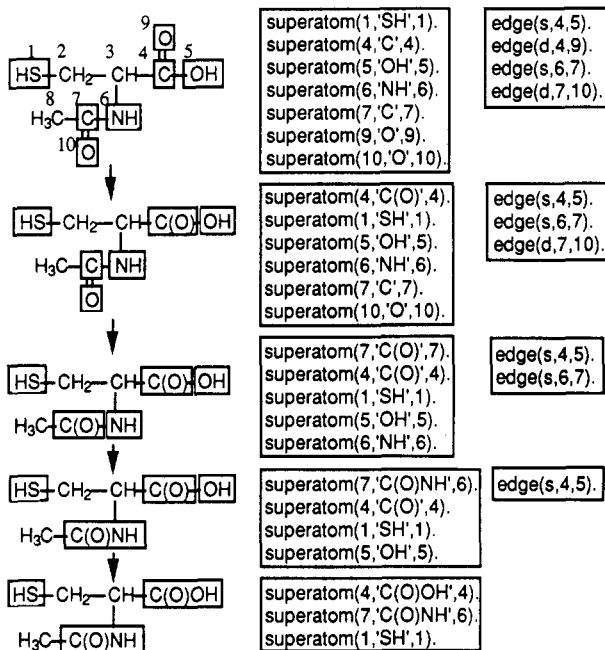


Figure 1. Prolog representation of molecular structure.

Figure 2. Perception for *N*-acetylcysteine by simple pair generation method.

as NH, OH, and Cl are considered to be elementary, because they are composed of only one heteroatom or a heteroatom and its neighboring hydrogen. Complex functional groups are made up of the simple pairs of them. For example, carbonyl $>C(=O)$ consists of the simple pair of the elementary functional groups $>C=$ and O. Bigger complex functional groups can be made by linking an elementary group or a complex group to a complex one. For example, the ester $C(=O)O$ is represented by the simple pair of the elementary O and the complex $C(=O)$, and the anhydride $C(=O)OC(=O)$ is of the complex $C(=O)O$ and $C(=O)$. Both the elementary functional groups and the complex ones are registered as "superatoms" in the program. Then all functional groups can be generated by the simple pair of the superatoms.

Representation of Molecular Structure. Prolog representation of a molecular structure is shown in Figure 1. All data are represented in terms of Prolog facts that make data processing very simple. Prolog facts represent some facts about objects and their relationships. An arbitrary but unique number is assigned to each non-hydrogen atom by the predicate **simple_atom**. Each connection between non-hydrogen atoms is cited once by the predicate **bond**. Its first argument shows the types of bond defined as s (single), d (double), t (triple), and ar (aromatic).

Description of the Program. Figure 2 shows an example of the perception for *N*-acetylcysteine with the change of the fact sets of superatoms and bonds. A flowchart is given in Figure 3, and the program is shown in Figure 4.

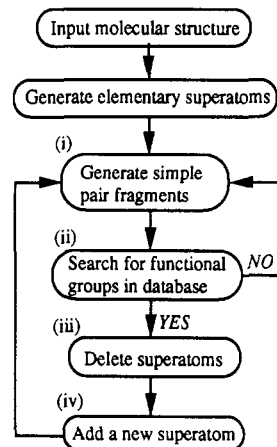


Figure 3. Flowchart for perceiving functional groups.

```

perception_of_groups:-
link(Bond_type,B,C),
superatom(A,X,B),
superatom(C,Y,D),
functional(Type,[X,Bond_type,Y],XY),
retract(superatom(A,X,B)),
retract_edge(Bond_type,B,C),
retract(superatom(C,Y,D)),
new_functional(Type,XY,A,D),
fail.

new_functional(center,XY,A,D):-
asserta(superatom(A,XY,D)),
new_functional(terminal,XY,A,D):-
asserta(superatom(A,XY,A)).

```

Figure 4. Program for perceiving functional groups.

1. Input Molecular Structure. The topological representation of a molecular structure can be input by the above scheme.

2. Generate Elementary Superatoms. The constituent atoms of functional groups are regarded as elementary superatoms. The connection table for the elementary superatoms is generated by searching for the input molecular structure. Heteroatoms and carbons with double or triple bonds are searched for by examining the second argument of the fact **simple_atom(A,X)** and are stored as the fact **superatom(A,X,A)**. The predicate **superatom** takes three arguments. For such elementary superatoms, the third argument is equal to the first one. The role of the third argument will be explained later. In addition, a search for the fact **bond(Bond_type A,B)** connecting **superatom(A,X,A)** to **superatom(B,Y,B)** registers the bonds linking elementary superatoms as the facts **edge(Bond_type,A,B)**. This results in reducing the number of searches. In this manner, two sets of clauses with the predicates **superatom** and **edge**, which are listed in the top squares in Figure 2, are generated as elementary superatoms and the bonds connecting them, respectively, for the example compound.

3. Generate Simple-Pair Fragments. In the present method, the simple-pair fragment is regarded as the basic structure constituting a complex functional group. The simple pair of elementary superatoms is generated on the basis of the connection table for the elementary superatoms. The four clauses with the predicate **edge**, listed in the top-right square in Figure 2, show that there are four simple pairs: [**'C',s,'OH'**], [**'C',s,'NH'**], and two [**'C',d,'O'**]. The corresponding new superatoms are generated by the three subgoal clauses shown in Figure 4i. The first clause, **link(Bond_type,B,C)**, means **edge(Bond_type,B,C)** or **edge(Bond_type,C,B)**. These three subgoal clauses search for the pair of superatoms having the right-hand position of the superatom **B** connected to the left-hand position of another superatom **C**, and then the simple

Table I. Database of Functional Groups for Rekker's Method

```
functional(terminal,['C','d','O'],'C(O)').
functional(terminal,['CH','d','O'],'COH').
functional(terminal,['C','t','N'],'CN').
functional(center,['S','s','S'],'SS').
functional(center,['S','d','O'],'SO').
functional(terminal,['N','d','O'],'NO').
functional(terminal,['C','d','S'],'CS').

functional(terminal,['C(O)','s','OH'],'C(O)OH').
functional(center,['C(O)','s','O'],'C(O)O').
functional(terminal,['C(O)','s','NH2'],'C(O)NH2').
functional(center,['C(O)','s','NH'],'C(O)NH').
functional(center,['C(O)','s','N'],'C(O)N').
functional(center,['SO','d','O'],'SO2').
functional(terminal,['NO','s','O'],'NO2').
functional(terminal,['N','d','CS'],'NCS').
functional(terminal,['S','s','CN'],'SCN').

functional(center,['NH','s','C(O)O'],'NHC(O)O').
functional(terminal,['C(O)O','s','NH2'],'C(O)ONH2').
functional(center,['C(O)NH','s','NH'],'C(O)NHNH').
functional(terminal,['SO2','s','NH2'],'SO2NH2').
functional(center,['NH','s','SO2'],'NHSO2').
functional(center,['N','s','SO2'],'NSO2').
```

pair fragment [X,Bond_type,Y] is found as a candidate of the functional groups.

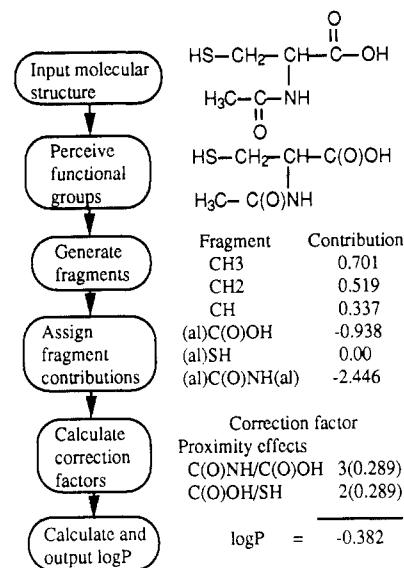
4. Search for Functional Groups in Database. The simple pair is searched for in the database of functional groups registered by the user in advance. If it is not detected, go to step 3. All functional groups have been stored in the database as shown in Table I. They are described by Prolog as a set of fact. Their general form is

functional(Type,[X,Bond_type,Y],XY).

The simple pair of superatoms is seen in the second argument of the predicate functional. The program in Figure 4ii retrieves the simple pair [X,Bond_type,Y] in the database of functional groups. When it is not found, an interpreter automatically backtracks and tries to generate the next simple pair. The backtracking is done by Prolog and is hidden from the user. In Figure 2, for example, no functional registered in Table I has the simple pair ['C','s','OH'] in the second argument. Then the interpreter backtracks and locates the next pair ['C','d','O'] comprising superatom(4,'C',4), superatom(9,'O',9), and edge(d,4,9). It matches the fact functional(terminal,['C','d','O'],'C(O)') in the database, and the variables Type and XY are instantiated to terminal and C(O), respectively, giving rise to the facts in the second row in Figure 2.

By the variable Type as the first argument in functional, functional groups are classified into terminal-XY and center-XY-. In the former, only one atom is connected to the surrounding atoms, for example, in the case of -CN and >C(O). In the latter, two atoms are connected to the surrounding ones, for example, in the case of >C(O)O- and >NC(O)N<.

5. Delete and Add Superatoms. The simple pair is inserted as a new superatom into the connection table. On the other hand, the bond and two superatoms constituting the new superatom are deleted from the connection table. After a simple pair is retrieved on the basis of the database of the functional groups, a new superatom with XY is generated and the two constituent superatoms X and Y are deleted from the list of superatoms. In Figure 4, the carbonyl >C(O) is first perceived, and superatom(4,'C',4), superatom(9,'O',9), and edge(d,4,9) are deleted. This is done by a built-in predicate retract as shown in three subgoal clauses of Figure 4iii. The predicate retract_edge deletes the fact edge(Bond_type,B,C) or edge(Bond_type,C,B). The predicate new functional adds a new superatom(4,'(O)',4) to the list of superatoms. This is done by (iv) in Figure 4. When the variable Type is instan-

**Figure 5.** General flow of program and example of calculation of log P by Rekker's method.

tiated to center, superatom(A,XY,D) is added, and when it is instantiated to terminal, superatom(A,XY,A) is added. The built-in predicate asserta enables us to add it at the beginning of the list.

6. Find the Next Functional Group. If all simple pairs of elementary superatoms are found, go to step 2, and the simple pairs of elementary and complex superatoms or complex ones generated in step 5 are searched for to perceive bigger functional groups. The last subgoal clause in Figure 4 is a built-in predicate fail. It forces the inference engine to backtrack until the next simple pair, superatom(7,'C(O)',7), is found. Having found all solutions, i.e., simple pairs, the routine perception_of_groups is finished, and it will be executed again to perceive larger functional groups. This is done by including superatoms generated in the previous level. In Figure 2, the carboxy -C(O)OH and the amido -C(O)NH- are perceived in this higher level. Two carbonyl groups perceived in the previous level are hence deleted from the set of superatoms. At last, three superatoms remain. They are C(O)NH, C(O)OH, and SH. They are the functional groups to be obtained for the example compound. The routine perception_of_groups will repeatedly be executed depending on the size of functional groups. For instance, it is executed three times to perceive urethane -NHC(O)O.

PROGRAM TO CALCULATE LOG P USING REKKER'S METHOD

The above procedure is applied to calculate log P by Rekker's method. The program is made of two parts: generation of substructure fragments and calculation of correction factors. A general flow of the program and an example of the calculation for N-acetylcysteine are depicted in Figure 5.

1. Input Molecular Structure. A molecular structure is input in a manner similar to that in the perception of functional groups shown in Figure 1. An aromatic atom is registered as Car for carbon and Nar for nitrogen.

2. Perceive Functional Groups. The database for the functional groups, which must be perceived by Rekker's method, is given in Table I. The same routine to perceive functional groups developed in the previous section can be applied.

3. Generate Fragments. In the example compound shown in Figure 5, the chemical structure is decomposed into six fragments, CH, CH₂, CH₃, SH, C(O)OH, and C(O)NH, with particular contribution values. Fragments are classified into

Table II. Examples of (a) Structures of the Fragments and Some Values for (b) Fragment Contributions and (c) Correction Factors

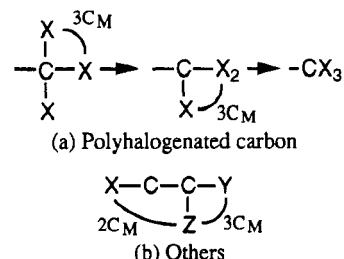
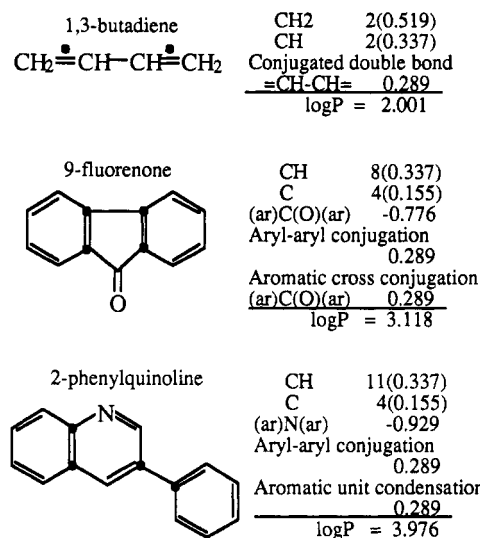
(a)	fragment('CH3'). fragment('CH2'). fragment('CH'). fragment('SH', ['Al']). fragment('C(O)OH', ['Al']). fragment(['Al'], 'C(O)NH', ['Al']).
(b)	contribution('CH3', 0.701'). contribution('CH2', 0.519'). contribution('CH', 0.337'). contribution('SH', ['Al'], -0.938'). contribution('C(O)OH', ['Al'], 0.00'). contribution(['Al'], 'C(O)NH', ['Al'], -2.446').
(c)	correction('1C_separation', 0.867'). correction('2C_separation', 0.578').

carbon atoms with attached hydrogen and functional groups. In Figure 5, there are three carbon-atom fragments, CH, CH₂, and CH₃, and three functional groups, C(O)OH, SH, and C(O)NH, linked to aliphatic carbon atoms. Functional group fragments have different contribution values according to whether their neighbor atoms are aliphatic (Al) or aromatic (Ar).

The fragments are generated as the facts with the predicate **fragment** shown in Table IIa. For the carbon-atom fragments, neighbor atoms are not taken into account, and the predicate **fragment** has a single argument. It is generated by examining the second argument of the fact **simple_atom(A,X)**. On the other hand, functional group fragments are registered by the following procedure. First of all, types of neighbor atoms, Al or Ar, are stored as facts. If aromatic atoms, for example, Car, are searched from the second argument of the fact **simple_atom(A,X)**, the fact **neighbor_atom(A,'Ar')** is registered. Otherwise, the fact **neighbor_atom(A,'Al')** is registered. Then the fragments including the types of the neighbor atoms can be generated by searching from the sets of facts with the predicates **neighbor_atom** and **superatom** in a manner similar to that of generating simple-pair fragments. First, consider a functional group composed of a single heteroatom, SH, or the terminal type, -C(O)OH, for example, and set aromatic or aliphatic characters in the second argument. Then its fragment is generated by finding the fact **superatom(A,X,A)**. For a functional group bonded to two or three neighbor atoms, such as a tertiary amine -N<, in which nitrogen atom is connected to, say, an aromatic atom and two aliphatic atoms, its fragment is registered as the fact **fragment('N', ['Al', 'Al', 'Ar'])**. When the functional group is composed of the central type, for example, C(O)NH, aromatic or aliphatic characters are set in the first and third arguments, as shown in the last example of Table IIa.

4. Assign Fragment Contributions. Some values of the fragment contributions are shown in Table IIb on the basis of ref 20. When the fragments are generated in the previous step, the predicate **contribution** gives the values in the last argument.

5. Calculate Correction Factors. Rekker introduced²⁰ five correction terms: proximity effects of polar groups, conjugated double bond, aryl-aryl conjugation, aromatic cross conjugation, and aromatic unit condensation. These corrections can be expressed as a multiple of the constant, which is called the "magic constant c_M ", having a value of 0.289. For the correction terms except proximity effects, the correction value $1c_M$ ($1 \times c_M$) is assigned to each pair of aromatic atoms or double bonds. Three example molecules are shown in Figure 7. The dots in that figure show the part which needs cor-

**Figure 6.** Rule for calculating proximity effects: (a) polyhalogenated carbon; (b) others.**Figure 7.** Three example molecules for calculation of log P.

rection factors for conjugated double bond, aryl-aryl conjugation, aromatic cross conjugation, and aromatic unit condensation. Hence, all correction factors introduced by Rekker are taken into account in this program.

Proximity Effects. When two polar fragments or functional groups in a molecule are separated by an aliphatic carbon atom, X-C-X, or two aliphatic carbon atoms, X-C-C-X, the correction factors for them are $3c_M$ and $2c_M$, respectively. These values are registered by the predicate **correction** shown in Table IIc. For the model compound shown in Figure 5, two proximity effects, C(O)NH-C-C(O)OH and C(O)OH-C-C-SH, are taken into account.

Rekker proposed¹⁹ that for a compound with a polyhalogenated carbon shown in Figure 6a, $3c_M$ is given to the path X-C-X according to the above rule. Then it is changed to CX₂, and again $3c_M$ is given to the path X₂-C-X. Hence $6c_M$ is the final value. For a compound with multiple fragment pairs shown in Figure 6b, $2c_M$ is given to the path X-C-C-Z and $3c_M$ to Y-C-Z. Although X-C-C-Y is also a fragment pair, it is not taken into account because the intermediate carbon atoms are already used. To avoid such duplicate enumeration, the node numbers of carbon atoms separating two polar fragments should be stored. The newly enumerated path is compared with those that have been stored in the program.

Conjugated Double Bond. The correction term for a pair of linear or cross-conjugated double bonds is generated by searching for paths such as =C-C= and =C-X, where X represents an aromatic carbon atom Car or functional groups including the carbonyl >C(O).

Aryl-Aryl Conjugation. When an aromatic ring is connected to another one by a nonaromatic single bond, the simple pair Car-Car is searched by using the facts **neighbor_atom(A,'Ar')**. If two aromatic rings are separated by a heteroatom, Car-X-Car, where X represents O, NH, S, or S-S, the fact **frag-**

ment('X', ['Ar', 'Ar']) is searched from the **fragment** set.

Aromatic Cross Conjugation. When two aromatic rings are separated by the functional group including the carbonyl >C(O), the correction term for aromatic cross conjugation is used. This correction term is generated by searching for the fact **fragment**('X', ['Ar', 'Ar']), where X stands for C(O), C(O)O, C(O)NH, and C(O)ONH.

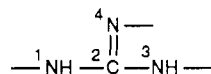
Aromatic Unit Condensation. To perceive the aromatic unit condensation, the path >Car-Car< in which all bonds have attribute ar (aromatic) is searched for.

6. Calculate and Output Estimated log P. The estimated log P value is obtained by summing all fragment contributions and all correction values.

DISCUSSION

It is found from the above presentation that the routine to perceive functional groups can directly be applied to the program to calculate log P. This extension can be done only by changing the database of functional groups. This is one of the advantages of programming in Prolog. It is easy to apply the present method to estimate other physical and physicochemical properties²¹ such as boiling point, heat capacity, molar refraction, molar volume, parachor, and so on. In addition, the interaction of functional groups, such as pairs of functional groups, can easily be obtained by using the predicate **superatom** as described in the calculation of proximity effects. Morita and Oka²² examined the correlation between functional group pairs and pharmacological action for 9000 synthetic drugs including nitrogen. It is clear that the present method is useful for systematically generating functional group pairs.

So far we have treated terminal functional groups and linear ones with two atoms linked to neighbor atoms. When we examine guanidines



there are three nitrogen atoms connected to the central carbon, 1-NH, 3-NH, and 4-N, and they have neighbor atoms. Strictly speaking, this structure cannot be represented by the fact **superatom**(A,XY,B). In this case, however, the isocyano group >C=N- may be treated as terminal type and the fact **superatom**(2,'CdN',2) can be generated. Then the guanidines may be perceived as **superatom**(1,'NHCdNNH',3).

The above two programs were developed by using personal computer PC-9801VX made by NEC. Its CPU is Intel 80286 driven by 10 MHz. They were written in Kyoto Artificial Brain Associates' Prolog-KABA with an extended tool WING.

CONCLUSION

A new method to perceive functional groups in a molecular structure by Prolog is proposed. In this method, the functional group structure is generated as the simple pair of superatoms. The program is presented, and it is shown that the complex functional groups are efficiently perceived by generating a simple pair repeatedly. This perception routine is applied to

calculate log P by Rekker's fragment method. Both methods to perceive functional groups and to calculate log P will be useful for structure-activity studies with large sets of compounds.

REFERENCES AND NOTES

- Bawden, D. Computerized Chemical Structure-Handling Techniques in Structure-Activity Studies and Molecular Property Prediction. *J. Chem. Inf. Comput. Sci.* **1983**, 23, 14-22.
- Corey, E. J.; Wipke, W. T.; Ceame III, R. D.; Howe, W. J. Techniques for Perception by a Computer of Synthetically Significant Structure Features in Complex Molecules. *J. Am. Chem. Soc.* **1972**, 94, 431-439.
- Chu, K. C. Application of Artificial Intelligence to Chemistry. Use of Pattern Recognition and Cluster Analysis to Determine Pharmacological Activity of Some Organic Compounds. *Anal. Chem.* **1974**, 46, 1181-1187.
- Chu, K. C.; Feldmann, R. J.; Shapiro, M. B.; Hazard, G. F., Jr.; Geran, R. I. Pattern Recognition and Structure-Activity Relationship Studies. Computer-Assisted Prediction of Antitumor Activity in Structurally Diverse Drugs in an Experimental Mouse Brain Tumor System. *J. Med. Chem.* **1975**, 18, 539-545.
- Crowe, J. E.; Lynch, M. F.; Town, W. G. Analysis of Structural Characteristics of Chemical Compounds in a Large Computer-Based File. Part 1. Noncyclic Fragments. *J. Chem. Soc. C* **1970**, 990-996.
- Esack, A.; Bersohn, M. A Program for Rapid and Automatic Functional Group Recognition. *J. Chem. Soc., Perkin Trans. 1* **1974**, 2463-2470.
- Bersohn, M.; Esack, A. Functional Group Discovery Using the Concept of Central Atoms. *Chem. Scr.* **1976**, 9, 211-215.
- Salatin, T. D.; Jorgensen, W. L. Computer-Assisted Mechanistic Evaluation of Organic Reactions. 1. Overview. *J. Org. Chem.* **1980**, 45, 2043-2051.
- Clocksins, W. F.; Mellish, C. S. *Programming in Prolog*, 3rd ed.; Springer Verlag: New York, 1987.
- Darvas, F.; Erdos, I.; Teglas, G. Influence of the Fragmental Systems on Calculated log P Values: a Theoretical Investigation Using a Logic-Based Expert System. *Pharmacochem. Libr. QSAR Drug Des. Toxicol.* **1987**, 10, 70-73.
- Contreras, M. L.; Deliz, M.; Galaz, A.; Rozas, R.; Sepulveda, N. Microcomputer-Based System for Chemical Information and Molecular Structure Search. *J. Chem. Inf. Comput. Sci.* **1986**, 26, 105-108.
- Takeuchi, K.; Kuroda, C.; Ishida, M. Prolog Program for Subgraph Enumeration and Calculation of Molecular Connectivity Indexes. *J. Comput. Chem.* **1989**, 10, 380-385.
- Moseley, C. W.; LaRoe, W. D.; Hamphill, C. T. Expert-System Rules for Diels-Alder Reactions. *ACS Symp. Ser.* **1986**, 306, 231-243.
- Wang, T.; Burnstein, I.; Corbett, M.; Ehrlich, S.; Evens, M.; Gough, A.; Johnson, P. Using a Theorem Prover in the Design of Organic Syntheses. *ACS Symp. Ser.* **1986**, 306, 244-257.
- Luinge, H. J.; van't Klooster, H. A. Artificial Intelligence Used for the Interpretation of Combined Spectral Data. *Trends Anal. Chem.* **1985**, 4, 242-243.
- Kleywegt, G. J.; van't Klooster, H. A. Chemical Applications of PROLOG Interpretation of Mass Spectral Peaks. *Trends Anal. Chem.* **1987**, 6, 55-57.
- Luinge, H. J.; Kleywegt, G. J.; van't Klooster, H. A.; van der Maas, J. H. Artificial Intelligence Used for the Interpretation of Combined Spectral Data 3. Automated Generation of Interpretation Rules for Infrared Spectral Data. *J. Chem. Inf. Comput. Sci.* **1987**, 27, 95-99.
- Kleywegt, G. J.; Luinge, H. J.; van't Klooster, H. A. Artificial Intelligence Used for the Interpretation of Combined Spectral Data. Part II. PEGASUS: a PROLOG Program for the Generation of Acyclic Molecular Structures. *Chemom. Intell. Lab. Syst.* **1987**, 2, 291-302.
- Rekker, R. F. *The Hydrophobic Fragmental Constants*; Elsevier: New York, 1977.
- Rekker, R. F.; de Kort, H. M. The Hydrophobic Fragmental Constant; An Extension to a 1000 Data Point Set. *Eur. J. Med. Chem.* **1979**, 14, 479-488.
- Lyman, W. J.; Reehl, W. F.; Rosenblatt, D. H. *Handbook of Chemical Property Estimation Methods*; McGraw-Hill: New York, 1982.
- Morita, K.; Oka, Y. Synthetic Drugs Containing Nitrogen with Special Reference to Functional Group Pair. *Kagaku Zokan (Kyoto)* **1979**, 79, 141-175.