The Generation and Logical Bubble-Up of Ring Screens for Structurally Explicit Generics. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 215-224.
(2) Shenton, K.; Norton, P.; Langdon, M. L.; Fearns, E. A. Graphical Retrieval of Patent Information. In *Proceedings of the 9th International Online Meeting, Learned Information*, London, Dec 1985; Learned Information: Oxford, 1986.
(3) Shenton, K.; Norton, P.; Fearns, E. A. Generic Searching of Patent Information. In *Chemical Structures. The International Language of Chemistry*; Worr, W. A., Ed.; Springer-Verlag: Berlin, 1988; pp 169-178.
(4) Fisanick, W. Storage and Retrieval of Generic Chemical Structures in Patents. U.S. Patent 4642762, Feb 1987.
(5) Fisanick, W. The Chemical Abstracts Service Generic Chemical (Markush) Structure Storage and Retrieval Capability. 1. Basic Concepts. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 145-154.
(6) Stiegler, G.; Maier, B.; Lenz, H. Automatic Translation of GENSAL Representations of Markush Structures into GREMAS Fragment Codes at IDC. Presented at the Fall ACS Meeting, Washington, DC, 1990.
(7) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 2. GENSAL. A Formal Language for the Description of Generic Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 151-161.
(8) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 4. An Extended Connection Table Representation for Generic Structures. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 160-164.
(9) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 6. An Interpreter Program for the Generic Structure Language GENSAL. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 66-70.
(10) Welford, S. M.; Ash, S.; Barnard, J. M.; Carruthers, L.; Lynch, M. F.; von Scholley, A. The Sheffield University Generic Chemical Structures Project. In *Computer Handling of Generic Chemical Structures*; Barnard, J. M., Ed.; Gower Publishing Company Ltd.: Aldershot, U.K., 1984; pp 130-158.
(11) Gordon, J. E.; Brockwell, J. C. Chemical Inference. 1. Formalization of the Language of Organic Chemistry: Generic Structural Formulas. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 117-134.
(12) Morris, C. W. *Foundations of the Theory of Signs*, 8th ed.; University of Chicago Press: Chicago, IL, 1953.
(13) Gillet, V. J.; Downs, G. M.; Ling, A. I.; Lynch, M. F.; Venkataram, P.; Wood, J. V.; Dethlefscn, W. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 8. Reduced Chemical Graphs and Their Applications in Generic Chemical Structure Retrieval. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 126-137.
(14) Welford, S. M.; Barnard, J. M.; Lynch, M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 5. Algorithmic Generation of Fragment Descriptors for Generic Structure Screening. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 57-66.
(15) Wittgenstein, L. Philosophische Untersuchungen. Blackwell: Oxford, 1953.
(16) Carnap, R. *Meaning and Necessity, A Study in Semantics and Modal Logic*. University of Chicago Press: Chicago, IL, 1947.
(17) Frege, G. Uber Sinn und Bedeutung. *Z. Philosophie Philosophische Kritik* **1892**, *100*, 25-50.
(18) Peirce, C. S. In *Collected Papers*; Hartshone, C., Weiss, P., Eds.; Harvard University Press: Cambridge, MA, 1931-1935; Vols. I-VI.
(19) Mockus, J.; Stobaugh, R. E. The Chemical Abstracts Service Chemical Registry System. 7. Tautomerism and Alternating Bonds. *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 18-22.
(20) Welford, S. M.; Barnard, J. M.; Lynch, M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 3. Chemical Grammars and their Role in the Manipulation of Generic Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 161-168.
(21) Church, A. *Introduction to Mathematical Logic*; Princeton University Press: Princeton, 1967; Vol. 1.

Computer Storage and Retrieval of Generic Chemical Structures in Patents. 12. Principles of Search Operations Involving Parameter Lists: Matching-Relations, User-Defined Match Levels, and Transition from the Reduced Graph Search to the Refined Search

WINFRIED DETHLEFSEN
BASF, Ludwigshafen/Rhein, Germany

MICHAEL F. LYNCH,* VALERIE J. GILLET, GEOFFREY M. DOWNS, and JOHN D. HOLLIDAY
Department of Information Studies, University of Sheffield, Sheffield S10 2TN, England

JOHN M. BARNARD
Barnard Chemical Information Ltd., 46 Uppergate Road, Stannington, Sheffield S6 6BX, England

Received November 16, 1990

This paper continues the establishment of a consistent framework for discussing and treating representations of generic structures described in Part 11 of this series (see preceding paper in the issue). In this part, the nature of search operations and the use of parameter lists representing both specifically and generically described parts within generic full structures as the basis for the matching operation prior to the refined search on the ECTR are considered. The nature of the matching-relations is identified, together with the concept of matching-paths for query structures in file structures. The possibility of extension of the matching operations in order to implement user-defined levels of search and a refined search are also discussed.

1. INTRODUCTION

The complexities associated with the retrieval of generic structures require a number of different levels of representation in order to provide efficient search. Consequently, several levels have been implemented, including fragment[1] and ring screens, and reduced graphs together with the use of parameters. The exact order of application of these screens is a matter of technical efficiency and may vary according to query and file characteristics. Ring screens have already been reported for the representation of full structures.[2] Reduced graphs were introduced as an additional screen,[3] but their role is now seen as being two-fold: both as screens and also as providing a preliminary mapping between the query structure and file structures, enabling more detailed comparisons to be carried out on the corresponding query and file partial structures via their ECTR representations. Parameters have so far been described for the representation of generically described partial structures, but in fact, they also provide a universally applicable concept that may be applied to other representations as well, as described below.

2. THE UNIVERSAL APPLICABILITY OF PARAMETER LISTS

As noted earlier, structural information relating to generic nomenclatural terms is expressed in GENSAL by a parameter list consisting of parameter values.[1] (The initial set of parameters is tentative and may be altered in a future system.) These values are numerical indications of the status of certain structural features and are referred to as **Enumerative Parameter Values (EPVs)**. Thus, in the parameter list of "C4–8 alkadienyl", the obligatory presence of 4–8 carbon atoms (C) and of two double bonds (E), the possible presence of ternary branching (T), and the obligatory absence of heteroatoms (Z) are reflected in the corresponding parameter values C⟨4–8⟩ E⟨2⟩ T⟨0–⟩ Z⟨0⟩. The structural features described by the parameters C, E, T, and Z form a subset of the whole parameter list, which contains further parameter values for the set of structural features selected for inclusion in GENSAL. At another level of description, the molecular formula range C⟨4–8⟩ Z⟨0⟩ characterizing the node in a reduced graph generated from "C4–8 alkadienyl" is also an excerpt from a parameter list, as described by Gillet et al.[3]

A specific partial structure may also be represented by a parameter list by reducing its specificity, by analogy with the treatment of a generically described partial structure. The parameter list can be derived from the partial connection table and is termed a specific (or **specific-derived**) **parameter list** to differentiate it from the usual generic parameter list. For instance, the specific parameter list for "$-CH{=}C(CH_3)-CH{=}CH_2$" contains the parameter values C⟨5⟩ E⟨2⟩ T⟨1⟩ Z⟨0⟩.

This specific parameter list then can easily be compared with a generic parameter list, e.g., with the list for "C4–8 alkadienyl". In this case, the specificity of the representation of the *specific* partial structure is reduced just to that level which is the best attainable level for *generic* partial structures denoted by h-variant expressions.[4] This level, therefore, is sufficient for a final decision on match or mismatch between both structures. This principle is also applicable to the comparison of simple full expressions, e.g., "$CH_2{=}C(CH_3)CH{=}CH_2$" and "C4–8 alkadiene". (Detail of the implementation of such comparisons is given in Section 5.)

Furthermore, not only structures denoted by simple expressions can be represented by a parameter list but also structures denoted by composite (full or partial) expressions, i.e., structures which are denoted by the whole AND/OR tree of an ECTR or by a part of it. The values of this parameter list are derived from the parameter values of the respective generic and specific parameter lists of the simple expressions within the AND/OR tree, by means of the bubble-up process, which has already been described for the accumulation of ring screens.[2] Ultimately, even a full generic structure of arbitrary complexity may be represented by means of a single parameter list, though at a low level of specificity. In summary, all kinds of partial structures can be represented and compared by means of parameter lists.

Such parameter lists may be used more advantageously for a screening comparison of those parts of the query and file ECTRs that have been recognized already as corresponding and possibly matching parts by other methods. In particular, therefore, complete parameter lists (or suitable excerpts) can be used as lists of node descriptors (termed **node parameter lists**), assigned to the nodes of the reduced graphs of query and file structures.

If a node of a reduced graph is derived solely from one distinct connection table or parameter list in the ECTR, then simply the respective specific or generic parameter list is used as the node parameter list. If a node of a reduced graph is constituted by the collapse of two or more partial structures, which may be connected structures (logical relationship: AND) and/or alternative structures (logical relationship: OR), then structural information for this node is accumulated, through the bubble-up process, by conflating the specific or generic parameter lists representing these partial structures. (Further details of the generation of reduced graph nodes are given in the following paper in this issue.) For instance, the nonring node which is generated by the collapse of the two alternative substituents "C1–7 alkoxy" (C⟨1–7⟩ E⟨0⟩ T⟨0–⟩ Z⟨1⟩) and "$-CH{=}C(CH_3)-CH{=}CH_2$" (C⟨5⟩ E⟨2⟩ T⟨1⟩ Z⟨0⟩) may be specified by an accumulated node parameter list with parameter values of C⟨1–7⟩ E⟨0,2⟩ T⟨0–⟩ Z⟨0–1⟩.

Node parameter lists can be compared by simple examination of the corresponding parameter values during matching operations. Consider the following example of node–node comparison of the node (a) for "C4–8 alkadiene" in a reduced query graph with the collapsed node (b) for "C1–7 alkoxy/$-CH{=}C(CH_3)-CH{=}CH_2-$" in a reduced file graph, showing that there is a match for each pair of parameter values, which is identity (for T), strict inclusion (for E and Z), or intersection (for C) between integer ranges. These matches result in an intersection between the whole parameter lists:

	(a)	C⟨4–8⟩	E⟨2⟩	T⟨0–⟩	Z⟨0⟩
compared with	(b)	⟨1–7⟩	⟨0,2⟩	⟨0–⟩	⟨0–1⟩
yields the intersection		⟨4–7⟩	⟨2⟩	⟨0–⟩	⟨0⟩

Details of the principles of such matches between parameter lists are given in the following sections. Attention must be paid to the fact that this match is merely a **node match** in a **node search**, i.e., in a search at the reduced screen level of representation given by reduced graphs the nodes of which are specified by node descriptors. In this example, the node match corresponds to a genuine **structure match**, which could be determined definitely only in a subsequent **refined search** at the level of the ECTR. But in other cases [e.g., if the same node parameter list as for node (b) is derived from "C1–7 alkyl/$-O-CH{=}CH-CH{=}CH_2-$"], a node match is not followed by a structure match.

Instead of this exact comparison of parameter lists on an algebraic basis, i.e., exact comparison of EPVs, the comparison can also be performed at a further reduced level of representation and implemented as a rapid bit string operation on parameter lists, with bit combinations representing **Reduced Parameter Values (RPVs)**. An indication of the status of the respective structural feature is given by a simple bit combination, which indicates whether this feature is obligatorily present, possibly present, or obligatorily absent. RPVs are an implementation of the concept of **Determinant-Screens**, as discussed in Section 5.4 of the preceding paper in this issue.

Parameters provide many useful functions within the GENSAL system; RPVs or EPVs are used as screens for the nodes of reduced graphs and for other representations. EPVs allow the implementation of user defined levels of search, and they can provide a definitive match for special types of partial structures. The latter two applications of parameters are discussed more fully following the details of the matching-relations for generic structures.

3. MATCHING-RELATIONS

The matching-relations that exist between *specific* query and *specific* file structures are well known; they are the relation of identity between molecular graphs and the relation of substructural embedment of one graph in the other. However, when either the query or the file structure, or both, are *generic* structures, being possibly composed of various partial structures, still other matching-relations must be taken into account in a differentiating manner. These matching-relations can exist between full structures and between partial structures, e.g., between the query and file partial structures from which query and file nodes are derived. Where such correspondences are made between partial structures of the query and a file structure, a *full* structure match is possible only if particular matching relations exist between the corresponding *partial* structures. For example, the matching relation of intersection between the generic full structures represented by the expressions "C_6H_5-O-alkyl" and "aryl-O-CH_3" is constituted by the matching relation of strict inclusion of phenyl within aryl and of methoxy within alkoxy.

As discussed already in Section 4.3 of the preceding paper in this issue, matching-relations may be conceived as relations between structures or, equivalently, between expressions; further, class relations are to be distinguished from the special matching-relation of substructual embedment. Only class relations between (full or partial) structures (or expressions) are described below, in a heuristic manner, from a structural point of view. (The principles of the algorithmic determination of these relations by comparison of expressions is discussed in Sections 4 and 5.) The concept of the relation of substructural embedment between specific (partial) structures can be extended in an analogous manner, by combination with class relations, to the concept of substructural embedment of a specific query (partial) structure in a generic file (partial) structure, e.g., of C_6H_5-O-CH_2- in C_6H_5-O-alkyl. This is not discussed here.

The matching-relation of identity (or, more precisely, **extensional identity**) as applied to *specific* (full or partial) structures is that of graph isomorphism. It is determined as an atom-by-atom match of (partial) connection tables. For *generic* structures, the extremely "strong" matching-relation of extensional identity alone is usually not required in practice. Because this would mean for generic *full* structures that the full file structures required must have exactly the same extension as the query structure; and for the generic *partial* expression "alkyl" in a query formulation, only "alkyl" would be the required file expression, but not an expression like "C1–5 alkyl" or "*t*-butyl". For generic structures, therefore, identity of specific partial structures is usually required only for a part of the query structure, in combination with "weaker" matching-relations for the other parts.

The matching-relation of **strict inclusion** of a specific query within a generic file structure implies that the query structure is identical with one of the specific structures embodied in the generic file structure. Strict inclusion of a generic query within a generic file structure implies that every specific which is a member of the query class of specifics is also a member of the file class *and* that at least one member of the file class is *not* a member of the query class. The relation can also be applied inversely such that a specific or a generic file structure is strictly included within a generic query structure. The expressions "*t*-butyl", "C4–8 *t*-alkyl", "*t*-alkyl", "alkyl", and "hydrocarbyl" form a series of strict inclusions, where each including structure (or expression) is a **strict structural generalization** of the included structure(s) [or expression(s)].

Usually, for full and partial structures, strict inclusion *or* identity is required, rather than strict inclusion alone. This matching-relation of *nonstrict* inclusion is termed **subsumption** to make a clear distinction. Thus, "*t*-alkyl" subsumes "*t*-butyl", "C4–8 *t*-alkyl", and "*t*-alkyl"; it is subsumed in "*t*-alkyl", "alkyl", and "hydrocarbyl".

Intersection of a generic query with a generic file structure exists if at least one specific is included by the query and also by the file structure, but the query and file structure are not related by either identity or strict inclusion. Examples: "C_6H_5-O-alkyl" and "aryl-O-CH_3" intersect by having the structure denoted by "C_6H_5-O-CH_3" in common; "C1–4 alkyl" and "C4–8 alkyl" intersect by having the four isomeric butyl radicals in common; "hydrocarbyl" and "cyclyl" intersect by having the carbocyclic radicals in common.

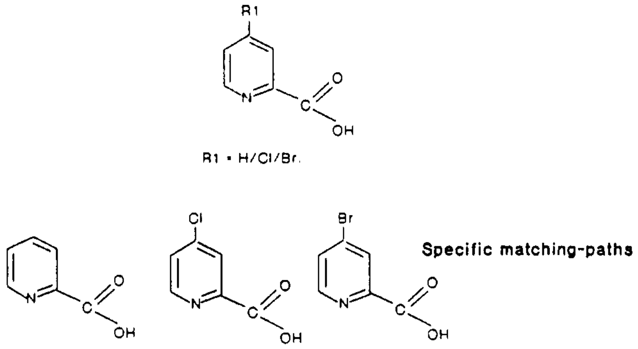

Figure 1. Specific matching-paths.

Usually, again, it is not intersection alone that is required as the matching-relation, but the matching-relation required is that of **community** or **common membership**. It exists if there is either identity or strict inclusion or intersection, i.e., if the query and file structure have at least one specific in common. Community is the most general level of search, and it is the usual matching-relation required for generic *full* structures. In most cases of generic full query structures then only intersecting generic full file structures are retrieved. For *partial* structures within full structures, however, the more restrictive matching relation of subsumption may be utilized for operating user-defined match levels, or even the most restricted level of identity may be required.

Finally, a query and file structure are **strange**, i.e., they do not match, if none of the previous matching-relations apply, i.e., if there is neither identity nor inclusion, and not even intersection.

4. MATCHING-PATHS

As discussed above, the matching-relations can be applied both to full structures and to the partial structures contained within full structures. Where generic full structures are concerned, it is useful to consider the concept of **matching-paths** through the query and file structures (or expressions) and to apply the matching-relations to these matching-paths. A non-h-variant generic full expression (previously referred to as a structurally explicit generic full expression by Downs[2]), i.e., a generic full expression containing exclusively specific partial expressions that are combined by p-, s-, or f-variation, can be resolved into the finite set of all those specific full expressions which are implied by the generic expression. This set is referred to as the set of all implied **specific matching-paths** through the generic full expression (or structure); an example is given in Figure 1 for an s-variant expression.

An h-variant generic full expression is a generic formula that contains at least one h-variant partial expression. Here only the p-, s- and f-variations can be resolved to give a finite set of full expressions. This set *may* contain *specific* full expressions (i.e., specific matching-paths through the full generic), but need not. However, at least one of the set must be an *h-variant generic* full expression, which denotes a finite or infinite class of homologous full specifics. This expression is a **generic matching-path** through the whole generic structure or expression. Figure 2 illustrates the specific and generic matching-paths for an h- and s-variant generic full expression. (Strictly speaking, such a path is a matching-path merely by potentiality, because it can factually become a *matching*-path in a strict sense only in a concrete comparison with paths in another structure.)

For a specific full query structure there are three different matching-criteria, given in terms of matching-relations and matching-paths:

(a) Identity with a specific full file structure.

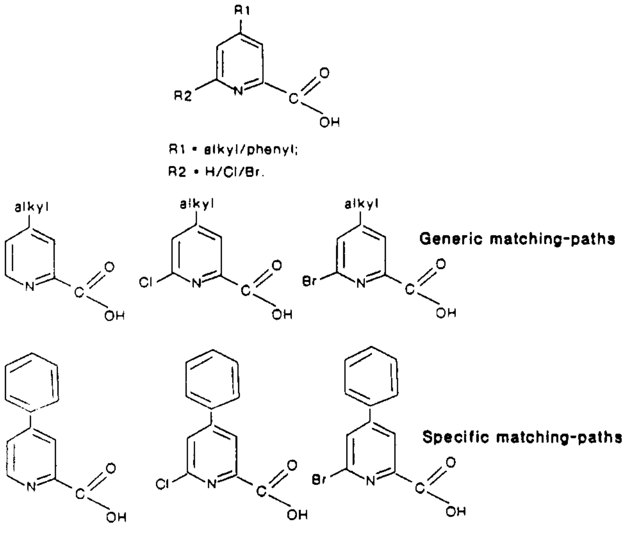

Figure 2. Specific and generic matching-paths.

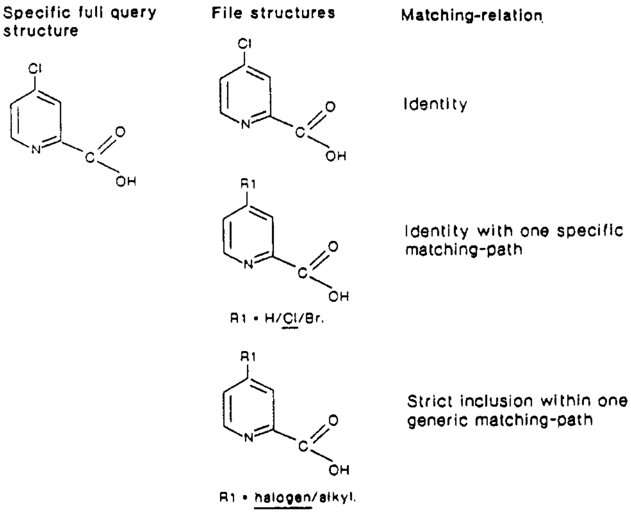
Specific full query structure
File structures
Matching-relation
Cl
Cl
Identity
R1
Identity with one specific matching-path
R1 = H/Cl/Br.
R1
Strict inclusion within one generic matching-path
R1 = halogen/alkyl.

Figure 3. Matching-criteria for a specific full-structure query.

(b) Identity with a specific matching-path through a generic full file structure.

(c) Strict inclusion within a generic matching-path through a generic full file structure.

These criteria are illustrated in Figure 3.

For a generic full query structure, there are also specific and/or generic matching-paths through this query structure that must be compared with matching-paths through the file structures. For a non-h-variant generic full query structure there are, in analogy with specific full query structures, three different matching criteria, illustrated in Figure 4:

(d) Identity of a specific query path with a specific full file structure.

(e) Identity of a specific query path with a specific file path.

(f) Strict inclusion of a specific query path within a generic file path.

For an h-variant generic full query structure, the matching criteria for the specific query paths are as in (d)–(f). The matching-criteria for the generic query paths are:

(g) A generic query path strictly includes a specific full file structure.

(h) A generic query path strictly includes a specific file path.

(i) A generic query path strictly includes a generic file path.

(j) A generic query path is identical with a generic file path.

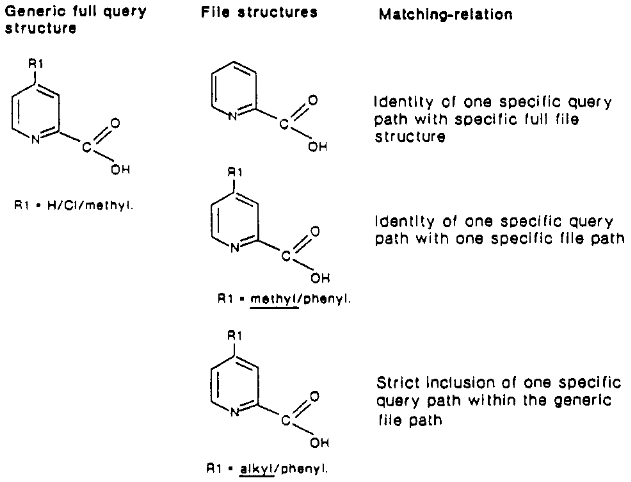
Generic full query structure
File structures
Matching-relation
R1
R1 = H/Cl/methyl.
Identity of one specific query path with specific full file structure
R1
R1 = methyl/phenyl.
Identity of one specific query path with one specific file path
R1
R1 = alkyl/phenyl.
Strict inclusion of one specific query path within the generic file path

Figure 4. Matching-criteria for a non-h-variant generic full-structure query.

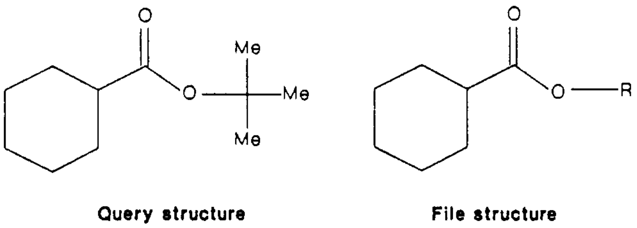

Figure 5. Query structure retrieving file structures for values of R as shown in Table I.

Table I. Values of R for File Structure Retrieved by the Query of Figure 5

(a) R = *t*-butyl/isopropyl
(b) R = *t*-alkyl (4–8)
(c) R = *t*-alkyl
(d) R = alkyl
(e) R = hydrocarbyl

(k) A generic query path is strictly included within a generic file path.

(l) A generic query path intersects with a generic file path.

These matching-criteria may be summarized for a *specific* full query structure as subsumption in the file structure and for a *generic* full query structure as common membership with the file structure.

The matching-criteria can also be applied to reduced graphs and are given in terms of the matching-paths through reduced graphs. A bare reduced graph match (without regard to a possible mismatch of node descriptors) is recorded if, for a *specific* full query, the reduced graph of the query is identical with a given matching-path through a file reduced graph; for a *generic* full query, a matching-path through the query reduced graph must be identical with a given matching-path through a file reduced graph.

5. IMPLEMENTING USER-DEFINED LEVELS OF SEARCH

User-defined levels of search may be implemented by applying matching-relations in a special manner to corresponding *parts* of the query and the file structure. The matching-relation for full structures is usually that of community; thus, the query shown in Figure 5 retrieves generic file structures having the formula indicated with the values of R in Table I.

It may be the case that the searcher in a special search is interested only in structures (a), (b), and (c), whereas he is

Table II. Parameter Lists and Their Values for Structures (a) to (g)

	A	C	E	Y	T	Q	RC	...	RZ	Z
(a1) *t*-butyl	4	4	0	0	1	0	0	...	0	0
(a2) isopropyl	3	3	0	0	0	0	0	...	0	0
(b) *t*-alkyl ⟨4–8⟩	4–8	4–8	0	0	1–	0–	0	...	0	0
(c) *t*-alkyl	4–	4–	0	0	1–	0–	0	...	0	0
(d) alkyl	1–	1–	0	0	0–	0–	0	...	0	0
(e) hydrocarbyl	1–	1–	0–	0–	0–	0–	0–	...	0–	0
(f) *t*-alkyl ⟨5–8⟩	5–8	5–8	0	0	1–	0–	0	...	0	0
(g) query: *t*-butyl	4	4	0	0	1	0	0	...	0	0

not at all interested in (d) and (e) because they are structurally too general for the special purpose of this search. The differences in structural generalization are clearly revealed within the partial structure representations of the ECTR. The alternative partial structures of (a) are represented as partial connection tables whereas (b)–(e) are represented by parameter lists where the different levels of generalization are indicated by differences in the integer ranges of particular parameters as shown in the shortened parameter lists of Table II. Table II also contains the specific-derived parameter lists of the corresponding partial query structure (g) and of the two alternative specifics within (a), *t*-butyl (a1), and isopropyl (a2); and it contains the generic parameter list of a mismatching structure (f), denoted by "*t*-alkyl ⟨5–8⟩". [The structural features evaluated in Table II as parameters are as follows: A = total non-hydrogen atom count, C = carbon atom count, E = alkEne unsaturations (double bonds), Y = alkYne unsaturations (triple bonds), T = ternary branch points within acyclic carbon chains, Q = quaternary branch points, RC = number of rings, RZ = number of ring heteroatoms, and Z = number of heteroatoms.]

Thus, whereas the matching relation for full structures is usually that of community, other matching-relations may be required and can be applied to partial structures within the overall strategy. The identity of two partial structures that are specific, i.e., given as connection tables, is finally determined only by an atom-by-atom search (whereas the nonidentity may be determined already by comparison of specific-derived parameter lists). The relation between two structure parts represented by parameter lists can be determined by examining the relation between the integer ranges of the corresponding parameter values of both lists. The matching-relations which can be applied to integer ranges, in analogy with the matching-relations for (full or partial) structures, are identity, strict inclusion, subsumption, intersection, or community, which is any of the above. (Here and in the following, the term integer range refers not only to genuine *ranges* of integers, but also to those *distinct* integers as they occur in specific-derived parameter lists and may occur in generic parameter lists.) Examples of these relations are

identity	⟨0⟩, e.g., ⟨3–4⟩	is identical to	⟨0⟩ ⟨3–4⟩
strict inclusion	⟨0⟩, e.g., ⟨2–3⟩	is strictly included within	⟨0–⟩ ⟨1–3⟩
subsumption	⟨0⟩, e.g., ⟨0–⟩	subsumes	⟨0⟩ ⟨0–1⟩
intersection	⟨4–⟩, e.g., ⟨1–2⟩	intersects with	⟨3–5⟩ ⟨2–4⟩
community is any of the above			
strangeness	⟨0⟩, e.g., ⟨1–2⟩	is strange with	⟨1–⟩ ⟨3–5⟩

The matching-relations between expressions—discussed in Section 3 in regard to the structural meaning of these expressions—can be determined then for (derivative or genuine generic) parameter lists in an algorithmic procedure; a particular matching-relation between two parameter lists is given if particular matching-relations exist between the corresponding parameter values of both lists:

Identity: two parameter lists are identical if, for every parameter, the corresponding integer ranges are identical.

Strict inclusion: the parameter list A strictly includes the parameter list B, i.e., A is a strict structural generalization of the parameter list B, if the integer range of each parameter of B is subsumed by the integer range of the corresponding parameter in A *and* if this subsumption is a strict inclusion for at least one pair of integer ranges so that the whole B is strictly included by A. Thus, as Table II shows, strict inclusion between generic nomenclatural expressions is exemplified by the following series of broadening generalizations: "*t*-butyl" is strictly included by "*t*-alkyl C⟨4–8⟩" which is strictly included by "*t*-alkyl" which is strictly included by "alkyl" which is strictly included by "hydrocarbyl".

Subsumption: the parameter list A subsumes the list B if, for each parameter, the integer range of A subsumes the corresponding integer range of B.

Intersection: two parameter lists (e.g., those of "C1–4 alkyl" and "C4–8 alkyl", or those of "hydrocarbyl" and "cyclyl") intersect if, for each parameter, the relation of identity, strict inclusion, or intersection exists between the corresponding integer ranges, *and* if additionally at least one of the two following conditions is satisfied: there is, for at least one parameter, intersection between the corresponding integer ranges (in the first example: between C⟨1–4⟩ and C⟨4–8⟩), or the direction of strict inclusion—which is an asymmetric relation—is inverted for different parameters (which is the case in the second example: the heteroatom count for "hydrocarbyl", Z⟨0⟩, *is* strictly *included* within the corresponding value Z⟨0–⟩ for "cyclyl", whereas the ring count for "hydrocarbyl", RC⟨0–⟩, strictly *includes* the value RC⟨1–⟩ for "cyclyl").

Strangeness: two lists are strange with one another if, for at least one parameter, the corresponding integer ranges are strange with one another.

Community: the relation of common membership exists if the two lists are not strange with one another or, more explicitly, if for each parameter subsumption or intersection exists between the corresponding integer ranges.

The relation between a specific partial structure and a generic partial structure represented by a parameter list may be established by generating a specific-derived parameter list for the specific structure. In this case the parameter list has integer values consisting of single integers and can be compared with the parameter list representing the generic partial structure as described above. There is no match if the specific and the generic parameter lists are strange with one another. A match is given if the generic parameter list strictly includes the specific parameter list. According to the respective generic parameter list, this match can be at a very high level of strict structural generalization (e.g., "hydrocarbyl" or "radical"). If generalizations exceeding a particular level are not wanted, then this search purpose can be achieved by utilizing the matching-relation of subsumption: only such generalizations are retrieved that are subsumed by the generic parameter list of a particular, user-defined generalization.

In practice, during a search such as that described in Figure 5 where the search purpose is to exclude such generalizations of "*t*-butyl" as are given in file structures by (d) or (e), the searcher must be able to express (e.g., by the syntactic means discussed in Section 5.8 of the preceding paper in this issue) that (c), i.e., "*t*-alkyl", is the broadest level of generalization which is desired. The search algorithm then retrieves every specific "*t*-butyl" and every parameter list which is, like (b) and (c), subsumed by "*t*-alkyl" and which strictly includes "*t*-butyl" but rejects file expressions like (d), (e), and (f): (d) is rejected because its parameter values A⟨1–⟩ C⟨1–⟩ T⟨0–⟩ are not subsumed by A⟨4–⟩ C⟨4–⟩ T⟨1–⟩ of (c); (e) is rejected for the same reason and additionally because of the values of E, Y, RC, RZ; (f) is rejected because it is not a strict structural generalization of "*t*-butyl", i.e., its integer range ⟨5–8⟩ for A and C does not include the corresponding ⟨4⟩ of "*t*-butyl", although ⟨5–8⟩ is subsumed by the corresponding ⟨4–⟩ of (c).

6. THE TRANSITION FROM THE REDUCED GRAPH SEARCH TO THE REFINED SEARCH

Reduced graphs have previously been introduced as screens for generic structures, where candidate structures are retrieved that match at the reduced graph level of representation, i.e., by a node match in a node search. Node matches, however, do not necessarily correspond with structure matches because of the collapse of information, e.g., the node derived from "alkoxy C1–6" matches with the node derived from "$-CH_2-CH_2-O-CH_3$" but this does not correspond with a structure match in a final, refined search at the level of the ECTR. Of course, in some cases node matches may correspond with structure matches, e.g., the node derived from "alkoxy C⟨1–6⟩" matches with the node derived from "$-O-CH_2-CH_2-CH_3$", and this does correspond with a structure match. The refined search is required to determine which node matches, if any, are also structure matches, and the starting point is the correspondence between query and candidate structures that is already provided as pairs of matching nodes.

The node search, therefore, should not be considered in isolation since it forms a preliminary and preparatory step toward the refined search; the concept and methods of the node search are directed toward the subsequent refined search. For example, the method of representing and comparing structures by means of parameter lists is used in the refined search and in the node search as well, for identifying structure matches and structure mismatches as early as possible in the sequence of search operations.

The following discussion outlines the steps involved in proceeding from the node search to the refined search. In this discussion, to simplify matters, the possibility of applying user-defined match levels is neglected, and structure matches are conceived always as matches between (partial) structures that are related to one another by the matching-relation of community. However, the more restrictive matching-relations, particularly user-defined match levels, can be applied analogously. It should be noted that this structure match is a match in terms of the GENSAL system, i.e., any structural differentiations that are not represented in GENSAL, e.g., stereochemistry, cannot be distinguished in the refined search.

6.1. NODE MATCHES AND STRUCTURE MATCHES

In the limit, a structure match can be determined only by resolving the query and the file node (in a pair of matching nodes) into their constituent distinct partial structures, and by performing an atom-by-atom search where the corresponding partial structures are represented by connection tables or by comparing parameter lists where one of the corresponding partial structures is represented by a nonderivative, genuine generic parameter list. In special cases of pairs of simply constituted nodes, belonging to special derivation-types (cf. Section 5.7 of the preceding paper in this issue), a resolution is not necessary, and the final comparison of parameter lists can be performed already at the level of the node search by means of the EPVs of complete node parameter lists. In these cases, the node match corresponds with the structure match, and the search has come, for this pair of nodes, to a definitive end, anticipated already in the node search.

The stages involved in proceeding from a node search to the refined search can be broadly summarized as:

Identification of a node match for the whole reduced graphs of query and file structures, without regard to a possible mismatch of node descriptors.

Identification of a match for those RPVs that are used as node descriptors (using the complete parameter list or a part of it).

Comparison of EPVs of complete parameter lists.

Evaluating the origins, i.e., the derivation types, of matching nodes to identify

node matches which coincide with structure matches

node matches which fail to satisfy this condition and must be submitted to a refined search

Resolving reduced graph nodes that arise from the superposition of alternative partial structures into correspondingly alternative reduced graph nodes, so that nodes represent either distinct partial structures or partial structures combined by AND logic only. The operations of matching RPVs and EPVs and evaluating the nodes for coincidence of a node match with a structure match can now be repeated.

Refined searching, i.e., the comparison of parameter lists or connection tables, accounting for the mutual attachment of partial structures.

If a mismatch between nodes is identified at any of these stages, then the candidate structure is rejected. The first two stages are described in more detail in a subsequent paper. Comparison of EPVs within reduced graph nodes provides a more detailed level of description and is used to identify structure matches and mismatches as early as possible in the succession of search operations, for example, in those cases where a structure match can already be identified during the node search. The conditions required for the coincidence of a node match with a structure match for a given query node/file node pair are as follows:

Neither of the two nodes in a pair is a-variant, i.e., derived by the collapse of any partial structures which are alternative to one another by p-, s-, or f-variation.

At least one of the nodes must be derived from one simple h-variant expression, i.e., from one parameter list in the ECTR.

EPVs for all parameters of a complete parameter list are used as node descriptors.

The possibility of a node match being coincident with a structure match requires, therefore, that the derivation-type of each of the nodes is known, i.e., the logical relationships between the partial structures which constitute a node, and also how those partial structures are represented: by parameter lists or connection tables or a combination of both. One of the nodes must be derived from a single genuine generic parameter list, i.e., from a simple generic nomenclatural term within GENSAL; its counterpart is derived also from a genuine generic parameter list, or it is derived from one simple specific partial structure, or it is derived from partial structures connected by AND logic, but it must not be derived from alternative partial structures, i.e., it must be non-a-variant.

Coincidence of node match with structure match for non-a-variant nodes is therefore possible if one node parameter list is a genuine generic parameter list, i.e., one that is derived from a simple generic nomenclatural term within GENSAL, and its counterpart is either a genuine generic parameter or a derivative parameter list, but must also be non-a-variant.

The derivative parameter list may be one of the following:

A specific-derived parameter list, derived from the connection table of one simple specific expression.

Specific-derived in a wider sense, i.e., from the interconnection of two or more simple specific expressions in a composite (but nonsegmented) specific expression, e.g., from "*t*-butyl—CH=C(*n*-propyl)—" given as one of several alternatives in "R_1—CH=C(R_2)—". Such a parameter list is derived by combining separate connection tables into one and then deriving the parameter list from this, or by deriving a specific-derived parameter list for each connection table then by the addition of these parameter list (by the bubble-up process).

Derived by the addition of genuine parameter lists, e.g., from a segmented expression like "alkyl-alkenylene-".

Derived by the addition of one (or more) genuine and one (or more) specific-derived parameter list, e.g., from a segmented expression like "*t*-butyl-alkenylene-".

The derivation-type of a node (and, consequently, of the node parameter list), which is decisive for the recognition of a coincidence of a node match with a structure match, can be derived and denoted in the way discussed in Section 5.7 of the preceding paper in this issue, and it can be assigned to a node as a node descriptor. For pairs of matching nodes (i.e., for pairs which are not revealed as mismatches by comparison of EPVs of complete node parameter lists), therefore, the coincidence of a node match with a structure match can be evaluated in the original reduced graph. If such a coincidence is not recognizable for a pair of matching nodes, then these nodes are resolved, i.e., the a-variant nodes, representing partial structures which are alternative to one another by p-, s-, or f-variation, are resolved into non-a-variant reduced graph nodes, and any structure matches which can be identified at this level are determined; the remaining node pairs are then subjected to further steps of the refined search. The exact order of implementing these steps, and whether all steps are necessary, e.g., the comparison of RPVs as a step preceding the comparison of EPVs, has yet to be determined.

In the transition from nodes of the reduced graph to representations in the ECTR, it is necessary to take account of the mutual attachment of partial structures. (Inversely connected partial structures, e.g., in "phenyl-NH-CO-O-cyclohexyl" and "phenyl-O-CO-NH-cyclohexyl", cannot be distinguished in a node search.)

6.2. THE MUTUAL CORRESPONDENCE OF QUERY AND FILE PARTIAL STRUCTURES

When *both* nodes in a matching node pair resolve to a combination of parameter lists and connection tables (e.g., "–alkenylene C⟨2–5⟩–COO–$CH_2CH(CH_3)_2$" matching with "–alkylene C⟨1–3⟩—CH=CH—COO—CH_2–alkyl C⟨3–6⟩"), then the refined search is no longer a simple case of performing either an atom-by-atom search or a comparison of parameter lists, because the proper correspondence between the partial structures within the nodes must be found. This problem may be alleviated, to some extent, by creating reduced graphs with an increased differentiation of node types, e.g., by focusing on the heteroatoms within an acyclic partial structure, since these often provide a natural division within partial structures.

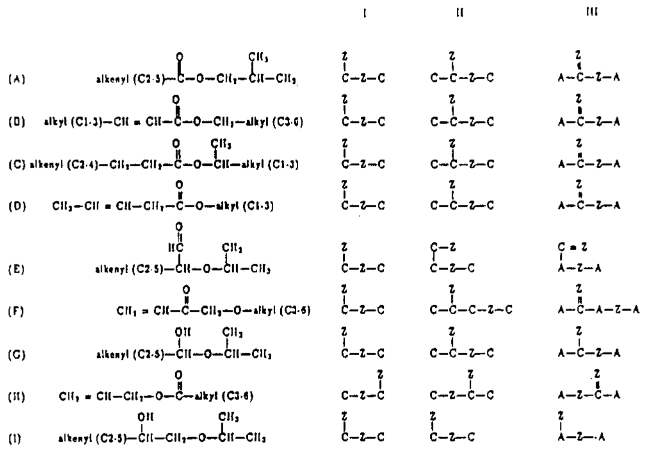

Figure 6. Structure expressions and reduced graphs at different levels of reduction.

Several different levels of reduction are illustrated, in increasing levels of differentiation, in Figure 6. Firstly, all of these full structures reduce to a single nonring node, if the reduction is on the basis of only ring and nonring components. Column I reduces on the basis of nodes that are aggregates of carbon atoms (C-nodes) and aggregates of heteroatoms (Z-nodes). Column II is based on a concept of heterofunctionality used in the GREMAS system as "degree of heteroorientation".[5] In a slightly modified application of this principle, in a reduced graph of the type shown in column II, each single carbon atom having a heterovalency of two or more is treated as a separate carbon area, i.e., as a separate C-node, with the exception only of carbon atoms which are substituted exclusively by two or exclusively by three halogen atoms. In column III, the carbon atoms with two, three, or four heterovalencies (C-nodes) are distinguished from aliphatic carbon chains with one or more carbon atoms (A-nodes), and additionally the bonds between the nodes are given.

The reduction according to heterofunctionality as given in column II assists in determining the proper correspondence between partial structures; the processes of the refined search are now explained using the examples in Figure 6 where structure A is the query structure. The modified node search now screens out structures E, F, and I by a comparison of node types; and D, G, and H are screened out by the Enumerative Parameter Values (EPVs) used as node descriptors. (The parameter value C⟨4⟩, counting the number of carbon atoms in the C-node derived from the isobutyl radical in structure A, for example, does not match with the parameter values C⟨1–3⟩, C⟨3⟩, and C⟨3⟩ for the corresponding C-nodes of structures D, G, and H, respectively.) The refined search is then necessary only for structures B and C. The final step of this refined search is now attained quite easily because the paired nodes in the graph of the query and file structures describe precisely the proper mutual correspondence of the heterofunctions and they prescribe precisely the required segregations and/or additions within carbon chains.

The required additions for the left-hand carbon chains of B and C have already been performed in creating the reduced graph of column II, and the coincidence of structure match with node match has already been anticipated for each file structure and query A. The connection tables of the central carboxy partial structure match at the atom-by-atom level. In matching the right-hand chains, the –CH_2– of B and the –$CH(CH_3)$– of C are searched as substructures in the partial structure –CH_2–$CH(CH_3)_2$ of A, taking into account the connections to the –O– of the carboxy group in all cases. This substructure search fails for C. For B the final comparison is of the genuine parameter list of alkyl C3–6 with the specific-derived parameter list of the remaining segregated partial structure $-CH(CH_3)_2$ in A and yields a structure match.

ACKNOWLEDGMENT

We gratefully acknowledge financial support by International Documentation in Chemistry—IDC GmbH, Chemical Abstracts Service, Derwent Publications, Ltd., and Questel S.A. We also thank the reviewers for useful comments, which added to the clarity of presentation.

REFERENCES AND NOTES

(1) Welford, S. M.; Barnard, J. M.; Lynch, M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 5 Algorithmic Generation of Fragment Descriptors for Generic Structure Screening. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 57–66.

(2) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 10. The Generation and Logical Bubble-Up of Ring Screens for Structurally Explicit Generics. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 215–224.

(3) Gillet, V. J.; Downs, G. M.; Ling, A. I.; Lynch, M. F.; Venkataram, P.; Wood, J. V.; Dethlefsen, W. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 8. Reduced Chemical Graphs and Their Applications in Generic Chemical Structure Retrieval. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 126–137.

(4) Dethlefsen, W.; Lynch, M. F.; Gillet, V. J.; Downs, G. M.; Holliday, J. D.; Barnard, J. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 11. Theoretical Aspects of the Use of Structure Languages in a Retrieval System. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 000–000.

(5) Meyer, E. Topological Searches for Classes of Compounds in Large Files—even of Markush Formulas—at Reasonable Machine Cost. In *Computer Representation and Manipulation of Chemical Information*; Wipke, W. T., Heller, S. R., Feldman, R. J., Hyde, E., Eds.; John Wiley & Sons: New York, 1974; pp 105–122.

Computer Storage and Retrieval of Generic Chemical Structures in Patents. 13. Reduced Graph Generation

VALERIE J. GILLET, GEOFFREY M. DOWNS, JOHN D. HOLLIDAY, and MICHAEL F. LYNCH*

Department of Information Studies, University of Sheffield, Sheffield S10 2TN, England

WINFRIED DETHLEFSEN

BASF, Ludwigshafen/Rhine, Germany

Received February 7, 1991

Criteria for creating reduced graph representations of full generic structures for screening in full structure and substructure searching are compared; ring/non-ring reduction is identified as the principal criterion. The current form of the Extended Connection Table Representation (ECTR), the internal representation of generic structures in the GENSAL system, is shown to be an AND/OR tree, in contrast with an earlier implementation, a logical graph. The role of the ECTR in facilitating the generation of reduced graphs from a wide range of generic structures, despite the complexity of their logical textures, is detailed. The structural descriptors associated with the nodes of the resultant graphs are detailed, together with their derivation. These descriptors are common, regardless of whether they are derived from specific or generic partial structures (PSs), thus ensuring the correctness of retrieval.

1. INTRODUCTION

Research efforts have been directed toward providing a topologically based system for the storage and retrieval of generic chemical structures for over a decade. The major contributors, to date, have been Sheffield University, Chemical Abstracts Service, International Documentation in Chemistry GmbH (IDC), and Derwent Publications Ltd., together with Questel SA and INPI (the French National Patent Office). The complexity of the problem is evidenced by the fact that the first publicly available system appeared on the market only in 1989, and the need for continuing research is evidenced by the comparison of this system with the fragmentation systems it is intended to replace.[1]

Many aspects of the procedures developed at Sheffield for the storage and retrieval of generic structures have already been presented in Downs et al.[2] and earlier papers. These include the definition of GENSAL, the formally defined language used to represent structures, and aspects of retrieval such as ring perception and screening, together with earlier studies on fragment screening. Fragment screening has now been extended to cover the full spectrum of structural variation found within generics and to provide a more accurate representation making full use of the logical relationships found within generics and will be the subject of a future paper. The concept of the reduced graph as a form of representation of generic structures has also been outlined in its earlier exploratory approach, first with regard to specific structures and then to generics of limited variability.[3] Reduced graphs have also been developed by Chemical Abstracts Service, although the criteria of reduction are different from those applied in the Sheffield approach.[4]

The techniques of graph reduction have now been substantially extended, and while one or two exceptions remain, the full variety of structures can be handled. The importance of reduced graphs both as screens and as an essential preparatory step toward the refined search, providing a gross correspondence between query and file structure prior to more detailed matching, is discussed in the previous paper in this issue.[5] This paper describes their generation from the internal representation of generic structures.

2. THE ECTR

Generic structures are of the internally as an Extended Connection Table Representation or ECTR.[6] The syntactic form of the ECTR has evolved since its earlier description, driven by the experience gained in developing software to translate this representation to other forms such as ring screens and reduced graphs.

0095-2338/91/1631-0260$02.50/0