# Algorithmic Generation of Chemical Abstracts Index Names. 1. General Design

J. MOCKUS,* A. C. ISENBERG, and G. G. VANDER STOUW

Chemical Abstracts Service, Columbus, Ohio 43210

Received January 19, 1981

The Chemical Abstracts Service (CAS) Chemical Registry System is a computer-based chemical information system that uniquely identifies chemical substances on the basis of their molecular structure. An algorithm has been developed for computer generation of Chemical Abstracts (CA) Index Names for organic compounds from Registry structure records. When programmed and installed in the CAS production system, the algorithm will support CA index preparation by generating names for a majority of the approximately 1400 new structures processed per working day at CAS.

## INTRODUCTION

Since 1907, CAS has published indexes in which chemical substances have been identified by chemical names alphabetically ordered in the index. (In Volume 92, the volume for the first half of 1980, there were approximately 306 055 different chemical substances indexed.) Before the introduction of computer processing into CA index production, it was necessary to name a substance each time it was selected as an index entry, regardless of whether that substance had previously been indexed and named. This resulted in much duplicative effort on the part of professional chemists on the CAS editorial staff, and it also meant that obtaining consistency in the indexes, i.e., having all the entries for a substance brought together at the same name, depended on staff skill in consistently applying the rules for naming substances.

The CAS Chemical Registry System was developed in part to help ensure greater consistency in CA indexes and to reduce the professional and clerical labor involved in their production.[1] The Registry System is a computer-based system for the unique identification of chemical substances on the basis of their structure and composition. The initial, experimental system, Registry I, began operation in 1964 and established the viability and validity of the registration concept for fully defined organic substances. In 1968, the scope of the system was increased as additional classes of substances were handled, and the system, now Registry II, began to be integrated into the CAS indexing operation. In 1974, the most recent version, Registry III, made major adjustments in the Registry structure records; these adjustments, and particularly the modification of the structure record to explicitly identify the ring systems present in a substance, were designed to increase the usefulness of the Registry records as a basis for automatic generation of chemical names and structure diagrams. As the use of the Registry System has been expanded, it has proved to be reliable and consistent as a structure identification method and has become an essential CAS production tool supporting CA index input and compilation. It has also found widespread interest and support in the scientific and technical community.

In a typical week, the Registry System processes over 35 000 substances. Some 28 000 of these are already present in the CAS Registry Master Structure File (which now contains records of over five million unique substances), and their Registry Numbers and CA Index Names are retrieved for further processing. The remaining 7000 substances are new to the CAS files and must be named before processing can continue. CAS nomenclature specialists derive the names of these new substances by applying the CA Index Name selection rules, rules within the framework of the International Union of Pure and Applied Chemistry (IUPAC) nomenclature rules, that are designed to ensure that each substance appears at a predictable and reproducible position in the "CA Chemical Substance Index".[2-5]

Although the Registry III system provides some automated support to the naming process, and this support helps to reduce the human effort involved in naming some 1400 substances per working day, there is still a great deal of time-consuming effort involved in name generation. The support which is currently provided includes the following: (1) retrieval of names for substances on file which have the same topology as the new substance being named, such as stereoisomers or isotopic variants, (2) retrieval of names of known ring systems contained in a new substance, and (3) retrieval of names for known components of addition compounds or polymers.[6]

Procedures have also been developed and put into operation to use the retrieved names for components as the basis for assembling candidate names for addition compounds and copolymers.[6] These procedures have been in use since early 1976 and, at present, are successfully generating names for over 400 substances per week. Although these procedures marked the first successful implementation of algorithmic name generation at CAS, they are limited to a special type of nomenclature, where a name is constructed from the names of components which are already known. This paper describes more recent work on an algorithm for the generation of systematic CA Index Names for organic compounds. (Procedures for the automatic generation of names for coordination compounds have also been defined.[7]) It is expected that programs which will be written based on this algorithm will be capable of generating correct names for a majority of the new substances encountered by CAS naming specialists.

## CA INDEX NAMES

The name generation algorithm is based on the rules for naming that are applied by CAS nomenclature specialists. These rules are given in the CAS "Chemical Substance Name Selection Manual" and are summarized in the "CA Index Guide".

A CA Index Name for an organic substance (see Figure 1) is usually based on an index heading parent name, which is generally formed from a molecular skeleton name, such as butane, a suffix denoting the principal function present, such as sulfonic acid, and, if required, a locant, such as 1-, citing the position of that function on the molecular skeleton. Following a comma, the substituents present are expressed in alphabetical order, as in 2,4-diamino-3-methyl-. The modification then cites any derivatives of the principal function, such as ethyl ester, and, if appropriate, stereochemical information, such as [*R*-(*R**,*S**)]-. The inverted form of the name, such as 1-butanesulfonic acid, 2,4-diamino-3-methyl-, ethyl ester, [*R*-(*R**,*S**)]-, is used for indexing purposes because the parent structure is emphasized.

Some organic substances, such as carbamic acid or imidodicarbonic acid, have names which express a chemical function but are not based on a molecular skeleton. These substances

0095-2338/81/1621-0183$01.25/0 © 1981 American Chemical Society

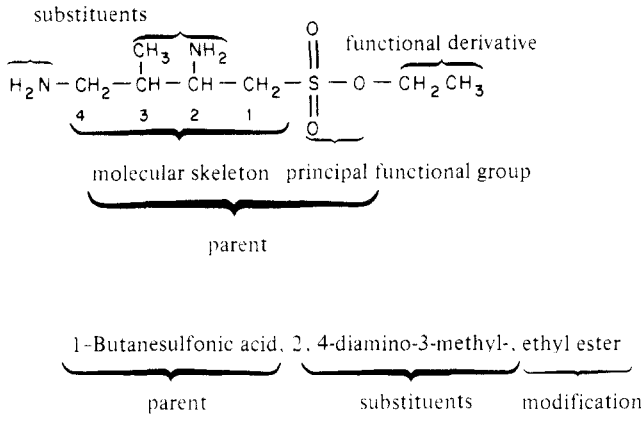



**Figure 1.** Chemical Abstracts index name.

are called functional parents. Their CA Index Names are likewise divided into parent, substituent, modification, and stereochemical sections.

Many organic substances can be named, unambiguously, in two or more ways. So that a unique CA Index Name will be obtained, the molecular skeleton to be named as the index heading parent is selected by application of the following set of principles, in the order given, until a decision is reached:

(1) greatest number of the principal chemical functional group; the basic order of precedence is, in descending order, acids, acid halides, amides, nitriles, aldehydes, ketones, alcohols/phenols, hydroperoxides, amines, and imines;

(2) preferred hetero atom content of the molecular skeleton;

(3) preferred ring system (as determined by 15 subprinciples);

(4) greatest number of acyclic hetero atoms (in the molecular skeleton);

(5) largest index heading parent (in terms of nonhydrogen atoms);

(6) greatest number of most-preferred acyclic hetero atoms;

(7) greatest number of multiple bonds;

(8) lowest locants in the heading parent for, successively, hetero atoms, principal groups (suffixes), all multiple bonds, and double bonds.

If the index heading parent selected by application of the above principles occurs more than once in the structure, further principles are applied to select one of those occurrences. The occurrence chosen is that which

(9) is in the most central position (if three or more occurrences are present in a linear arrangement);

(10) possesses the most substituent groups;

(11) has the lowest locants for substituents on the heading parent;

(12) is capable of being "multiplied" so that the index heading parent expresses the maximum number of occurrences;

(13) when a choice still exists, leads to the name which appears the earliest in the sequence of CA Index Names (in inverted form).

The 13 principles of index heading parent name selection are the most important of the CA Index Name selection rules, but they are just one of the many sets of rules implemented in the name generation algorithm. Other sets of rules provide for

* restoration of tautomeric and alternating bonds (since the Registry III System normalizes these bonds as part of the registration process, the explicit single and double bonds must be regenerated before analysis for naming; this process is discussed later);

* identification of ring assemblies (a ring assembly in CAS index nomenclature is a set of two or more identical ring systems which are directly linked by acyclic single bonds, such as 1,1'-biphenyl);

* assignment of indicated and added hydrogen (the "extra" hydrogen, expressed by a term such as 2*H*, cited to specify the position of a saturated atom necessary to completely define some ring systems, as in 2*H*-pyran);

* selection of numbering schemes for molecular skeletons;

* selection of names for compound radicals (i.e., radicals made up of two or more individually named radicals, such as hydroxymethyl);

* suppression of unnecessary locants (straightforward application of the naming rules would often lead to a name such as benzene, 1-chloro-, in which the locant is unneeded and is therefore deleted to give simply benzene, chloro-);

* identification and naming of functional derivatives (i.e., esters, anhydrides, oximes, etc.);

* construction of names (i.e., locant placement, punctuation, elision, alphabetization, etc.).

Still other sets of rules are implemented in the name generation algorithm via tables or reference files rather than algorithmic procedures. These provide for

* identification of functional groups;

* name fragments (for use in name assembly) for ring systems, chains, functional groups, and hetero units;

* locant numbering schemes for ring systems, functional groups, and hetero units.

## THE NAME GENERATION ALGORITHM

Although the name generation algorithm is quite large, containing over 300 routines and subroutines, it is not nearly as complex as might be expected, largely because of the modular design approach that was taken to the overall problem. This strategy entailed looking at the operations that had to be performed in order to name an organic compound and repetitively splitting them into smaller and simpler operations. Eventually, even the most complicated procedure could be broken down into subprocedures that could be defined in terms of computer-language statements and thus are amenable to programming. A number of these low-level procedures necessarily perform very intricate and unusual computational functions, but they are at least small enough to be understandable.

The algorithm names a structure in the same manner as would a nomenclature specialist, by identifying candidate index heading parents and then successively applying the applicable nomenclature principles to eliminate the less-preferred candidates. The algorithm operates on a systematic step-by-step basis; while a chemist can often apply nomenclature rules without conscious effort, the algorithm has no such ability (nor can it have one) and must systematically collect numerous items of information about the structure during the analysis.

The algorithm is organized as two phases, a setup phase and an analysis phase. The setup phase consists of five steps, explained in greater detail later in this paper, which prepare the structure for naming:

* bond restoration, which regenerates explicit single and double bonds from the normalized tautomer and alternating bonds of the Registry III structure record;

* functional group identification;

* chain domain identification, a preparatory step for the identification of chains during the analysis phase;

* ring name preparation, in which ring system names retrieved from Registry files are used to create name

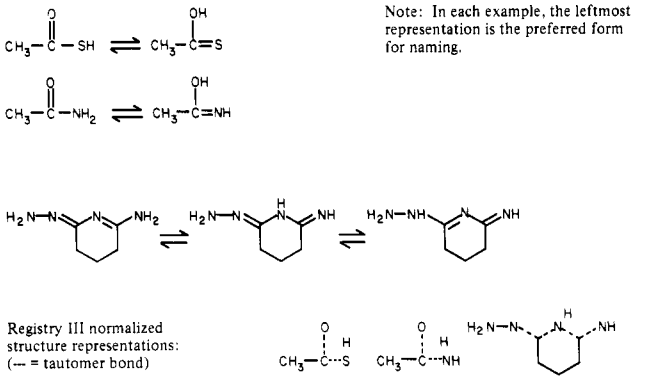



**Figure 2.** Examples of tautomers.

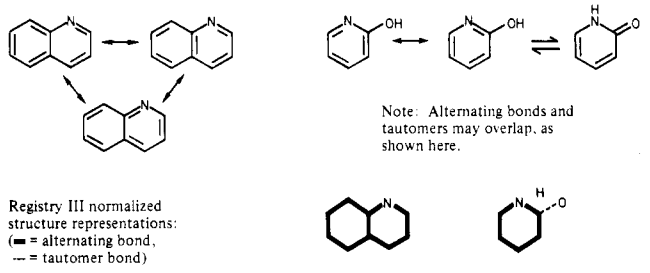



**Figure 3.** Examples of alternating bonds.

fragments for construction of ring system, ring assembly, and cyclic radical names;

* ring assembly identification.

The analysis phase of the algorithm is less amenable to a stepwise description. Although the analysis phase is basically organized about the 13 principles of index heading parent name selection and applies these in sequence as needed by the structure being named, many of the analysis functions may be applied at a number of points in the overall analysis. For example, the basic rule that "indicated hydrogen is assigned the lowest possible locants" could be applied to a ring system comprising a parent molecular skeleton at any of six different points in the algorithm, depending on other features of the structure; the application of this rule to ring systems in substituents or functional derivatives would occur at still other points in the algorithm. Accordingly, the discussion of the analysis phase is divided into three general areas:

* selection of the index heading parent—the mainline of the analysis phase where the 13 principles are applied;

* name generation—the procedures that perform the detailed analysis of the structure as parent, substituent, and modification and build the data structure for naming;

* name assembly—the procedures that actually construct the names, working with the data structure from the analysis and extensive tables of name fragments.

An annotated example of algorithmic name generation is presented in the Appendix to this paper.

## THE SETUP PHASE

**Bond Restoration**: The CAS Registry System normalizes tautomeric and alternating bond situations as part of the registration process so that any of the possible chemically equivalent representations of a substance will always lead to the same Registry structure record.[8] In a tautomeric situation, an equilibrium involving single/double bond shifts coupled with hydrogen migration, the Registry replaces the explicit single and double bonds with special tautomer bonds and associates the migrating hydrogen with groups of atoms rather than just single atoms (see Figure 2). Similarly, the alternating single and double bonds conventionally used to depict resonant or

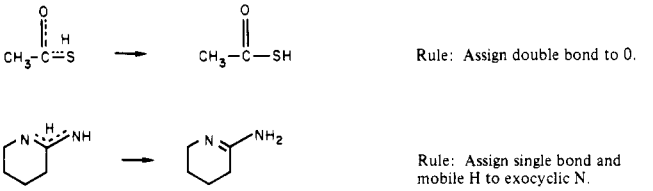



**Figure 4.** Examples of denormalization.

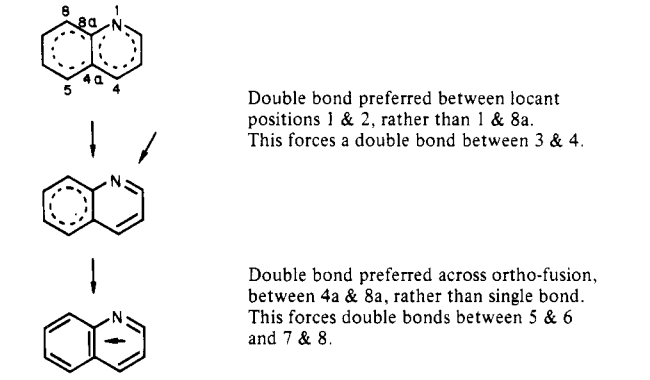



**Figure 5.** Example of unscrambling.

"aromatic" bonds are replaced with special alternating bonds (see Figure 3). These phenomena are quite common; about 25% of all structures in the CAS Registry Master File have tautomeric bonds, and about 90% of the ring-containing structures have alternating bonds.

A CA Index Name for an organic substance describes a specific (and preferred) bond pattern in a tautomeric structure, so the name generation algorithm must restore the single and double bonds that were normalized and fix the migrating hydrogen. Similarly, alternating single and double bonds must be restored from the normalized alternating bonds. The bond restoration process is the first major step in the initial or setup phase of the algorithm and is done in two stages, denormalization and unscrambling.

Denormalization is done by two procedures that operate according to several structural considerations, the basic rule being "assign the maximum number of double bonds, where possible, to atoms in the order of precedence O, S, Se, Te, and terminal N". The first procedure is intended to reduce the processing effort spent on bond restoration; it handles only small tautomeric situations matching one of five generic patterns but does so very rapidly. Most acyclic tautomer situations are handled here, along with some simple, common cyclic ones. After a number of setup and data-gathering steps, the second procedure handles the remaining acyclic tautomer situations; it operates by simply assigning double bonds as needed, following the basic structural preference rule (see Figure 4).

Unscrambling is a much more complicated process that handles tautomeric situations according to nomenclature rules as well as structural considerations and "aromatic" alternating bonds according to CAS input graphic standards. Since many of the rules and standards include the phrase "where possible", the unscrambling procedures must be decision-making procedures with the ability to backup and change decisions that prove to be incorrect. Annotated examples of unscrambling are given in Figure 5.

Since the bond restoration process must be the first major step in the algorithm and since it must be guided by nomenclature considerations, there is a problem of circularity. In some cases, single and double bonds could be assigned in several ways, each leading to a different analysis and a different name for the structure; the correct bond assignment leading to the preferred name cannot be determined without

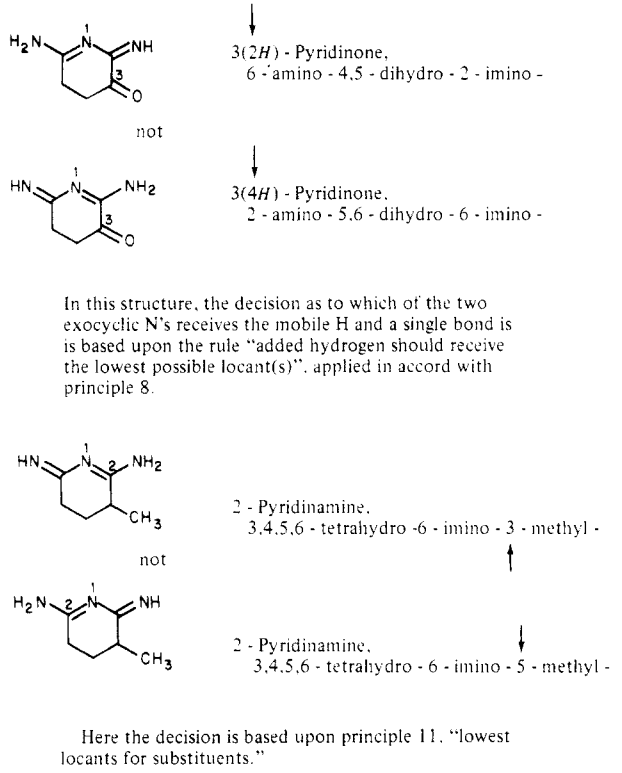



**Figure 6.** Examples of unscrambling during the analysis phase.

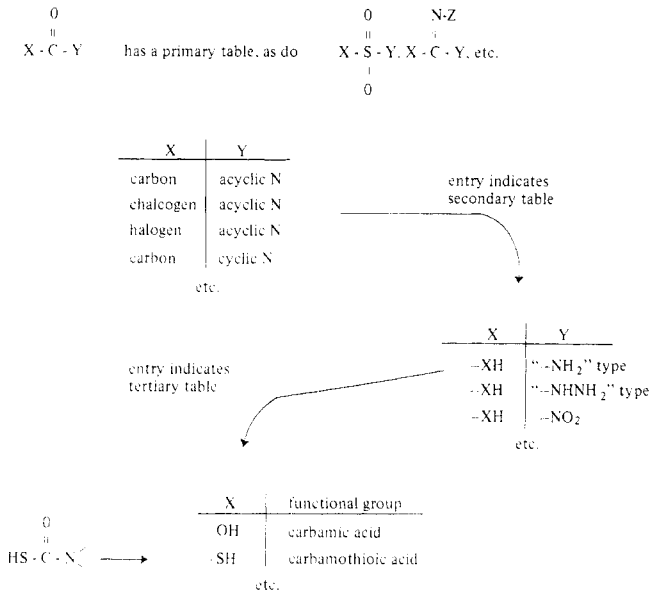



**Figure 7.** Example of functional group identification.

prior knowledge of that name. Most of these cases involve the placement of bonds and tautomeric hydrogen atoms within ring systems in accordance with rules of the type "lowest locants for xxx", which affect primarily the names for the ring systems. Since these rules do not affect the identification of functional groups, the bonds are arbitrarily unscrambled at this point to allow verification of the overall bond restoration process. They are subsequently renormalized and unscrambled at the appropriate points in the analysis phase, when all of the information needed to compare the alternative bond assignments becomes available (see Figure 6).

The remaining cases involve placement of bonds and tautomeric hydrogen atoms within the acyclic portion of a structure, and these affect several aspects of naming. Since these cases comprise only a fraction of a percent of tautomeric structures, the algorithm simply rejects such structures. Although procedures could have been developed to handle most of these problem cases, the effort involved would have been far too great to be justified by the small increase in capability. (Cost-benefit considerations affected the design of many of the more complicated—and more interesting—procedures in the algorithm; a basic point of our design philosophy was to develop procedures to handle the bulk of, but not necessarily all of, the naming effort.)

**Functional Group Identification.** Functional group identification is the second major step in the setup phase of the algorithm. The procedure uses a table-driven approach to recognize and identify functional groups. It scans the acyclic portion of the structure, looking for substructures characteristic of functional groups, such as hetero atoms with unsaturated bonds. When a key substructure such as C=O is found, the identifier determines the generic nature of its attachments, which can be represented as X and Y in X—C(=O)—Y. The particular combination of X and Y, such as X = chalcogen and Y = acyclic nitrogen, is used to select an entry in the primary table associated with the key substructure; this entry indicates a secondary table and specifies any further analysis of X and Y to be done. More specific information about X and Y, such as X = generic hydroxy (–XH) and Y = amino nitrogen, is used to select an entry in the secondary table, such as generic carbamic acid. This entry usually indicates yet another table, a tertiary table that handles chalcogen and halogen variations of the identified generic functional group, thus finally arriving at "carbamothioic acid" when X = –SH and Y = $-NH_2$ (see Figure 7).

After the identification process is completed, the functional group list that was built is sorted by order of precedence and partitioned into sets of functional groups of equal rank. The highest ranking functional groups are the principal functional groups that are used in the generation of candidate index heading parent structures according to principle 1.

If the highest ranking functional group is a functional parent (as was the case in the example) rather than an ordinary functional group, no special action is necessary at this point. Later, though, during the analysis phase, special procedures will be used to generate a name based on a functional parent instead of the usual molecular skeleton parent.

**Chain Domain Identification.** There are four basic types of chains in organic nomenclature: hydrocarbon chains, homogeneous hetero chains, heterogeneous hetero chains (containing alternating atoms of a group 4A element and a chalcogen), and organic hetero chains ("a" chains) named by replacement ("a") nomenclature (see Figure 8 for examples of these types of chains). If elemental variations are considered, there are more than 30 distinct varieties of chains.

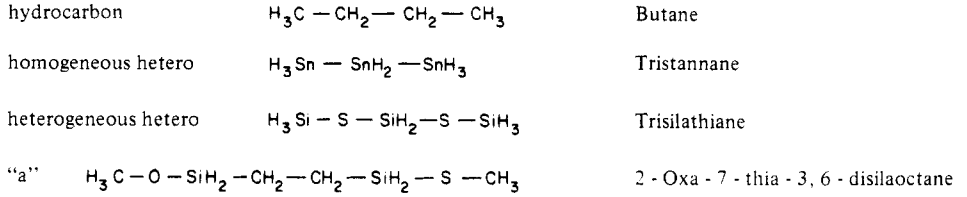
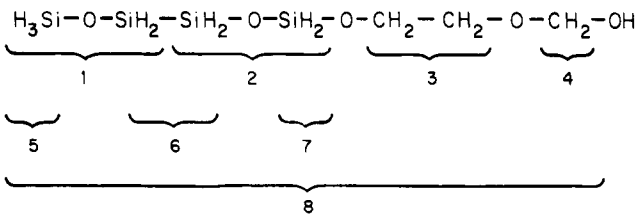
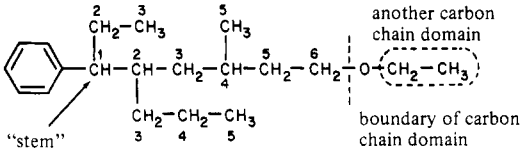



**Figure 8.** Types of chains.




Domains 1 and 2 define territories in which heterogeneous hetero Si-O-Si chains can be found, 3 and 4 territories in which hydrocarbon chains can be found, 5-7 homogeneous hetero Si chains, and 8 "a" - chains. Note that the domains simply group connected atoms meeting eligibility criteria; an atom may belong to several domains, to just one, or to none at all.

**Figure 9.** Examples of chain domains.




After numbering outwards in concentric layers from the "stem" atom, tracing back from the highest-numbered atom(s) identifies the longest chain(s) from the "stem"; here, a 6-atom chain is found. Note that the numbering ends at the boundary of the carbon chain domain.

**Figure 10.** Substituent chain identification.

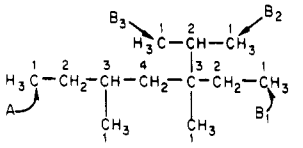
The basic design approach used in the algorithm separates chain processing into two functions: identification of the atoms that might belong to a particular variety of chain and identification of the chains themselves. As a result, only 14 major routines and a few small subroutines are needed to handle the basic chain-building process; the major routines could be built around a few basic techniques, so the overall process was quite easy to design and implement.

The third step of the setup phase is the identification of "chain domains", where a domain is a group of connected acyclic atoms meeting certain criteria that define a territory in which chains of a given variety are sought (see Figure 9). Seven identification procedures are used, with the identification of homogeneous hetero chain domains being subdivided, for convenience and efficiency, among four procedures.

Once domains are identified according to the applicable nomenclature rules, chain identification in the analysis phase of the algorithm is largely reduced to simply finding strings of atoms within a particular domain:

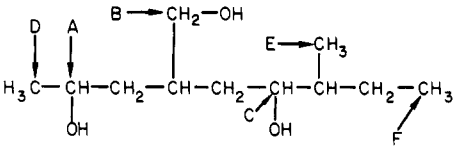
* Substituent chains are quite simple to find, since the starting point is the chain "stem" atom by which the chain is attached to the underlying structural unit. The procedure assigns level numbers to atoms in the domain, with the stem as level 1 and numbering outward the successive concentric layers of atoms assigned level 2, 3, etc. After numbering, tracing back down the levels from the atoms assigned the highest level identifies the longest chains in the domain that start from the stem atom (see Figure 10).

* Nonfunctional parent chains (chains bearing no functional groups, acceptable as parents only when a function expressible as a suffix is not present in the structure) are found with a procedure that also identifies only the longest chains existing within a domain. Here, an "outside-in" numbering is used, beginning with the assignment of level 1 to all "terminal" atoms ("terminal" when considering only the atoms in the domain), then working inward, with an atom assigned a level only after all of its outer attachments have been processed. The atoms with the highest level provide the starting point for tracing chains back out to the level 1 atoms (see Figure 11).



After numbering inwards from terminal atoms to identify the central atom, the longest chains are found by tracing out from the center. Here, three alternative 7-atom chains are found, running between atom A and atoms B1 - B3.

**Figure 11.** Nonfunctional parent chain identification.



1. The initial step identifies atoms bearing principal functional groups (A, B and C), then finds the best chain between them. No choice can be made on the number of principal groups, since A-B, A-C and B-C all bear two groups.

2. The A-B chain is extended from atom A to include atom D, giving the 5-atom chain B-D.

3. The A - C chain is likewise extended to include atom D. From atom C, the chain could be extended to either atom E or atom F; the former would add only two atoms while the latter would add three, so the extension is to atom F. The result is the 9-atom chain D-F.

4. The B - C chain is likewise extended to include atom F, giving the 7-atom chain B - F.

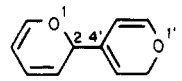
5. The parent chain is the 9-atom chain D-F, the longest of the three chains.

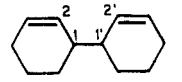
**Figure 12.** Parent chain identification.

* Parent chains bearing functional groups are more difficult to find, though only because the procedure includes many look-ahead techniques (in order to substantially increase processing efficiency). The procedure initially searches for chains (of any length) that terminate at atoms bearing functional groups; those chains bearing the greatest number of functional groups are then "extended" to the limits of the domain, and only the longest of these chains are kept (see Figure 12).

Seven chain identification procedures are required, three handling "a" chains and four handling all other types. "A" chains must have their own procedures because of problems inherent in their definition: "a" chains can be terminated only with atoms of certain elements and must contain at least four hetero units; as a consequence, the "a" chain procedures must check each generated atom string and accept as "a" chains only those meeting these criteria.

**Ring Name Preparation.** Prior to the development of the CAS Registry III System, preliminary investigations of the possibility of algorithmic name generation had concluded that the most difficult problems would lie in the area of naming ring systems and assigning ring locants. (Reports in the literature of other work on name generation have, in fact, concentrated on the problems of naming ring systems.[9,10]) The Registry III System was thus designed to isolate and identify ring systems, allowing a "building block" approach to name generation, whether performed by CAS chemists or by the



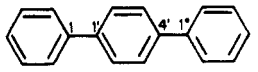




**Figure 13.** Examples of ring assemblies.

algorithm. Algorithmic structure display, where a chemical structure diagram is created from a structure record, also benefited from this approach.[11]

When a ring system is present in the structure being processed, the Registry III record contains a Ring Identifier that is used to retrieve the ring's structure record from the Registry Ring File so that the complete connection table for the structure can be built. Subsequent processing in the ring name preparation step, the fourth setup step, is as follows:

* Ring system data needed for preferred ring system comparisons (principle 3) and ring atom locant data are retrieved from internal tables (for common rings) or from the Ring Data File, a computer-readable file used for support of algorithmic name generation.

* CA Index Names for the ring parents are retrieved either from internal tables or from the Registry Nomenclature File.

* The ring parent names are examined and "abnormal" names describing ions, unusual valences, or unusual bonding are eliminated to leave the "normal" parent names—one name is the usual case, but some types of ring nomenclature have one name for saturated rings and another for unsaturated rings (for example, the names phospholane and 1*H*-phosphole for a $C_4P$ ring or the names pyridine and piperidine for a $C_5N$ ring).

* The ring in the structure is similarly examined for "abnormalities" (which, if present, lead to rejection of the structure for naming by a chemist), and if necessary, its state of saturation is determined so that the proper ring parent name can be selected.

* If indicated hydrogen is cited in the ring parent name, information is collected to allow reassignment of the indicated hydrogen during the analysis phase.

Subsequent processing in the analysis phase collects all of the data necessary to describe the ring as it appears in the structure being named, such as locants for indicated hydrogen or unsaturation, the use of the ring in parent or substituent, and, if appropriate, ring assembly descriptive data. The name assembly routine works with this data and the "root" of the proper ring parent name to build the final name:

* For a $C_4N$ ring, for example, the ring name procedures would retrieve the two names 1*H*-pyrrole and pyrrolidine and provide the information needed for the subsequent generation of names such as 2*H*-pyrrole, 3*H*-pyrrol-2-yl, and 1-pyrrolidinyl.

* For a $C_5S$ ring, four names would be retrieved: 1*H*-, 2*H*-, and 3*H*-thiopyran and thiopyrylium; all but the second would be recognized as "abnormal" names, leaving 2*H*-thiopyran as the "normal" name from which 4*H*-thiopyran, 2*H*-thiopyran-3-yl, 4,4'-bi-2*H*-thiopyran, etc., would be generated.

**Ring Assembly Identification .** A ring assembly in CAS index nomenclature is a linear "chain" of two or more identical rings joined by acyclic single bonds, not necessarily at equivalent positions (see Figure 13). A ring assembly may be a parent molecular skeleton or a substituent and is essentially treated as if it were an isolated ring system.

Identification of ring assemblies is the final step of the setup phase of the algorithm. In theory, it might be better to search for ring assemblies at appropriate points in the analysis phase, while the parent, substituent, and modification portions of the structure were being identified and analyzed, but the procedures involved would be quite complex. In practice, a simple identification procedure provides adequate capabilities, since only a few percent of the structures in the CAS Registry Master File have ring assemblies; about two-thirds of these are 1,1'-biphenyl and most of the rest are other two-component assemblies.

The preneed identification approach allows the construction of a ring data table containing an entry for each ring system and each ring assembly present in the structure. The table entries provide access to information such as ring parent names, ring atom locant tables, comparison data for principle 3, etc., as well as storage for data generated during the analysis, describing the rings as they are used in the structure being named. An unusual descriptive approach is used in which an ordinary ring system is described as a one-component ring assembly; this is done so that the many algorithm procedures that work with cyclic structural units can be generalized to handle both ring systems and ring assemblies. (Some routines specific to ring systems or ring assemblies are still needed to handle name construction, for example, or assign primes to ring assembly component ring locants.)

Ring assemblies are identified by building a connectivity matrix in which the nodes are ring systems. Node–node connections are added to indicate potential ring assembly linkages, i.e., acyclic single bonds between similar ring systems (with identical indicated hydrogen or unsaturation specifications, if applicable). Chain-building techniques are used to find potential ring assemblies, indicated by connected chains of nodes; branched chains indicating overlapping, alternative ring assemblies lead to rejection of the structure for naming by a chemist, while linear chains lead to further checks and, eventually, the addition of a ring assembly entry to the algorithm's data tables.

Ring assemblies involving rings with indicated hydrogen are occasionally difficult to identify, since the indicated hydrogen must be at the same positions on each of the component rings and not all of the rules governing indicated hydrogen assignment can be applied at this point. Here, as elsewhere in the algorithm, a "fail-safe" approach is used: if there is the slightest doubt about the correctness of the analysis, a warning message is issued so that the generated name can be manually reviewed.

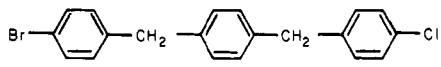
## THE ANALYSIS PHASE

**Selection of the Index Heading Parent Structure.** The generation of a set of candidate index heading parents is the initial step in the analysis phase. Potential candidates based on molecular skeleton parents are created at one of four points in the procedure, depending on the nature of the structure being named. In the order in which they are attempted, these are as follows:

* If functional groups are present in the structure, molecular skeletons bearing the greatest number of the principal functional group are created concurrently with the application of principle 1 to these candidates.

* Otherwise, nonfunctional hetero atom molecular skeletons (hetero chains and/or heterocycles) are sought prior to the application of principle 2.

* If no candidates bearing functional groups or containing hetero atoms can be found, carbocyclic molecular skeletons (ring systems or ring assemblies) are sought prior to the application of principle 3.



Benzene, 1 - [(4 - bromophenyl)methyl] -4 -[(4 - chlorophenyl)methyl] -

not Benzene, 1 - bromo - 4 -[ [ 4 - [(4 - chlorophenyl)methyl]phenyl]methyl] -

not Benzene, 1 -[ [ 4 - [(4 - bromophenyl)methyl]phenyl]methyl] - 4 - chloro -

**Figure 14.** Principle of centrality.

* As the last resort, carbon chain molecular skeletons are sought concurrently with the application of principle 5.

After a set of candidates has been generated, the relevant nomenclature principles are successively applied, and the less-preferred candidates are deleted. The application of principles 2–7 is straightforward and largely requires just counting the occurrences of various structural features and comparing the counts.

Principle 8, "lowest locants for structural features in the index heading parent", has a dual purpose and results in both inter- and intracandidate comparisons. If the molecular skeleton of a candidate has several possible locant numbering schemes, principle 8 is applied to select the preferred numbering schemes. After the candidates have been individually processed, the intercandidate comparisons are performed, and the less-preferred candidates are deleted.

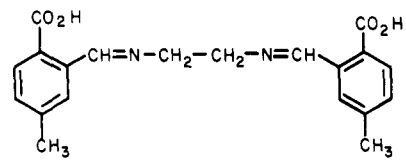
After principle 8 is applied, the preferred index heading parent has been selected. If two or more equivalent candidates are present, the remaining principles (9–13) are applied to select the preferred occurrence of the heading parent within the structure. As an initial step in this selection, the atoms of the structure are marked to indicate, for each candidate, whether they belong to the parent, substituent, or modification portion of the candidate's index name.

Principle 9, "centrality" (see Figure 14), is applied via techniques adapted from chain-building procedures. A connectivity matrix is built in which the nodes correspond to the candidates and the groups of atoms connecting them. An "outside-in" level numbering leads to the identification of the central candidate(s), and others are deleted.

Principle 10, "greatest number of substituents", is a simple "count and compare" procedure. Since only the number of substutuents is of interest, the procedure simply scans the connection table of the structure looking for bonds between atoms with "parent" flags and atoms with "substituent" flags.

If the index heading parent candidates are polyfunctional and they either contain substitutable functional groups such as amides or amines or have conjunctive unit molecular skeletons, the next step in the analysis determines how locants are to be formed for substituents on the functional groups or conjunctive unit chains, since this must be known for principle 11 to be applied. Ordinarily, primes are used to distinguish the substitutable positions, but superscript locant indexes are needed if primes would be ambiguous or could be misinterpreted, if the molecular skeleton is unsymmetric or unsymmetrically substituted, and if the substitution in question is unsymmetric (consider, for example, *N*,*N*′,*N*″-trimethyl- and $N^1$,$N^1$,$N^2$-trimethyl-1,2,4-benzenetriamine). Only a few of the necessary analysis procedures were developed, since the determination of symmetry proved to be quite a difficult task; structures that cannot be handled are flagged for human review and meanwhile receive locants with primes, the more common treatment.

Principle 11, "lowest locants for substituents", is another place where both inter- and intracandidate comparisons are made. Here, the substituent identification routines of the parent name generator procedures are used to provide lists of substituents and their locants. Again, if the molecular skeleton



multiplicative name: Benzoic acid, 2, 2' - [1, 2 - ethanediylC bis(nitrilomethylidyne)] bis [4 - methyl-

normal substitutive name: Benzoic acid, 2 -[[[[2 -[[(2 - carboxy-C 5 - methylphenyl)methylene]amino]C ethyl]imino]methyl] - 4 - methyl-

**Figure 15.** Principle of multiplication.

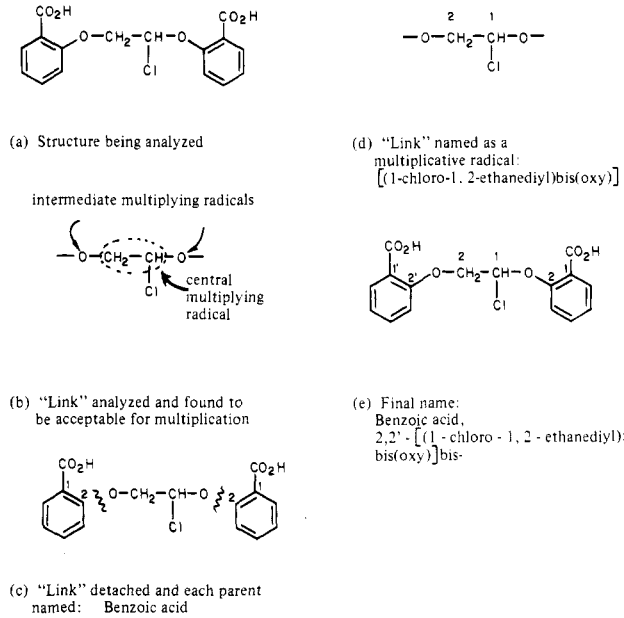



**Figure 16.** Generation of a multiplicative name.

of a candidate has several possible locant numbering schemes, an intracandidate comparison is applied to select the preferred numbering schemes before the intercandidate comparisons are performed and the less-preferred candidates are deleted.

Principle 12, "multiplication" (see Figure 15), is only partially implemented in the algorithm so that multiplication greater than doubling cannot be handled, and only limited sets of central and intermediate multiplying radicals can be handled. A preliminary study of the problem showed that this partial implementation would handle 80–90% of all multiplicatively named structures, which comprise only about 5% of all organic compounds; a more comprehensive implementation would have been much more difficult to develop and would not have been cost effective. Structures with three or more parent candidates at this point and those with two candidates that cannot be handled by the principle 12 procedure are flagged and sent on to principle 13; a subsequent review of the flagged structures by a chemist allows multiplicative names to be manually generated where appropriate.

The principle 12 procedure operates (see Figure 16) by first identifying the "link" between the two parent candidates and analyzing the link in terms of the rules for naming multiplying radicals. The link is then "detached" from the two candidates (by temporarily altering the connection table describing the structure) so that they can each be named by the principle 13 procedure; note that a full name (parent, with substituent and modification where applicable) is generated, not just a parent name. If the generated names are identical, the two candidates are likewise identical and a multiplicative name is appropriate;

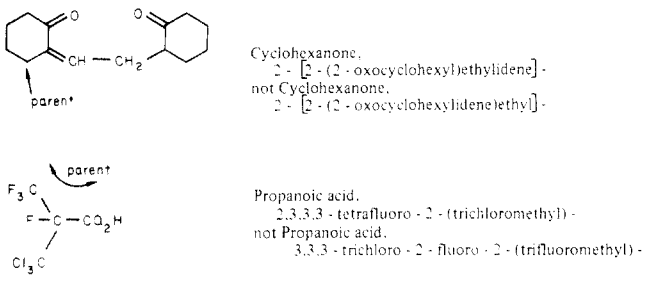



**Figure 17.** Principle of "earliest index position".

after the link is "reattached", the multiplicative name is constructed from the previously generated candidate name and the multiplicative radical name(s) describing the link. If the two candidates are not identical, multiplicative nomenclature cannot be used and processing continues with principle 13.

Principle 13, "earliest index position" (see Figure 17), is the final step in the parent name selection process and is the point where the names are generated. Like the earlier principles 8 and 11, it must make both intra- and intercandidate comparisons to select the preferred numbering scheme for the candidate's molecular skeleton and then to select the preferred candidate. Comparisons are made by applying to the generated names the sort-key procedure used for the alphabetization of names in the "CA Chemical Substance Index" to the generated names.

If the index heading parent is based on a functional parent, a simpler set of procedures is used for the analysis. Since only a small fraction of organic substances are named as functional parents, the algorithm's capabilities in this area were not extensively developed. Complex substances named as functional parents are rejected by named algorithm to be named by a chemist.

**Name Generation.** The actual name generation process is performed by three supervisory procedures; which one is used depends on whether the parent is a chain, a ring, or a functional parent. These procedures control the construction of the nomenclature tree data structure that describes the structure being named and use the name assembly procedure, discussed later, to build the names.

The nomenclature tree data structure is the link between the analysis phase of the algorithm and its name assembly procedure. A tree structure is the appropriate format for the name generation analysis data, since a chemical structure is named in this very way—as a parent to which are attached substituents, to which are attached more substituents, etc. (see Figure 18).

Each node in the tree describes a single nomenclature/structural unit:

* The parent node is the base node of the tree; it may describe either a molecular skeleton (ring, ring assembly, conjunctive unit, or chain) and, if present, attached principal functional groups or a functional parent, depending upon the nature of the index heading parent.
* Each substituent node describes a substituent radical and specifies how and where it is attached to the node above it.
* A modification header node carries control information for the name assembly procedure; it is attached below the parent, with a subtree below it describing the modification—several different subtree structures are needed to describe the various types of functional derivatives in accordance with the basic nomenclature tree approach (see Figure 19).

The analysis phase obtains the name for a full or partial structure by invoking the name assembly procedure and indicating a tree node. The assembly procedure builds several names, adding them to the node's data block. The primary

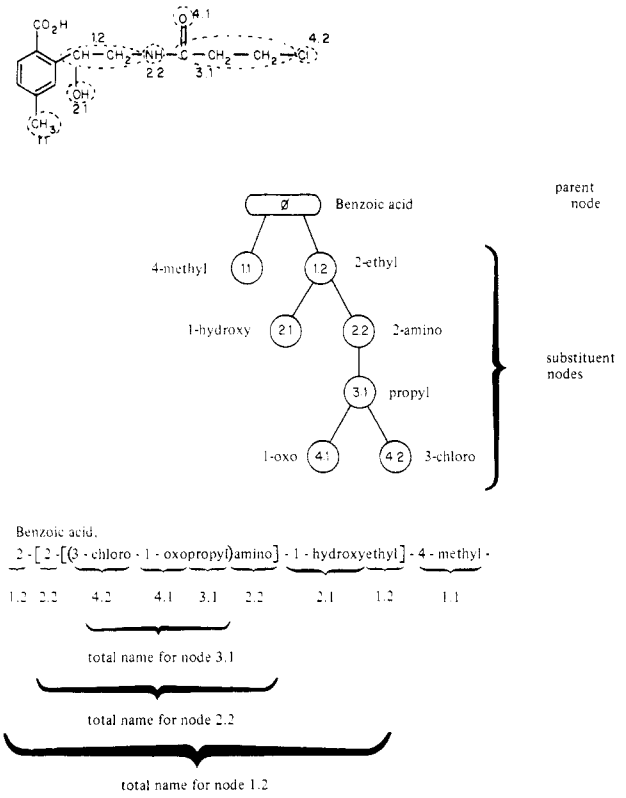



**Figure 18.** Nomenclature tree data structure.

name returned is a composite name describing the node and the subtree below it. The secondary names are intermediate names created during the name assembly process and are saved so that a new primary name can easily be created if the node is later renamed with an alternative numbering scheme.

Given a candidate index heading parent, the appropriate name generator procedure first sets up a node describing the parent to serve as the base of the nomenclature tree. All of the necessary data are on hand, coming from sources such as the functional group table, the ring data table (and associated ring parent names), and the structural feature data collected during the application of principles 1–8. After the parent node is built, processing is as follows:

* Each substituent attached to the parent is processed, resulting in the addition of a substituent node (which might be the top node of a substituent subtree) below the parent node.
* If the parent molecular skeleton has alternative locant numbering schemes, the corresponding names are generated and compared according to principle 13 ("earliest index position"), and only the best numbering scheme is retained.
* Each functional derivative is processed, resulting in the addition of a modification header node and attached subtree below the parent node.
* If the parent molecular skeleton still has alternative locant schemes, the corresponding names are again generated and compared.
* "Single atom fragments" in the connection table (describing hydrates, salts, etc.) are processed and appropriate terms are added to the modification portion of the name.
* Finally, if a text descriptor is present, the stereochemical information it contains is properly formatted and added as the stereochemistry portion of the name.

Substituent processing is an extremely complex operation, despite its use of only two supervisory procedures handling acyclic and cyclic substituents. The complexity comes about

because the last rule applicable to substituent analysis is "earliest index position of the total name"; accordingly, the analysis of a substituent must include the analysis of its substituents, which in turn must include the analysis of their substituents, etc. As a result, the two substituent analysis supervisors must be recursive, able to invoke both themselves and their counterparts.

Acyclic substituent processing begins with the identification of a set of alternative substituent chains, then applies the appropriate nomenclature rules (e.g., greatest number of hetero atoms, longest chain, greatest number of multiple bonds, etc.) to select the preferred substituent chain. At this point, a substituent node is built describing the chain, but its processing is then temporarily suspended while its substituents are analyzed and named (see Figure 20). When processing of the chain resumes, the subtree describing its substituents has been completed, and a total name for the chain can be assembled. If alternative chains are still possible, their total names are compared according to the final rule, "earliest index position"; the node and attached subtree describing the preferred substituent chain are retained and the analysis and name data for the less-preferred chains are discarded.

Cyclic substituent processing begins by simply identifying the ring or ring assembly that is the substituent and accessing the appropriate ring data table entries. If alternative locant numbering schemes are possible, nomenclature rules provide for preference on the basis of lowest locants for indicated hydrogen (if present), the radical attachment point, substituents, etc., and these rules are applied successively as needed. As with an acyclic substituent, processing of the cyclic substituent is temporarily suspended, late in the procedure, while its substituents are analyzed and named.

Modification processing uses data provided by the functional group identifier as a starting point; this data specifies the nature of the functional derivative and cites the functional derivative atom attached to the functional group. Each type of functional derivative has its own analysis and naming procedure, and these require three general approaches to processing:

* anhydrides require the functional derivative to be described as a parent plus substituents; to simplify the algorithm, only readily identifiable anhydrides are handled: symmetrical anhydrides and unsymmetrical anhydrides in which the secondary acid is a functional parent or a simple monobasic acid (specifically, a ring or carbon chain molecular skeleton with one acid group); extension of the procedure to handle polyfunctional acids or other molecular skeletons would not be cost effective, since it would increase the algorithm's capabilities only slightly but would require a very complex secondary acid parent candidate generation procedure followed by a complete implementation of the 13 principles;

* esters of acid parents require the esterifying alcohol residue of the functional derivative to be described as a radical; almost all of the analysis and naming can be done by use of the two substituent processing procedures, with the ester procedure itself simply supervising;

* hydrazides, hydrazones, and oximes are described by modification terms with substituent prefixes, and their procedures are merely greatly simplified versions of the chain substituent processor.

The nomenclature rules governing locants state explicitly when locants are to be cited, allowing locants to be omitted only when there would clearly be no ambiguity in the resulting name. The implementation of these rules was simplified by the creation of an "inverted" set of rules that specified when locants were not to be cited. The name generation procedures thus assume that locants will be cited, and all contain a number of locant suppression routines to identify situations where locants are not needed. Most of these routines are quite complex, since they must determine whether the substructure under consideration is symmetric in some respect; the presence of symmetry is the general indication that a locant may not be needed. Examples of names and structures in which locants can be suppressed (i.e., where no rule states that locants must be cited) are shown in Figure 21.

**Name Assembly.** The name assembly routine builds a name for the nomenclature tree structure it is given, determining the appropriate name fragment for each node in the tree and combining these fragments with proper ordering, locants, and punctuation. It consists of four major subroutines to handle index parent, substituent, modification, and stereochemistry portions of the name.

Most processing follows a basic three-step approach: collect the various nomenclature bits and pieces describing the node and its substituents, select the appropriate assembly rule, and assemble the bits and pieces according to the rule.

The various assembly rules are the core of the name assembly procedure. The typical "rule" is really an ordered list of nomenclature data items so that assembly consists primarily of building a character string by retrieval and concatenation of specified substrings in the required order. The substrings may be punctuation elements, locants, multiplier prefixes, name fragments built for higher order nodes, or name terms (name root, bonding suffix, etc.) describing the node being processed.

The name terms used by the name assembly procedure come from four major sources:

* Ring names are built by the addition, as necessary, of indicated hydrogen or multiple bond specifications and name or bonding suffixes to name stems derived from the ring parent names retrieved from the ring reference file.

* Chain names are generated by specific procedures for the four types of chains and are essentially concatenations of length, node, and bond descriptors retrieved from tables.

* Functional parent, principal functional group, and hetero unit names are retrieved from tables.

* Special case and exception procedures are present in many places throughout the assembly procedure; these handle, for example, the replacement of the systematic "methyloxy" by "methoxy" and "benzenyl" by "phenyl".

The index parent name is defined by the top node of the nomenclature tree structure that was built during analysis. The index parent is usually a molecular skeleton, a ring, ring assembly, conjunctive unit, or chain, to which functional groups may be attached. It is named by first naming the molecular skeleton via a provided ring name or a generated chain name and then, if there are functional groups, adding locants as needed and appending the appropriate functional suffix. If the index parent is a functional parent, its name is retrieved from a table.

A substituent node is named by combining a base name stem, which may be a provided ring name, a generated chain name, or a retrieved hetero unit name, with a suffix describing its bonding to the node above it, and then, if there are substituent nodes attached below it, adding their names and locants, properly ordered, as prefixes. When all substituent nodes attached to the parent have been named, their names and locants are similarly ordered and combined to give the substituent portion of the index name.

Modification nodes are named in several ways; the details of the naming procedure differ greatly, since they depend on the type of functional derivative being handled—acid anhydride, ester, hydrazone, or oxime, etc. The general techniques

are similar to those described above, with the final assembly rules bringing in the descriptive modification terms such as "monoanhydride with" or "ester".

## PLANS FOR THE FUTURE

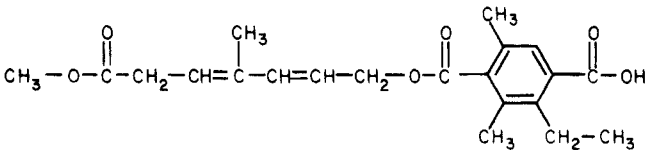
Developing the systematic organic name generation algorithm was a substantial task, with 6 work years resulting in about 3000 pages of algorithm procedures and descriptive material. Programming the algorithm will likewise require a sizeable effort, estimated as 8–10 work years. Once the programming is completed, a period of operation of the program in parallel with the manual naming process will be used to discover and correct errors in the algorithm and its implementation. We expect that the algorithm's accuracy in analysis and naming will be sufficient for it to substantially reduce the effort needed to create CA Index Names for organic substances while maintaining the present quality of indexing.

## ACKNOWLEDGMENT

The development of the CAS Chemical Registry System was substantially supported by the National Science Foundation, which also supported a portion of the initial CAS investigations into algorithmic name generation. Chemical Abstracts Service, a division of the American Chemical Society, gratefully acknowledges this support.

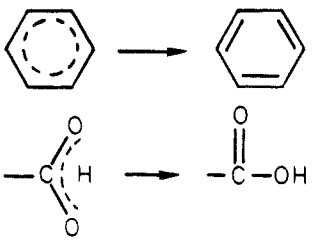
## APPENDIX—AN EXAMPLE OF ALGORITHMIC NAME GENERATION

Structure to be named:

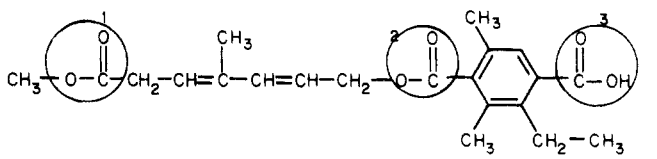


(1) Connection tables and data structures are set up.

(2) Denormalization fixes acyclic tautomer bonds and benzene ring bonds, regenerating fixed single and double bonds from the normalized bonds of the input Registry III structure record.

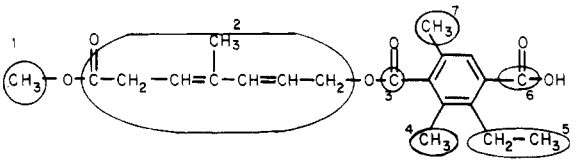


(3) Functional group identification finds the principal functional group to be "carboxylic acid"; there are three principal groups present.



The data generated for each group includes its identity, the atoms comprising it, the "base" atom to which it is attached, and data about any substituents or functional derivatives that might be present. Here, principal groups No. 1 and No. 2 have ester functional derivatives. Additional data for each group, such as name fragments (here, "carboxylic acid" and "oic acid"), reside in reference tables.

(4) Chain domains are identified. The structure has seven carbon domains, but no "a"-chain, homogeneous hetero, or heterogeneous hetero domains and no acceptable chalcogen domains.

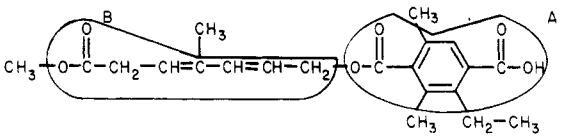


(5) The ring name procedure retrieves the ring parent names cyclohexane and benzene for the $C_6$ ring, corresponding to the "normal" and "maximally unsaturated" forms of the ring. Analysis of the ring as present in the structure indicates that benzene should be used.

(6) Since there is only one ring system present in the structure, there can be no ring assemblies.

(7) The 13 principles for index name selection are applied in order and as necessary.

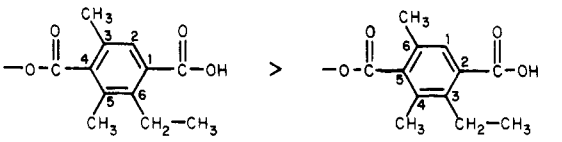
(a) Principle 1—"maximum number of principal functional groups". Possible parents are



Candidate A has two principal groups and is preferred over candidate B with one. (Due to look-ahead tactics, candidate B is never actually generated.)

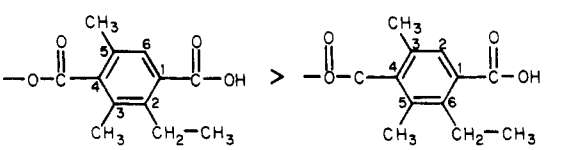
(b) Since there is just one candidate, application of principles 2–7 is not necessary.

(c) Principle 8—"lowest locants for. . ."; the only applicable subprinciple is "8.2"—"lowest locants for principal groups". Since the parent candidate has 12 alternative locant numberings, principle 8.2 will be applied to eliminate the less-preferred numbering schemes. The four schemes that are retained have principal groups at locants 1 and 4, while the eight schemes discarded have higher locants such as 2 and 5, etc.



(d) Since there is just one candidate, principles 9 ("centrality") and 10 ("maximum number of substituents") are not applicable.

(e) Principle 11—"lowest locants for substituents". Since the parent candidate has four alternative locant numberings, principle 11 will be applied to eliminate the less-preferred numbering schemes. The scheme that is retained has substituents at locants 2, 3, and 5, while the three schemes discarded have higher locants such as 3, 5, and 6, etc.



(f) Since there is just one candidate, principle 12 ("multiplication") is not applicable.

(8) Principle 13—"earliest index position". At this point, a name generator is used to build a nomenclature tree describing the structure so that a name can be built for it by the

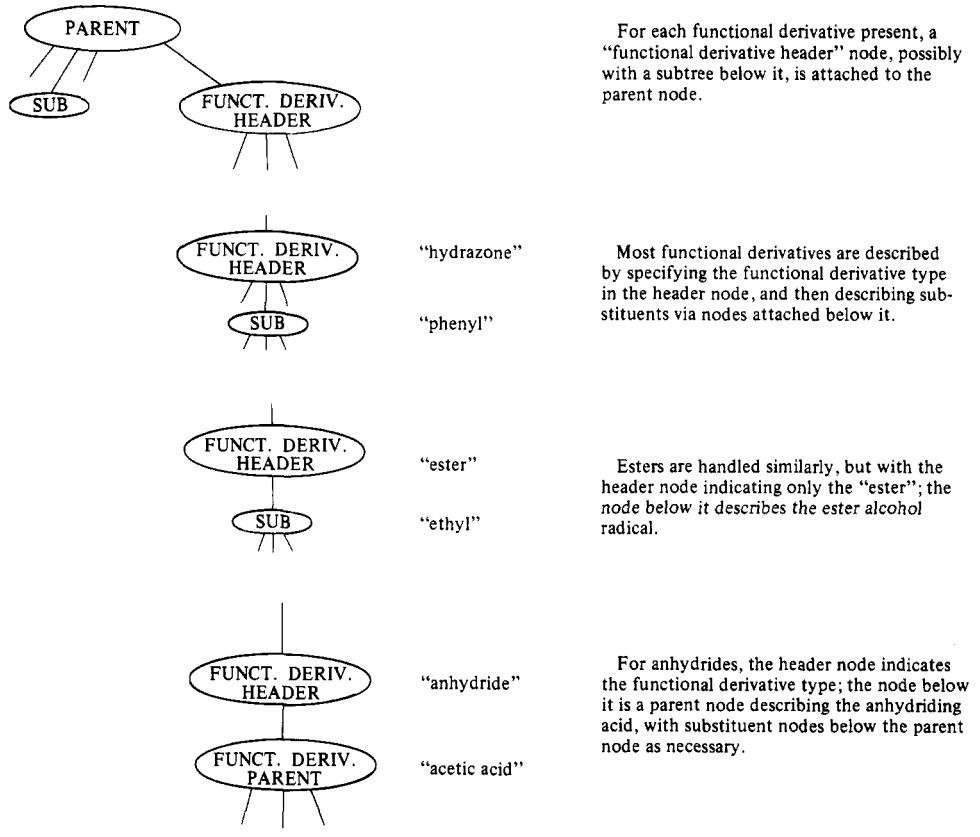
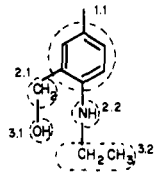



**Figure 19.** Nomenclature tree data structures for functional derivatives.

The sequence of processing steps for the substituent subtree shown is as follows:

analyze node 1.1: phenyl
analyze node 2.1: methyl
analyze node 3.1: hydroxy
name node 3.1: "hydroxy"
name node 2.1 + subtree: "3 - (hydroxymethyl)"
analyze node 2.2: amino
analyze node 3.2: ethyl
name node 3.2: "ethyl"
name node 2.2 + subtree: "4 - (ethylamino)"
*name node 1.1 + subtree: "4 - (ethylamino) - 3 - (hydroxymethyl)phenyl"*




Shown schematically, the processing sequence corresponds to a preorder scan of the tree. Nodes are built and analyzed (A) while going down the tree, and named (N) when coming back up.

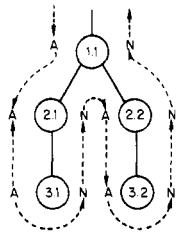
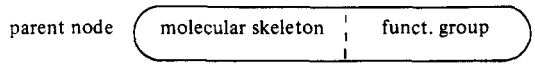



**Figure 20.** Substituent processing.

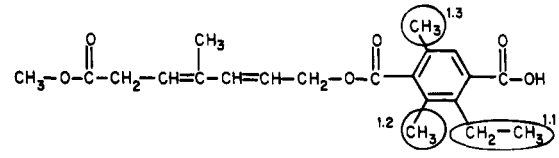
name assembly routine. The nomenclature tree is begun with a parent node describing the index heading parent. The data describing the parent has been collected during the application of the nomenclature principles. Initial nomenclature tree structure:




parent data: parent type = ring system + principal group
ring system name = benzene

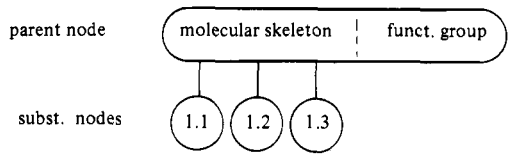
block: principal group multiplier = di
principal group suffix = carboxylic acid
principal group locants = 1,4

Substituents are then identified and processed; a substituent node is generated for each substituent and added below the parent node in the nomenclature tree.

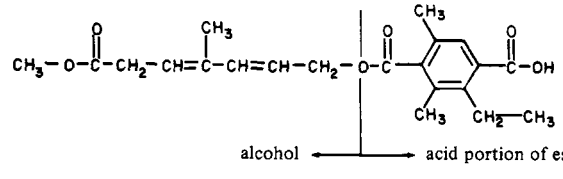


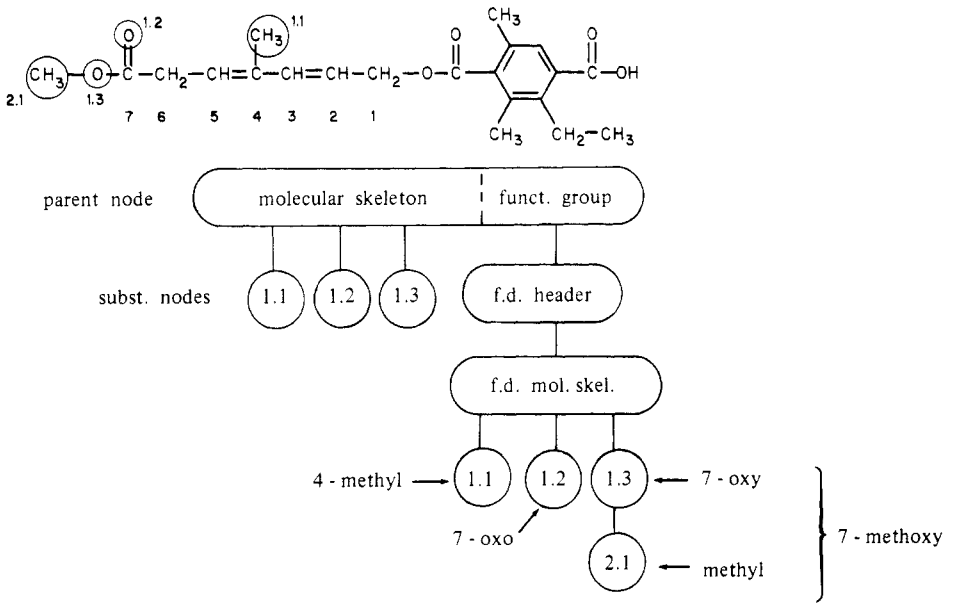

typical substituent data block:
substituent node 1.1
type = carbon chain
length = 2
attached to parent node 0
attached to locant 2
attached by single bond

Nomenclature tree structure so far is



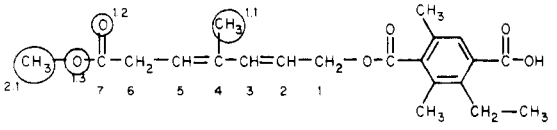

(9) Functional derivatives are identified and processed. Since the functional derivative is an ester, the sought-for "functional derivative parent" is the best molecular skeleton attached to the "link" oxygen of the esterified acid group.



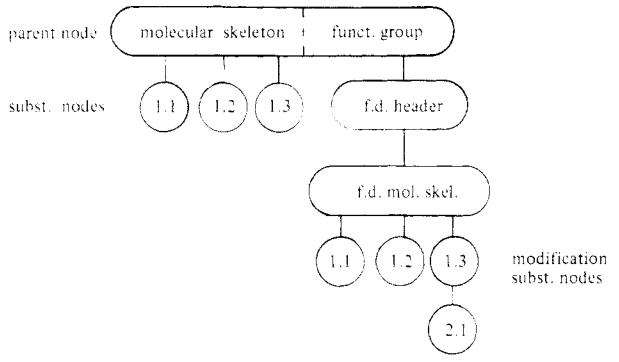



(a) The functional derivative parent is found to be a 7-carbon chain. A functional derivative header node is attached below the parent node, describing the functional derivative as an ester; below it is attached a functional derivative parent node, describing the 7-carbon chain molecular skeleton.

(b) Substituents of the functional derivative parent are identified and processed.

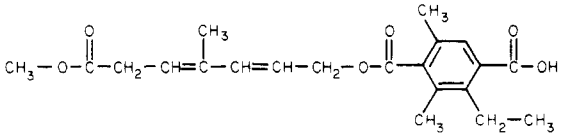


Final nomenclature tree structure:




(c) Locant suppression procedures decide that the functional derivative locant cannot be suppressed—since there are two principal groups, only one functional derivative, and the parent is not symmetric, a locant must be used.

(10) The name assembly procedure builds the name for the structure. Processing of the nomenclature tree begins with the parent node, then handles its substituent nodes, and finally deals with the functional derivative subtree.



(a) **Parent.** The parent is a cyclic molecular skeleton with attached principal groups. The naming data created by the analysis phase provides the following information:

| | |
|---|---|
| ⟨molecular skeleton name⟩ | = benzene |
| ⟨principal group multiplier⟩ | = di |
| ⟨principal group suffix⟩ | = carboxylic acid |
| ⟨principal group locants⟩ | = 1,4 |

The assembly rule selected is

⟨principal group locants⟩-⟨molecular skeleton name⟩ ⟨multiplier⟩ ⟨suffix⟩

The parent naming procedure concatenates the name terms in the order specified, capitalizing the first alphabetic character of the molecular skeleton name, to get

1,4-Benzenedicarboxylic acid

(b). **Substituent.** Each of the substituents is named, and these names are then ordered and combined to get the substituent portion of the name. For the ethyl radical at the 2-position, naming is as follows:

| analysis data: | | naming data: | |
|---|---|---|---|
| attached to locant 2 | | ⟨locant⟩ | = 2 |
| type: carbon chain<br>length: 2 atoms | } | ⟨chain stem⟩ | = eth |
| unsaturation: none | | ⟨unsaturation⟩ | = (null) |
| attached by single bond | | ⟨bond suffix⟩ | = yl |

The assembly rule is

⟨substituent term⟩ = ⟨chain stem⟩ ⟨unsaturation⟩ ⟨bond suffix⟩

The generated substituent name is

⟨locant⟩ = 2 ⟨substituent term⟩ = ethyl

Naming of the two methyl groups is similar. Ordering of the substituent terms leads to recognition of the duplicate methyl terms, and these are combined with use of a multiplier. The resulting substituent portion of the name is

2-ethyl-3,5-dimethyl-

(c) **Modification.** The functional derivative parent is named

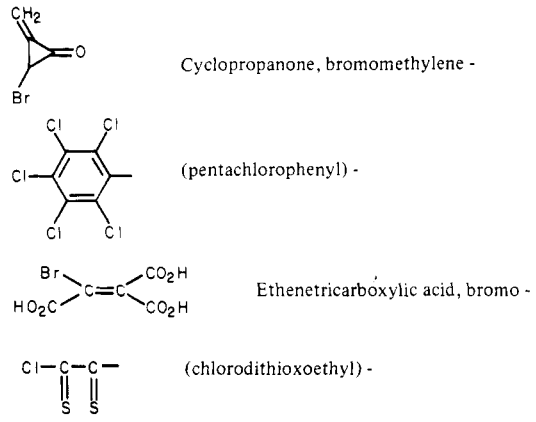
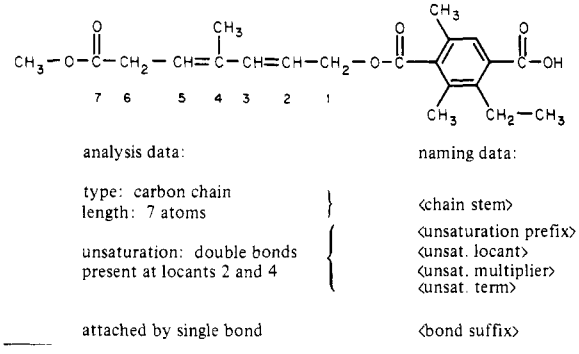



**Figure 21.** Examples of locant suppression.

as a radical and prefixed with its substituent terms, as required by the rules for naming ester derivatives.




The assembly rule is

<unsat. loc.> - <chain stem> <unsat. prefix> <multiplier> <term> <suffix>

The functional derivative parent name is

2,4-heptadienyl

Substituents on the "functional derivative parent" are as shown:

Each substituent is named, and these names are then ordered and combined. During the processing, the attachment of a "methyl" to an "oxy" is recognized as a special case, and the term "methoxy" is substituted for the compound radical name "methyloxy". The functional derivative substituents prefix is

7-methoxy-4-methyl-7-oxo-

The full functional derivative description is

7-methoxy-4-methyl-7-oxo-2,4-heptadienyl

The modification portion of the name includes the locant specifying the modified principal group and an "ester" descriptor, and is

4-(7-methoxy-4-methyl-7-oxo-2,4-heptadienyl) ester

(d) Stereochemistry. Given the "2:*Z*,*Z*" text descriptor from the structure's Registry III record, the stereochemistry naming procedure creates

(*Z*,*Z*)-

(11) The full name of the structure is

1,4-Benzenedicarboxylic acid, 2-ethyl-3,5-dimethyl-, 4-(7-methoxy-4-methyl-7-oxo-2,4-heptadienyl) ester, (*Z*,*Z*)-

## REFERENCES AND NOTES

(1) Dittmar, P. G.; Stobaugh, R. E.; Watson, C. E. "The Chemical Abstracts Service Chemical Registry System. I. General Design". *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 111–121.
(2) Rowlett, R. J., Jr.; Tate, F. A. "Computer-Based System for Handling Chemical Nomenclature and Structural Representations". *J. Chem. Doc.* **1972**, *12*, 125–128.
(3) Donaldson, N.; Powell, W. H.; Rowlett, R. J., Jr.; White, R. W.; Yorka, K. V. "Chemical Abstracts Index Names for Chemical Substances in the Ninth Collective Period (1972-1976)". *J. Chem. Doc.* **1974**, *14*, 3–15.
(4) Rowlett, R. J., Jr. "*CA* [*Chemical Abstracts*] Nomenclature". *Chem. Eng. News* **1975**, *53* (18), 3, 46–47.
(5) "Chemical Abstracts Index Guide", 1977, Appendix IV.
(6) Vander Stouw, G. G.; Gustafson, C.; Rule, J. D.; Watson, C. E. "The Chemical Abstracts Service Chemical Registry System. IV. Use of the Registry System to Support the Preparation of Index Nomenclature". *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 213–218.
(7) LeMasters, J. T.; Vander Stouw, G. G., paper in preparation.
(8) Mockus, J.; Stobaugh, R. E. "The Chemical Abstracts Service Chemical Registry System. VII. Tautomerism and Alternating Bonds". *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 18–22.
(9) Conrow, K. "Computer Generation of Baeyer System Names of Saturated, Bridged, Bicyclic, Tricyclic, and Tetracyclic Hydrocarbons". *J. Chem. Doc.* **1966**, *6*, 206–212.
(10) Van Binnendyk, D.; MacKay, A. C. "Computer-Assisted Generation of IUPAC Names of Polycyclic Bridged Ring Systems". *Can. J. Chem.* **1973**, *51*, 718–723.
(11) Dittmar, P. G.; Mockus, J.; Couvreur, K. M. "An Algorithmic Computer Graphics Program for Generating Chemical Structure Diagrams". *J. Chem. Inf. Comput. Sci.* **1977**, *17*, 186–192.