Computer Storage and Retrieval of Generic Chemical Structures in Patents. 10. Assignment and Logical Bubble-Up of Ring Screens for Structurally Explicit Generics

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

Department of Information Studies, University of Sheffield, Sheffield S10 2TN, U.K.

Received October 18, 1988

The rings perceived within and across the partial structures of structurally explicit generics are analyzed to produce ring-screen information that complements the existing fragment screens. This paper presents and discusses the format of these ring screens. The resultant bit vectors for each partial structure are accumulated to retain their logical relationships by means of a "bubble-up" of the vectors. The principles of the bubble-up are outlined and explained by means of several examples. The bubble-up governs the superimposition of the vectors into two global vectors that reflect the variant and invariant parts of the entire structure and enhances the differentiation. This is achieved without generating all the specific possibilities covered by the generic.

INTRODUCTION

The previous paper in this series[1] outlines the ring perception algorithm developed for use in the RINGGEN program, with related papers giving a review of previous algorithms[2] and presenting the theoretical basis for the extended set of smallest rings (ESSR).[3] To avoid extensive repetition, the reader is referred to them for the definition and description of many of the terms and concepts used here.

This paper shows how the perceived ESSR rings are represented as screens and the manner in which the logical relationships between the screens from various parts of the structure are retained. These subjects are treated more fully in reference 4.

The various alternatives and combinations of structurally explicit generics are held in an extended connection table representation (ECTR).[5] This provides a logical treelike framework in which the internal nodes represent the logical relationship (AND or OR) between the branches, and the leaf nodes contain the partial structure connection tables. The root partial structure is a special node from which the rest of the tree emanates. Connections to other partial structures are designated parental if they go up the tree toward the root or child if they go down the tree away from the root. As a consequence of the early stages of the work presented here, the form of the ECTR outlined in reference 5 has been changed so that the tree structure conforms to that of a strict AND–OR tree.[7] Among other advantages, this has made ring perception across partial structures easier.

RING SCREENING

The usual strategy for searching files of specific structures is to eliminate those structures that cannot possibly match by means of a rapid **screening** search and then to apply a more painstaking **atom-by-atom** match (originally proposed by Ray and Kirsch[8]). Screening, in the context of serial file searching, involves the mapping of selected structural features onto a predefined bit string (bit vector), normally by means of a screen dictionary of these features. The structure is analyzed, and each structural feature, or fragment, is looked up in the screen dictionary. If an entry is found, then it will have an associated number that refers to the bit position of the bit string that needs to be set to 1 (or TRUE) to indicate the presence of that fragment in the structure. Screening has been adopted at Sheffield as a first stage before more sophisticated and novel matching strategies.

The fragment generator FRAGGEN[6] records the presence or absence of certain discriminatory atom and bond sequences and augmented atoms in vector form by means of a bit string. The positions in the bit string correspond to particular screen numbers given in a subset of the CAS ONLINE screen dictionary.[9] The resultant bit string can thus be regarded as a vector for the purposes of matching and retrieval. The inclusive OR operation determines whether each bit in the query vector is present in a file structure vector. If this is not so, then the query vector cannot be identical with, or a subset of, the file structure vector, and no match is possible.

To obtain the maximum effectiveness from such a screen, it is necessary to select those characteristics that will contribute most to the discriminatory performance of the screen. Characteristics occurring in a majority of structures will add little to the descrimination of the screen, while characteristics found in very few structures, although very discriminating, will be of no relevance to the majority of structures. The problem of defining a suitable set of descriptors for screening is therefore one of including enough to sufficiently encompass all structures while including only those that are necessary to provide optimum discrimination. A previous series of Sheffield papers[10-14] considers such optimization with respect to specific structures and has contributed to the basis for the CAS ONLINE screening[15] among others.

The purpose of the ring analysis by RINGGEN is to provide information specifically about the cyclic parts of a structure in such a way as to complement that already present in the fragment analysis. There is thus no requirement to give sequence or localized bonding information.

The ring-count and type-of-ring screens used in the CAS ONLINE dictionary are not appropriate for use with generics, not least because of such factors as variable attachment positions within rings and variable numbers of substituents. It is necessary, therefore, to develop a new approach that accommodates the vastly more complex environment of generic structures. Preferably, such an approach should be compatible with the existing vector representation and matching operations used for the fragment descriptors.

There are currently 1696 bits available in the vectors, with the first 1123 being used to represent the fragment screens. A convenient number of bits is thus available for further information such as ring screens. It should be noted, however, that the vector length is arbitrary and can be altered to accommodate as many descriptors as are required.

The ring perception algorithm of RINGGEN[1] will determine the nullity and then find the extended set of smallest rings (ESSR)[3] for specific and structurally explicit generics. The rings found within and across each partial structure connection table (the **intra-** and **inter-PS** rings) are stored as atom and

0095-2338/89/1629-0215$01.50/0

bond sequences, together with their size and the ring connectivity of each atom. This gives more information than is required for the ring screening outlined here, so that any future requirements and extensions to the screening will not be inhibited by a lack of information from the original ring analysis.

To complement the fragment screens, the most useful information includes that of ring size, aromaticity, the number and type of heteroatoms, the type of ring system, and the number of rings present. The resultant format of the ring screens produced by RINGGEN is outlined in the next section.

Analysis of the ring set to determine which characteristics are present can be done directly without recourse to a dictionary. This is the major difference between fragment-screen and ring-screen assignment. Each ring in the set can be analyzed separately and the appropriate ring-screen bits assigned to the vectors.

In a generic structure, such as that declared in a chemical patent, there will usually be some part that is common to all the alternative structures covered, which will be referred to as the **invariant** part of the structure. Those parts not common to all alternatives are the **variant** parts. Descriptors used to categorize such a structure can therefore be split into those occurring wholly within the invariant part, i.e., those that MUST be present in the structure, and those that occur at least partly within the variant parts of the structure, i.e., those that MAY be present. As discussed by Welford et al.,[6] it is necessary to represent a generic structure by two vectors to reflect this differentiation for screening purposes and subsequent matching operations (in contrast to the single vector required for specific structures). The first vector represents the MUST descriptors, while the second represents all POSSibly present descriptors (i.e., the union of MUST and MAY). The differentiation between variant and invariant parts, and hence the loading of the MUST and POSS vectors, are obviously dependent upon the nature of the descriptors used. The fragment screen may well contain fragments common to parts of the ECTR that do not have common ring-screen descriptors.

For the purposes of increased precision in matching, it is necessary to differentiate between variant and invariant parts as accurately as possible and to reflect this partitioning in the MUST and POSS vectors. The division in FRAGGEN is simply between those fragments present in the root partial structure (MUST) and those occurring in any of the other partial structures (MAY), a differentiation that is inadequate as a general solution. In contrast, the ring screening for each partial structure is accumulated within a framework in which the logic between what must and what may be present in the entire structure is retained. This accumulation of ring-screen information into the top-level MUST and POSS screens is referred to as the **bubble-up** principle. The development and implementation of this principle for ring screening are outlined later.

Welford et al. also discuss the matching operations required to determine whether a file structure is relevant to a particular query. For substructure queries (specific or generic) the criterion for a successful match is whether all the descriptors that must be present in the query are also present in the file structure. Therefore, the query MUST vector should be a subset of the file structure POSS vector, i.e.

$$Q[MUST] \subseteq FS[POSS]$$

where Q is the query and FS is the file structure. This is referred to as the **forward match**.

Full-structure searching additionally requires identity of a specific query with a specific file structure, or with one of the specifics covered by a generic file structure, or it requires identity of at least one specific covered by a generic query with a specific file structure, or with at least one of the specifics covered by a generic file structure. Therefore, the file structure MUST vector should be a subset of the query POSS vector, i.e.

$$FS[MUST] \subseteq Q[POSS]$$

This is referred to as the **reverse match**.

Table I. Ring-Screen Format[a]

bit#	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4
1505	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																							startpoint (1527) ⟹	3
																							Rs	Ac
											ring-composition fields													
1529	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			1	2	>2	1	2	>2	1	2	>2	1	2	>2	1	2	>2	0	1	2	>2	4		
	A	NA		N			O			S			P			Oh			Fp			Rs	Ac	A
1553	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1	2	>2	1	2	>2	1	2	>2	1	2	>2	1	2	>2	0	1	2	>2	5			
	NA		N			O			S			P			Oh			Fp			Rs	Ac	A	NA
1577	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	2	>2	1	2	>2	1	2	>2	1	2	>2	1	2	>2	0	1	2	>2	6				1
		N			O			S			P			Oh			Fp			Rs		A	NA	
1601	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	N			O			S			P			Oh			Fp			Rs	Ac	A	NA		N
	2	>2	1	2	>2	1	2	>2	1	2	>2	1	2	>2	0	1	2	>2	7				1	2
1625	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	>2	1	2	>2	1	2	>2	1	2	>2	1	2	>2	0	1	2	>2	>7				1	2	>2
			O			S			P			Oh			Fp			Rs	Ac	A	NA		N	
1649	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	2	>2	1	2	>2	1	2	>2	1	2	>2	0	1	2	>2	1	2	3	4	5	6	7	8
		O			S			P			Oh			Fp			number-of-rings field							
1673	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	>31
											number-of-rings field													

[a] Rs = ring size, Ac = all-carbon ring, A = aromatic, NA = not aromatic, N, O, S, P = specific heteroatom fields, Oh = other heteroatom, Fp = fusion point.

RING-SCREEN FORMAT

The ring-screen format that has been devised comprises six individual 23-bit composition fields (corresponding to ring sizes 3–7 and more than 7) followed by a 32-bit number-of-rings field (to record explicitly the nullity for that structure). The ring screen is thus allocated the last 170 bits of the vector.

As mentioned earlier, the information contained within the ring screen complements that in the fragment screen and represents different graph-invariant characteristics relevant to the rings present in a structure. Obviously, a prerequisite is an invariant ring set from which to determine these characteristics, hence, the stress on this aspect in the development of the ESSR.

For each of the ring-size categories the composition is analyzed further to determine whether or not a ring is all-carbon or aromatic, whether it contains certain heteroatoms, and if so how many, and how many fusion points there are with other rings. These are all graph-invariant characteristics of the ring set and complement the more localized atom and bond environment information given in the fragment screen.

The format for each ring-composition field is

Rs	Ac	A,NA	1N,2N,>2N	1O,2O,>2O	1S,2S,>2S	1P,2P,>2P	1Oh,2Oh,>2Oh	0Fp,1Fp,2Fp,>2Fp

where Rs is ring size, Ac is all-carbon ring, A is aromatic, NA is not aromatic, N, O, S, and P are specific heteroatom fields, Oh is other heteroatom, and Fp is fusion point. Table I gives a complete ring-screen format.

Ring Size. The use of individual ring-composition fields for various ring sizes implies that if at least one bit within a field is set, then a ring is present. However, it is necessary to have

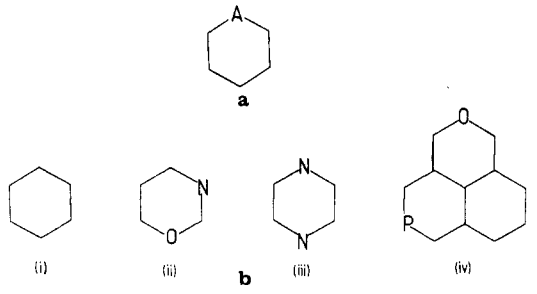

Figure 1.

an *explicit* ring-size bit to cater for the generic atom-type **Any** that may be used in queries. Since **Any** implies a single atom, the ring in which it is situated will not vary in size. Any attempt to make the atom types explicit in the query MUST vector would require every file structure MUST vector to have the Ac, N, O, S, P, and Oh bits set, thus masking any atom-type information from other rings of that size. It would, in effect, be an attempt to include variant information in the invariant screen.

For instance, in Figure 1a, the MUST vector will indicate a six-edged ring with no fusion points. As a query in a substructure search, this will retrieve any structure containing a six-edged ring. For instance, all of the file structures in Figure 1b would be retrieved. The forward match of substructure search will find the six-edge ring bit set in each of these cases, and also the 0 Fp bit, since it is the query MUST that is compared with the file structure POSS (the POSS fusion-point assignment is explained later). Thus, substructure search using such a query will retrieve all file structures containing six-edged rings irrespective of their atom-type composition and their relationship with other rings in a complex.

Full-structure search is a little more discriminating. The reverse match will compare the file structure MUST vector with the query POSS. The POSS vector for this query will indicate a six-edged ring with no fusion points, but will also have the All-C, 1 N, 1 O, 1 S, 1 P, and 1 Oh bits set. In Figure 1b, only the MUST vectors for file structures i and ii will be a subset of this POSS vector. File structure iii will be eliminated since it has 2 N, and file structure iv will be eliminated because it has >2 Fp.

It can be seen from this that structure ii is a false drop—the query stipulated the presence of only one Any atom in the ring. However, this example illustrates the complementary basis on which the ring screens are designed. The file-structure MUST fragment screens will include atom sequences with two heteroatoms. These will fail to match any of the query fragment screens, and hence structure ii will be eliminated.

There is also the question of representing large rings that exceed the array size for the ring perception algorithm. At present the partial structure connection table size limit of 32 vertices is also used as the limit for the size of ring that can be stored in the ring sequence array (and is also used as the limit for the number of bits in the number-of-rings field of the screens). This causes no problems for intra-PS rings since their maximum size is 32 vertices, which can be stored complete and will result in the >7 vertices ring-composition field being set in the screens. However, for inter-PS rings it is quite possible for linkages of two or more inter-PS part-ring paths to generate rings of more than 32 vertices. The lengths of the paths are checked before linkage is attempted; if the total length exceeds 32, then a flag is set to indicate "ring present with more than 32 vertices", and no attempt is made to link the paths.

For diagnostic purposes, if such a flag is encountered in the ring list, then no attempt is made to print the contents of the sequence array but the message "ring of more than 32 vertices found" is printed instead.

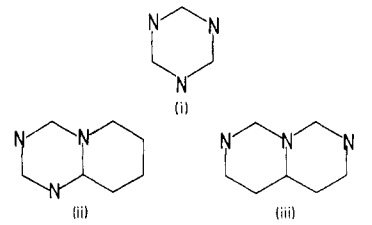

Figure 2.

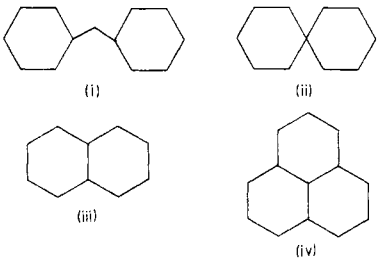

Figure 3.

For screening purposes, the flag will lead to the ">7 vertices" bit being set, but no attempt is made to further analyze the ring composition since the ring is incomplete. Rings with between 8 and 32 vertices will lead to the same bit being set and also to analysis and setting of the ring composition.

All Carbon. The Ac bit represents the presence of "at least one all-carbon ring". Difficulties arise in trying to copy the 1,2 and >2 categories of the heteroatom fields since these refer to atoms present *within* a single ring, while >1 all-carbon rings require comparison *between* different rings. Such a comparison would complicate the present procedures used to accumulate and set all information in the composition fields for structures crossing more than one partial structure.

Aromaticity. The A and NA bits indicate the presence, or otherwise, of aromatic rings. Several of the perception algorithms in the review paper[2] have supplementary code to detect aromaticity. Although most aromatic rings are six-edged, these two bits are also present for the other ring sizes to allow the same assignment procedures to be used for each ring-composition field.

Heteroatoms. Differentiation between N, O, S, and P will cater for nearly all occurrences of heteroatoms within rings. Rare situations involving other heteroatoms are covered by the Oh field. Similarly, the 1,2 and >2 counts in the heteroatom fields are sufficient to cover the majority of ring systems. More importantly, they differentiate between certain structures that pass through both the fragment screens and the relaxation search procedure,[16] as shown by the examples in Figure 2. If structure i is the query, then both file structures ii and iii will be retrieved by the fragment screening, since the query fragments are a subset of both. Application of the relaxation algorithm would also retrieve both structures since the neighborhood environment of each heteroatom is common to all. However, the N heteroatom field of the ring screens will match only structure ii to structure i. There is no six-edged >2 N ring in structure iii.

Fusion Points. The occurrence of ring atoms with a ring connectivity of more than 2, i.e., fusion points, is given by the relevant augmented atoms in the fragment screen. However, no indication is given as to their numbers within a ring or ring complex. The fragments assigned could well be identical when structures with ring systems such as in structures i, iii, and iv of Figure 3 are considered. The fusion-point field will separate all these structures by setting (i) to 0 Fp, i.e., solitary; (ii) to 1 Fp, i.e., spiro-fused; (iii) to 2 Fp, i.e., ortho- or endo-fused; and (iv) to >2 Fp, i.e., 1 Fp and 2 Fp combinations, and peri-fusions.

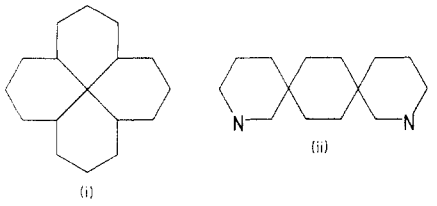

Figure 4.

The Fp screen is intermediate between the CAS "type of ring" screens (double/triple ring-connectivity sequences) and the more generalized reduced-graph cyclic/acyclic categories. The analysis looks at each ring individually and sets the fusion-point field accordingly. Hence, for Figure 4, structure i will be regarded as four six-edged all-carbon rings, each with three fusion points (i.e., peri-fused). Unfortunately, (ii) will be classified as two six-edged 1 N rings with one fusion point and one six-edged all-carbon ring with two fusion-points, i.e., not a spiro-fused ring. Correct detection of spiro-fusions can be accomplished by a procedure that looks for atoms with a ring connectivity of 4 which are common to only two rings found in phases 1 and 2 of the ring perception algorithm (see reference 1). Such a procedure correctly eliminates Figure 4i as a spiro-fused system since the central atom is common to all four phase 1 rings. For Figure 4ii, the procedure correctly recognizes two spiro-fusions and no others; hence, all three rings are designated spiro-fused.

Number of Rings. Following the composition fields in the number-of-rings field. The first 31 bits explicitly represent the presence of 1–31 rings, while the last bit denotes 32 or more rings present. Note that if there are no rings present, then none of the ring-screen bits will be set. The number of rings is taken as the nullity for the structure, rather than the ESSR cardinality, due to the ease of calculation.

BUBBLE-UP

After an initial format for the ring-screen vectors had been designed, it was realized that no procedures were available to retain the logical relationship between partial structures in the vectors accumulated for the whole ECTR. As mentioned earlier, FRAGGEN simply regards the root partial structure as MUST and anything below it as POSS. In response to this lack of discrimination, the bubble-up principle was developed and incorporated into RINGGEN, as described in this section.

For ring screening, the differentiation between variant and invariant parts of the ring analysis is accomplished by selective incorporation of local vectors into global MUST and POSS vectors.

The intra-PS ring and inter-PS part-ring path perception is conducted on a depth-first tour of the ECTR. At any one partial structure, ring information from within that partial structure is set in local MUST and POSS vectors on the way down the ECTR. On the way back up, the inter-PS rings are constructed and the appropriate screens assigned to the vectors. Since an inter-PS ring is completed only at the topmost partial structure in which it occurs, the logical relationships between the various parts are retained by the use of flags to indicate whether the ring should be assigned to the POSS vector or to both POSS and MUST vectors.

The order of partial structure numbering and the partitioning into partial structures depend upon how the GENSAL expression is declared on input; this is largely at the discretion of the analyst or user. The bubble-up within RINGGEN is designed to provide a uniform analysis independent of this input. This is achieved by the procedures written to accumulate the local vectors into the global vectors in a progressive manner on the way back up from the depth-first trace of the ECTR. Once back at the root partial structure, the resultant

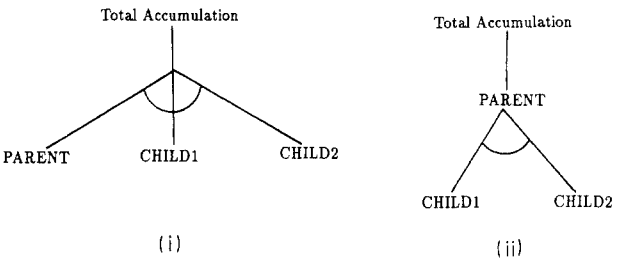

Figure 5.

two vectors represent the MUST and POSS information for the whole generic structure, no matter how that may have been split and represented as an ECTR.

Assignment to the *local* MUST or POSS vectors depends upon whether the rings (intra- and inter-PS) occur in combination with each other or are alternative to each other. The bubble-up is achieved by successively ANDing and ORing these local vectors with the *global* vectors as they are completed on the way back up. Whether ANDing or ORing is used at a particular level will depend upon the logical relationship between the partial structures. This is shown more clearly by consideration of the examples given in Figures 7i, 8i, 9i, and 10. Associated with each of these is a representation of the accumulative framework by which the individual partial structure ring-composition screens and number-of-rings ranges are bubbled-up to produce the top-level RINGGEN ring screens (Figures 7ii, 8ii, 9ii, and 11, respectively). It is important to note that *the logical* **operations** *performed to assign correctly the MUST and POSS vectors at a particular level are determined by, but different from, the logical* **relationship** *between vectors at different levels.*

The actual framework in which the partial structures are processed is similar to an AND–OR tree. Each accumulative framework has been drawn as an AND–OR tree, with all leaf nodes being partial structures and all internal nodes being either AND or OR relationships. Processing during the bubble-up is slightly different in that the unique relationship between parent and child is retained by performing assignment of inter-PS rings and related vector operations while processing the parent partial structure. The parent is thus implicitly in an AND relationship to its children rather than explicitly.

For example, in Figure 5, structure i is actually processed as structure ii. This processing relationship corresponds more faithfully to the relationship within the original generic structure and makes ring perception across partial structures easier to accomplish. It is, however, transparent to the user and can be represented diagramatically more simply as an AND–OR tree.

Each edge of the tree has been overlaid by a representation of the number-of-rings and ring-composition information accumulated up to that level. The number-of-rings range is enclosed in parentheses, the MUST ring-composition field by braces, and the POSS ring-composition field by square brackets. Each of the rings drawn in the ring-composition fields indicates the presence of *one or more* rings of that size, the numbers in the middle of each ring indicate the number of fusion points for that type of ring, and the large rings with two dotted edges indicate a ring, or rings, with more than seven edges.

Each junction point in the accumulative framework tree denotes the logical *relationship* between the branches; combinations (AND) are joined by an *arc*, while alternatives (OR) are not. Directly underneath each junction point are given the logical *operations* required (ADD, AND, or OR) to accumulate each set of fields for the next level up. Those operations performed on the number-of-rings ranges are enclosed in parentheses, those on the MUST ring-composition vectors are enclosed in braces, and those on the POSS ring-compo-

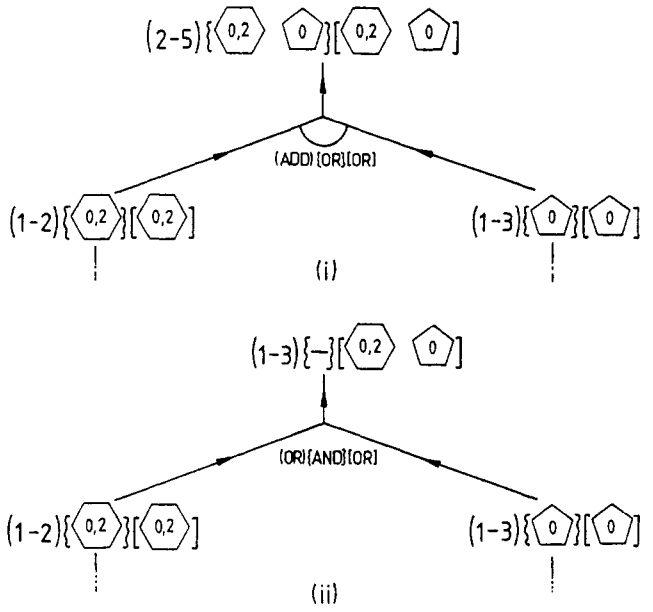

Figure 6.

sition vectors in square brackets.

For each partial structure, the *local* ring-composition vector is ORed into both the POSS and MUST accumulative bit strings, thus accomplishing the implicit AND **relationship** between this partial structure and anything coming from its children. At subsequent levels of bubble-up the POSS vectors are always ORed, but the MUST vectors are either ANDed or ORed depending on whether they were alternatives or combinations, respectively.

Accumulation of the number-of-rings ranges can be accommodated within the same framework, but the logical operations are slightly different. The ORing of ranges is equivalent to the logical ORing of two vectors. The logical AND, however, implies the intersection of two vectors; this is not what is required here. For the accumulation of distinct combinations, what is required is the addition of each element of one range to each element of the other range(s); hence, the operation is referred to as ADDing. Procedures to OR and ADD ranges have thus been developed for RINGGEN.

For example, in Figure 6, structure i represents the combination of two children, while structure ii represents the same two children given as alternatives. For Figure 6i the combination produces a structure with one or two six-edged rings AND one to three five-edged rings. ORing the ring-composition vectors would correctly produce MUST and POSS vectors with the six- and five-edged ring bits set. However, ORing the number-of-rings ranges, if they were represented as vectors, would result in a combined range of one to three rings present, which is incorrect. There *must* be at least one ring from the left branch and one ring from the right branch, so one ring overall is not possible. Similarly, the maximum number *poss*ible is two from the left branch and three from the right branch. Hence, the overall combined range is two

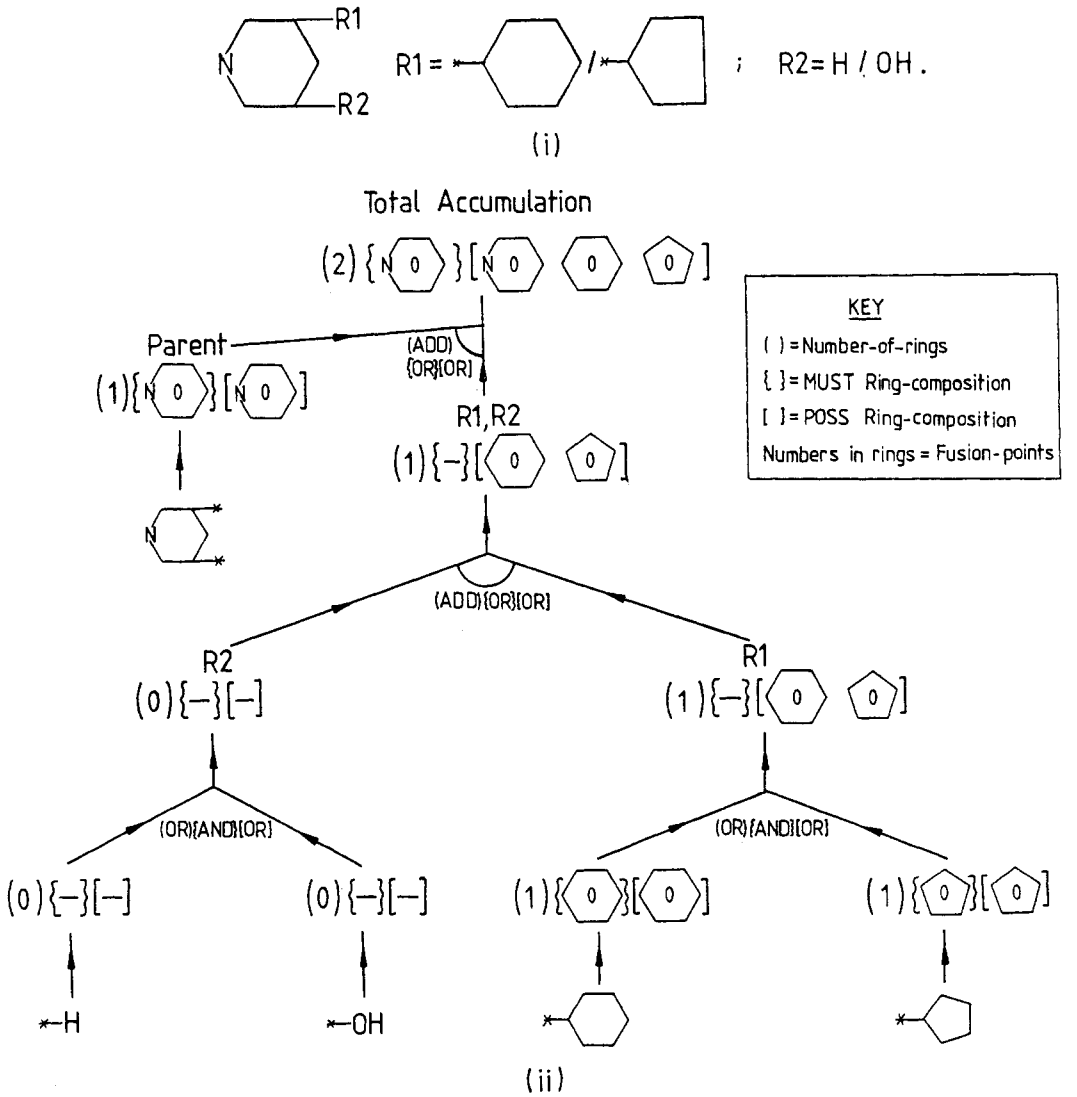

Figure 7.

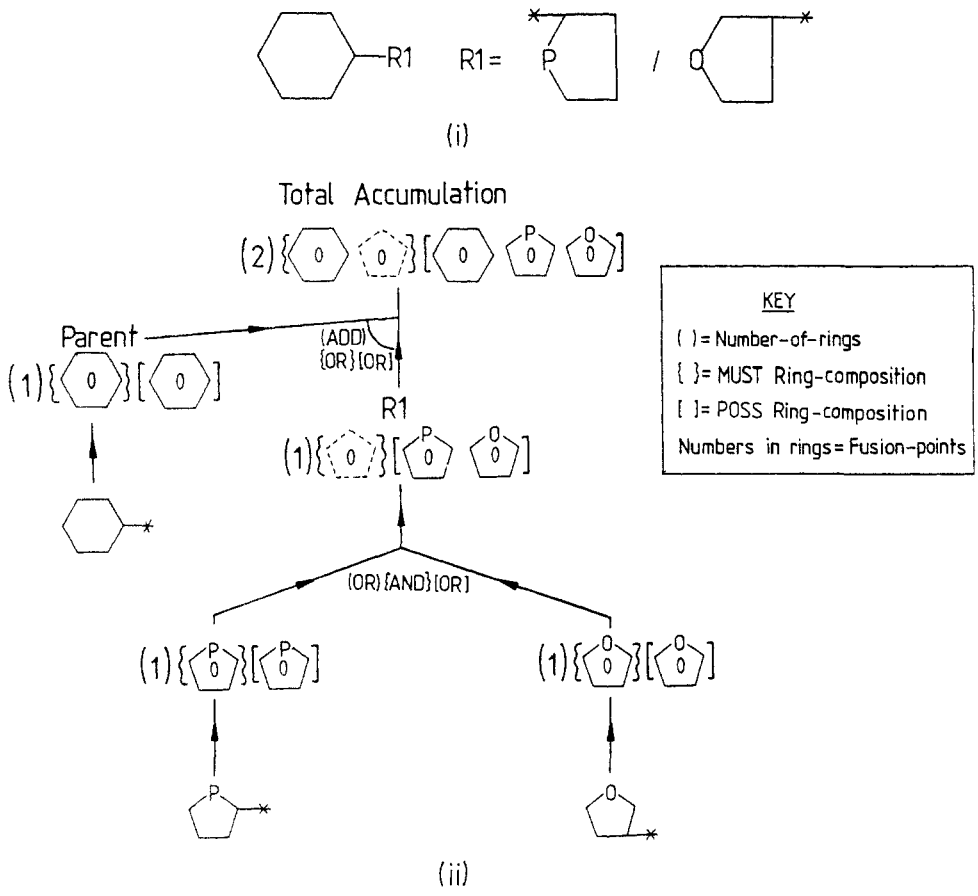

Figure 8.

to five rings, which is equivalent to *add*ing the lower bounds of each range to give the new lower bound and adding the upper bounds of each to give the new upper bound.

For Figure 6ii, the ring-composition MUST vectors have to be ANDed to let only common rings pass. There are none, so the MUST vector is empty. The ring-composition POSS vectors can be ORed as before. For the number-of-rings ranges the alternatives are to have either one to two rings from the left branch *or* one to three rings from the right branch. The overall range is therefore one to three rings, which is equivalent to ORing the two ranges.

To cater for the ADDing and ORing operations on ranges, the ranges are stored as a linked list of lower and upper bounds and not as vectors. The ranges can then be correctly accumulated within the bubble-up, with assignment to the vectors being left until the root has been reached, when they can be incorporated directly into the global vectors.

The reason for highlighting relationship versus operation should now be clear. Branches for combination are in AND relationship to each other, but the MUST vector operation required is OR (with ADD required for the ranges). Conversely, alternative branches are in OR relationship to each other, but the MUST vector operation required is AND (with ORing required for the ranges). As stated before, the POSS vector operation is always OR.

The structure in Figure 7i is an example in which the variables are independent. In this case, R1 has two separate alternatives, a five- or a six-edged all-carbon ring, which are separate from the two acyclic alternatives for R2. The accumulative framework in Figure 7ii shows how, for R1, the alternative relationship results in no common ring coming through the MUST ring-composition field. When combined with acyclic R2 alternatives, all fields remain the same.

Combination with the parent results in addition of the number-of-rings ranges to give a total of two, the appearance of the parental six-edged 1 N ring in the overall MUST ring-composition field, and the occurrence of all three rings in the overall POSS ring-composition field. None of these rings have any fusion points.

The structure in Figure 8i similarly contains an R1 with two cyclic alternatives. However, in this case the ring contained in each alternative is of the same size but different composition. The effect on the accumulation can be seen in Figure 8ii, in which only the presence of a five-edged ring can be indicated in the MUST ring-composition field. This is represented by the dotted outline of a five-edged ring. The presence of two distinct five-edged rings, one with a phosphorus and one with an oxygen atom, is indicated in the POSS ring-composition field.

Combination with the parent incorporates the parental six-edged all-carbon ring into both ring-composition fields. As with the example in Figure 7, whichever alternative is chosen there are always two rings present, and none of the rings have any fusion points.

The structure in Figure 9i contains dependent variables. R1 and R2 can either have independent acyclic existence or can combine to give a cyclic alternative containing both intra- and inter-PS rings. The accumulative framework for this is given in Figure 9ii.

The combination of R1 + R2 with the parent produces an infinite inter-PS ring of 10 edges, two alternative embedment inter-PS rings of 8 edges, and an inter-PS 6-edged ring. R1 + R2 also contains a 6-edged intra-PS ring.

The presence of the 10- and 8-edged inter-PS rings is indicated by the large ring with two dotted edges, depicting a ring with more than 7 edges. Both these rings have more than two fusion-points. The six-edged all-carbon inter-PS ring has more than two fusion points, while the six-edged all-carbon intra-PS ring has just two fusion points. These are represented in the ring-composition fields by a six-edged all-carbon ring

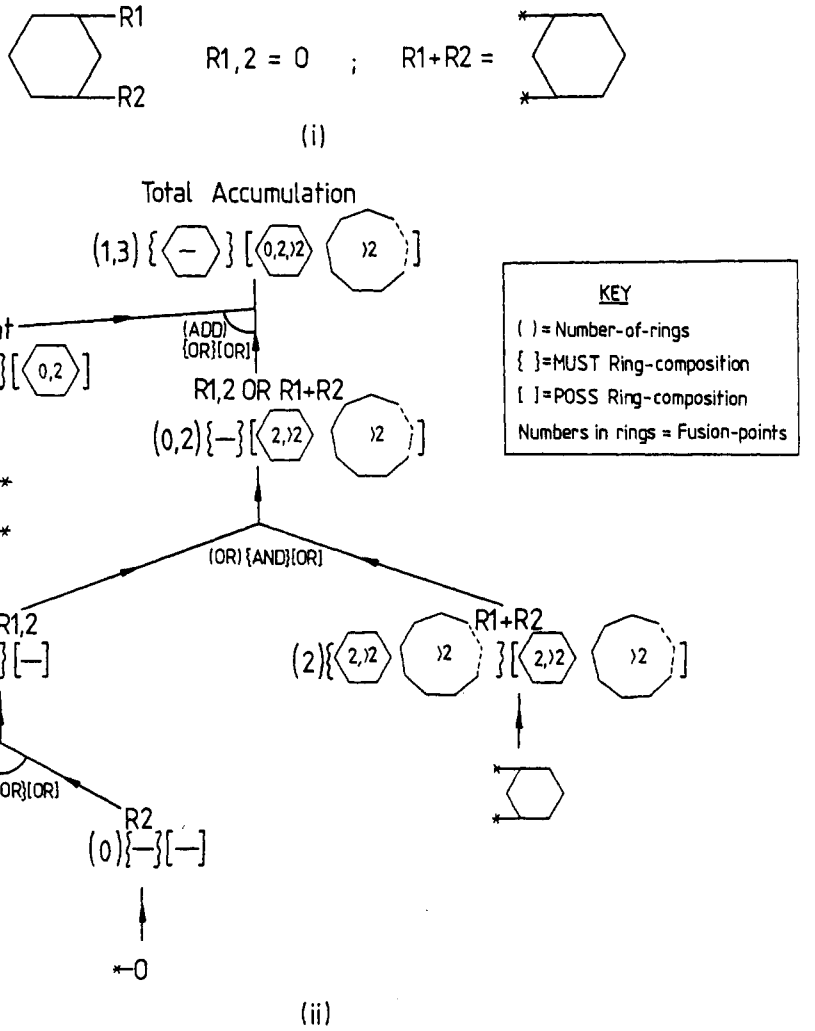

Figure 9.

with both two and more than two fusion points. Combination with the acyclic separate existence alternatives for R1 and R2 results in the loss of these rings from the MUST ring-composition field and a number-of-rings range indicating either nought or two rings.

Notice that for the parent the edges leading to R1 and R2 are variable; i.e., they can either by cyclic or acyclic. As a result the POSS ring-composition field six-edged all-carbon ring contains nought and two fusion points, but in the equivalent MUST ring this variable information cannot be represented and so the number of fusion points remains blank.

Combination of parent and children leads to an overall MUST ring-composition field containing the parental six-edged all-carbon ring with no fusion points and to an overall POSS ring-composition field containing a six-edged all-carbon ring with nought, two, and more than two fusion points and an all-carbon ring with more than seven edges with more than two fusion points. The total number-of-rings range indicates that there can be one or three rings.

Due to the difficulty in prediction of the number of rings in an ESSR, only the nullity number of rings in each partial structure, plus the nullity number of inter-PS rings crossing partial structures, is set in the number-of-rings range. For example, the combination of R1 + R2 with the parent in Figure 9i leads to a structure with six ESSR rings when the infinite region and alternative embedment regions are taken into consideration. However, in Figure 9ii the nullity is used to give two rings for R1 + R2 and one ring for the parent, giving a total nullity of 3, and it is these numbers that are carried through the bubble-up of the number-of-rings ranges to give the three in the total accumulation.

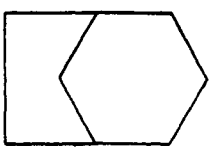

Figure 10.

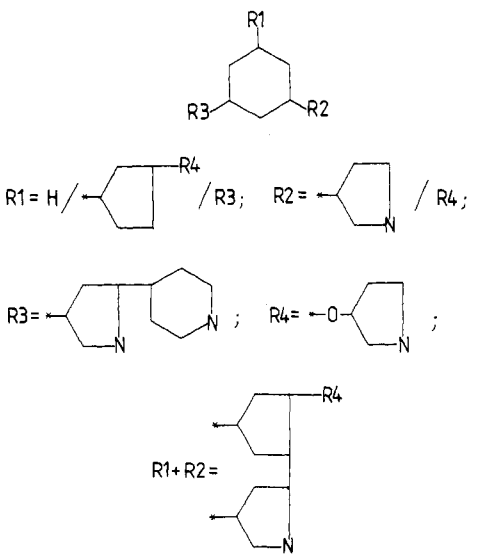

Figure 11.

Notice that use of the nullity can lead to the possibility of there being more MUST ring-composition rings than are given

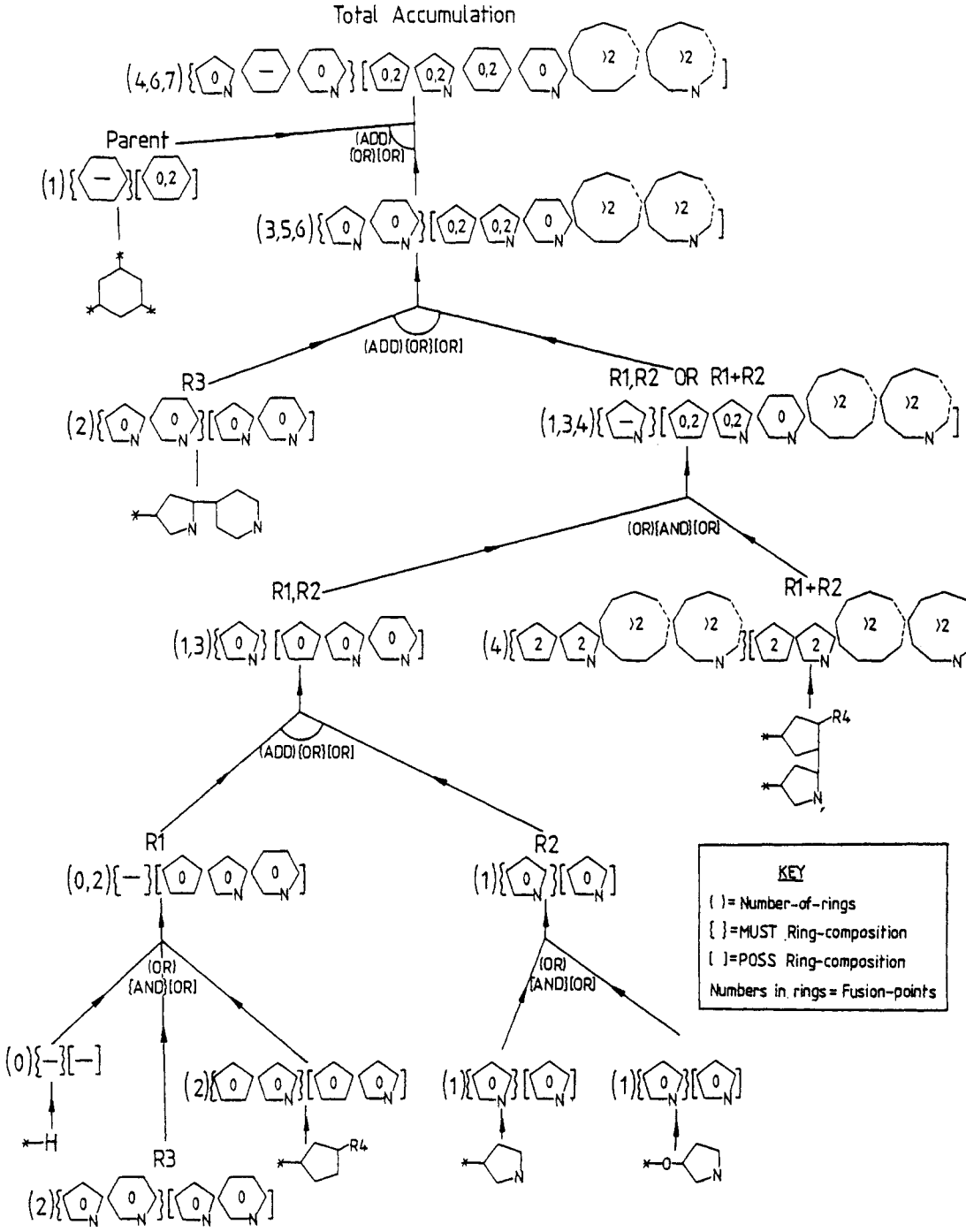

Figure 12.

in the number-of-rings field. For instance, the structure in Figure 10 has an ESSR comprising one five-, one six-, and one seven-edged all-carbon ring, which will lead to three separate and distinct entries in the MUST ring-composition field. However, the nullity is only 2, and so the number-of-rings range will indicate one less ring than MUST be present! Obviously the POSS ring-composition field can contain more rings than are indicated in the number-of-rings range due to the superimposition of several alternatives.

A generic environment containing a combination of dependent and independent variables is shown by the test structure in Figure 11. This contains quite a complex arrangement of parent/child and child/child relationships. R3 occurs directly attached to the parent, with no alternatives, and also as one of the alternatives for R1, while R4 occurs directly attached to one of the alternatives for R1 and to the combined R1 + R2 and also exists as one of the alternatives for R2.

The operation of the bubble-up is shown by the accumulative framework given in Figure 12. Starting from the bottom left of the tree, the acyclic alternative of hydrogen for R1 ensures that no rings are regarded as MUST for the R1 accumulation. The two cyclic alternatives for R1 each contain two ring types, but the five-edged ring with one nitrogen atom is common to both so that the accumulated POSS ring-composition field contains three ring types, all with no fusion points.

The alternatives for R2 both contain one five-edged ring with one nitrogen atom, and so this comes through both the MUST and POSS ring-composition fields. In combination with R1 this ring appears in both the ring-composition fields of the next level of accumulation. The number-of-rings ranges are added to give one or three rings, all of which have no fusion points.

The combined existence R1 + R2 contains two five-edged intra-PS rings, one all-carbon and the other with a nitrogen atom, both with two fusion points, and both of which occur

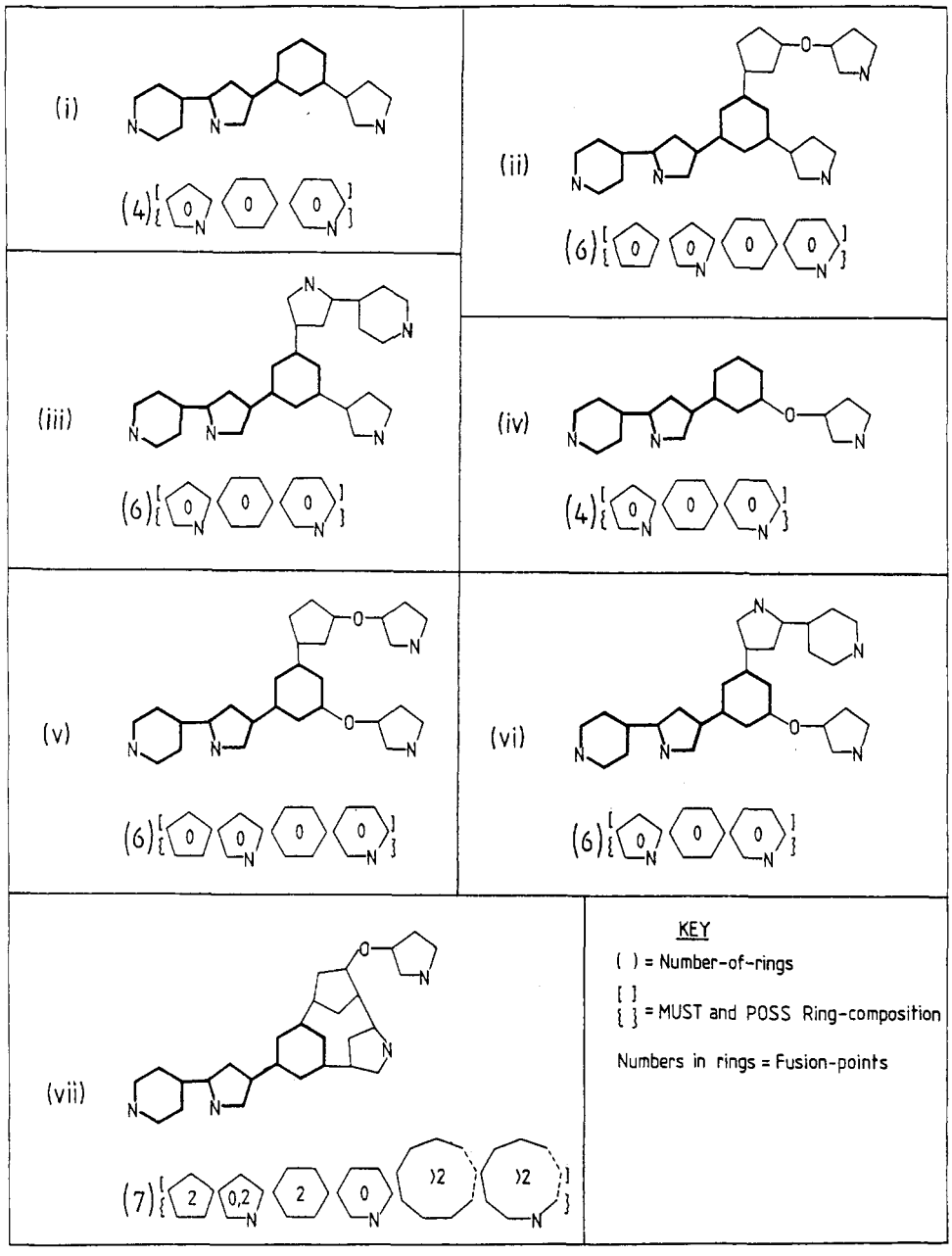

Figure 13.

in the ring-composition fields without being superimposed. There are also several inter-PS rings, each with more than seven edges and more than two fusion points. If the structure is linked with the parent in the projection used for Figure 11, then a 9-edged finite all-carbon ring and a 13-edged infinite ring with one nitrogen will be produced. This projection will also have six alternative embedment regions, giving two 10-edged, two 11-edged, and two 12-edged rings, one of each pair being all-carbon and the other with one nitrogen atom. In the ring-composition fields the two large rings, indicating rings with more than seven edges, one all-carbon, one with a nitrogen atom, and both with more than two fusion points, represent the superimposition of all these inter-PS rings.

When the alternative existences of R1, R2, and R1 + R2 are accumulated, only the presence of a five-edged ring with a nitrogen atom is common to all and hence passes through the MUST ring-composition field. Note that since the number of fusion-points is not the same, the fusion-point field of this ring becomes blank. The overlaying of five-edged rings is indicated in the POSS ring-composition field by the presence of nought and two fusion points. The number-of-rings range is extended to contain one, three, and four rings.

The final two junction points before the total accumulation are both combinations, indicating that R3 must always occur with the parent. The two rings from R3 therefore join the ring from the parent in the overall MUST ring-composition field. Note that the presence of a five-edged ring with a nitrogen in *both* R3 and (R1, R2 OR R1 + R2) is obscured in the overall representation because the ring-composition fields do not indicate the numbers of rings present of each size.

Figure 13 gives all the possible specifics that can be generated from Figure 11. Processing each of these specifics through phases 1–3 of the ESSR algorithm (see reference 1) will give the number-of-rings and ring-composition information indicated under each structure. Since these are specifics, the MUST and POSS bit strings are identical and so have been shown bracketed together. Outlined in bold is the part of each structure that is common to all seven specifics, i.e., the parent plus R3. This defines the invariant part covered by the generic; the other parts are the variant parts.

Note that in terms of the rings, there is also always the bottom-right five-edged ring with a nitrogen atom. This is the ring, mentioned in the previous note, that is obscured in the bubble-up of the generic. Note also that the number-

of-rings given for Figure 13vii is 7, which is the nullity, whereas 14 rings would be included in the ESSR due to the presence of the alternative embedment regions.

Each of these specific ring analyses can be seen to be included within the total accumulation of Figure 12, while the combination of all of them equals the total accumulation. In particular, it can be seen that the differentiation between variant and invariant parts has been successfully achieved. ANDing all the specific vectors (ORing the number of rings) gives the identical results to the MUST total accumulation of Figure 12, while the ORing of all the specific vectors (ADDing the number of rings) produces an identical screen to the POSS total accumulation.

The bubble-up will therefore successfully accumulate the ring screens derived from the ESSR processing within and between all the partial structures of the ECTR without the generation of all the specifics.

One final point is that in Figure 12 the overall number-of-rings range is 4, 6, or 7. Once the total accumulation has been achieved, the lower bound of this range is set in the MUST number-of-rings vector. For the POSS number-of-rings vector it is the upper limit that is of significance, and so every bit between 0 and 7 is set. This enables a test for inclusion to be used for both forward and reverse comparisons.

The original idea was to assign the actual ring range to the MUST vector and the 0 to maximum range to the POSS vector. However, this would introduce alternatives into the MUST vector, a contradiction that would have to be accommodated by using an inclusive AND matching operation on that part of the vector rather than the inclusive OR used for the fragment screens and ring-composition screens (i.e., the test would be for intersection rather than inclusion). Although perfectly possible, the necessity for such specific MUST range information is questionable and will have to await wider testing. However, the information is there should it ultimately be required.

CONCLUSIONS

The ring-screen format presented in this paper is designed specifically to complement the existing fragment screens. This does not preclude it from being of use and interest elsewhere. Further information can be added as required to optimize it for particular applications. One important aspect is that assignment of the ring-screen information to the bit vectors does not require a dictionary look-up.

The logical relationships between the local vectors of structurally explicit generics can be superimposed into global MUST and POSS bit vectors by means of the bubble-up. ANDing and ORing bit vectors representing the ring-composition information are readily accomplished. However, the number-of-rings information needs to be given as ranges that have to be separately ADDed and ORed. The operations performed on the ring-composition bit vectors and number-of-rings ranges are incorporated into the bubble-up and are governed by the logical relationship between branches of the AND–OR tree. Once back at the root, the number-of-rings ranges are assigned to the global bit vectors.

The bubble-up principle is appropriate for general information accumulation in an ECTR environment and is not restricted just to the ring screens outlined here.

ACKNOWLEDGMENT

This research was made possible through funding provided by IDC (Internationale Dokumentationsgesellschaft für Chemie mbH), whose staff also made many useful suggestions. We thank Drs. S. M. Welford and J. M. Barnard for valuable discussions.

BIBLIOGRAPHY

(1) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 9. An Algorithm To Find the Extended Set of Smallest Rings in Structurally Explicit Generic Structures. *J. Chem. Inf. Comput. Sci.* (third of four papers in this issue).

(2) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Review of Ring Perception Algorithms for Chemical Graphs. *J. Chem. Inf. Comput. Sci.* (first of four papers in this issue).

(3) Downs, G. M.; Gillet, V. J.; Holliday, J. D.; Lynch, M. F. Theoretical Aspects of Ring Perception and Development of the Extended Set of Smallest Rings Concept. *J. Chem. Inf. Comput. Sci.* (second of four papers in this issue).

(4) Downs, G. M. Computer storage and retrieval of generic structures in patents: ring perception and screening to extend the search capabilities. Ph.D. Thesis, University of Sheffield, March 1988; Chapter 5 (Screen generation and the bubble-up).

(5) 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.

(6) Welford, S. M.; Lynch, M. F.; Barnard, J. M. 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.

(7) Nilsson, N. J. *Principles of Artificial Intelligence*; Springer: Berlin, 1982; p 40.

(8) Ray, L. C.; Kirsch, R. A. Finding chemical records by digital computers. *Science (Washington, D.C.)* **1957**, *126*, 814.

(9) *CAS ONLINE screen dictionary for substructure search*, 2nd ed.; Chemical Abstracts Service: Columbus, 1981.

(10) Crowe, J. E.; Lynch, M. F.; Town, W. G. Analysis of structural characteristics of chemical compounds in a large computer-based file. Part 1. Non-cyclic fragments. *J. Chem. Soc. C* **1970**, 990–996.

(11) Adamson, G. W.; Lynch, M. F.; Town, W. G. Analysis of structural characteristics of chemical compounds in a large computer-based file. Part 2. Atom-centred fragments. *J. Chem. Soc. C* **1971**, 3702–3706.

(12) Adamson, G. W.; Lynch, M. F.; Town, W. G. Analysis of structural characteristics of chemical compounds in a large computer-based file. Part 3. Statistical association of fragment incidence. *J. Chem. Soc., Perkin Trans. 1* **1972**, 2428–2433.

(13) Adamson, G. W.; Cowell, J.; Lynch, M. F.; Town, W. G.; Yapp, A. M. Analysis of structural characteristics of chemical compounds in a large computer-based file. Part 4. Cyclic fragments. *J. Chem. Soc., Perkin Trans. 1* **1973**, 863–865.

(14) Adamson, G. W.; Creasey, S. E.; Eakins, J. P.; Lynch, M. F. Analysis of structural characteristics of chemical compounds in a large computer-based file. Part 5. More detailed cyclic fragments. *J. Chem. Soc., Perkin Trans. 1* **1973**, 2071–2076.

(15) Feldmann, A.; Hodes, L. An Efficient Design for Chemical Structure Searching. 1. The Screens. *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 147–152.

(16) von Scholley, A. A Relaxation Algorithm for Generic Chemical Structure Screening. *J. Chem. Inf. Comput. Sci* **1984**, *24*, 235–241.