A Computer-Oriented Linear Canonical Notational System for the Representation of Organic Structures with Stereochemistry

Krishna K. Agarwal*

Department of Computer Science, Louisiana State University in Shreveport, Shreveport, Louisiana 71115

Herbert L. Gelernter

Department of Computer Science, State University of New York at Stony Brook, Stony Brook, New York 11794

Received May 11, 1993*

Computer algorithms and programs have been developed for a simple and effective notational system for organic molecules. The main features of the system are as follows: (1) The name produced is canonical and linear and consists of two parts. The first part represents the constitution of the molecule, the second its stereochemistry. Two molecules that are constitutionally alike but stereochemically distinct (diastereomers) have identical constitutional representations but different stereochemical descriptors. (2) Atoms which are constitutionally equivalent are identified, as are those that are stereochemically equivalent. (3) Generalized Huckel-resonant substructures of the molecule are identified by the system, allowing different resonant forms of the same molecule to be given identical names. (4) The program accepts as input an easy to write stereochemical descriptor of the molecule which is in fact a noncanonical form of the canonical name. (5) Molecules which are mirror images of one another (enantiomers) are identifiable from their canonical names. (6) The smallest number of asymmetric carbons at which two diastereomers differ can readily be computed from their names. Few systems offer the depth, breadth, and algorithmic correctness in addition to easy bidirectional human–machine communication of organic molecules including their stereochemistry. Although the notational system described here was developed to provide a complete and independent canonical stereochemical descriptor of a molecular structure for the SYNCHEM2 organic synthesis discovery program and has been successfully used for several years, the stereochemical descriptors may be used in conjunction with other standard nomenclature systems to extend their range to include stereochemistry.

INTRODUCTION

Many methods have been developed and are now in use for uniquely representing organic molecules.[1-3,16-28] Each has features that recommend it for one application or another, and yet it may lack several other desirable features. Among the older systems, the Wiswesser notation,[1] for example, has the advantage of linearity, conciseness, and structure elucidation, with properties that make it quite suitable for indexing and substructure search. On the other hand, it is defined by a large number of complex rules which make it difficult to use, its canonicity is questionable, and it does not include an accepted system for describing stereochemistry. Another old method, the Morgan system,[2] is canonical, well-suited for computer manipulation, and in an extension by Wipke and Dyott,[3] linear and capable of representing stereochemistry. It is, however, nonconcise, and the notations give no clues as to the chemical nature of the structure without decoding, a time-consuming and difficult procedure for the human chemist without a computer terminal at his or her immediate disposal. Among the more traditional nomenclature systems, the IUPAC system in conjunction with the Cahn–Ingold–Prelog rules[4-6] for stereochemistry is concise, linear, and highly descriptive of the named structure, but it is not canonical, and the large and rather incoherent set of defining rules make this system somewhat unsatisfactory for computer generation, manipulation, and decoding. Indeed, the complexity of the rules makes it exceedingly difficult to translate a structure diagram into a unique IUPAC name, manually or by machine.[17-19] Computer translation of an IUPAC name to a connection table has been attempted by Kirby et al.[20-25] This too has turned out to be very complex, and it is unclear whether the work is complete, i.e., does the system handle every possible input? Handling the stereochemistry of molecules has been done only for a subset of IUPAC names.[24] Furthermore, name correction requires manual intervention in some cases.[25] All these considerations make the IUPAC nomenclature system unsuitable for bidirectional human–machine communication such as that necessary in SYNCHEM2, an automatic synthetic planning system which can manipulate stereochemical organic molecules with no manual intervention whatsoever. SMILES,[26-28] a simplified molecular input line entry system, is convenient for bidirectional human–machine communication, but cannot handle stereochemistry or Huckel-resonance.

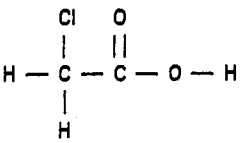


Figure 1.

The notational system proposed here, while lacking in full measure some of the desirable features mentioned above, has none of the listed disadvantages and has some further advantages, as well. For example, the notational algorithm identifies constitutionally and stereochemically equivalent atoms of the organic structure, and enantiomerism and diastereomerism with respect to a given set of molecules are apparent from the canonical names. Furthermore, that part of the name which describes the molecule's stereochemistry (the canonical parity vector) can be used with any of the other

* Abstract published in *Advance ACS Abstracts*, March 15, 1994.

0095-2338/94/1634-0463$04.50/0

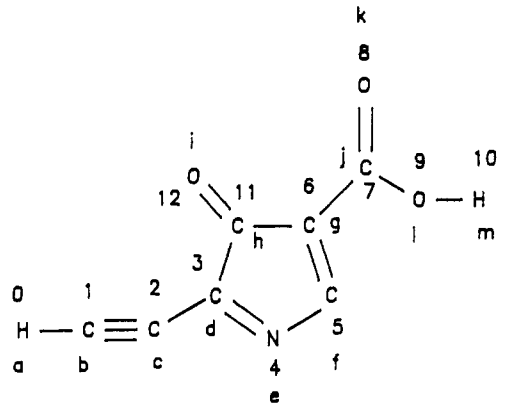


Figure 2.

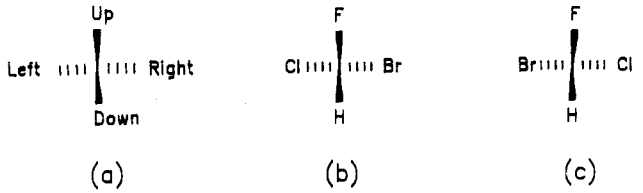


Figure 3.

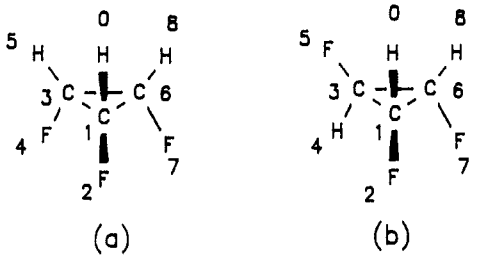


Figure 4.

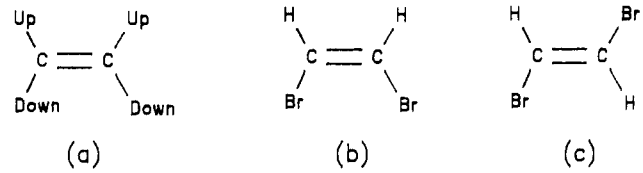


Figure 5.

notational systems. Finally, the canonical name is easy to hand-decode into a structural formula, and, in fact, a noncanonical form of the name is easily written by inspection of the structural formula. This noncanonical name is conveniently used as input to the computer algorithm for canonization. Few systems offer the depth, breadth, and algorithmic correctness that our system does. In addition, it allows for easy bidirectional human–machine communication of organic molecules including their stereochemistry.

SYNCHEM2 NOTATIONAL SYSTEM

Names produced by our system (called SNS, for SYNCHEM2 notational system) consist of two parts, a SLING (for SYNCHEM2 linear input graph) and a CPV (for canonical parity vector). The SLING,[7] which is a representation of the chemist's structural formula, is linear, reasonably compact in its hydrogen-free version, easily interconvertible with the structural formula, and can contain complete local stereochemical information. SLINGs produced by SNS are canonical, but the SLING may also be used in noncanonical form for the user's input to the program. We remark here that a major part of the work done by SNS is invested in canonization of the input SLING, a problem which we conjecture belongs to that "intractable" class known as NP-complete. A noncanonical SLING for a given molecule is easily constructed by inspection of the structural formula. Thus, starting at any atom in the molecule, proceed along any bonded sequence of atoms, listing the type-label of each atom on the path in the order in which it is scanned. Singly bonded atoms along the path were listed consecutively, without intervening characters or blanks. Doubly or triply bonded atoms are separated by = or *, respectively. Atoms whose type-labels (usually, the chemical symbol) consist of multiple letters are enclosed in parentheses. Thus, NI represents a nitrogen atom singly bonded to an iodine atom, while (NI) represents a nickel atom. (Alternately, Ni may be used and parentheses may be dropped. We retain the parentheses for the sake of clarity.) Since branching molecules require the user to specify multiple paths from a given atom, whereas each atom may be cited only once in the SLING, integers are used to permit the path to return to a previously cited atom to begin a new branch. A negative integer specifies the number of edges (bonds) that must be retraced in the path to begin the branch. A positive integer or 0 indicates the atom from which the branch starts by its sequential position in the SLING, beginning with 0. That is, 0 refers to the first atom in the SLING, 1 to the next, and so on. If type-labels have been defined for superatoms, these are of course treated as if they were single-atom symbols. Thus, the string (Cl)CH-1H-1C=O-1OH is a SLING for the structure in Figure 1 using negative integer references, while the string (Cl)CH1H1C=O4OH is a SLING for the same molecule using references of the second kind.

Both kinds of reference integers may be mixed in the same SLING, so that (Cl)CH-1H-1C=O4OH and O=COH-2C-(Cl)4H-1H are also SLINGs for the molecule in question. Molecules with rings clearly require some provision for specifying a bond back to a previously cited atom. The slash symbol followed by an integer (as in .../6...) is used to specify a return to a previously cited atom via a bond, where negative and non-negative integer references have the same meaning as for branch indications. Thus, HCH-1CH-1H-1CH-1H-1/1 is a possible SLING for cyclopropane, as is also the string HCH-1CH-1H-1CH-1H-1/-2. It should be noted that a retraced edge is ignored in counting the number of edges back to the previously cited atom for ring closure or to begin a branching path. Whenever an integer appears in the SLING, whether or not it is preceded by a /, the algorithm's "cursor", which indicates the atom to which the next atom in the SLING sequence must be attached, is moved to the atom to which the integer refers. The presence of a / before the integer indicates that a bond exists between that atom and the one to which the cursor had pointed previously; that is, the cursor moves back along a previously untraced bond. The absence of the / indicates that there is no previously untraced bond between those atoms. If the integer is negative, the cursor moves to the atom which is that number of edges previously traced (but not previously retraced) back along the path. If the integer is non-negative, the cursor moves directly to the atom whose sequence number is that integer, the number of edges traced or retraced being irrelevant in this case. These points are illustrated by the following three SLINGs for the structure in Figure 2:

SLING1: HC*CC=NC=CC=O-1OH-3C=O-1/3

SLING2: HC*CC=NC=CC=O-1OH-3C=O-1/-4

SLING3: HC*CC=NC=CC/3+7=O6C=O-1OH

The first two SLINGs derive from the path which visits each atom for the first time in the order indicated by the

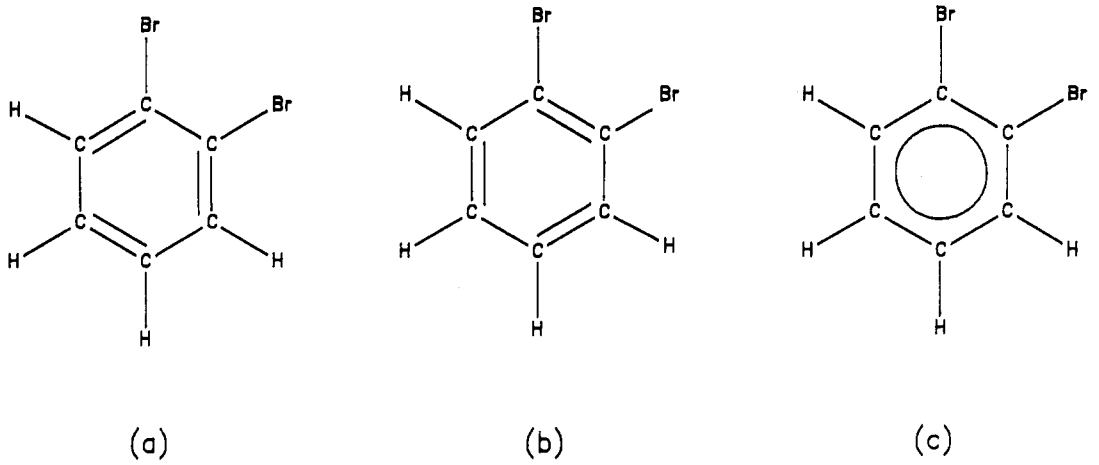


Figure 6.

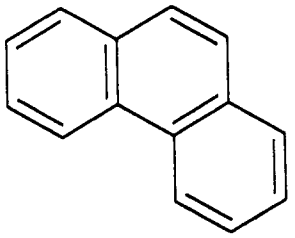


Figure 7.

integers 0–12. The third SLING is derived by visiting the atoms in the order indicated by the labels a–m. Note that while the + sign is not necessary to specify a non-negative integer that follows a symbol in the SLING which is not also an integer, it must be introduced explicitly as a separator whenever integers are cited serially. Thus, the ...+7=O indicates that an oxygen atom is doubly bonded to the carbon atom labeled 7 in the second path after the ring has been closed with a single bond to the carbon atom labeled 3. It is usually most convenient to trace branches in a chemical graph by backtracking with negative integers, while ring closures (as well as the starting node of a ring junction in a polycyclic system) are generally most easily treated with non-negative integer references. Of the three SLINGs listed for the chemical graph in Figure 2, the first is in what might be called normal (but not canonical) form. SNS will accept any of the three as input (or, indeed, any of the multitude of other legal SLINGs) for conversion to the canonical SLING and parity vector. By imposing a conventional ordering on the sequence in which the bonds from an asymmetric center are traced, the SLING may be used to convey stereochemical information. The bonds at a tetrahedral carbon upon which the stereoschema in Figure 3a has been superimposed are traced in the sequence corresponding to up, down, left, and right, or even permutations of that ordering.

Thus, the stereomer in Figure 3b could be represented by the stereoSLINGs FCH-1(Cl)-1(Br), (Cl)CH-1F-1(Br), or any other even permutation of the ligands, while any odd permutation of the ligands (HCF-1(Cl)-1(Br), for example) is a stereoSLING for its enantiomer, Figure 3c. As a further illustration, consider the diastereomers in Figure 4. The atoms will be visited in the order indicated by the integers 0–8.

A stereoSLING for the first could be given as HCF-1CF-1H-1CF-1H-1/1. Notice that we have arbitrarily defined the up direction to be the hydrogen-substituted face of the ring. The bonds of the first tetrahedral carbon are clearly traced in the order up, down, left, and (ring closure) right, while the remaining carbons have their bonds traced in an even permutation of that sequence, right, down, up, and left. A stereoSLING for the other diastereomer could be written as HCF-1CH-1F-1CF-1H-1/1. To specify cis–trans isomerism about an olefin bond, the ligands are traced in the order up, down, olefin (or any even permutation of that sequence). An arbitrarily selected up orientation at one of the olefin carbons determines the up direction at the other olefin carbon (Figure 5a).

Noting that the sequence olefin, up, down is an even permutation of the standard ordering; the complete olefin functionality in Figure 5a might be traced in the order up1, down1, olefin1, olefin2, up2, down2. Thus, HC(Br)-1=CH-1(Br) is a stereoSLING for *cis*-dibromoethane (Figure 5b), as is also (Br)CH-1=C(Br)-1H, which is obtained if the molecule is oriented with the bromine atoms up. A stereoSLING for *trans*-dibromomethane (Figure 5c) is HC(Br)-1=C(Br)-1H. Finally, where stereo asymmetry must be specified for a nitrogen atom, the ligands are traced in the counterclockwise order at the base of a tetrahedron, where the nitrogen atom is assumed to be at the apex. Since a given bond, which is only represented once in a SLING, might have to convey information relating to the stereochemistry of two asymmetric centers (if both atoms joined by the bond are asymmetric), it may be impossible to derive a valid SLING which reflects the true stereochemistry at each end of the bond because of conflicting requirements for bond ordering. The difficulty is circumvented by introducing the ~ character in the SLING to indicate an inversion of parity at that node. The inversion character may be introduced wherever the SLING path cursor points to the node where parity inversion is to be specified. Thus, HCF-1(Cl)-1(Br) is equivalent to the string FCH-1(Cl)-1~(Br), as are also the strings HCF-1(Cl)-1(Br) and HCF-1~(Cl)-1(Br). More compact SLINGs may be obtained by omitting hydrogen atoms bonded to carbon atoms if they are not needed to specify stereochemistry. Using this compacting convention, the string (Cl)CC=O-1OH is a valid SLING for the molecule in Figure 1, CCC/0 is a SLING for cyclopropane (note that in the absence of a leading hydrogen, the ring is closed to the 0th atom in the SLING), and the molecule in Figure 2 becomes (less impressively) C*CC=NC=CC=O-1OH-3C=O-1/2.

A molecular notational system must satisfy two fundamental requirements. First, names produced by the system must be distinct for different molecules. Since the name SNS generates for a molecule is in the form of a valid SLING for that molecule, and since only one molecule can be constructed from a given SLING, SNS gives the same name for two molecules only if they are in fact the same. The second requirement specifies that a unique name must be generated for a given molecule from any of the many possible valid ways

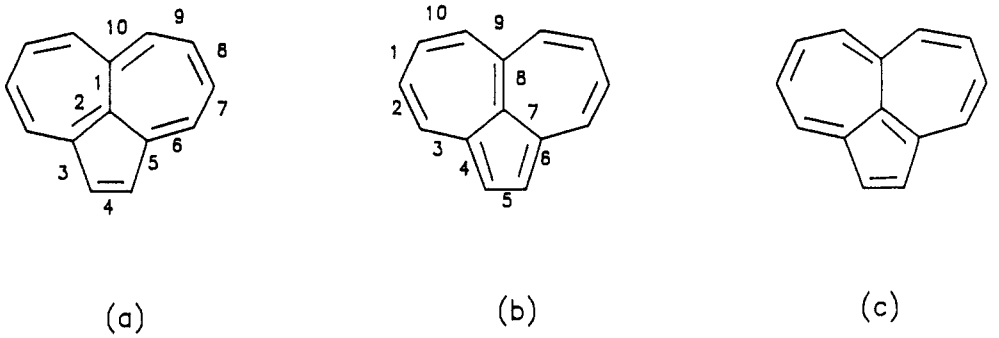


Figure 8.

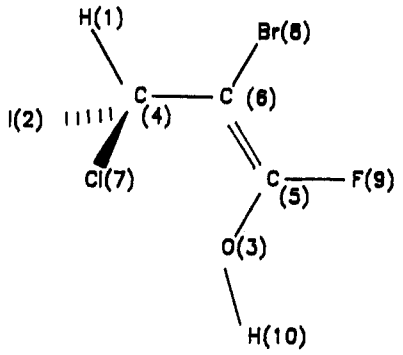


Figure 9.

Node-number	Atom	Up	Down	Left	Right	In	Out
4	C	7:1	2:1	1:1	6:1		
2	I	4:1					
1	H	4:1					
5	C	3:1	9:1	0	0	6:2	
3	O	5:1	10:1				
10	H	3:1					
7	Cl	4:1					
8	Br	6:1					
6	C	4:1	8:1	0	0	0	
9	F	5:1					5:2

Figure 10.

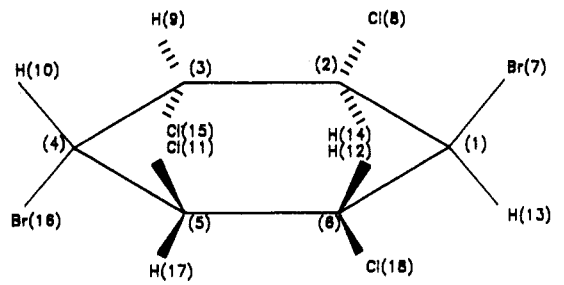


Figure 11.

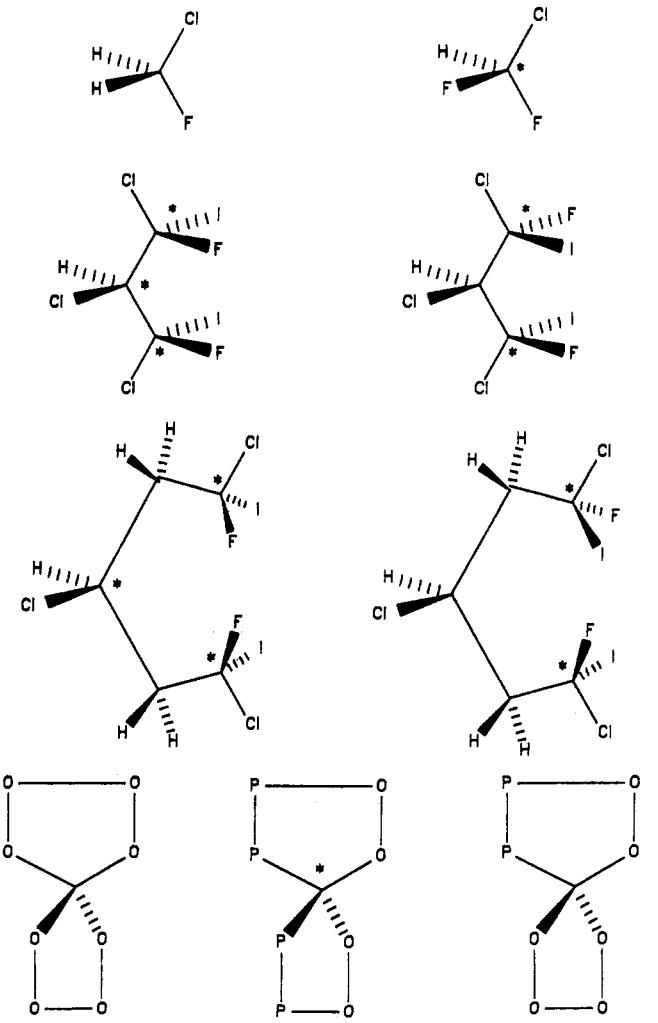


Figure 12.

of representing its structure to the system. That is, the same name must be produced for any of the possible descriptions of the molecule that can be used as input to the system. Thus, a system accepting SLINGs as input should produce the same canonical name for each of the following valid SLINGs for difluoromethane: HCH-1F-1F, HCF-1H-1F, CH-1H-1F-1F, FCF-1H-1H. SNS produces a canonical SLING by defining a unique path along which the molecule is traversed in generating the name. Once this path is determined, and providing that conventions have been established for the use of the ~,/ and for the use of negative and non-negative integers for backtracking and ring closure, only one possible SLING can result. In this way, a canonical SLING is generated for any molecular structure with well-defined fixed bonds. The occurrence of bond resonance in an organic molecule complicates the problem of consistent naming, since different resonant bond configurations of the same molecule must be given the same name. For example, both a and b in Figure 6 are valid structural representations of 1,2-dibromobenzene, but in fact neither configuration truly represents the physical system, which may be thought of as a hybrid of the two structures.

Such structures are often drawn as in Figure 6c to emphasize the point that the bonds are in fact indistinguishable. Not surprisingly, we deal with the situation in SNS by defining all of the carbon–carbon bonds on this molecule to be delocalizable bonds, introducing the new character < to represent these bonds in the SLING. Canonical names for structures containing delocalizable bonds are then generated exactly as for molecules having only fixed bonds. With this notational extension, a SLING for 1,2-dibromobenzene becomes (Br)C < C(Br-1 < C < C < C < C </1. In many instances (where benzene rings occur in isolation, for example), it is easy to identify the resonant bonds. In others, however, the process is not nearly so straightforward, especially in the case of fused or bridged polycyclic systems (Figure 7).

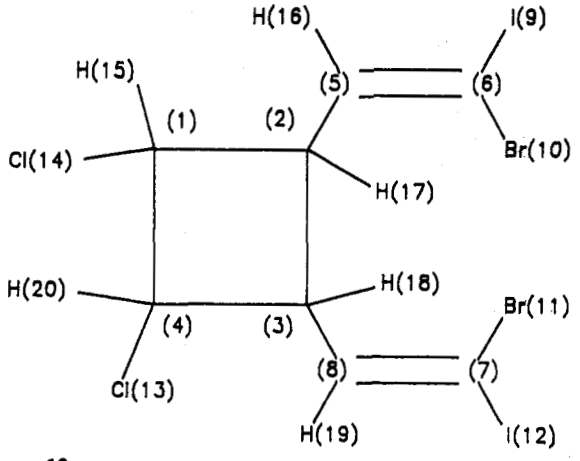


Figure 13.

Because it can become diffult for the user to recognize an occurrence of a substructure containing a cycle or cycles of delocalizable bonds, and *a fortiori* because complex structures sometimes produce differences of opinion among chemists in the designation of these substructures, the specification and identification of delocalized bonds should be a function of the notational system. In particular, the notational system must be able to recognize and so label a consistent class of delocalizable bonds so that the canonizing algorithm can start with a common classification of bond types for each of the possible representations for a delocalized bond configuration. The ability of the notational system to do this is especially important to the SYNCHEM2 organic synthesis discovery program, which must consistently name large numbers of diverse molecular species whose structures it has itself created and do so without any human intervention.[8] Although a precise definition of what constitutes an aromatic, resonant bond cycle is elusive and is occasionally the subject of lively discussion among chemists, we have chosen to use the Huckel $4n + 2$ rule (at least for those cases where $n \leq 3$) as our predictor for the presence of cycles of delocalizable bonds. Following this notation, SNS defines as delocalizable those bonds that can occur in cycles of alternating single and double bonds of size 6, 10, or 14, either in the original configuration or in any configuration that can be obtained by interchanging single and double bonds in cycles already discovered to be delocalizable. By this definition, all carbon–carbon bonds in 1,2-dibromobenzene (Figure 6) are delocalizable. Similarly, all carbon–carbon bonds in Figure 7 can be shown to be delocalizable. First, the bonds numbered 1–10 in the configuration of Figure 8a are identified as delocalizable in the given 10-cycle. Then, by shifting bonds in the alternating bond cycle of Figure 8a, the bonds numbered 1–10 in the configuration of Figure 8b are seen to form a 10-cycle of alternating single and double bonds, so that all of the carbon–carbon bonds in the structure are determined to be delocalizable. Note that still another bond configuration, that of Figure 8c, need not be analyzed, since all potentially delocalizable bonds have already been identified. Certain aromatic structures (furan and pyrole, for example) do not require the designation of delocalizable bonds for satisfactory structural representation. With slight modification, however, the algorithm above may be used as a discovery mechanism for all aromatic structures in a molecule. See Boivie[9] for more details regarding the treatment of delocalizable bonds, SLINGs, and stereoSLINGs.

INTERNAL REPRESENTATION OF THE MOLECULE

To decrease the complexity of generating a canonical description for a molecule, SYNCHEM2 uses an internal description for a molecule called the topological structural description (TSD for short). A TSD is simply a tabular representation for a molecule. Its conveys the same stereochemical information that a SLING does.

Node number	LOCAL vector	Initial ASI number
1	(1, 4, 4, 0, 0, 0)	5'
2	(1, 4, 4, 0, 0, 0)	5'
3	(1, 4, 4, 0, 0, 0)	5'
4	(1, 4, 4, 0, 0, 0)	5'
5	(1, 4, 2, 1, 0, 0)	1'
6	(1, 4, 2, 1, 0, 0)	1'
7	(1, 4, 2, 1, 0, 0)	1'
8	(1, 4, 2, 1, 0, 0)	1'
9	(19, 1, 1, 0, 0, 0)	13'
10	(18, 1, 1, 0, 0, 0)	11'
11	(18, 1, 1, 0, 0, 0)	11'
12	(19, 1, 1, 0, 0, 0)	13'
13	(17, 1, 1, 0, 0, 0)	9'
14	(17, 1, 1, 0, 0, 0)	9'
15	(27, 1, 1, 0, 0, 0)	15'
16	(27, 1, 1, 0, 0, 0)	15'
17	(27, 1, 1, 0, 0, 0)	15'
18	(27, 1, 1, 0, 0, 0)	15'
19	(27, 1, 1, 0, 0, 0)	15'
20	(27, 1, 1, 0, 0, 0)	15'

Figure 14.

Figure 9 shows an arbitrary molecule. A SLING for the molecule is HC(C1)-1I-1C(Br)-1=COH-2F. The TSD for this molecule is shown in Figure 10. Note that the atoms of Figure 9 have been numbered arbitrarily. The columns of the table have been labeled according to the skeletons of Figures 3 and 5.

From a TSD with randomly numbered atoms we will generate a canonical numbering for the atoms. The procedure which produces the canonical numbering will also identify "constitutionally equivalent" (CE) atoms as also "stereochemically equivalent" (SE) ones.

Consider the molecule of Figure 11 for example. By careful examination it may be determined that the atoms may be divided into the following "stereochemical equivalence" classes. These classes indicate which atoms are equivalent to each other, taking into consideration the asymmetry about each carbon atom. These classes are {1,4}, {2,3}, {5,6}, {7,16}, {8,15}, {9,14}, {10,13}, {11,18}, and {12,17}. Clearly, atoms in different classes are not stereochemically equivalent.

On the other hand, if stereochemistry about the atoms was ignored, we obtain the following "constitutional equivalence" classes for the molecule of Figure 11: {1,4}, {2,3,5,6}, {7,16}, {8,11,15,18}, {9,12,14,17}, and {10,13}. As in the case of stereochemical equivalence, atoms in distinct classes are not constitutionally equivalent.

For more precise definitions of stereochemical and constitutional equivalence, refer to Ugi,[10] Davis,[11] Agarwal,[12] Rucker,[13] and Figueras.[14]

Another aspect considered by the canonical-numbering algorithm is the detection of centers of asymmetry in a molecule. Asymmetry is defined to be present at an atomic site if a new molecule is obtained by inverting the site (i.e., exchanging the positions of any two ligands).

Figure 12 shows sites of asymmetry by marking them with asterisks ("*"). Although not illustrated in Figure 12, sites of asymmetry may arise at olefinic carbons also, depending on the nature of the neighboring atoms.

The canonical numbering produced by our algorithm depends entirely on the molecule's constitution and not its stereochemistry. Therefore, the algorithm produces identical canonical descriptors for diastereomers. However, the stereochemical appendages to these canonical descriptors will be distinct for the diastereomers. This implies that diastereomerism between molecules can be easily detected by comparing their canonical descriptors.

Furthermore, the algorithm affords the detection of "chiral-antipodes" or "enantiomers", i.e., molecules which are mirror images of each other (only as far as stereochemistry is

First Iteration (steps 1 to 5)

Atom number	Input ASI	OLD-ASI	NASIV	NEW-ASI
1	5′	5′	(5, 5, 9,15)	7′
2	5′	5′	(1, 5, 5,15)	5′
3	5′	5′	(1, 5, 5,15)	5′
4	5′	5′	(5, 5, 9,15)	7′
5	1′	1′	(1, 5,15)	1′
6	1′	1′	(1,11,13)	3′
7	1′	1′	(1,11,13)	3′
8	1′	1′	(1, 5,15)	1′
9	13′	13′	(1)	13′
10	11′	11′	(1)	11′
11	11′	11′	(1)	11′
12	13′	13′	(1)	13′
13	9′	9′	(5)	9′
14	9′	9′	(5)	9′
15	15′	15′	(5)	17′
16	15′	15′	(1)	15′
17	15′	15′	(5)	17′
18	15′	15′	(5)	17′
19	15′	15′	(1)	15′
20	15′	15′	(5)	17′

Second iteration (steps 2 to 5)

Atom number	OLD-ASI	NASIV	NEW-ASI
1	7′	(5, 7, 9,17)	7′
2	5′	(1, 5, 7,17)	5′
3	5′	(1, 5, 7,17)	5′
4	7′	(5, 7, 9,17)	7′
5	1′	(3, 5,15)	1′
6	3′	(1,11,13)	3′
7	3′	(1,11,13)	3′
8	1′	(3, 5,15)	1′
9	13′	(3)	13′
10	11′	(3)	11′
11	11′	(3)	11′
12	13′	(3)	13′
13	9′	(7)	9′
14	9′	(7)	9′
15	17′	(7)	19′
16	15′	(1)	15′
17	17′	(5)	17′
18	17′	(5)	17′
19	15′	(1)	15′
20	17′	(7)	19′

Third Iteration (steps 2 to 5)

Atom number	OLD-ASI	NASIV	NEW-ASI	Algorithm ends with OUTPUT-ASI
1	7′	(5, 7, 9,19)	7′	7′
2	5′	(1, 5, 7,17)	5′	5′
3	5′	(1, 5, 7,17)	5′	5′
4	7′	(5, 7, 9,19)	7′	7′
5	1′	(3, 5,15)	1′	1′
6	3′	(1,11,13)	3′	3′
7	3′	(1,11,13)	3′	3′
8	1′	(3, 5,15)	1′	1′
9	13′	(3)	13′	13′
10	11′	(3)	11′	11′
11	11′	(3)	11′	11′
12	13′	(3)	13′	13′
13	9′	(7)	9′	9′
14	9′	(7)	9′	9′
15	19′	(7)	19′	19′
16	15′	(1)	15′	15′
17	17′	(5)	17′	17′
18	17′	(5)	17′	17′
19	15′	(1)	15′	15′
20	19′	(7)	19′	19′

Figure 15.

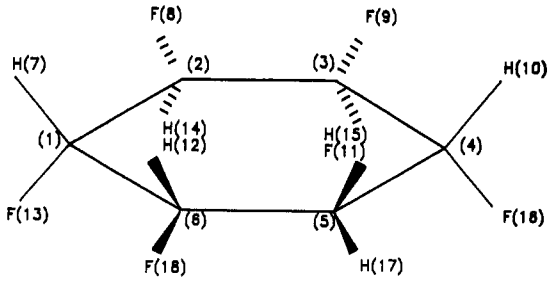
concerned, ignoring any higher-order asymmetries). This also means that we can detect whether a molecule is achiral, i.e., its own mirror image. Another feature provided by the mechanism is the detection of the smallest number of asymmetric carbons at which two diastereomers differ.

Finally, a TSD to SLING algorithm is used to make the canonical name linear, thereby imparting it the secondary (though important) attributes of readability and ease of manipulation.

CANONIZATION ALGORITHM

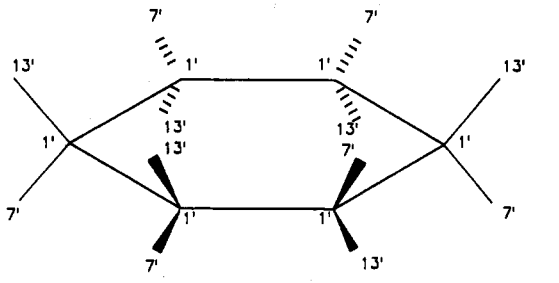
In broad outline, the canonization algorithm generates an invariant set of numberings for the atoms of a given molecule. Stereochemical considerations are disregarded during this process so that the aforementioned properties can be determined. (This statement will become clearer at the end of our discussion.) For reasons of efficiency, the set of numberings produced should be as small as possible. However, the set of numberings should be identical for molecules of identical constitution, else the algorithm cannot possibly come up with a canonical name. The algorithm discussed below has all these properties.

The atoms of the input molecule are numbered arbitrarily but uniquely. This allows us to construct an arbitrary TSD for it. To construct a canonical TSD, a systematic way of numbering the atoms must be used. If a molecule has n carbon atoms, they will be numbered 1,2, ... n in any numbering we produce. Similar decisions are made for the other atoms by



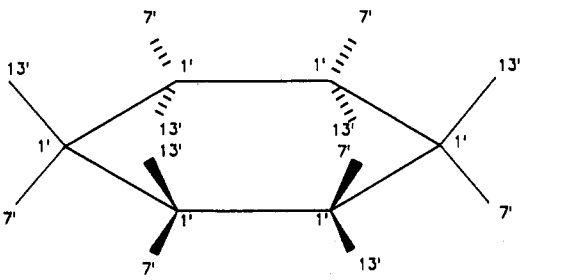
INPUT NUMBERING

Figure 16.



INITIAL ASI ASSIGNMENT

Figure 17.



STARTING ASI ASSIGNMENT

Figure 18.

considering their relative positions in the list obtained by stringing out the columns 5, 6, 7, 8, 1, 2, 3, and 4 of the periodic table. The advantage of using a list of this type is that molecules which are constitutionally similar but differ by atoms in the same column of the periodic table will have similar names.

The decision-making for assigning a range of numbers to the atoms of a molecule is done by the INITIALIZER. Consider the molecule of Figure 13, in which atoms have been arbitrarily numbered.

The initial atomic sequence index (ASI) numbers are assigned by constructing a LOCAL information vector. The LOCAL vector of length 6 is formed with the following information: (1) atom type number from the list constructed from the periodic table; (2) valence of the atom; (3) number of multiplicity 1 bonds emanating from the atom; (4) number of multiplicity 2 bonds; (5) number of multiplicity 3 bonds; and (6) number of resonant bonds.

Suppose that n_1 of the atoms have the smallest LOCAL vectors. Then each of these n_1 atoms will be assigned the Initial ASI number 1′, and all of them will be "tied" at the same Initial ASI value. If n_2 atoms have the next larger LOCAL vector, each of these n_2 atoms will be tied at the Initial ASI value of $n_1 + 1'$. Initial ASI values are assigned similarly to the remaining atoms. Figure 14 shows the LOCAL vectors and the Initial ASI values for each of the atoms.

The Initial ASI assignment can usually be refined further by taking into consideration the Initial ASI assignments of neighboring atoms. Such a refinement helps tremendously in reducing the number of numberings that the algorithm has to explore and hence reduces the time consumed in naming a molecule. The procedure that makes this refinement of ASI values uses an iterative (or relaxation) technique and is called the DIFFERENTIATOR.

The DIFFERENTIATOR is basically the same algorithm discussed by Ugi,[10] Davis,[11] and Agarwal.[12] The input to this procedure is the TSD of the molecule and an ASI assignment. Using these inputs, this algorithm attempts to differentiate atoms from each other by comparing the ASI values of their neighbors. The output of the algorithm is thus a new ASI assignment which is a refinement of the input ASI assignment.

We define a neighbor ASI vector (NASIV) for each atom to be a vector containing the ASI values for each of its neighbors in nondescending order. A neighbor is any atom connected to the atom under consideration by a bond, the bond may be single, double, triple, or resonant.

A summary of the DIFFERENTIATOR algorithm follows:

Step 1: Set OLD-ASI assignment to the Input ASI assignment.

Step 2: Form the NASIV for each atom using the OLD-ASI assignment.

Step 3: Compare the NASIV's for all atoms with identical OLD-ASI values. Consider the atoms tied at the OLD-ASI value of n_1'. If n_2 of these have the same smallest NASIV's, assign them the NEW-ASI value of n_1'. If n_3 atoms have the next higher NASIV's, assign them the NEW-ASI value of $n_1 + n_2 + 1'$. This process is repeated until each atom in the molecule has been assigned a NEW-ASI value.

Step 4: If the NEW-ASI value for each atom is identical to its OLD-ASI value, then return the NEW-ASI assignment as output. Else, continue with step 5.

Step 5: Replace the OLD-ASI value of each atom by its NEW-ASI value and continue with step 2.

Although the above description of the DIFFERENTIATOR may seem complicated, conceptually it is a simple procedure. A "trace" of the algorithm is provided in Figure 15.

The inputs to the algorithm are the TSD for the molecule of Figure 13 and the Initial ASI assignment of Figure 14.

The trace is fairly simple to understand if each step of the DIFFERENTIATOR is followed through. During the first iteration, the NEW-ASI value of 1′ was assigned to the atoms numbered 5 and 8 since their NASIV (1,5,15) was smaller than the NASIV for atoms numbered 6 and 7 (1,11,13). Since two atoms (numbered 5 and 8) were assigned the NEW-ASI value 1′, atoms 6 and 7 were assigned the NEW-ASI value 3′. The remaining part of the table is self-explanatory.

Several points about the DIFFERENTIATOR algorithm are in order:

(i) The Output ASI assignment produced by the algorithm is either the same as its Input ASI assignment or is a "finer" partition of the Input ASI assignment.

(ii) If the Input ASI assignment is the Initial ASI assignment for a molecule as produced by the INITIALIZER, then in most cases (but not all), the Output ASI assignment is such that two atoms receive identical Output ASI assignments if they are constitutionally equivalent. Examples of molecules in which non-constitutionally equivalent atoms get the same Output ASI assignments will be discussed further on in this paper.

(iii) The use of the DIFFERENTIATOR in conjunction with the canonical algorithm (as discussed later) contributes

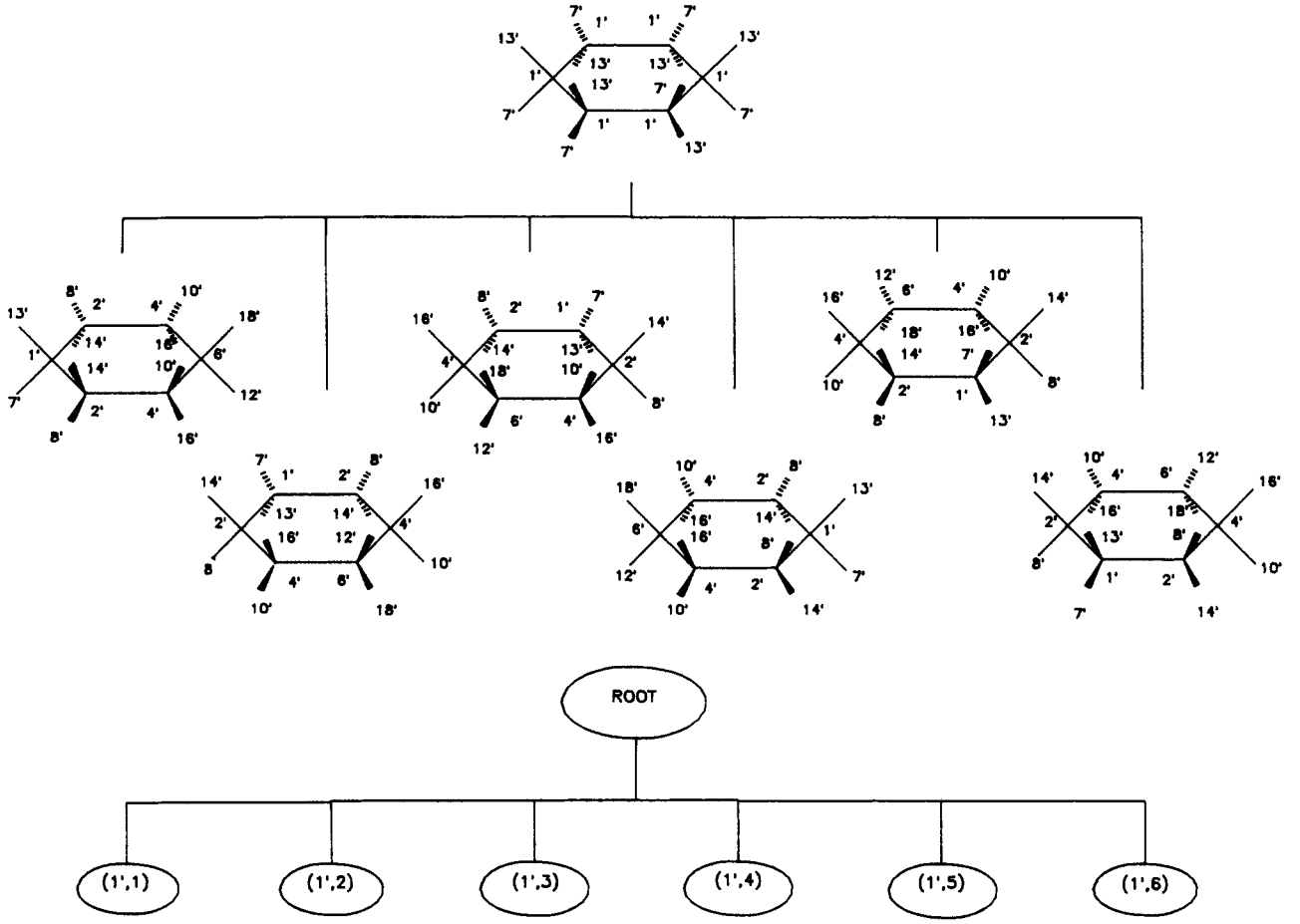


Figure 19.

to the relative efficiencies afforded by our algorithm over those discussed in Davis,[11] Morgan,[2] Wipke,[3] and Randic.[15] Without its usage, the canonical algorithm will be much more inefficient.

Now we discuss the CANONIZER. This algorithm will generate a canonical TSD for the constitution of the molecule and a canonical "parity vector" to describe its stereochemistry. Along with the statement of the algorithm we will trace through a fairly simple example to illustrate its simplicity. The molecule with its atoms numbered randomly is shown in Figure 16.

Step I: Using the INITIALIZER, generate the Initial ASI assignment for the molecule.

The Initial ASI assignment for our example is shown in Figure 17.

Step II: Perform the DIFFERENTIATOR with the Initial ASI assignment as its input. The output of this step is called the Starting ASI assignment.

The Starting ASI assignment for our example is shown Figure 18.

For our example, the Starting ASI assignment turns out to be identical to the Initial ASI assignment. In general, it will usually be a refinement of the Initial ASI assignment.

Step III: If the Starting ASI numbers assigned to each atom are distinct, stop with the information that this molecule has no CE (constitutionally equivalent) atoms and no SE (stereochemically equivalent) atoms. All carbon centers are asymmetric. Furthermore, the canonical TSD and parity vector can be formed quite easily using the same techniques discussed later on in this paper.

If the Starting ASI assignment is such that at least two atoms have identical values assigned to them, continue with step IV.

In our example the Starting ASI values are identical for several atoms, and so we proceed to the next step.

Step IV: In this step the ambiguities represented by our Starting ASI assignments are resolved. The result of this step will be a set of Final ASI assignments. Each of these assignments will be such that no two atoms have the same ASI values. For generating the Final ASI assignments we use an "ASI-TREE". Every node of the ASI-TREE represents an ASI assignment. Each of the leaf nodes (those at the lowest level) represents a Final ASI assignment. The root node of the tree (at the highest level) is initialized with the Starting ASI assignment obtained from step II. The number of nodes in the ASI-TREE in the next level equals the number of nodes in the molecule which are "tied" at the smallest ASI at the current level in the ASI-TREE. (The first time this step is executed, we look for ties in the Starting ASI assignment.)

Traversing left-to-right at the next level in the ASI-TREE, a different atom in the molecule will retain the smallest ASI tie value. All other atoms will be assigned an ASI value 1 larger. This ASI assignment will be input to the DIFFERENTIATOR, and the output obtained will form a new node at the next level.

This process of developing the ASI-TREE continues until all Final ASI assignments are obtained; that is, the ASI-TREE has all of its leaf nodes.

In the example under consideration, the first application of step IV yields the partial ASI-TREE of Figure 19. A concise notation for the ASI-TREE is also illustrated in this figure. Note that (a',b) indicates that the atom originally numbered b is now assigned the ASI value a'. Also note that the Starting

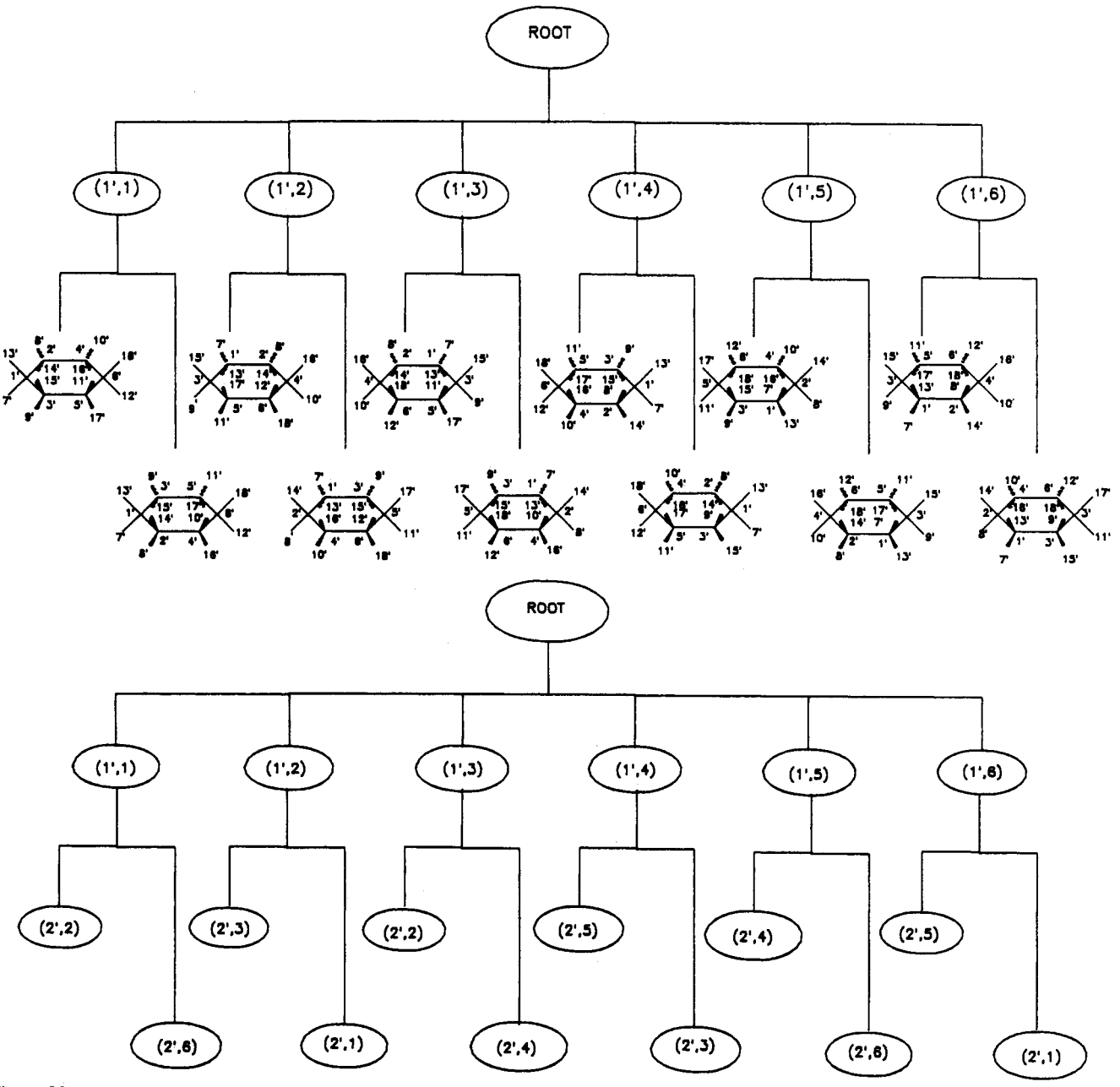


Figure 20.

ASI assignment is such that six atoms are tied at the ASI value of 1′. Hence level 1 of the tree (the root is level 0) has six nodes. In each node, a different atom is selected to retain the ASI value 1′. The remaining atoms which had the ASI value 1′ are assigned the ASI value 2′. This assignment is then run through the DIFFERENTIATOR to obtain the six level 1 ASI assignments shown in Figure 19. Each level 1 node in the ASI-TREE still has ties in the ASI assignments. Hence step IV is repeated another time. This results in the complete ASI-TREE of Figure 20. Note that there are 12 distinct Final ASI assignments, each being represented by a leaf node of this tree.

Step V: With each Final ASI assignment, form a TSD and a parity vector for the molecule.

A Final ASI assignment is used to construct a new TSD by simply substituting the ASI value for the atom in place of its original number in the original TSD. Each row of the new TSD is rearranged so that the list of neighbors appears in descending order. For a tetrahedral carbon if sorting the row misrepresents the stereochemistry of the atom, the corresponding entry in the parity vector is assigned a value of −1, otherwise it is assigned a value of +1. For olefin carbons, parity values of −2 and +2 are assigned. These rules are summarized in Figure 21, where arbitrary ASI values have been selected for the purpose of illustration.

Note that it is in this step where the stereochemistry of a molecule is isolated into a parity vector and only its constitution is retained in a TSD. This is done in order to generate identical descriptors for diastereomers while distinguishing them with their stereochemical descriptors.

For the example under discussion, the TSDs and parity vectors generated are shown in Figure 22. Each of the 12 Final ASI assignments generates the same TSD T1. The leaf nodes of the ASI-TREE in Figure 22 are labeled with the TSD and parity vector generated by the corresponding Final ASI assignment. Each parity vector is of length 6 since there are six carbon atoms in the molecule. If a carbon atom has the value i in a final ASI assignment, its parity is stored in position i in the parity vector.

In this example, only one TSD was generated by all Final ASI assignments. This will not be the case in general. An example in which several distinct TSDs are generated by the CANONIZER is discussed later on in this paper.

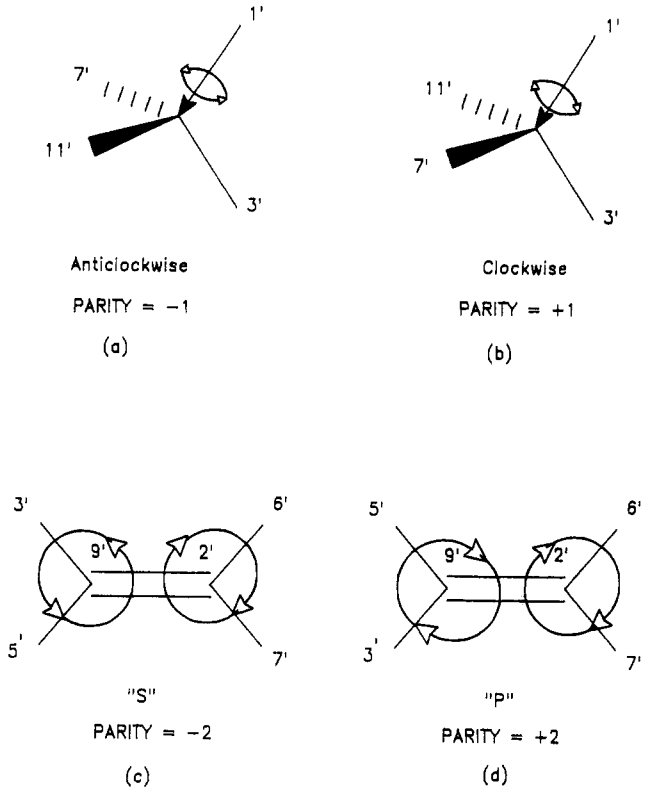


Figure 21.

Step VI: In this step we form a table. This table has an entry for each atom in the molecule. The *i*th entry in the table is for the atom with number *i* in the original TSD. Entry *i* in the table holds the following items:

(1) The smallest ASI value assigned to atom *i* in any of the Final ASI assignments on the ASI-TREE. Let *i'* be this ASI value.

(2) The minimal TSD obtained for the molecule in any of the Final ASI assignments in which atom *i* was assigned the ASI value *i'*. Note that TSDs are compared a row at a time to obtain the minimal one. Let the minimal TSD for atom *i* be T(*i*).

(3) Among all the TSDs T(*i*) for the atom *i*, save the smallest parity vector P(*i*).

(4) Initially, every atom is assumed to be asymmetric. Atom *i* is marked symmetric if the following conditions are true:

(a) There are two or more Final ASI assignments such that *i* was assigned the ASI value *i'* and the TSD T(*i*) was obtained.

(b) The parity for *i* is inverted in two parity vectors which are produced with the TSDs T(*i*). For a tetrahedral carbon these parity vectors are identical in all positions except position *i*. For a trihedral carbon, the two parity vectors are identical except for atom *i* and its trihedral carbon.

For our example, the table in Figure 23 is obtained due to this step. T1 and P1–P6 mentioned in Figure 23 are identical to those in Figure 22. The CE and SE class number columns will be explained in the next step. Referring back to the ASI-TREE of Figure 22, we see that atom 1 was assigned the lowest value of 1' in only the two leftmost Final ASI assignments. The TSDs produced were identical to T1 and therefore minimal. However, the two parity vectors were such that they differed in more than just position 1, thereby keeping atom 1 marked "asymmetric". For atom 1 to be marked symmetric, P1 and P2 should have differed in position 1 only. Other carbon atoms are marked similarly.

Step VII: As the canonical TSD, select the minimal TSD obtained among any of the Final ASI assignments of the ASI-TREE. Among all the parity vectors that were derived from the Final ASI assignments that led to the canonical TSD, select the minimal one as the canonical parity vector. Assign two atoms the same CE class number if and only if they have the same smallest ASI values and identical minimal TSDs, as observed from the table built in the preceding step. Assign two atoms the same SE class number if they have the same CE class number and identical smallest parity vectors. This information is also present in the table built in step VI.

In our example, the canonical TSD is T1 and P1 is the canonical parity vector. Note that CE class numbers are assigned by comparing the vectors (smallest ASI, minimal TSD) for the atoms. The SE class numbers are then assigned by comparing the vectors (CE class number, minimal parity vector) for the atoms. This decision was simply a matter of convenience.

It may be mentioned here that each Final ASI assignment of the ASI-TREE is generated in a "depth-first" manner, traversing the tree left to right. The table discussed in step VI is also built and updated as we generate each Final ASI assignment. The effect of combining steps V and VI in this fashion enables us to save a large amount of storage space which would otherwise be required for the ASI-TREE.

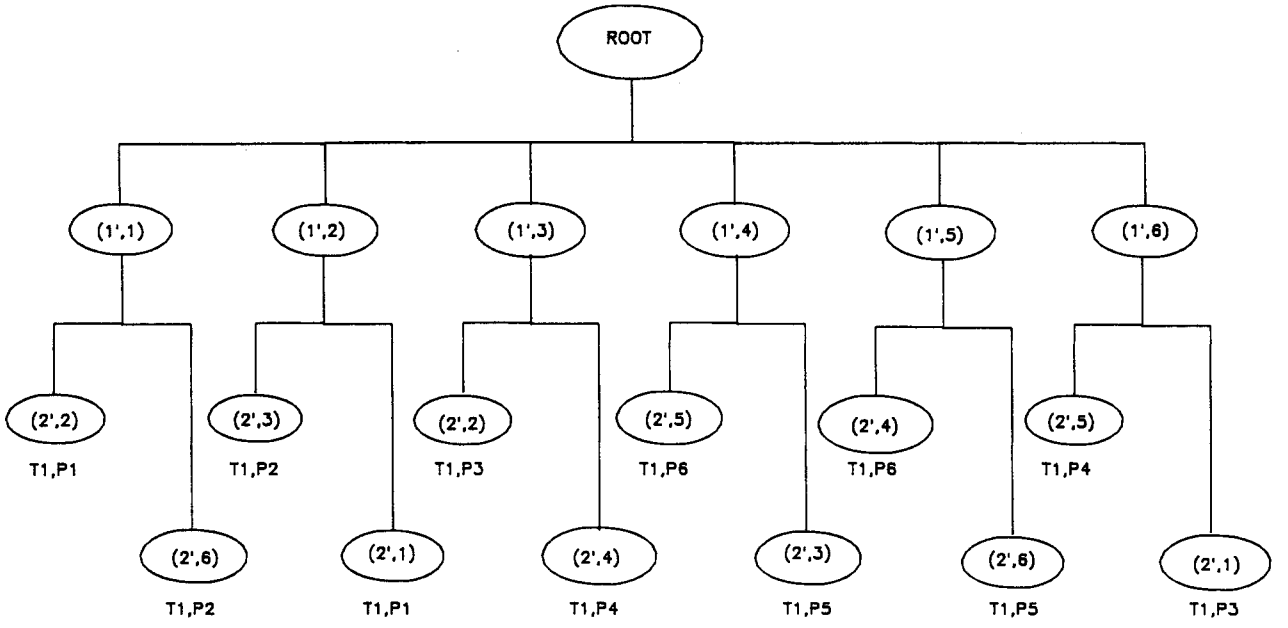
Before we leave this example we make the observation that the given molecule is chiral (i.e., not its own mirror image). This piece of information can also be derived from the parity vectors P1–P6, which accompany the canonical TSD T1. All we must do is select any of these parity vectors and observe whether its complement lies in this set. For example, the complement of P1 is (+1, +1, +1, +1, −1, −1). This parity vector is distinct from P1 to P6. Hence, we conclude that the molecule is chiral. In case a parity vector contains +2's and −2's, its complement is obtained by inverting only the +1 and −1 entries, if any.

A SECOND EXAMPLE

As a second example of the canonization algorithm, consider the molecule of Figure 24. Its atoms have been randomly numbered. After steps I and II of the CANONIZER, the Starting ASI assignment obtained (it turns out to be the same as the Initial assignment) is shown in Figure 25. Note that the Starting ASI assignment is such that non-CE atoms were assigned identical Starting ASI values. Hence, this is an example in which the DIFFERENTIATOR algorithm is too weak to distinguish non-CE atoms. The complete ASI-TREE obtained for this example is shown in Figure 26. Rather than discussing the entire ASI-TREE in detail, we will discuss some of the nodes on the ASI-TREE, indicating their relationships with the other nodes. For our discussion, let us assume we have the CE classes of the molecule, although the CANONIZER will provide us with this information. The CE classes (using the atom numbers of Figure 24) are

CE class 1:	{2,10}
CE class 2:	{6,14}
CE class 3:	{4,5,9,13}
CE class 4:	{3,12}
CE class 5:	{1,11}
CE class 6:	{8,16}
CE class 7:	{7,15}

On examining the Starting ASI assignment, we discover



THE ASI–TREE FOR ALL ASI ASSIGNMENTS OF EXAMPLE I

T1:

Node	Atom	Up	Down	Left	Right
1	C	13:1	7:1	3:1	2:1
2	C	14:1	8:1	4:1	1:1
3	C	15:1	9:1	5:1	1:1
4	C	16:1	10:1	6:1	2:1
5	C	17:1	11:1	6:1	3:1
6	C	18:1	12:1	5:1	4:1
7	F	1:1			
8	F	2:1			
9	F	3:1			
10	F	4:1			
11	F	5:1			
12	F	6:1			
13	H	1:1			
14	H	2:1			
15	H	3:1			
16	H	4:1			
17	H	5:1			
18	H	6:1			

P1: (-1,-1,-1,-1,+1,+1)

P2: (+1,-1,-1,+1,-1,-1)

P3: (-1,+1,+1,-1,-1,-1)

P4: (+1,+1,+1,-1,-1,+1)

P5: (+1,+1,-1,+1,+1,-1)

P6: (-1,-1,+1,+1,+1,+1)

Figure 22.

that the ASI values for class 1 and class 2 atoms turned out to be identical. Since atoms 2, 6, 10, and 14 all have the same smallest ASI value 1′, the first level of the ASI-TREE will break the tie in favor of each of these atoms in turn. We examine what happens along the (1′,2) and (1′,6) nodes only of the ASI-TREE since similar ASI assignments will appear for the (1′,10) and (1′,14) nodes. This is because atom 10 is CE to atom 2 and atom 14 is CE to atom 6.

The ASI assignments for the nodes (1′,2), and (1′,6) are shown in Figures 27 and 28, respectively. Note that the DIFFERENTIATOR algorithm has been used after breaking the tie in each case. Also, the atom numbering of Figure 24 is implicit in both Figures 27 and 28.

From Figure 27 it may be seen that the ASI-TREE must be developed below node (1′,2) by creating two new nodes, (2′,6) and (2′,14), below it to break the tie at 2′. The two Final ASI assignments obtained after the DIFFERENTIATOR are depicted in Figure 29.

Similarly, nodes (2′,2) and (2′,10) below node (1′,6) in the ASI-TREE correspond to the Final ASI assignment of Figure

Original atom number	Atom type	Min. ASI value assigned	Corresponding Min. TSD	Corresponding Min. parity vector	CE class No.	SE class No.	Asymetric center?
1	C	1′	T1	P1	1	1	Yes
2	C	1′	T1	P1	1	1	Yes
3	C	1′	T1	P3	1	5	Yes
4	C	1′	T1	P6	1	3	Yes
5	C	1′	T1	P6	1	3	Yes
6	C	1′	T1	P3	1	5	Yes
7	H	13′	T1	P1	13	13	
8	F	7′	T1	P1	7	7	
9	F	7′	T1	P3	7	11	
10	H	13′	T1	P6	13	15	
11	F	7′	T1	P6	7	9	
12	H	13′	T1	P3	13	17	
13	F	7′	T1	P1	7	7	
14	H	13′	T1	P1	13	13	
15	H	13′	T1	P3	13	17	
16	F	7′	T1	P6	7	9	
17	H	13′	T1	P6	13	15	
18	F	7′	T1	P3	7	11	

Note: (a) The canonical TSD is T1.
(b) The canonical parity vector is P1.

Figure 23.

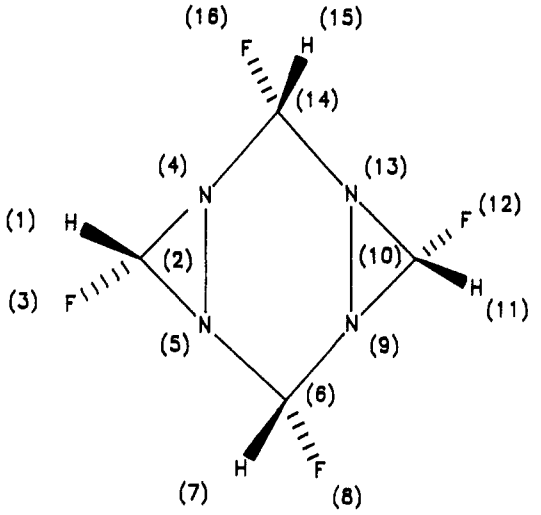


Figure 24.

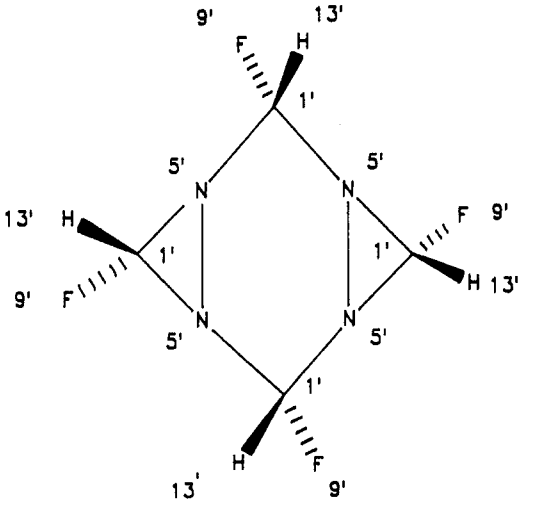


Figure 25. Initial-ASI assignment and the Starting-ASI assignment

30. Two distinct TSDs, T1 and T2, were obtained with the ASI-TREE because of the inability of DIFFERENTIATOR to distinguish between non-CE atoms, as illustrated by the Starting ASI assignment produced for the molecule. Had atoms 2 and 6 been distinguished in the Starting ASI assignment, only one of the TSDs would have been generated.

The table obtained from steps VI and VII of the CANONIZER is shown in Figure 31. Note that CE and SE atoms have been correctly identified and the centers of asymmetry properly indicated. T1 and P1 are the canonical TSD and parity vector, respectively. The molecule is achiral, as evidenced by the existence of both P1 and its inverse, i.e. P2, with the canonical TSD T1.

THE NAME FOR THE MOLECULE

The CANONIZER discussed above provides us with the following information for the molecule: (1) the canonical TSD, (2) the canonical parity vector, (3) the set of parity vectors accompanying the canonical TSD, (4) the CE class numbers for each atom in the canonical TSD, (5) the SE class numbers for each atom in the canonical TSD, (6) the asymmetry information for each carbon atom in the canonical TSD, and (7) the information about its chirality.

In this section we discuss the "name" for the molecule which will be constructed from this information. The name discussed below will consist of several sections, the most important parts of it being the "canonical SLING" and the "parity bitstring". These parts describe the constitution and stereochemistry of the molecule canonically. Other parts of the name will be the parity vectors, the CE class numbers, the SE class numbers, the "asymmetry bitstring", and the "chirality bit". Other desirable features (such as ring structure descriptions) can be easily appended to the linear name thus generated, as and when the situation demands. If similar molecules have similar names, it is easier to identify the similarity (see Randic[16]).

The canonical SLING is generated by traversing the canonical TSD in an orderly, depth-first-search fashion. The SLING thus obtained will describe the constitution of the molecule linearly and canonically. The technique of generating the SLING from a TSD is rather straightforward and will not be dealt with in detail. Examples will be discussed later on in this report. Further information may be obtained from Agarwal[12] and Boivie.[9]

The canonical parity vector has been described earlier to be a "correction" to the canonical TSD (since the rows of the canonical TSD were sorted into decreasing order, thus destroying the stereochemical information contained therein). The parity bitstring is thus simply a coding of this parity vector. If the parity of the *i*th carbon in the canonical SLING is a −1 (as determined from the canonical parity vector), the *i*th position of the parity bitstring will hold a 0. This signifies that the stereochemistry of the *i*th carbon is falsely represented by the canonical SLING and must be inverted to restore the correct stereochemistry for that carbon. Similarly, a 1 in the *i*th position of the parity bitstring indicates that the stereochemical information contained in the canonical SLING for the *i*th carbon is correctly represented.

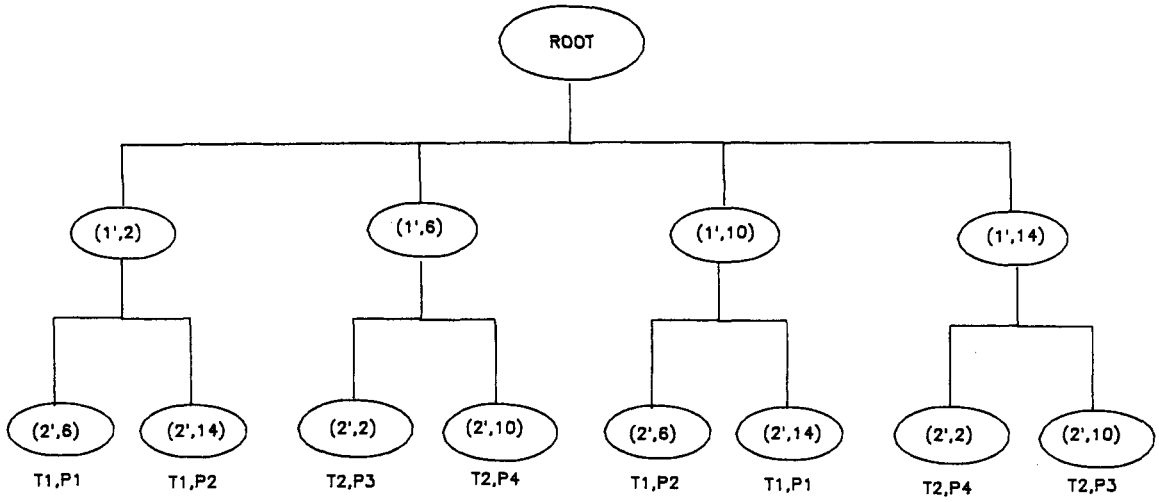
Note that for an olefinic pair of carbons, at most one entry in the parity bitstring needs to be marked with a FALSE (0) parity bit.

The CE information is translated into a CE string. This string lists the atoms in the canonical SLING which are constitutionally equivalent to each other. For example, if atoms 5 and 8 of the canonical SLING are CE and 7, 9, and 12 are also CE, the CE string will be 5,8,,7,9,12.

Note that CE atoms appear consecutively and distinct classes are separated by double commas. SE strings are constructed similarly.

The asymmetry bitstring consists of 1's and 0's. A 1 in position *i* indicates that the *i*th atom in the canonical SLING is asymmetric. If this carbon were symmetric, the *i*th position of the asymmetry bitstring will hold a 0.

The chirality of the molecule can be represented by one bit of information.



THE ASI–TREE FOR ALL ASI ASSIGNMENTS OF EXAMPLE II

T1:

Node	Atom	Up	Down	Left	Right
1	C	13:1	9:1	6:1	5:1
2	C	14:1	10:1	7:1	5:1
3	C	15:1	11:1	8:1	6:1
4	C	16:1	12:1	8:1	7:1
5	N	6:1	2:1	1:1	
6	N	5:1	3:1	1:1	
7	N	8:1	4:1	2:1	
8	N	7:1	4:1	3:1	
9	F	1:1			
10	F	2:1			
11	F	3:1			
12	F	4:1			
13	H	1:1			
14	H	2:1			
15	H	3:1			
16	H	4:1			

T2:

Node	Atom	Up	Down	Left	Right
1	C	13:1	9:1	6:1	5:1
2	C	14:1	10:1	7:1	5:1
3	C	15:1	11:1	8:1	6:1
4	C	16:1	12:1	8:1	7:1
5	N	7:1	2:1	1:1	
6	N	8:1	3:1	1:1	
7	N	5:1	4:1	2:1	
8	N	6:1	4:1	3:1	
9	F	1:1			
10	F	2:1			
11	F	3:1			
12	F	4:1			
13	H	1:1			
14	H	2:1			
15	H	3:1			
16	H	4:1			

For T1: P1: (+1,-1,+1,-1)
P2: (-1,+1,-1,+1)

For T2: P3: (-1,+1,-1,+1)
P4: (+1,-1,+1,-1)

Figure 26.

The complete name for the molecule is obtained by concatenating the canonical SLING, the parity bitstring, the CE string, the SE string, the asymmetry bitstring, and the chirality bit separated by suitable delimiters.

PROPERTIES OF THE NAME

Properties of a name can be classified as intramolecular and intermolecular. Intramolecular properties pertain only

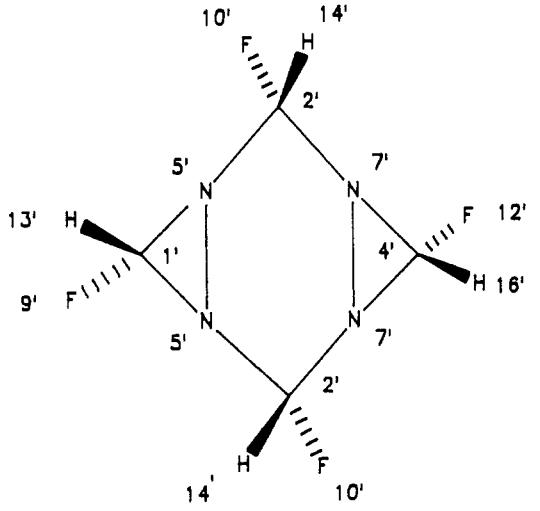


Figure 27.

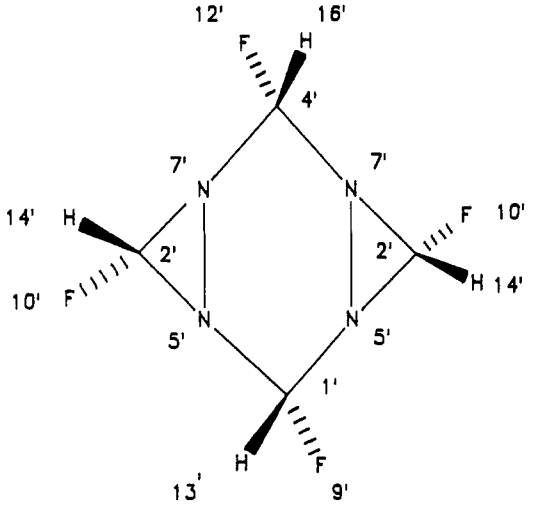


Figure 28.

to the given molecule. For example, chirality, equivalence between atoms, etc., can be determined entirely from the given molecule. Intermolecular properties relate two or more molecules. Given two molecules, we would like to determine (a) whether they are constitutionally identical but stereochemically distinct; (b) whether they are enantiomers, i.e., mirror images of each other; and (c) what is the least number of asymmetric carbons that must be inverted to convert one molecule into the other. There are other properties (such as substructure information) which we have not taken into consideration for our canonical name. However, we discuss the properties which our canonical names afford.

The nine molecules of Figure 32 were input as SLINGs to our notational algorithms. They represent all possible diastereomers for hexafluorocyclohexane. The SLINGs were written by traversing the bonds of the molecules according to the order shown in Figure 33. The numbers on the skeleton correspond to the positions of the atoms in the SLING. Atom 0, 5, 8, 11, 14, and 17 of the SLINGs for the molecules are all on the same side of the ring. The input SLINGs for the molecules are shown in Figure 34. For each molecule 12 ASI numberings were generated by the algorithm. The canonical SLING obtained in each case was CH-1F-1CF-1H-1CF-1H-1CH-1F-1CH-1F-1CH-1F-1/0. The structure which corresponds to this SLING is shown in Figure 35. The canonical parity bitstrings for the nine molecules are

(a) 111000	(d) 110001	(g) 101101
(b) 111001	(e) 111101	(h) 110011
(c) 111011	(f) 111111	(i) 110101

Using the interpretation that a 0 implies a FALSE representation of the stereochemistry, we see that the parity bitstring for molecule a suggests that the carbons 9, 12, and 15 of Figure 35 should be inverted. This indeed reconstructs molecule a of Figure 32. The molecule f of Figure 32 is fully represented by just the canonical SLING, and its parity bitstring contains only 1's.

The CE information for all nine molecules turns out to be identical, as it should be, since all are diastereomers. The CE string for each case is 0,3,6,9,12,15, where atoms 0, 3, 6, 9, 12, and 15 of the canonical SLING are all CE. Note that the CE string can be extended to list non-carbon CE classes also.

For the nine molecules, the SE strings obtained were as follows:

(a) 0,3,6,9,12,15	(d) 0,9,,6,15,,3,12	(g) 0,3,6,9,12,15
(b) no SEBs	(e) no SEBs	(h) 0,9,,3,6,,12,15
(c) no SEBs	(f) 0,15,,3,12,,6,9	(i) 0,3,,6,15,,9,12

where "no SEBs" stands for "no stereochemically equivalent brothers".

These can be verified against the structures shown in Figure 32.

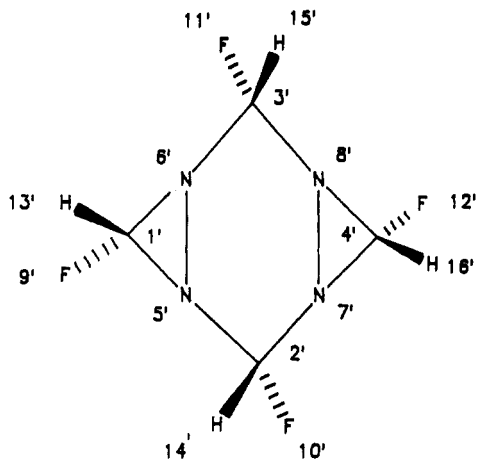
In each of the nine cases the asymmetry bitstring is 111111, indicating that each of the six carbons is asymmetric.

The parity vectors obtained by the notational algorithms for the nine cases are

(a)	(1)	(+1,−1,+1,−1,+1,−1)	(e)	(1)	(+1,+1,−1,−1,+1,−1)
	(2)	(−1,+1,−1,+1,−1,+1)		(2)	(−1,−1,+1,+1,−1,+1)
(b)	(1)	(+1,−1,+1,−1,−1,−1)		(3)	(+1,+1,−1,−1,−1,+1)
	(2)	(−1,+1,−1,−1,−1,+1)		(4)	(−1,−1,+1,−1,−1,−1)
	(3)	(−1,+1,−1,+1,−1,−1)		(5)	(−1,−1,−1,+1,−1,−1)
	(4)	(+1,−1,+1,−1,+1,+1)		(6)	(+1,−1,−1,−1,+1,+1)
	(5)	(−1,+1,−1,+1,+1,+1)		(7)	(−1,+1,−1,−1,+1,+1)
	(6)	(+1,−1,+1,+1,+1,−1)		(8)	(+1,−1,+1,+1,−1,−1)
	(7)	(−1,+1,+1,+1,−1,+1)		(9)	(−1,+1,+1,+1,−1,−1)
	(8)	(+1,+1,+1,−1,+1,−1)		(10	(+1,+1,+1,−1,+1,+1)
	(9)	(+1,+1,−1,+1,−1,+1)		(11)	(+1,+1,−1,+1,+1,+1)
	(10)	(−1,−1,+1,−1,+1,−1)		(12)	(−1,−1,+1,+1,+1,−1)
	(11)	(−1,−1,−1,+1,−1,+1)	(f)	(1)	(−1,+1,−1,−1,+1,−1)
	(12	(+1,−1,−1,−1,+1,−1)		(2)	(+1,−1,+1,+1,−1,+1)
(c)	(1)	(+1,−1,−1,−1,−1,−1)		(3)	(+1,−1,−1,−1,−1,+1)
	(2)	(−1,−1,−1,−1,−1,+1)		(4)	(−1,−1,−1,−1,−1,−1)
	(3)	(−1,+1,−1,−1,−1,−1)		(5)	(−1,+1,+1,+1,+1,−1)
	(4)	(+1,−1,+1,−1,−1,+1)		(6)	(+1,+1,+1,+1,+1,+1)
	(5)	(−1,+1,−1,+1,+1,−1)	(g)	(1)	(−1,−1,+1,+1,−1,−1)
	(6)	(+1,−1,+1,+1,+1,+1)		(2)	(+1,+1,−1,−1,+1,+1)
	(7)	(−1,+1,+1,+1,+1,+1)	(h)	(1)	(−1,−1,+1,−1,−1,+1)
	(8)	(+1,+1,+1,+1,+1,−1)		(2)	(+1,+1,−1,−1,−1,−1)
	(9)	(+1,+1,+1,+1,−1,+1)		(3)	(−1,−1,−1,+1,+1,−1)
	(10)	(−1,+1,+1,−1,+1,−1)		(4)	(+1,−1,−1,+1,+1,+1)
	(11)	(+1,−1,−1,+1,−1,+1)		(5)	(−1,+1,+1,−1,+1,+1)
	(12)	(−1,−1,−1,−1,+1,−1)		(6)	(+1,+1,+1,+1,−1,−1)
(d)	(1)	(−1,−1,+1,−1,+1,+1)	(i)	(1)	(−1,−1,−1,−1,+1,+1)
	(2)	(+1,+1,−1,+1,−1,−1)		(2)	(+1,−1,−1,+1,−1,−1)
	(3)	(−1,−1,−1,+1,+1,+1)		(3)	(−1,+1,+1,−1,−1,−1)
	(4)	(+1,−1,−1,+1,+1,−1)		(4)	(+1,+1,+1,−1,−1,+1)
	(5)	(−1,+1,+1,−1,−1,+1)		(5)	(+1,+1,−1,+1,+1,−1)
	(6)	(+1,+1,+1,−1,−1,−1)		(6)	(−1,−1,+1,+1,+1,+1)

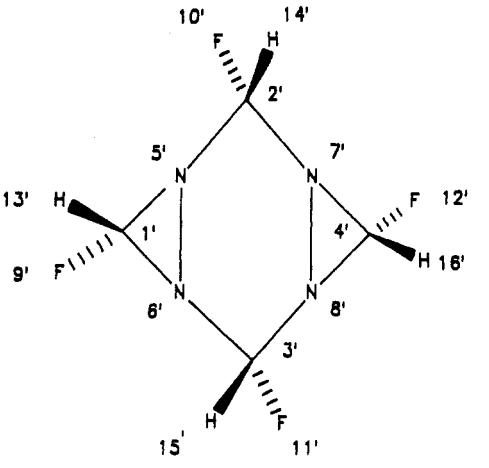
Note that since there are six centers of asymmetry, we have a total of $2^6 = 64$ parity vectors, each of which must belong to one of the molecules. Hence, the parity vectors are divided into equivalence classes.

Since the inverse of (+1,−1,+1,−1,+1,−1) is (−1,+1,−1,+1,−1,+1) and since both parity vectors belong to molecule a, it



RESULTING DESCRIPTORS — T1,P1

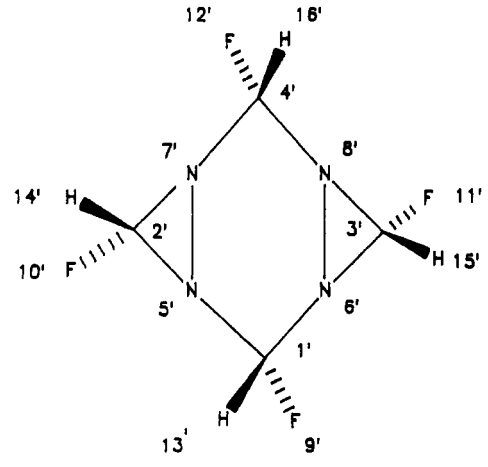
(a)



RESULTING DESCRIPTORS — T1,P2

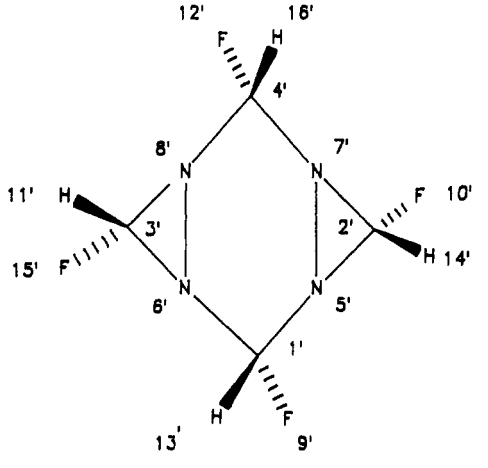
(b)

Figure 29.



RESULTING DESCRIPTORS — T2,P3

(a)



RESULTING DESCRIPTORS — T2,P4

(b)

Figure 30.

Original atom number	Atom type	Min. ASI value assigned	Corresponding Min. TSD	Corresponding Min. parity vector	CE class No.	SE class No.	Asymetric center?
1	H	13′	T1	P2	13	13	
2	C	1′	T1	P2	1	1	Yes
3	F	9′	T1	P2	9	9	
4	N	5′	T1	P2	5	5	
5	N	5′	T1	P1	5	7	
6	C	1′	T2	P3	3	3	Yes
7	H	13′	T2	P3	15	15	
8	F	9′	T2	P3	11	11	
9	N	5′	T1	P2	5	5	
10	C	1′	T1	P2	1	1	Yes
11	H	13′	T1	P2	13	13	
12	F	9′	T1	P2	9	9	
13	N	5′	T1	P1	5	7	
14	C	1′	T2	P3	3	3	Yes
15	H	13′	T2	P3	15	15	
16	F	9′	T2	P3	11	11	

Note: (a) The canonical TSD is T1.
(b) The canonical parity vector is P2.

Figure 31.

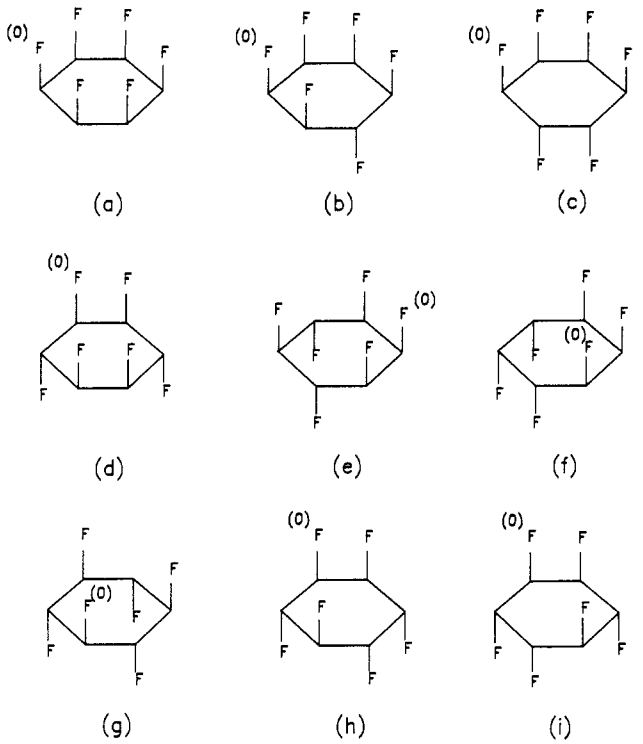
is achiral. For a similar reason, molecules b through g are also achiral. Molecules h and i are mirror images of one another. This follows from the fact that the inverse of any parity vector for molecule h can be found among the set of parity vectors for molecule i, and vice versa.

Since the parity vector (+1,−1,+1,−1,+1,−1) for molecule a differs from (+1,−1,+1,−1,−1,−1), a parity vector of molecule b, in only one place, only one carbon atom in molecule a needs inversion to transform it into molecule b. Molecules h and i have the "least distance" of 2, as is seen by comparing the vector (−1,−1,+1,−1,−1,+1) of molecule h with vector (−1,−1,−1,−1,+1,+1) of molecule i. Other vectors of molecule i differ in two or more places also.

Proofs of correctness of the notational algorithms and the properties listed above can be found in Agarwal.[12]

ANALYSIS OF THE NOTATIONAL ALGORITHMS

The efficiency of the notational algorithms depends mainly on the number of ASI assignments produced by the CANONIZER for the given molecule. This number is larger than or equal to the total number of automorphisms (1–1, onto mappings, without considering stereochemistry) of the molecule to itself. Because of the DIFFERENTIATOR algorithm, in most cases, the number of ASI assignments equals



Note: The starting nodes in constructing the stereo-SLINGS in Figure (34) using the skeleton of Figure (33) are labelled (0) in the above molecules.

Figure 32.

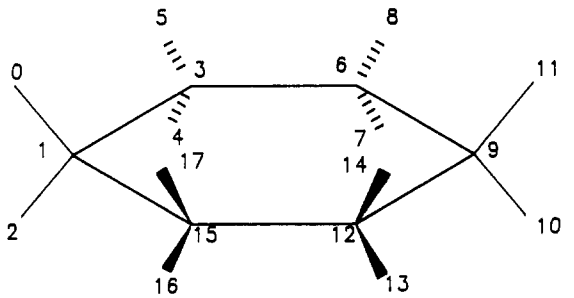


Figure 33.

```
(a) FCH-1CH-1F-1CH-1F-1CH-1F-1CH-1F-1CH-1F-1/1
(b) FCH-1CH-1F-1CH-1F-1CH-1F-1CF-1H-1CH-1F-1/1
(c) FCH-1CH-1F-1CH-1F-1CH-1F-1CF-1H-1CF-1H-1/1
(d) FCH-1CH-1F-1CF-1H-1CH-1F-1CH-1F-1CF-1H-1/1
(e) FCH-1CH-1F-1CF-1H-1CH-1F-1CF-1H-1CH-1F-1/1
(f) FCH-1CF-1H-1CF-1H-1CF-1H-1CH-1F-1CH-1F-1/1
(g) FCH-1CF-1H-1CH-1F-1CF-1H-1CH-1F-1CF-1H-1/1
(h) FCH-1CH-1F-1CF-1H-1CF-1H-1CH-1F-1CF-1H-1/1
(i) FCH-1CH-1F-1CF-1H-1CH-1F-1CF-1H-1CF-1H-1/1
                ↑       ↑       ↑       ↑       ↑       ↑
Position ——→ 0       5       8       11      14      17
in SLING
```

Figure 34.

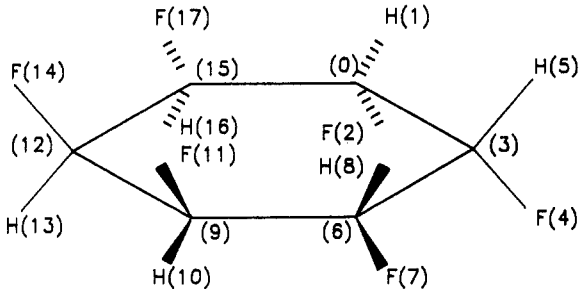


Figure 35.

the number of automorphisms. Hence, the more "symmetric" the structure of the molecule, the larger the time taken by the notational algorithms to develop the name for the molecule.

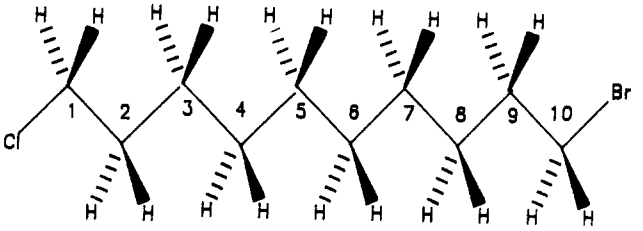


Figure 36.

Node	Atom	Up	Down	Left	Right
1	C	Cl:1	H:1	H:1	2:1
2	C	1:1	H:1	H:1	3:1
3	C	2:1	H:1	H:1	4:1
4	C	3:1	H:1	H:1	5:1
5	C	4:1	H:1	H:1	6:1
6	C	5:1	H:1	H:1	7:1
7	C	6:1	H:1	H:1	8:1
8	C	7:1	H:1	H:1	9:1
9	C	8:1	H:1	H:1	10:1
10	C	9:1	H:1	H:1	Br:1

Figure 37.

Molecules which have several monovalent atoms of the same type can have a large number of automorphisms. Consider, for example, the molecule of Figure 36. Although, there are no CE carbons in this molecule, there are several other CE atoms. In fact, there are 1024 ($=2^{10}$) automorphisms of the molecule. The CANONIZER will have to generate 1024 ASI assignments, a prohibitively large number, to generate a canonical TSD for this molecule. However, this problem can be rectified simply by "compacting" the input TSD for the CANONIZER. One such compacted TSD for the given molecule is shown in Figure 37. Using this TSD, the CANONIZER has to generate only one ASI assignment for the remaining atoms. As output we obtain a canonical TSD and CE and SE class information only about the 10 carbon atoms. However, it is a simple matter to generate the CE and SE class information for the monovalent atoms.

Note that certain TSDs (such as a TSD for HCl) cannot be compacted.

Another point to be noted is that the number of ASI assignments can be reduced further if the stereochemistry information is taken into consideration by the CANONIZER while it is generating the ASI assignments. However, a serious drawback of doing this is that the CANONIZER will be unable to discover the CE and SE classes and the asymmetry and chirality information. Also, diastereomers will not be identifiable because the constitutional and stereochemical information will be merged into one common name.

CONCLUSIONS

The notational algorithms described in this paper have been in use for several years as a part of SYNCHEM2, a noninteractive organic synthesis-planning computer program which works retrosynthetically and includes stereochemistry of the molecules and chemical transforms (see Gelernter[8] for more details). All notational programs are written in PL/I and have been found to be quite satisfactory for all our needs. The main advantage provided by our system is that it permits bidirectional communication between humans and computers. The linear name produced has been shown to be canonical. The first part of the name represents the constitution of the molecule and can be used independently of the second part if stereochemistry need not be taken into consideration. The second part represents the stereochemistry of the molecule and can be used in conjunction with any standard nomenclature system. Two molecules that are constitutionally alike but stereochemically distinct (diastereomers) have identical constitutional representations but different stereochemical de-

scriptors and can be so identified by matching the first parts of their names. They can also be distinguished from each other because their stereochemical descriptors are different. Atoms within a single molecule are identified with their constitutionally equivalent counterparts. Similarly, atoms within a single molecule are identified with their stereochemically equivalent counterparts. Generalized Huckel-resonant substructures of the molecule are identified by the system, allowing different resonant forms of the same molecule to be given identical names, a feature rarely found in other algorithmic notational systems. The program accepts as input an easy to write stereochemical descriptor of the molecule which is a noncanonical form of the canonical name. Molecules which are mirror images of one another (enantiomers) are identifiable from their canonical names by comparing their parity vectors. This task is difficult to carry out algorithmically in most other notational systems. The smallest number of asymmetric carbons at which two diastereomers differ can readily be computed from their names in our system. All the properties discussed above have been incorporated into the programs, and the algorithms used by our programs have been shown to be mathematically correct. Few systems offer the depth, breadth, and algorithmic correctness (not to mention the ease of bidirectional human–machine communication) that our system does. Further development of the algorithms can be done in the areas of improving their efficiency and incorporating higher-order three-dimensional properties of molecules along the lines discussed by Cahn, Ingold, and Prelog.[6]

ACKNOWLEDGMENT

We gratefully acknowledge the contributions made by R. H. Boivie, H. W. Davis, and J. E. Searleman. Professor H. L. Gelernter originated the SYNCHEM and SYNCHEM2 projects at the State University of New York at Stony Brook for automating the synthetic planning process for organic molecules. We thank the various private and governmental organizations that have provided support for this work.

REFERENCES AND NOTES

(1) Smith, E. G. *The Wiswesser Line-Formula Notation*; McGraw-Hill: New York 1968.
(2) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures. *J. Chem. Doc.* **1965**, *5*, 105–113.
(3) Wipke, W. T.; Dyott, T. M. Stereochemically Unique Naming Algorithm. *J. Am. Chem. Soc.* **1974**, *96*.
(4) Cahn, R. S.; Ingold, C. K. Specification of Configuration about quadricovalent Asymmetric Atoms. *J. Chem. Soc.* **1951**, *612*.
(5) Cahn, R. S.; Ingold, C. K.; Prelog, V. The specification of Asymmetric Configuration in Organic Chemistry. *Experientia* **1956**, *12*.
(6) Cahn, R. S.; Ingold, C. K.; Prelog, V. Specification of Molecular Chirality. *Angew. Chem. Int. Ed. Engl.* **1966**, *5*, 385–415.
(7) SLINGs, without their stereochemistry, are similar to the linear representation called SMILES, a simple discussion of which can be found in the following: *Chem. Eng. News* **1992**, *70* (12), 17–19.
(8) Gelernter, H. L.; et al. Empirical Explorations with SYNCHEM2. *Science* **1977**, *197*, 1041–1049.
(9) Boivie, R. H. Ph.D. Thesis, Department of Computer Science, State University of New York at Stony Brook, 1977.
(10) Ugi, I.; Marquarding, D.; Klusacek, H.; Gokel, G.; Gillespie, P. Chemistry and Logical Structures. *Angew. Chem. Int. Ed. Engl.* **1970**, *9*.
(11) Davis, H. Master's Thesis, Department of Computer Science, State University of New York at Stony Brook, 1974.
(12) Agarwal, K. K. Ph. D. Thesis, Department of Computer Science, State University of New York at Stony Brook, 1976.
(13) Rucker, G.; Rucker, C. Computer Perception of Constitutional (Topological) Symmetry. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 187–191.
(14) Figueras, J. Automorphism and Equivalence Classes. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 153–157.
(15) Randic, M. On Canonical Numbering of Atoms in a Molecule and Graph Isomorphism. *J. Chem. Inf. Comput. Sci.* **1977**, *17*.
(16) Randic, M. Representation of Molecular Graphs by Basic Graphs. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 57–69.
(17) Raymond, K. W. A LISP Program for the Generation of IUPAC Names from Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 270–274.
(18) Wisniewski, J. L. AUTONOM: System for Computer Translation of Structural Diagrams into IUPAC-Compatible Names. 1. General Design. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 324–332.
(19) Goebels, L.; Lawson, A. J.; Wisniewski, J. L. AUTONOM: System for Computer Translation of Structural Diagrams into IUPAC-Compatible Names. 2. Nomenclature of Chains and Rings. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 216–225.
(20) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 1. Introduction and Background to a Grammar-Based Approach. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 101–105.
(21) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 2. Development of a Formal Grammar. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 106–112.
(22) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 3. Syntax Analysis and Semantic Processing. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 112–118.
(23) Cooke-Fox, D. I.; Kirby, G. H.; Lord, J. D.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 4. Concise Connection Tables to Structure Diagrams. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 122–127.
(24) Cooke-Fox, D. I.; Kirby, G. H.; Lord, M. R.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 5. Steroid Nomenclature. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 128–132.
(25) Kirby, G. H.; Lord, M. R.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 6. (Semi)-automatic Name Correction. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 153–160.
(26) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.
(27) Weininger, D. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97–101.
(28) Weininger, D. SMILES. 3. Depict. Graphical Depiction of Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 237–243.