Computer Coding of Configuration[†]

John Figueras
399 Baker's Pond Road, Orleans, Massachusetts 02653

Received August 25, 1995[⊗]

A program is described that accepts graphics input to a computer of conventional 2D representations of stereoisomers and makes configurational assignments. The present version of the program handles asymmetric carbon, sulfur, and phosphorus atoms and isomerism at double bonds. Fischer projections, ball-and-stick diagrams using bold and broken directional bonds, and conventional planar diagrams for isomerism at double bonds may be used as input.

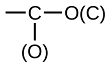
INTRODUCTION

The aim of this work is extension of the recognition capabilities of graphics-based structure input programs to accommodate stereoisomers, both asymmetric atoms and *cis*−*trans* double bonds. The approach is that used by Cahn−Ingold−Prelog (CIP) in which priorities of substituents at a stereogenic atom are induced through a series of rules, and assignment of configuration at the stereo atom is based upon the sequence of substituents taken in order of priority. Razinger and Perdih[1a] approached the problem of assigning configurations starting from a canonically-labeled graph using the canonicalization procedure of Shelley and Munk[1b] to isolate topologically nonequivalent atoms and applying parity vectors to those atoms. A similar approach was taken by Contreras[2] *et al.* in which N-tuple atom descriptors were extended to incorporate chirality. In both of these papers, the principal focus was design of a stereoisomer generator that produced an exhaustive set of stereoisomers and included an algorithm for classifying the output from the generator. The goal of my work described here is the assignment of configurations to asymmetric atoms, double bonds, and substituted rings using conventional Fischer projections or D2.5 drawings (with directional bold and broken bonds) or explicit drawings of cis−trans olefins as computer input: that is, the user may draw a conventional 2D representation of a stereoisomer; the program will examine the representation, isolate the stereogenic centers, and assign configurations **R** or **S** in the case of asymmetric atoms, **r** or **s** in the case of pseudoasymmetric atoms, and **E** or **Z** in the case of olefinic bonds. In this regard it is quite different in aim from the aforementioned works, requiring analysis of drawings created by a user's graphics input. A routine of this type has applications to programs that deal with C13 assignments, organic synthesis, and information retrieval.

The graphics input routine used in this work is a mouse-driven program developed for Macintosh and MS-DOS computers and provides the usual facilities for creating rings, drawing connected atoms, modifying bond types, introducing heteroatoms and substituents, etc. The routine produces a connection table and the atom coordinates required for making configuration assignments.

SUBSTITUENT PRIORITIES

The procedure for making configuration assignments to stereogenic atoms requires the assignment of priorities to atoms attached to the stereogenic center. CIP priorities are based strictly on atomic number. If two attached atoms have the same atomic number, the procedure moves to the next outer shells of atoms, finds the largest atomic number in each shell, and terminates if the atomic numbers are unequal (thus allowing priority assignment); otherwise it moves to the next shells and iterates the process. To maintain comparisons based only on atomic number, various subterfuges are invoked in the CIP treatment of ring substituents, unsaturated groups, and aromatic rings. For example, "phantom" atoms are used to saturate multiple bonds, so that $-C{=}O$ becomes



If we compare $-CH_2OH$ with this expanded form of $-C{=}O$, it is evident that the carbonyl group has higher priority than the hydroxymethyl group based on the following argument: We start with the first comparison and equal atomic numbers (carbon atoms). We move to the next shell, and again atomic numbers match (oxygen atoms). We proceed to the next shell and now encounter a mismatch, H (in OH) and the (phantom) C. Since C has higher atomic number than H, $-C{=}O$ has higher priority than $-CH_2OH$. This example suggests that priority increases with unsaturation and that use of bond patterns in addition to atomic numbers for establishing priorities offers a more convenient way of handling multiple bonds than the CIP use of phantom atoms, providing a method requiring no structure modifications and one much more easily implemented in a computer program. To this end, a table of codes (Table 1) is used to describe bond type at each atom. These codes and atomic number are merged into a composite **atomCode** given by the formula

$$\text{atomCode} = 100^{*}\text{atomic no.} + 10^{*}\text{bondCode} + \text{non-hydrogen degree}$$

Non-hydrogen degree contains information related to the degree of substitution at an atom, distinguishing, for example, oxygen in an OH group (atomCode 801) from oxygen in a methoxy group (atomCode 802), thus assigning higher priority to the methoxy group, consonant with CIP ranking. AtomCodes were designed to produce the same priorities as

[†] Dedicated to Prof. Nelson J. Leonard on the occasion of his eightieth year.

[⊗] Abstract published in *Advance ACS Abstracts,* February 15, 1996.

0095-2338/96/1636-0491$12.00/0 © 1996 American Chemical Society

Table 1. Bond Codes for Priority Assignment

bond type	bond count	bond code
single	n^a	0
double	1	1
double	2	2
triple	1	3
aromatic	2	4
aromatic	3	5

[a] n is the degree of the atom. Bold and broken bonds are counted as single bonds.

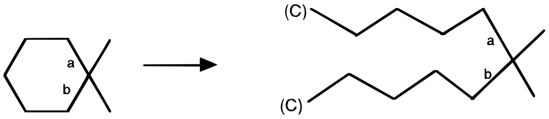


Figure 1. Treatment of rings in the CIP procedure. Each bond a and b is opened to produce a chain terminated with phantom carbon atoms. Priority assignment then proceeds normally as in the case of acyclic chains.

the CIP procedure, but minor deviations are expected for certain structures, as described later

An example dealing with priority assignment for two alkane chains of different length introduces a new consideration:

$$-CH_2CH_2CH_2CH_3 \text{ vs } -CH_2CH_2CH_3$$

In this example, the first three atoms match for atomic number. Thereafter, the residual carbon atom in the butyl group must be paired with terminal hydrogen in the propyl group; this establishes priority for butyl over propyl. This example suggests that, all other factors being equal, longer groups should have priority over shorter groups. Group length appears in the computer program as a counter, a variable named **level**, that is incremented when processing moves to the next outer shell of atoms.

The CIP procedure was originally intended for manual assignment of configurations. It has been suggested that CIP does not readily lend itself to reduction to a computer program;[1a] however, a complete implementation of the CIP *method*—including use of phantom atoms and virtual replacement of multiple bonds with multiple single bonds—applied to acyclic structures is described in the Contreras paper.[2] The use of phantom atoms to treat unsaturation is, however, an unsatisfactory complication in CIP. Equally unhappy is the treatment of stereogenic ring atoms; each ring bond at the stereogenic atom is broken, and a phantom atom is added to the end of each chain. In this way, the ring is opened and replaced by two chains, one chain containing the atoms encountered in a clockwise tour of the ring, the other containing the results of a counterclockwise tour (Figure 1). The two chains are compared until atoms differing in atomic number are encountered, or the end of chains is reached. In the computer program, independent tours are launched, starting at each atom attached to the stereogenic central atom, and comparisons are pursued until the ends of the tours are reached (or atoms differing in atomCodes are encountered). For ordinary substituents (if no differentiation based on atomCodes occurs), an end of a tour is a terminal atom (no atoms are left in the chain); for a ring, an end of a tour is reencounter with the central atom.

To begin ascertaining priorities, the atomCodes for atoms directly attached to the stereogenic center are collected. If the set of atomCodes is nonredundant, the program proceeds to configuration assignment. If the set of atomCodes is redundant, the redundant subset is isolated, and the next outer shell of atoms attached to each atom in the redundant subset is extracted. The maximum atomCode associated with each shell is determined; if the set of maximum atomCodes is redundant, the redundant subset is once again extracted, and the process is repeated with the next outer shell until a nonredundant subset is obtained or the supply of attached atoms runs out.

A counter variable, **level**, is incremented when the program advances to the next shell. The counter measures the length of the chain of atoms at each stage. When searching has been completed, associated with each atom in the original redundant set will be an atomCode and value for **level**. Priority within the redundant set is determined first by sorting on atomCode in decreasing order, then within redundant atomCodes sorting on **level** in decreasing order**.** If the combinations of atomCodes and levels are nonredundant, the program proceeds to configuration assignment. If all candidate atoms have been explored and codes and levels are still redundant, no configurational assignment can be made because of the presence of at least two identical substituents.

Implementation of the above steps in Pascal makes extensive use of the Pascal *set* data structure for exploring successive atom shells in the search for a nonredundant set of atomCode descriptors. The following is a pseudo-code outline of the Pascal procedure, **GetPriorities**. Let **S** be the index of the stereogenic atom and let **n** be the degree of **S**, including hydrogen atoms. For asymmetric carbon atoms, **n** would have a value[4] of 4; for asymmetric sulfur or nitrogen, **n** would have a value of 3; for isolated C−C or C−N double bonds, **n** would have a value of 2; for isolated N−N double bonds, **n** would be a degenerate case with a value of 1 (the double bond is excluded from the degree counts in the last two cases). It is assumed that atomCodes have been assigned to all atoms before **GetPriorities** is called.

(1) Associate a set, **currSet[i]**, $i = 1..n$, with each of the **n** atoms attached to **S**. Initially each **currSet[i]** contains the index of a neighbor to **S**. Associate an array **priorityList[i]**, $i = 1..n$, which contains the indices of the **n** neighbors to **S**. Associate a record, **codeList[i]**, $i = 1..n$, with each of the **n** atoms attached to **S**. Initially, each **codeList[i]** contains the atomCode of the *i*th neighbor to **S** and the current value for **level** (= 1).

(2) Sort **codeList** in decreasing order of atomCodes, rearranging the corresponding order of **priorityList** and **currSet** during the sort.

(3) Examine **codeList** for atomCode collisions. If there are none, exit the procedure, returning **priorityList** to the calling program.

(4) Isolate the subset of **codeList** containing redundant atomCodes. Because the list is sorted, this is easily done by finding the limits inside the array at which the redundant set starts (a variable named **start**) and ends (a variable named **finis**), since the redundant codes will be contiguous owing to the sort. Subsequent processing can then be carried out between the limits **start** and **finis**.

(5) For each **currSet[i]**, **i = start**..**finis**, create a new set, **nextSet[i]**, containing the atoms (indices) attached to each element in **currSet[i]. NextSet[i]** contains the next outer shell associated with the atom recorded in **priorityList[i]**.

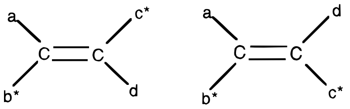


Figure 2. *Cis*−*trans* isomerism is assigned relative to the spatial distribution of atoms of highest priority, indicated as starred atoms.

(6) Copy **nextSet** into **currucet**. For each **currSet[i]** (**i** = **start**..**finis**), find the maximum atomCode associated with the atoms in **currSet[i]**; record the maximum atomCode in **codeList[i]** and increment the variable **level** in **codeList[i]**.

(7) Sort **codeList** in decreasing order of atomCodes, rearranging the corresponding order of **priorityList** and **currSet** during the sort. If there are any redundant codes in **codeList** after sorting, do a nested sort of the redundant codes on **level**.

(8) If there are any items in **codeList** that are redundant with respect to both atomCode and **level**, select the subset containing the redundant **codeList** records (i.e., recompute **start** and **finis**, the bounds of the subset) and return to step 5, else exit and return **priorityList** to the calling program.

In order to keep the preceding description uncongested, I have omitted discussion of one important variable, an array named **Parent** that keeps track of the parent of each element in **currSet**; if **i** is in **currSet**, then **Parent[i]** contains the index of the atom in the previous shell to which **i** is attached. The **Parent** array is used to prevent backward references from appearing in **nextSet** when advancing to the next shell of atoms (step 5). The **Parent** array is updated whenever a new element is added to **nextStep** in step 5.

As the very last step, **GetPriorities** scans **codeList** and returns a boolean flag **OK** = TRUE if there are no collisions, otherwise it sets this flag to FALSE.

cis−*trans* ISOMERS AT A DOUBLE BOND

The routine **getPriorities** is applied at each end of an isolated double bond to obtain two substituents of highest priority, one at each end of the double bond. Assignment of configuration (**Z** or **E**) is based upon the spatial relationship of the two high-priority substituents; namely, these two substituents are farther apart in the *trans* isomer than in the *cis* isomer, Figure 2. Parenthetically, this relationship can exist only for structures that are reasonably drawn, requiring the user to show some common sense in building structures; the structure entry program can assist here by creating bonds to a standard length. The program is not restricted to carbon−carbon double bonds; it will make **Z**, **E** assignments to carbon−nitrogen and nitrogen−nitrogen double bonds (*syn*−*anti* isomerism). The designation of *cis* or *trans* is usually applied to the whole structure, which is not practical for computer application based on connection tables. Therefore, the program associates the **Z**, **E** assignments with each of the two atoms participating in the double bond.

Although it is simple to determine computationally which structure is *cis* and which is *trans* when one has representation of both isomers on hand, it is not so simple when one deals with a single isomer. The procedure about to be described uses the carbon−carbon double bond distance as a reference for determining the spatial relationship between the key atoms. Substituent coordinates about one atom in the double bond are translated toward the other doubly-bonded atom by an amount equal to the double bond length. This is done for both key atoms at the double bond. Figure 3 portrays the consequence of this translation: the distance between the key substituents in the *cis* isomer will be less than or equal to the double bond length, whereas the distance between the key substituents in the *trans* isomer will be larger than the double bond length. A number of empirical trials were undertaken with large variations in bond angles at each end of the double bond; it was found that a distance parameter could be determined that satisfactorily differentiates between the two isomers in all reasonable drawings. This procedure not only permits discrimination between the two isomer types, but also, because it is based on *distance*, does so in a way that is independent of spatial orientation of the structure.

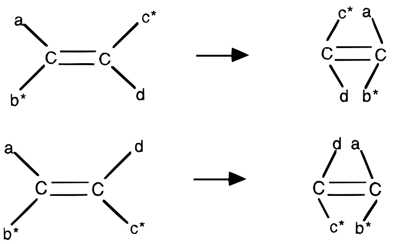


Figure 3. Mutual translation of substituents at a double bond exaggerates distances between key (starred) atoms relative to double-bond length, allowing facile identification of *cis* and *trans* relationships.

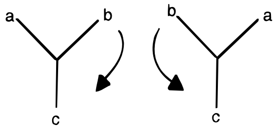


Figure 4. Sequences of substituents in order of decreasing priority, a > b > c. The substituents are viewed fom the side opposite the group of lowest priority. Clockwise sequence (left figure) is denoted by **R** and counterclockwise sequence (right figure) by **S**.

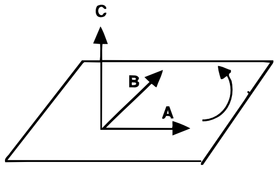


Figure 5. The right-hand rule. Vector **C** is the vector product of vectors **A** and **B**. **C** is normal to the plane containing **A** and **B**, and has a positive *z*-component for counterclockwise generation of the angle from **A** to **B**.

ASYMMETRIC CARBON ATOMS

The CIP procedure[2] assigns configuration at asymmetric carbon atoms by noting the rotation sequence—clockwise or counterclockwise—of substituents in order of decreasing priority, when viewed from the side opposite the substituent of lowest priority (Figure 4). We note that only two substituents (e.g., **a** and **b**, Figure 4) are needed to determine the direction of the sequence. A computationally convenient approach for assigning this direction is available from vector algebra. Consider two vectors **A** and **B** along the bonds joining substituents **a** and **b** to the stereogenic atom, Figure 5. The vector product **C** = **A** × **B** is a vector normal to the plane containing **A** and **B** and having a *z*-direction (up or down) depending upon the right-hand rule; i.e., if the angle subtended by vectors **A** and **B** is formed from **A** to **B** in counter-clockwise fashion (as in Figure 5), the vector **C** will be directed upwards, and conversely. We need compute only the component C_z in the z direction from the formula

$$C_z = A_xB_y - A_yB_x$$

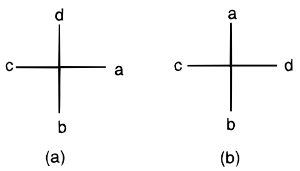


Figure 6. Fischer projections. The right−left pair is in front of the paper and the other pair behind the paper. Substituent priority is a > b > c > d. In part (a), substituent d with lowest priority is behind the plane; the clockwise sequence a, b, and c is an **R** configuration. In (b) the sequence is still clockwise (apparent **R**), but now the lowest priority substituent, d, is in front, so configuration must be reversed to **S**.

where A_x, A_y, B_x, and B_y are the components of the **A** and **B** vectors and note the sign of C_z. Normally, when **C** is directed upwards (counterclockwise rotation order from **A** to **B, S** configuration), C_z is positive, otherwise, for clockwise order C_z is negative (**R** configuration). Unfortunately, the coordinate system generally used for computer screen graphics is upside-down, with origin at the upper left corner of the screen and the *y*-coordinate increasing in the downward direction of the screen. The consequence of this is that the sign of the C_z component computed on the basis of screen graphics is inverted: a counterclockwise order from **A** to **B** (**S** configuration) generates a *negative* value for C_z, and a *positive* value for C_z is associated with clockwise order (**R** configuration). The components A_x, A_y, ... for computing C_z are easily obtained from the $x-y$-coordinates of the substituents and the stereogenic atom; for example, $A_x = a_x - S_x$ where a_x and S_x are the *x*-coordinates of substituent **a** and stereogenic atom S. The two substituents of highest priority are used for the **A** and **B** vectors unless they are collinear. For two collinear vectors, the plane containing vectors **A** and **B** is indeterminate, and vector **C** does not exist; in this case, the two substituents of next highest priority are used. Vector **A** always corresponds to the substituent of higher priority in the (**A**, **B**) pair.

The major problem in applying the foregoing to configuration is deducing the relationship between the computations in 3D to a structure represented in 2D. The Fischer projection in Figure 6a represents a general structure with substituents **a** and **c** in front of the plane and substituents **b** and **d** behind the plane of the paper. Since **d**, the substituent of lowest priority lies behind the plane, the clockwise order of decreasing priority from **a** to **b** to **c** determines directly the **R** configuration. It is instructive to interchange substituents **a** and **d** as in Figure 6b. A single interchange inverts configuration, so structure **6b** has **S** configuration. We observe, however, that substituents **a**, **b**, and **c**, in terms of their coordinates in the plane of the paper, remain in clockwise order, which implies that C_z has an incorrect sign; this is remedied by using a multiplier **Parity** = −1. The consequence of placing the lowest priority substituent to the right or left is that it now occupies a position in *front* of the plane of the remaining atoms; therefore, the sequence of groups—clockwise or counterclockwise—will be reversed when the substituents **a**, **b**, and **c** are viewed from the side opposite the **d** substituent. In the computer program, the position of **d** is tested, and **Parity** is assigned a value −1 for a left/right placement of **d**; otherwise **Parity** has a value +1.

The program handles asymmetric centers at trivalent sulfur and phosphorus (sulfoxides, phosphines) in much the same way. It is assumed that an unshared pair of electrons, playing the role of lowest priority substituent, occupies one vertex of a tetrahedron. A dummy, low priority atom is attached to the stereogenic center to force the three substituents to appear as three highest priorities, and priorities are determined. If no directed bonds are present, a Fischer projection is assumed in which two of the three substituents must be collinear. If they are collinear along the horizontal axis, therefore in front of the plane, the lowest priority electron pair substituent must be located behind the plane and the priority sequence (clockwise or counterclockwise) corresponds to **S** or **R**, directly; in this case, **Parity** = +1. On the other hand, if two of the three substituents are collinear vertically, and therefore behind the plane, then the lowest priority electron-pair must appear in front of the plane, and the priority sequence will now give incorrect configuration, which must be inverted: i.e., **Parity** = −1. A compact method is available for determining whether two substituents are collinear vertically: compare the *x*-coordinates for all possible pairs of the three substituents (easily and quickly done in two nested loops); if any pair of x-coordinates is equal, vertical collinearity has been found (set **Parity** = −1), else set **Parity** = +1.

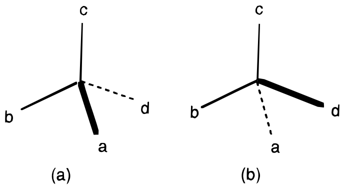


Figure 7. Directed bonds. Priorities a > b > c > d. In (a) the lowest priority substituent is behind the plane (broken bond); the clockwise sequence a, b, and c corresponds to **R** configuration. In (b) the lowest priority substituent is in front of the plane (bold bond); configuration deduced from the clockwise sequence as it appears on paper must be inverted (**S**).

Different considerations apply if the user elects to represent 3D arrangements using bold (or wedge) and broken bonds, in which the bold bond represents a substituent above (or in front of) the plane containing the remaining substituents and a broken bond represents a substituent below (or in back of) the same plane. In Figure 7a, atom **d** of lowest priority is attached to a broken bond and therefore lies behind the plane, and we view the sequence **a**, **b**, and **c** from the side opposite the lowest priority atom. This clockwise sequence therefore represents an **R** configuration with a negative value for C_z (in our upside-down coordinate system). If we append the lowest priority atom with a *bold* bond, Figure 7b, that atom now lies in front of the plane, and the configuration deduced from the sequence of substituents (clockwise as it appears on paper) must be inverted to yield an **S** configuration.

If a stereogenic atom carries one or more directed bonds (bold and/or broken) but the atom of least priority is attached by an ordinary single bond, the strategy in the program is to interchange the attached atom with one attached by a broken bond (if available), setting **Parity** to −1. If a broken bond is not available, interchange is made with the substituent attached with a bold bond and **Parity** is set to 1 (two inversions required: one for the interchange and one to reverse the direction of viewing substituent order along the axis of the bold bond). These observations lead to the following procedure, *when a stereogenic atom appears with directed bonds*:

(1) Determine the bond type linking the lowest priority atom to the stereogenic center.

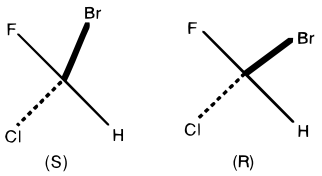


Figure 8. Anomalous configuration assignments by the program resulting from inconsistencies in placement of the Br substituent. The problem was solved by using closest atom pairs with neighboring priorities (here, F and Cl) for computing the C_z component.

(2) If the bond type is bold, set **Parity** = −1. Else if the bond type is broken set **Parity** = 1.

(3) If the bond type to the lowest priority atom is 1 (an ordinary single bond), then interchange that atom and the atom linked by the broken bond and set **Parity** = −1. In the event that a broken bond is not available, interchange the lowest priority atom and the atom attached to the bold bond and set **Parity** = 1. The interchange is effected by swapping the coordinates of the two atoms.

(4) Compute C_z, as discussed earlier, and the product **Parity***C_z. If the product is negative, assign the **R** configuration; else assign **S**.

As mentioned earlier, if the pair of substituents of highest priority are collinear, the direction vector does not exist, and the second highest priority pair must be used. An analogous problem occurs with atoms bearing directed bonds if the pair of substituents of highest priority in the molecular diagram are *almost* collinear, in which case the direction vector is not very accurately computed and may, in fact, occur with an incorrect sign, giving an incorrect configuration. This is exemplified in Figure 8 where two diagrams obviously of the same configuration are assigned opposite configurations by the program as a result of small differences (exaggerated here for clarity) in placement of the bromine substituent. In this example, if the pair of next lower priority (Cl, F) had been used instead of highest priority pair (Br, Cl) to compute the normal vector, consistent results for the two diagrams would be obtained. To circumvent this problem, the program computes the distance between the atoms of highest priority (in the example, the Br, Cl pair) and between the atoms of second highest priority (in the example, the Cl, F pair) and uses that pair having the smaller separation. In the case of Fischer projections, there are fewer degrees of freedom in constructing the diagram, and a simple test for collinearity (zero *x*-components or zero *y*-components for both vectors) may be used.

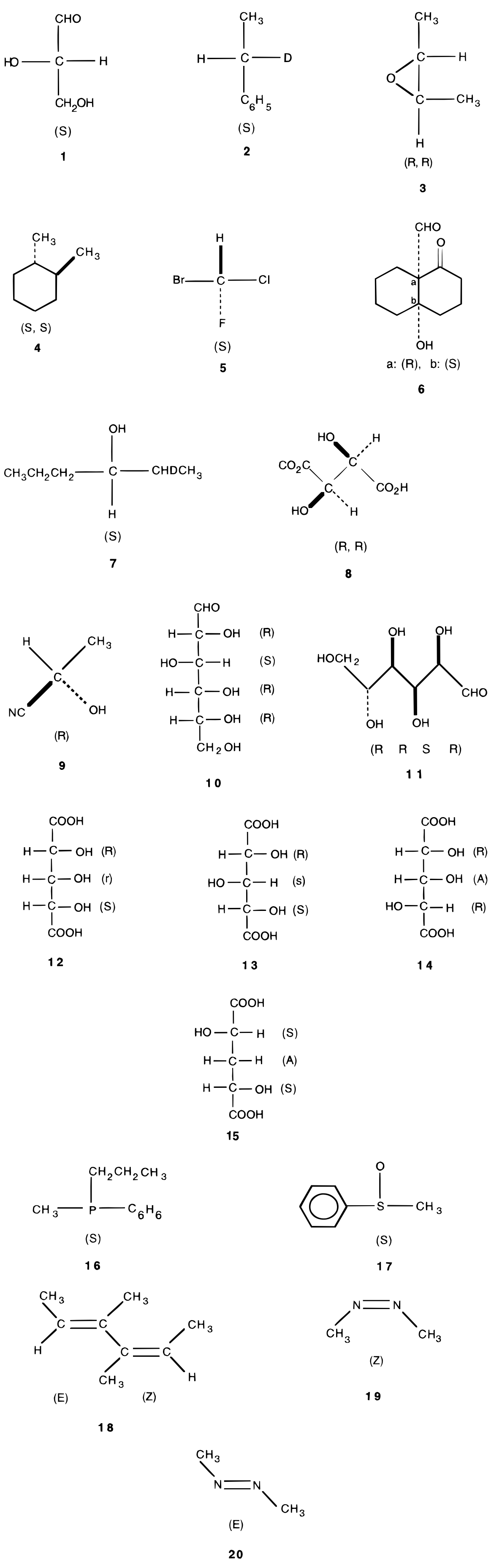
PSEUDO-ASYMMETRIC CARBON ATOMS

The first pass through configuration assignment will not suffice to assign configurations to pseudo-asymmetric atoms, since these by definition become asymmetric as a consequence of configuration differences between otherwise identical substituents. After all configurations for clearly stereogenic atoms have been assigned, atomCodes are augmented with configuration information in the form

$$\text{atomCode} = 10^*\text{atomCode} + \text{configuration code}$$

Configurations **S**, **R**, **E**, and **Z** are assigned configuration codes 1, 2, 5, and 6, respectively. Because of this reevaluation of atomCodes, it is possible that atomCodes once identical for two substituents are now different, with consequent dissymmetry. The routine for extracting configurations is re-entered with the new set of atomCodes (a

Chart 1



flag is transmitted to the program to let it know what is afoot); configurations **r** and **s** (lower case for pseudoasymmetric atoms) are assigned by the same procedures already discussed. The configuration codes give the **R** configuration priority over **S**, in agreement with a suggestion by Eliel and Wilen.[3]

RESULTS AND DISCUSSION

The program was written in Think Pascal on a Macintosh computer. The program includes a graphics interface to facilitate structure input. Many examples were checked against those in Eliel and Wilen,[3] with full agreement. A number of these examples appear in structures **1**−**20**. These examples demonstrate the power of the program to cope with a range of different representations of stereoarrangement. Structures **1** and **2** demonstrate applicability to Fischer projections. Structures **3**, **4**, and **6** are examples of application to cyclic structures, including directed bonds. Stick diagrams with directed bonds appear in structures **5**, **8**, and **9**. Structures **10** and **11** demonstrate successful application of the program to different representations of the open-chain form of D-glucose. Structures **12**−**15** demonstrate application to the stereoisomerism of trihydroxyglutaric acid, including configuration assignment to pseudoasymmetric atoms. Structures **16** and **17** illustrate application to a phosphine and a sulfoxide. The last examples, **18**−**20**, treat doubly-bonded atoms: *cis*−*trans* isomerism in a diene and in azo compounds. Although no timing runs were made, the program implemented on a 486 DX2-66 machine is fast; configurations in all examples were instantly reported by the program.

For purposes of this program, a stereogenic center is one for which all substituents are pairwise nonequivalent; i.e., no two substituents at a stereogenic center can be in the same equivalence class. Methods for assigning atoms to equivalence classes have employed canonicalization of a connection table to induce a unique descriptor for each atom in the structure,[1a] or—more rigorously—by automorphic mappings of a structure onto itself.[5] These methods are global; assignment of an atom to an equivalence class reflects the whole structure in which the atom exists. The approach taken in the current program is quite different in that the local environment of each atom defined by its substituents is explored, and no global determination of equivalence classes is required. Atoms immediately attached to a center that are identical on the basis of local descriptors (atom codes) are starting points for a phased breadth-first search along each substituent chain until unequivocal differences are found or one branch runs out of atoms. The search could conceivably extend over the whole structure before a conclusion is reached.

The principal restriction in application of this program is the likelihood that the priority rules, though close to CIP in many respects as indicated by the agreement in assignments by the two methods reported here, do not in fact replicate CIP priorities completely. For example, according to Eliel and Wilen[3] (p 112), *tert*-butyl, cyclohexyl, and *sec*-butyl groups occur in priority sequence between acetylenic and terminal ethylenic groups; the new program would not produce such a sequence, because unsaturation gives ethylenic carbon atom priority over the cyclohexyl group. For this reason, the program may not be applicable to chemical nomenclature as currently practiced. This does not detract from its use in applications such as organic synthesis or C13 data handling, where computer-generated internal representation of configuration suffices without necessarily conforming to external naming requirements.

The program could be modified to handle allenes and cumulenes. This might be done either within the framework of the present program (by appropriate virtual translation of a terminal atom to generate a phantom asymmetric carbon atom in the case of cumulenes with even number of double bonds or a phantom isolated double bond in the case of cumulenes with odd number of cumulative double bonds) or by generalizing the procedure **getPriority** to allow assignment of priorities to any collection of atoms, not just those that happen to be connected to a single stereogenic atom. The problem of assigning *cis*−*trans* configurations to groups attached to achiral atoms on rings (if the ring atoms are chiral, *cis*−*trans* relationships will be implied in assigning **R** and **S** configurations) is difficult when more than two substituents are present on a ring. Eliel and Wilen describe and recommend the use of a reference substituent against which all other substituents can be compared and designated as *cis* or *trans*. The unambiguous designation of such a reference in a computer program is a special problem and probably would require generation of a canonical connection table to induce a unique label for a reference atom.

REFERENCES AND NOTES

(1) (a) Razinger, M.; Perdih, M. Computerized Stereochemistry: Coding and Naming Stereoisomers. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 290−296. (b) Shelley, C. A.; Munk, M. E. An Approach to the Assignment of Canonical Connection Tables and Topological Symmetry Perception. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 247−250.

(2) Contreras, M. L; Rozas, R.; Valdivia, R.; Agüero,R. Exhaustive Generation of Organic Isomers. 4. Acyclic Stereoisomers with One of More Chiral Carbon Atoms. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 752−758.

(3) Eliel, E. L.; Wilen, S. H. *Stereochemistry of Organic Compounds*; Wiley Interscience: New York, NY, 1994.

(4) The program allows use of contractions "CH" and "HC" and directed bonds at ring vertices with hydrogen suppressed, combinations that result in an apparent degree 3 for carbon. The program deals with these cases by temporarily adding a "phantom" hydrogen atom to allow **getPriorities** to proceed normally.

(5) Prepare a copy of the structure. Select an atom x from the original and a different atom y from the copy. Atoms x and y are equivalent if after superimposing x on y, the original is superimposable on the copy. Implementation of the algorithm is described in the paper by the author, Figueras. J. Automorphism and Equivalence Classes. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 153−157.

CI950097M