ISE-TR-81-20

A GRAPH DATABASE FOR STORAGE OF CHEMICAL STRUCTURES ORGANIZED BY BCT

by

Yuzuru Fujiwara *

and

Takashi Nakayama

July, 22, 1981

INSTITUTE
OF
INFORMATION SCIENCES AND ELECTRONICS

UNIVERSITY OF TSUKUBA

SUMMARY

A method is presented for organizing a graph database, which is based on the representation of chemical structures in terms of BCT (block-cutpoint tree). It is useful for quick substructure search, for saving memory as well as for convinience in structure generation. The database consists of four files: a master file, a bit sequence file of fixed length records which gives block components of compounds, a BCT file which gives the BCT-structures of compounds, and a block file which specifies the blocks. These files are organized recursively and hierarchically, which makes it easy to process structural information of compounds.

INTRODUCTION

Information about chemical structures is essential in many application systems, such as substructure search, structure elucidation from spectral data, drug design, and synthesis route analysis.

A typical database of chemical structures has the data in the form of connection tables or adjacency matrices. Databases of this type are independent of applications, and therefore universal, but are not necessarily efficient for structural data processing such as access to data through a substructure as a key or searching a series of compounds with similar structures.

One of the major problems in computer handling of chemical structures is the explosive increase of time and space complexities due to the increase in the number of atomic combinations. Application systems should be designed to reduce these complexities. The solutions are in two directions: to improve processing algorithms and to improve the method of representing chemical structures. Many efforts have been made in both directions, and a typical method in the second direction is the screen system for substructure search

[1-3].

Our chemical structure database organized by BCT (block-cutpoint tree)[4] is also one of the attempts in the latter direction. This database is universal and independent of particular applications. It is constructed so that time and space complexities may be substantially reduced, for effective use of information about chemical structures.

In a chemical structure database (CSDB), the representation of structure should not use codes that simply identify compounds and require an algorithm to assign unique and unambiguous codes. These codes are of little use from the database point of view unless they are recursive and suitable for screening. The present database has taken this into account and does not deal with atoms directly, but uses a block consisting of several atoms as a meaningful operational unit. A block is an intermediate concept between atoms and molecules, which are BCT's by the present representation.

A superblock composed of simple blocks is introduced as a high-order operational unit which is easily defined and modified for flexible and effective processing of structures.

REPRESENTATION OF CHEMICAL STRUCTURES

In recent work we presented a hierarchical representation scheme of chemical structures by means of the block-cutpoint tree[5]. Using this representation scheme, an intermediate representation is generated in the form of a tree whose nodes consist of blocks and cutpoints, where blocks (biconnected components of a graph) correspond to ring systems or linear systems of two vertices. The strict representation is described by connectivity information between blocks and block structures in the block dictionary. A brief description of BCT representation of chemical structures is given below:

1. Block-Cutpoint Tree. Chemical structures are regarded as graphs in this representation method, so atoms and bonds correspond to vertices and edges respectively. A vertex of a connected graph G is called a "cutpoint" if its removal brings about the disconnectivity of G, and a "block" is a maximal subgraph of G which contains no cutpoints. Let $\{B_i\}$ be the set of blocks of G, and $\{c_j\}$ be the set of cutpoints of G; then a block-cutpoint graph of G is defined as a graph whose vertex set is $\{B_i\} \cup \{c_j\}$ and whose connectivity

is defined between $\{B_i\}$ and $\{c_j\}$, where c_j is adjacent to B_i if and only if c_j is one of the vertices of block B_i. The block-cutpoint graph is always a tree, so it is called a block-cutpoint tree (hereafter a BCT). Cutpoints and blocks of a graph and its BCT are shown in Figure 1. Black nodes in Figure 1 are cutpoints.

2. BCT Representation of Chemical structures. The algorithm which finds the blocks and cutpoints of a graph was described in our previous paper[5], and two other algorithms have also been presented[6,7]. A BCT is constructed easily from those blocks and cutpoints. Let BCTGEN be the name of the program which finds the blocks and cutpoints and constructs a BCT. Input data for BCTGEN are adjacency matrices or connection tables of graphs. Given the data for a chemical structure, BCTGEN generates a BCT as its intermediate structure, and describes it in canonical form where the center of the tree is the root of the tree. This was implemented according to Edmonds' algorithm[8]. Block nodes (a member of the vertex set of a BCT can be called a "node", so that there are block nodes and cutpoint nodes in a BCT) are given identification numbers by BCTGEN and their internal structures are known in the block dictionary (described later); in other words,

the key item of block dictionary occurs in attributes of a BCT record.

Further information about the connectivity of those constituent blocks is needed in order to describe a chemical structure completely. This is expressed by a connectivity matrix which represents the connectivity among blocks: A row and a column of connectivity matrix represent the constituent blocks $B_1, \cdots, B_m$. The element of connectivity matrix c_{ij} represents the identification number of the cutpoint in B_i which connects B_i with B_j if B_i is connected with B_j, otherwise c_{ij} is zero. The element c_{ij} also represents the connectivity of block B_j in relation to block B_i. Thus the connectivity matrix is not symmetrical (usually $c_{ij} \neq c_{ji}$). Morgan's algorithm[9] is used as the numbering scheme for the vertices of a block. The way of selecting cutpoints in a block is not always unique, so the following method for specifying cutpoints is prepared. Suppose that $\{p_1, \cdots, p_k\}$ is the set of cutpoints in the block, where p_i is the identification number. The set $\{p_1, \cdots, p_k\}$ is selected so that it gives the minimal value when the sequence of any permutation of $p_1, \cdots, p_k$ is evaluated lexicographically. Further, the following rule is applied because it is possible that

equivalent vertices (vertices belonging to the same orbit) are selected as cutpoints at the same time: The assignment order of cutpoints in a given block is determined in relation to the blocks which connect with the given block. That is to say, cutpoints are assigned in depth-first order to each connecting vertex of adjacent block node in the BCT canonical form, and a smaller number is assigned to a prior (i.e., deeper) vertex in an orbit. Figure 2 shows an example of cutpoint assignment for vertices of blocks.

BCT representation of the structure in Figure 1 is shown in Table I. The tree code consists of four lines. The first line indicates the node number assigned in depth-first order in BCT canonical form. The second line is the tree code computed by Edmonds' algorithm, and the figure in each column represents the number of descendants (including itself) of the node which is specified by the number above it in the first line. The third line shows the identifiers of the block nodes specified by the first line. These are key values of the block dictionary, but in this sample we have substituted alphabetic symbols; in practice they are registry numbers.

The connectivity matrix is shown in the lower part

of Table I. Row/column numbers correspond to the numbers of blocks indicated in the first line of the tree code. The numbering of vertices in a block is shown in Figure 2. Therefore the first row of connectivity matrix indicates that block 2 is connected with block 4 at vertex 1, and with block 7 at vertex 2. The second row represents that block 4 is connected with block 2 at vertex 1, and with block 6 at vertex 4. Rows from the third to the seventh are interpreted similary. Actually the connectivity matrix is linked to the tree code in a list form, as shown in the fourth line in Table I.

The hierarchical representation of chemical structures is realized through this use of BCT. This BCT representation of chemical structures gives a record type of CSDB (BCT) which is an intermediate structure representation, and at the same time, a record type (BCF) which represents block components of a chemical structure and neglects the connectivity among blocks. These two record types make it possible to access CSDB flexibly and rapidly.

CONSTRUCTION OF CSDB

The diagram of a CSDB (Chemical Strucutre Database) which is constructed on the basis of BCT representation of chemical structures is shown in Figure 3. Figure 3 shows that the CSDB has a network structure which allows a loop (recursive reference); on the other hand, the access path from a compound through an intermediate representation in block terms to internal structures of blocks is given, and this reflects the hierarchy of BCT representation. That is to say, the characteristic of CSDB is a network structure incorporating both hierarchy and recursiveness. Record types in the CSDB are explained below.

1. MASTER. The MASTER file has all the attributes of compounds except structural data: CAS registry number, compounds name, molecular formula, molecular weight and identification number as key items of the CSDB. New items can be added in order to record physical properties of various kinds as attributes of compounds.

2. BCT. The BCT file is one of the two files which contain structural data for compounds. Structure is one of the attributes of compounds, and the BCT file has the records of BCT representation of chemical

structures, where internal structures of constituent blocks are known in the block dictionary (BD). The key item of the BCT record is common to MASTER and BCF. Cross-reference between these files is accomplished by that key item, and the line between these three record types in Figure 3 shows it.

The upper part of Table I shows the BCT representation of the structure in Figure 1. The fourth line represents the connectivity matrix; here, a number without underline indicates the length of the subsequent underlined subsequence. Each underlined subsequence corresponds to the row of the connectivity matrix shown in the lower part of Table I in row number order. For example, subsequence 4 1 7 2 means that block 2 is connected with block 4 at vertex 1, and with block 7 at vertex 2. The other lines are explained in the previous section. The correspondence of BCT to BD is n : 1.

3. BCF. The BCF (Block Component File) record represents the block components of chemical structures: it indicates what blocks are contained in a compound, so it is regarded as a kind of fragment file. The BCF record format is shown in Figure 4. The record length is fixed at 1024 bits excluding the identifier filed.

A bit position or a tuple of some bit positions corresponds to identifiers of constituent blocks of a compound.

The record is divided into two fields: a simple block field and a superblock field. A simple block is a block defined graph-theoretically as described before; the term "block" without the qualifier "simple" is used in the following discussion to mean a simple block. A superblock is a subgraph which consists of connected plural blocks, and is used mainly for representing substructures which contain acyclic parts (BCT representation is not always advantageous for representing acyclic substructures). Each field is further subdivided into two parts: a core part and an extension part. A core part contains blocks/superblocks of high frequency in compounds, and each bit position in a core part corresponds to an identifier of a block/superblock. 416 blocks and 160 superblocks are assigned to core parts of block and superblocks.

An extension part contains blocks/superblocks which are of comparatively lower frequency in compounds. It consists of a modifier part of 64 bits and a bit position (abbreviated to bp) part of 160 bits. A block /superblock in an extension part is expressed by

a tuple of bits, (m,p), where m is the value of a modifier part and p is an on-bit position of a bp part. A tuple (m,p) means that the record is stored logically as the p-th record in the m-th page of a BD/SBD file. In Figure 4, the record indicates that blocks i, (m_1,j) and superblocks k, (m_2,l) are the components of the structure.

A modifier is defined so that it reflects chemical and topological features of a chemical structure. The contents of modifiers are shown in Table II. When the number of on-bits of a bp part is plural, the logical sum of modifiers corresponding to those on-bits is assigned to the modifier part: $m = m_1 \oplus m_2 \oplus \cdots \oplus m_k$, where m_i is a modifier corresponding to a single on-bit (a constituent block/superblock, $i = 1, \cdots, k$) and $\oplus$ is an operator for logical sum. In this case, the extension part becomes ($m, p_1, \cdots, p_k$). This would seem to indicate that blocks/superblocks $p_1, \cdots, p_k$ are stored in the same page (page m), but in fact the contents of $p_1, \cdots, p_k$ are placed in pages $m_1, \cdots, m_k$ respectively by making a thesaurus of modifiers. If the number of blocks/superblocks of the same modifier exceeds 160, the 7th and 8th bytes of a modifier are used for extending pages. This value is regarded as

the second figure of the page number of modulus 160. Therefore, $2^{16} \times 160$ blocks/superblocks can be expressed in a modifier. An example of a BCF record is shown in Figure 5. Blocks B_1, B_2, B_3, B_5 and B_6 are members of a core part; block B_4 is a member of an extension part and is also superblock SB_1. Modifiers m_1 and m_2 are as follows (byte 1-6):

m_1 = 0 0 1 0 0 0 0 0 | 0 0 0 1 0 0 0 0 | 1 0 0 0 0 0 0 0 |
0 0 0 0 0 0 0 1 | 1 0 0 0 0 0 0 0 | 0 0 0 0 1 0 0 0

m_2 = 0 1 0 0 1 0 0 0 | 0 1 0 0 1 0 0 0 | 1 0 0 0 0 0 0 0 |
0 0 0 0 1 0 0 1 | 1 0 0 0 0 0 0 0 | 0 0 0 0 1 0 0 0

where vertical lines (|) are inserted to make it easy to see every byte by separating the bit sequence.

BCT and BCF are indexed sequential files. A modifier of the same format as that of superblocks is computed for each compound, and is used for indexing the storage page for compounds. The storage page consists of 160 records corresponding to the BCF record format. A page suffix is inserted when the number of compounds of the same modifier exceeds 160. The suffix is also put in the modifier itself (the value in the 7th and 8th byte of the modifier, as noted earlier).

4. BD. The BD (Block Dictionary) is a file of the contents of blocks (structure code) which are the

elements for describing BCT/BCF records. A block code is in the form of a connection table. The numbering method for atoms (vertices) is based on Morgan's algorithm. The constituent unit of the BD file is a page of 160 records corresponding to the BCF record format. The procedure for registering blocks is as follows: The modifier is computed for a given block, then the logical page for the modifier is consulted. The block is registered if it is not found. If the page itself has not yet been set up, the block is registered after it is set up. If the number of registered blocks of the page exceeds 160, a new page is added with a suffix which is put in the 7th and 8th byte of the modifier. The access to the file is managed by the modifier thesaurus which is also a directory to the BD file.

5. SBD. The SBD (Superblock Dictionary) gives the contents (structures) of superblocks which appear in BCF records. In this sense, the SBD plays the same role as the BD in the CSDB, but it is not strictly an independent file. Rather, it is defined as another phase of the BCT/BCF file, because the structures of superblocks are equivalent to ordinary compounds. SBD and BCT/BCF hold a record in common, and the key item

(the identifier field in Figure 3) of the record contains flag bits for identifying the membership of the record:

bit 1 ••• on if the record is a member of BCT/BCF

bit 2 ••• on if the record is a member of SBD

bit 3-32 ••• registry number of the record

Though the number of superblocks should be much larger than that of whole compounds in the literal sense of superblocks, the size of SBD is maintained as many as that of BD according to the statistical check about query structures because the essential sense of using superblocks exists in the point that it increases the power to describe profieles of chemical structures in practical aspect.

The access to the SBD is managed by the modifier thesaurus which is also a directory to the SBD file. A recursive construction is used for the common part of SBD and BCF in which superblocks are explicitly represented (see Figure 3).

CONCLUSION

A chemical structure database (CSDB) which makes it easy to access substructures in various aspects can be formed by using blocks and superblocks as intermediate dascribing elements to represent chemical structures. In particular, the representation by bit sequence makes it easy to abstract specified substructures, which is an operation corresponding to imaging for attributes of relational database model. Any combination of Boolean logic operations, including negation, is easily executed. The Markush type of representation is important with regard to patent information, and can be handled by BCF.

The present method for organizing a CSDB is useful for quick substructure search. When the query structure is represented by a BCT form, the characteristic of the CSDB is fully displayed for substructure search as described below; the query structure is input in the form of a connection table.

(1) Search Pattern Making. The BCT representation of the query structure and a structural description with the same format as the BCF record are made from the input connection table.

(2) BCF search. If the query structure is registered

in SBD as a superblock, the bit pattern representing it is searched in BCF. If the query structure is not a member of SBD, the bit pattern representing its block components is searched as a screening step, and then the connectivity among blocks is checked by consulting the BCT file.

Further, this CSDB is suitable for structure inference from spectral data, molecular design, and other application systems.

REFERENCES

1. M.F.Lynch, in J.E.Ash and E.Hyde, Chemical Information Systems, Ellis Horwood, 1975, p.177.
2. M.Milne, D.Lefkovitz, H.Hill and R.Powers, J.Chem. Doc., 12 (1972) 183.
3. W.Graf, H.K.Kaindl, H.Kniss, B.Schmidt and R.Warszawski, J.Chem.Inf.Comput.Sci., 19 (1979) 51.
4. F.Harary, Graph Theory, Addison-Wesley, Reading, Massachusetts, 1969.
5. T.Nakayama and Y.Fujiwara, J.Chem.Inf.Comput.Sci., 20 (1980) 23.
6. I.C.Ross and F.Harary, Manag.Sci., 1 (1955) 251.
7. A.V.Aho, J.E.Hopcroft and J.D.Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.
8. R.G.Busacker and T.L.Saaty, Finite Graphs and Networks: An Introduction with Applications, McGraw-Hill, 1965.
9. H.L.Morgan, J.Chem.Doc., 5 (1965) 107.

TABLE LEGENDS

Table I. BCT representation of a chemical structure.

Table II. Contents of block / superblock modifier.

FIGURE LEGENDS

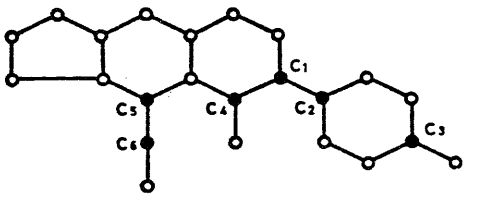
Figure 1. Cutpoints, blocks and BCT of a graph.

Figure 2. Example of cutpoint assignment for vertices of blocks.

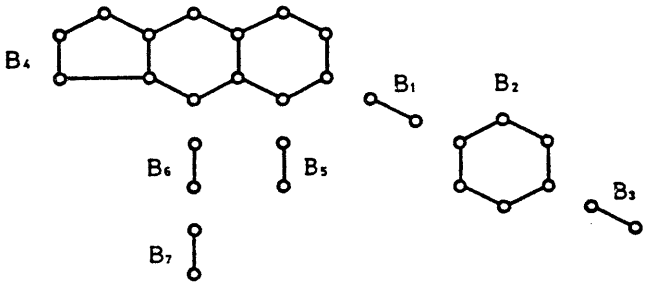
Figure 3. Block diagram of CSDB.

Figure 4. Record format of BCF.

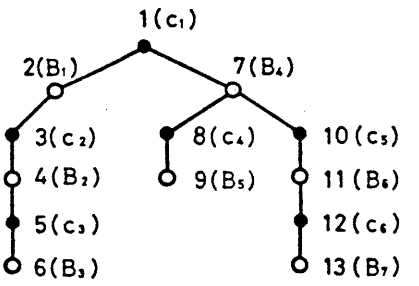
Figure 5. Example of BCF record.



cutpoints



blocks



BCT

Figure 1

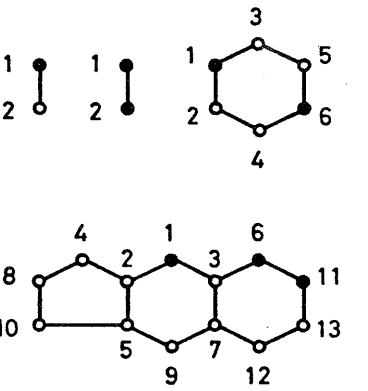


Figure 2

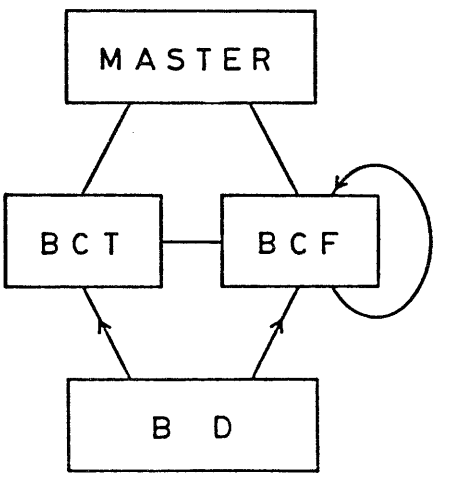


Figure 3

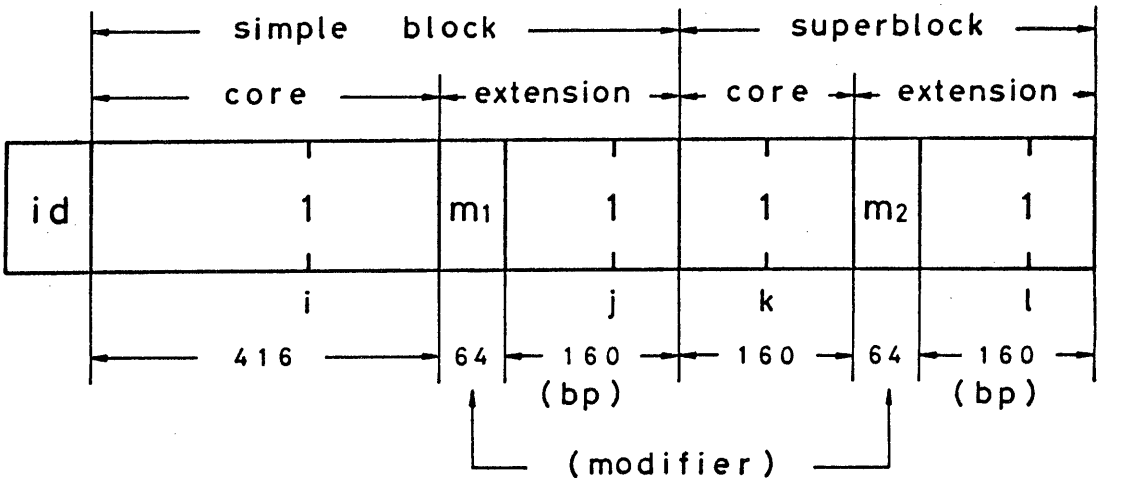
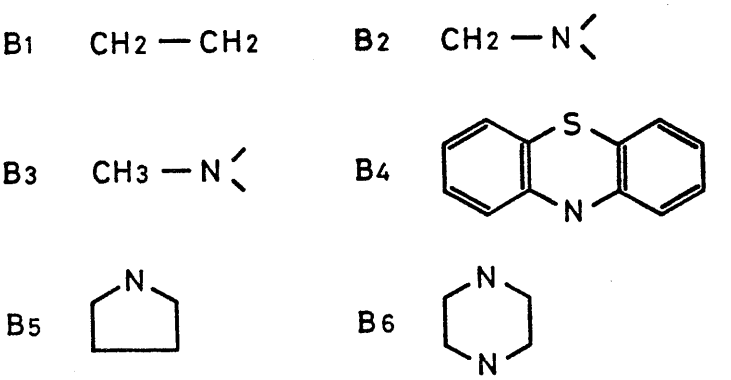
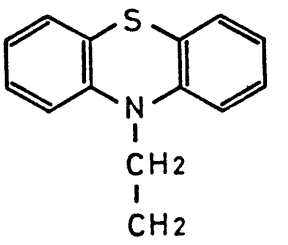


Figure 4



simple blocks



superblock

Figure 5(1)

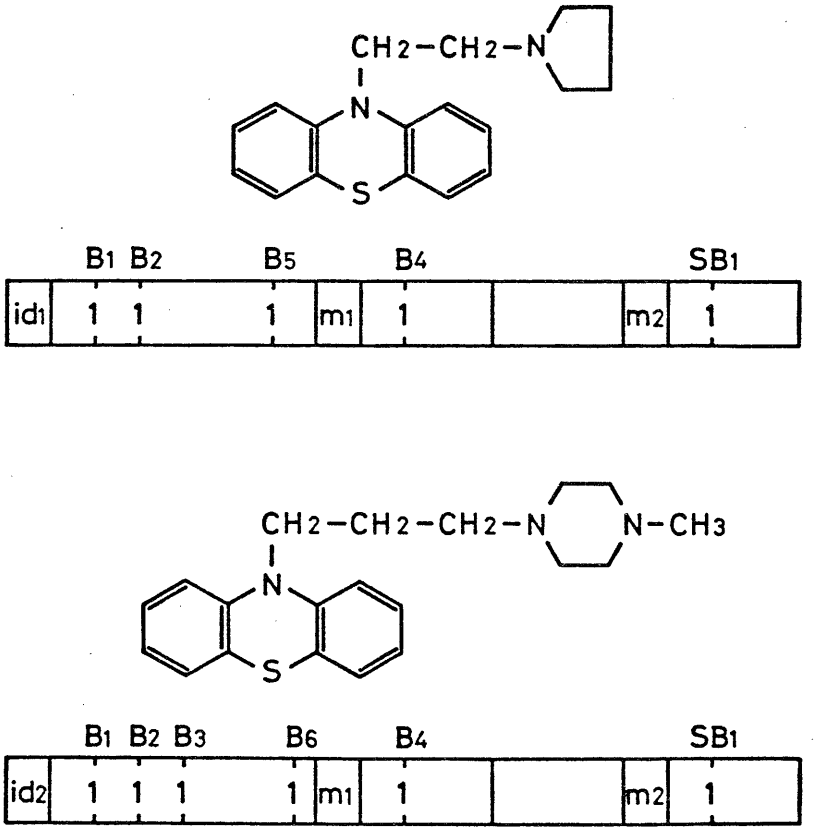

CH2-CH2-N
N
S
B1 B2 B5 B4 SB1
id1 1 1 1 m1 1 m2 1
CH2-CH2-CH2-N N-CH3
N
S
B1 B2 B3 B6 B4 SB1
id2 1 1 1 1 m1 1 m2 1

Figure 5(2)

BCT code

1	2	3	4	5	6	7	8	9	10	11	12	13
13	5	4	3	2	1	7	2	1	4	3	2	1
	a		*b*		*c*	*d*		*c*		*e*		*c*

4 4 1 7 2 4 2 1 6 4 2 4 1 6 2 11 9 6 11 1 2 7 1 4 7 1 13 2 2 11 1

connectivity matrix

	2	4	6	7	9	11	13
2	0	1	0	2	0	0	0
4	1	0	4	0	0	0	0
6	0	1	0	0	0	0	0
7	11	0	0	0	6	1	0
9	0	0	0	1	0	0	0
11	0	0	0	1	0	0	2
13	0	0	0	0	0	1	0

Table I

simple block modifier

byte	bit		
1	1	number of rings	= 1
	2		= 2
	3		= 3
	4		= 4
	5		= 5
	6		= 6
	7		= 7
	8		≥ 8
2	1	ring type	three-ring
	2		four-ring
	3		five-ring
	4		six-ring
	5		seven-ring
	6		eight-ring
	7		nine-ring
	8		ten or more-ring
3	1	core / extension	= 0 / 1
	2	not used	
	3		
	4		
	5		
	6		
	7		
	8		
4	1	number of double bonds	= 1
	2		= 2
	3		= 3
	4		≥ 4
	5	number of triple bonds	= 1
	6		≥ 2
	7	number of aromatic bonds	= 0
	8		≥ 1
5	1	number of nitrogen atoms	= 1
	2		= 2
	3		= 3
	4		= 4
	5		= 5
	6		≥ 6
	7	number of oxygen atoms	= 1
	8		= 2
6	1		= 3
	2		= 4
	3		= 5
	4		≥ 6
	5	number of sulphur atoms	= 1
	6		= 2
	7		= 3
	8		≥ 4

Table II(1)

superblock modifier

byte	bit		
1	1	number of acyclic block types	= 1
	2		= 2
	3		= 3
	4		≥ 4
	5	number of cyclic block types	= 1
	6		= 2
	7		= 3
	8		≥ 4
2	1	number of acyclic blocks	= 1
	2		= 2
	3		= 3
	4		≥ 4
	5	number of cyclic blocks	= 1
	6		= 2
	7		= 3
	8		≥ 4
3	1	core / exeleton	= 0 / 1
	2		
	3		
	4		
	5		
	6		
	7		
	8		
4	1	BCT skeleton id No.	
	2		
	3		
	4		
	5		
	6		
	7		
	8		
5	1	number of nitrogen atoms	= 1
	2		= 2
	3		= 3
	4		= 4
	5		= 5
	6		≥ 6
	7	number of oxygen atoms	= 1
	8		= 2
6	1		= 3
	2		= 4
	3		= 5
	4		≥ 6
	5	number of sulphur atoms	= 1
	6		= 2
	7		= 3
	8		≥ 4

Table II(2)

INSTITUTE OF INFORMATION SCIENCES AND ELECTRONICS
UNIVERSITY OF TSUKUBA
SAKURA-MURA, NIIHARI-GUN, IBARAKI 305 JAPAN

REPORT DOCUMENTATION PAGE	REPORT NUMBER ISE-TR-81-20

TITLE

A Graph Database for storage of Chemical Structures Organized by BCT

AUTHOR(s)

Yuzuru Fujiwara [Institute of Information Sciences and Electronics, University of Tsukuba]

Takashi Nakayama [Central Research Laboratory, Kuraray Co. Ltd.]

REPORT DATE	NUMBER OF PAGES
July 22, 1981	26
MAIN CATEGORY	CR CATEGORIES
5 Mathematics of computation	5.39

KEY WORDS

Graph database / Chemical graph / Graph isomorphism / Substructure search

ABSTRACT

A method is presented for organizing a graph database, which is based on the representation of chemical structures in terms of BCT (block-cutpoint tree). It is useful for quick substructure search, for saving memory as well as for convinience in structure generation. The database consists of four files: a master file, a bit sequence file of fixed length records which gives block components of compounds, a BCT file which gives the BCT-structures of compounds, and a block file which specifies the blocks. These files are organized recursively and hierarchically, which makes it easy to process structural information of compounds.

SUPPLEMENTARY NOTES