Representation of Chemical Structures in Knowledge-Based Systems: The StAR System

C. A. G. Tonnelier,*,† J. Fox,‡ P. Judson,† P. Krause,‡ N. Pappas,§ and M. Patel†

LHASA UK Ltd., School of Chemistry, The University of Leeds, Leeds LS2 9JT, UK, Imperial Cancer Research Fund, Lincoln's Inn Fields, London, WC2A 3PX, UK, Logic Programming Associates Ltd., Trinity Road, London, SW18 3SX, UK

Received June 30, 1995⊗

As part of the StAR project, for the design of a computer system to support risk assessment, it has been necessary to develop a graphical language for the representation of generic structures. These structures are used by rule writers to describe toxicophores for risk assessment. Until now, the input of generic structures in knowledge-based systems has very often been by means of a SMILES-like linear notation. The new StAR graphical language allows the use of a wide range of Markush features including atom lists, bond lists, G-groups, and superatoms. In addition, the StAR graphical language allows rule writers to input rules easily to the knowledge base *via* a user-friendly graphical interface. This language is well-suited for the representation of chemical features required for expert general toxicity systems, chemical reaction systems, and substructure database systems.

INTRODUCTION

Expert systems, computerized systems which mimick the thinking and reasoning of human experts, are increasingly being used in biochemical and chemical applications,[1] e.g. to assess the toxicity of compounds. These systems use sets of rules to make predictions in novel situations rather than relying on complex calculations. Another area of future promise is their use to predict the metabolic transformations a compound is likely to undergo in living entities. All these knowledge-based systems used in chemistry have in common that they apply structure−activity relationships to chemical compounds. Knowledge-based systems all require the input, manipulation, and pattern matching of chemical graphs or substructures. In the case of Computer Assisted Organic Synthesis (CAOS) programs they link the presence of reaction patterns (or retro-reaction patterns called retrons) with specific reactivities;[2−5] in the case of toxicity prediction system they associate a chemical substructure with a particular toxic endpoint.[6,7]

RULE DERIVATION AND CHEMICAL PATTERN REPRESENTATIONS

Both human experts and computer systems are capable of deriving from specific information (or facts) the rules to be included in the knowledge base.[8] This mainly involves the derivation, from a set of congeneric structures, of the common substructure responsible for inducing the studied activity. The rule induction process is based on the reasonable supposition that compounds have similar biological activities or chemical reactivities, because they possess common structural features (Figure 1). Most computer rule induction systems identify toxic/active substructures by systematic generation of common substructures[9] or statistical analysis of fragment distribution in active and inactive compounds.[10] Once these structural features have been identified, the human expert can generalize these in the form of a generic structure, also called a Markush structure or pattern. Markush structures, containing generic parts, represent a set of specific structures. The generic structure can be of variable complexity, ranging from a simple substructure to a highly parametrized structure.

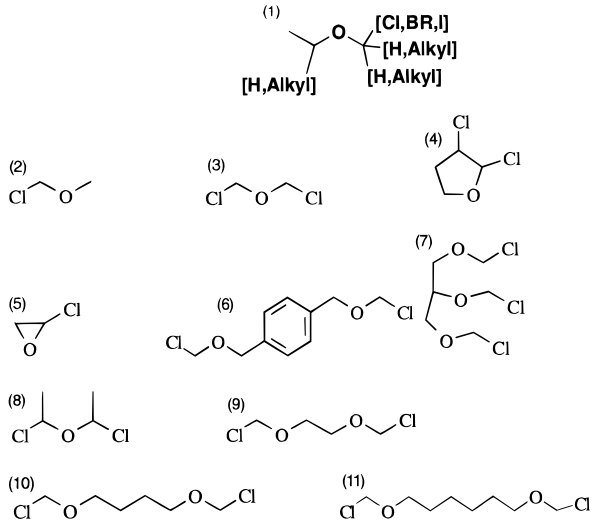

Figure 1. Example of derivation of a activity-related substructure. The common substructure or alert (**1**) supposedly responsible for conferring carcinogenicity was derived by a human expert from a set of tested active compounds (**2**−**11**) and from mechanistic considerations.

In most knowledge base systems, chemical patterns are represented, as far as the rule writer is concerned, by a character string following SMILES-like notation[11] or even in some cases by combinations of built-in generic terms.[12] DEREK, a system for toxicity prediction, and LHASA, a system for synthesis planning, both used a SMILES-like notation called PATRAN[13,14] for the coding of chemical patterns in their knowledge base (Figure 2). To complete PATRAN patterns, an English-like language called CHMTRN allows the association of structural inquiries to refine the structure specificity of rules (Figure 3).

† The University of Leeds.
‡ Imperial Cancer Research Fund.
§ Logic Programming Associates Ltd.
⊗ Abstract published in *Advance ACS Abstracts,* December 15, 1996.

S0095-2338(96)00094-7 CCC: $14.00

```
1D*PATTERN
    O[EPS=2]-C[HETS=1;HS=1]=C[HETS=0;HS=1]-C-C%[FUSION= ALKYLARYL]C-
    O[EPS=2]-C[HETS=2;HS=1](-@4)-@1
END*PATTERNS
```

Figure 2. Example of a PATRAN pattern representation of a bisfuranoid used in a rule of the DEREK knowledge base.

```
1D*PATTERN
    N[CHARGE=EITHER;ARYL=NO;RINGS=NO]
    C[ARYL=YES;FGNOT=NITRO,ISOCYANATE,NITROSO,DIAZO,AZIDE]%
    C%C%C%C%C@2
END*PATTERNS

KILL IF THERE ARE THREE METHYL GROUPS ON ATOM*1
FOR EACH ATOM ALPHA TO ATOM*1 OFFPATH DO N*SUBS
    IF SPECIFIED*ATOM 1 IS NOT A HYDROGEN*ATOM
    BEGIN GOT*SUBS
        IF SPECIFIED*ATOM 1 IS A CARBOXYL GROUP
        BEGIN CHK*AMIDE
            KILL IF THERE IS A NITROGEN ATOM ALPHA TO &
                SPECIFIED*ATOM 1 OFFPATH
            KILL IF THERE IS MORE THAN ONE OXYGEN ATOM &
                ALPHA TO SPECIFIED*ATOM 1 OFFPATH
        BLKEND CHK*AMIDE
        IF SPECIFIED*ATOM 1 IS A METHYL*ATOM THEN EXIT GOT*SUBS
        KILL IF SPECIFIED*ATOM 1 IS NOT A CABOXYL GROUP OR: &
            SULPHUR OR: SILICON
    BLKEND GOT*SUBS
N*SUBS ENDDO
```

Figure 3. Example of CHMTRN tests associated with a PATRAN pattern representation of aniline-related compounds used in a rule of the DEREK knowledge base.

CHEMICAL STRUCTURES IN KNOWLEDGE BASE SYSTEMS

Three aspects of chemical structures are to be considered in the context of knowledge base systems: representation, manipulation, and processing. The representation, or structural language, relates to the model used and dictates the types of chemical structures the knowledge base editor will accept, e.g., simple structures and simple or complex Markush structures. Structure manipulations are those operations the rule writer performs by means of an interface on the structures of the knowledge base, i.e., input, editing, and modifications. Structure processing is comprised of all the operations the computer performs on the internal representation of chemical structures, i.e., pattern matching, chemical perception, storage, *etc*. A suitable internal representation is crucial for good performance of the system. A forthcoming paper will focus on the StAR internal representation of structures.

KNOWLEDGE INPUT PROCESS

In the case of DEREK, LHASA, and similar systems, the role of the human expert is essential for good system performance. The expert analyzes specific information published in the literature, compiles it, and summarizes it according to the generalized knowledge of the domain and his own knowledge. The result of the extraction phase is that the expert is able to generate a rule representation suitable for input into the system and to add this to the knowledge base. During the coding phase, the expert translates the rules to the knowledge base system language and adds them to the knowledge base. This phase is followed by a series of tests and modifications to ensure the correctness of the results given by the modified knowledge base.

The coding of the chemical structures is effected by a linear description. The structures are coded using a fixed language with a given vocabulary and grammar. The rules are entered using a text editor, then compiled, and interpreted by the system. The coding, compilation, testing, debugging, editing, and correcting of the rules is a time consuming process. It is understandable that rule writers are looking for more friendly and appropriate tools for building and maintenance of knowledge bases. Another limitation of these traditional languages comes from the fact that they are designed to exclusively represent organic compounds. The manipulation of chemical knowledge, especially in the risk assessment context, is not only limited to organic chemistry but can also be applied to inorganic chemistry.

THE STAR PROJECT

The end of the century is marked by increasing regulations, controls, and developments of standards. Consequently, risk assessment, e.g., evaluation of chemical toxicity, has become an important factor in policy and safety decision making. Expert systems for toxicity prediction such as DEREK are valuable tools to organizations and individuals for evaluating hazards associated with particular situations or products.

Computers are particularly good at processing large amounts of data and at reaching logical conclusions from a set of arguments. In the risk assessment context, data are made up of sets of experiences or experiments. These data, e.g., data on confirmed carcinogens, often need reworking by a human expert before being input in the system knowledge base.

An argument is an assessment derived by the system from its sets of rules in a given context. The communication of details of its reasoning and of its assessment can be a difficult matter especially when addressing nonexperts of the domain.

The StAR project[15] intends to address three major problems faced by computer systems for toxicity prediction:

1. identifying and considering the arguments for and against a proposed risk to reach an overall impartial assessment,
2. communicating its conclusions about risk and reasoning in an unambiguous form understandable by any individual,
3. facilitating knowledge input by providing the rule writer with tools which make the writing of rules quick and easy.

In order to reach and communicate an assessment of risks, StAR utilizes the Logic of Argumentation which has been described elsewhere by Krause *et al.*[16] StAR uses general rules from the domain of toxicity combined with rules relating toxicity to the presence of specific structural alerts. In the chemical context, the knowledge base of the risk assessment system is constituted of structure−activity rules aimed at predicting the toxicity of chemicals. To facilitate the knowledge representation, StAR makes extensive use of the structural features of chemical representations. It provides user-friendly interfaces to render the input, editing, and correction of this structural information, simple, fast, and natural. As described in this paper, StAR provides the rule writer with three essential tools to facilitate the knowledge manipulation: a graphical language to help the expert formatting the structural data in a way suitable for rule formulation and knowledge base input; a chemical editor for drawing in structural patterns that are components of structure-activity rules; a rule editor for the production of the structure−activity rules. Adaptable to domains other than

chemical risk assessment, these tools might be the basis of new standards for the electronic manipulation and storage of chemistry-related knowledge.

EXTENDED MARKUSH STRUCTURES

As previously described by Dethlefsen *et al.*,[17] Markush structures can be divided into four categories.

1. Substituent Variations. Structures of this variation contain a list of alternative substituents on a ring or a chain. For example, a substituent may be specified by a set of possible instances such as halogen atom *X = Cl, I, Br, F*.

2. Frequency Variations. Structures of this type include a component for which repetition within the structure is variable. For example, structures containing chains of variable size parametrized by *−(CH2)n−* where the value of *n* is a defined interval of integer values.

3. Position Variations. This type of variation occurs, for example, when a group is allowed to be attached at more than one alternative position in a structure, e.g., a methyl with two possible attachment points on benzaldehyde allows the representation in one structure of *o*-, *m*-, and *p*-methylbenzaldehyde.

4. Homology Variations. This type of variation occurs when a standard nomenclatural term is used to describe a family of compounds, e.g., "alcohol 2−10C".

Based on experience acquired with DEREK and LHASA, the use of simple Markush structures to describe patterns for structure−activity rules is too limiting. In most cases, the patterns have to be associated with an additional series of structural restrictions to ensure the accuracy of the rule. A Markush representation would probably suffice in most of the cases if we only wanted to describe structures of a same family. However, in the rule induction process, we seek to describe an extended representation of a set of substructures sharing a congeneric core. The rule writer often includes generalizations, e.g., methyls generalized in alkyls or restrictions, e.g., no hydrogens on certain atoms, in the pattern. These generalizations and restrictions are based on external knowledge such as mechanistic considerations, chemical properties, negative information, or personal knowledge. In order to be able to provide the rule writer with appropriate graphical tools we considered the following three options:

1. force the expert to condense each structure rule into a single Markush pattern, without allowing any structural test-like inquiries,
2. retain the PATRAN and CHMTRN languages and provide appropriate graphical input systems and interpreter to automatically generate the rules,
3. allow the expert to condense the rule into an extended Markush pattern halfway between single generic structures and a pattern associated with tests.

The first option is too limiting in terms of the information that can be included in a single pattern. The second option, complicated to implement, would still result in the time-consuming elaboration of complex rules. The development of an appropriate structural language and implementation of complementary graphical interfaces satisfies both the rule writer's needs and the structure−activity rule requirements, and this is the route that has been followed in the StAR project.

THE STAR STRUCTURAL LANGUAGE

The StAR structural language describes generic organic and inorganic chemical structures by graphical means. One important aspect of the language is that the structures that are graphically-entered generate patterns coded on Molfiles directly understandable by the StAR system, i.e., substructures which StAR can use for pattern matching. The Psi graphical interface used by StAR is an extension of the drawing package in STN express developed by Hampden Data Services [Hampden Data Services Ltd., 49 Stoney Street, Nottingham NG1 1LX, UK].

The language includes five important features to allow the graphical definition of accurate structure−activity patterns: the generic definition of atoms and bonds; the use of the NOT operator to refine atomic environment; the definition of lists of atoms, bonds, and groups; the use of the OR, NOT, and AND operators; and the use of variable points of attachment. The generic definition of atoms and bonds allows the description of families of compounds. If required, patterns can be more precisely described by using the NOT operator to pose conditions on the environment of specific atoms. The definition of atoms, bonds, and groups in the form of lists allows the general description of substructures dissimilar by one functional group, atom, or bond type. In addition, the StAR structural language offers the possibility of combining the item definitions in using the OR, NOT, and AND operators. Finally, to allow the description of Markush structure of position, the language will also support atom or list of atoms or groups, variably attached at more than one point of the pattern. A detailed description of the StAR structural language is given as an annex to this paper.

CHEMICAL STRUCTURE PROCESSING

Standard perceptions, e.g., cycle and aromaticity perceptions, performed by chemical knowledge base systems on chemical structures have been the object of many publications.[14,18,19] They extensively make use of the fact that chemical structure diagrams are mathematical graphs. More of a problem are the operations a chemical expert system has to perform on Markush substructures contained in its knowledge base. All the features of the structural language must be "understood" by the expert system in order to be utilized in the processing phase, i.e., assessment of chemical toxicity in our context. For example, the use of superatoms relies on the perception by the system of the corresponding structural features or chemical properties.

Pattern matching is at the heart of the knowledge processing. The expert system routinely matches the generic structures in its knowledge base on specific structures. The pattern matching of Markush structures on chemical compounds is a difficult problem that has been studied for years.[20] The matching problem is a serious limitation to the use of Markush structures in expert systems. If not restricted to reasonable generic structures, the matching of patterns can quickly become complex and even unmanageable. Within the StAR project, we have limited the complexity of the structural language to a reasonable but sufficient level. For example, we do not allow the definition of fragments within fragments.

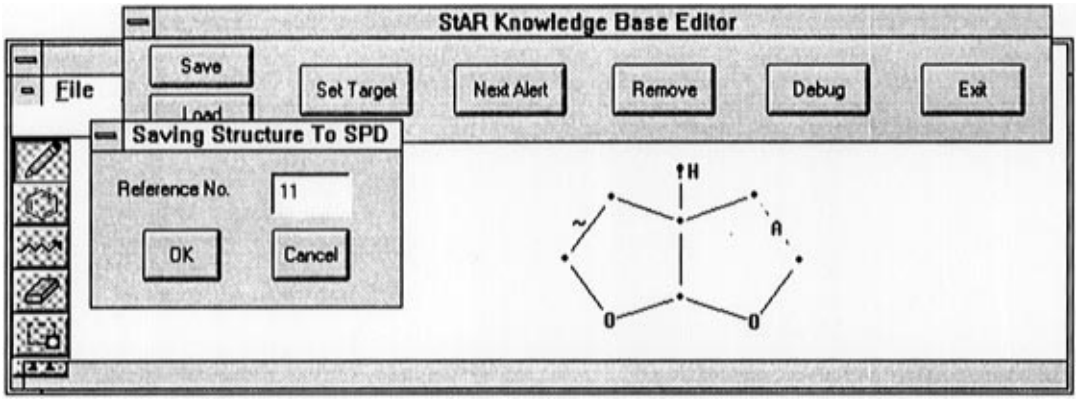

Figure 4. The rule writer can use the StAR Chemical Editor to put pattern structures into the system and edit patterns already in the rulebase.

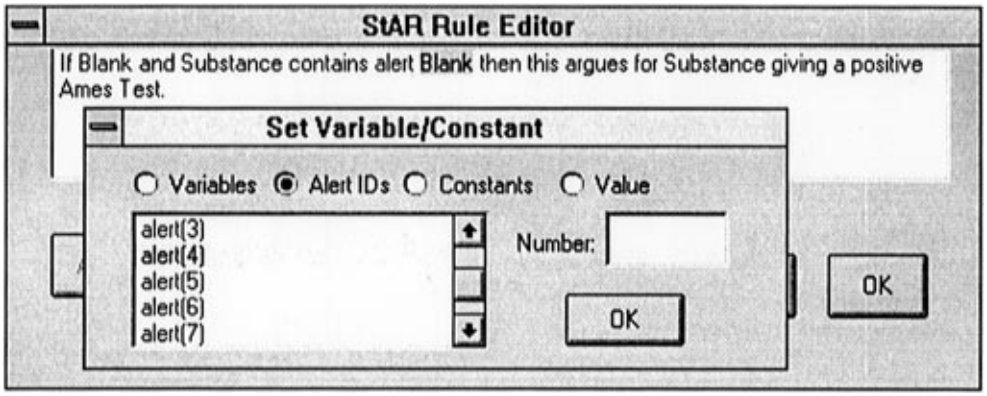

Figure 5. The rule writer can use the StAR Rule Editor to link structural alerts with specific endpoints.

STAR KNOWLEDGE BASE EDITOR

The StAR knowledge base editor allows the rule writer to describe structural alerts, i.e. substructures considered responsible for activity induction, and to formulate structure−activity rules electronically. The Psi chemical editor accepts any Markush structure defined in the StAR structural language. The rule writer can use this chemical editor to put pattern structures into the system and also to check and to edit patterns already in the database (Figure 4). Once the pattern is stored, the rule writer links it with a specific activity by entering a rule, e.g., "IF pattern #x THEN activity #y". The rule editor, specifically customized for the need of the StAR risk advisor, presents a series of menus making the creation of such rules fast and easy (Figure 5).

STAR PROJECT RESULTS

Started three years ago, the StAR project is now near completion. A carcinogenicity risk advisor has been implemented on a PC platform using the StAR technology. The StAR structural language has been used to represent the chemical structures (or structural alerts) stored in its database. A chemical editor and a rule editor have also been implemented. By using the chemical editor we have been able to graphically enter a series of active patterns. We have used the rule editor to link these structural alerts with toxic endpoints. The first version of the StAR carcinogenicity risk advisor is now linked with a limited but accurate knowledge base. It can already be used to assess the carcinogenicity of certain classes of compounds. For example, Figure 6 shows the alert summary and the StAR report obtained when processing a query structure. The system found evidence both for and against the realization of human carcinogenicity. The system has detected the presence of a structural alert for peroxisome proliferator activity in the query structure, but other rules in the knowledge base questioned the relevance of this activity to Humans.

CONCLUSION

The features of the StAR structural language have proved to be appropriate for the representation of structural alerts stored in the prototype version of the carcinogenicity risk advisor. In this prototype, the combination of the structural language, the chemical editor and the rule editor have eased the chemical knowledge manipulations as expected. Work is now under way to apply these graphical tools to the constitution of a full-scale knowledge base for carcinogenicity risk assessment.

The structural language developed within the scope of the StAR project is particularly appropriate for the representation of structural alerts. Combined with graphical computer tools, it provides a promising solution for chemical knowledge manipulation. As such, this model could be used at the core of other types of systems that require chemical knowledge manipulation and representation. With a few additions, the technology could be used for the implementation of databases of Markush substructures searchable by specific compounds or of databases of structures searchable by Markush substructures. Another possible application is the adaptation of the StAR structural language to represent the transformation patterns used in systems for Computer Assisted Designed Organic Synthesis (CADOS).[5] Reaction editors could then be used to manipulate reaction information graphically. Such

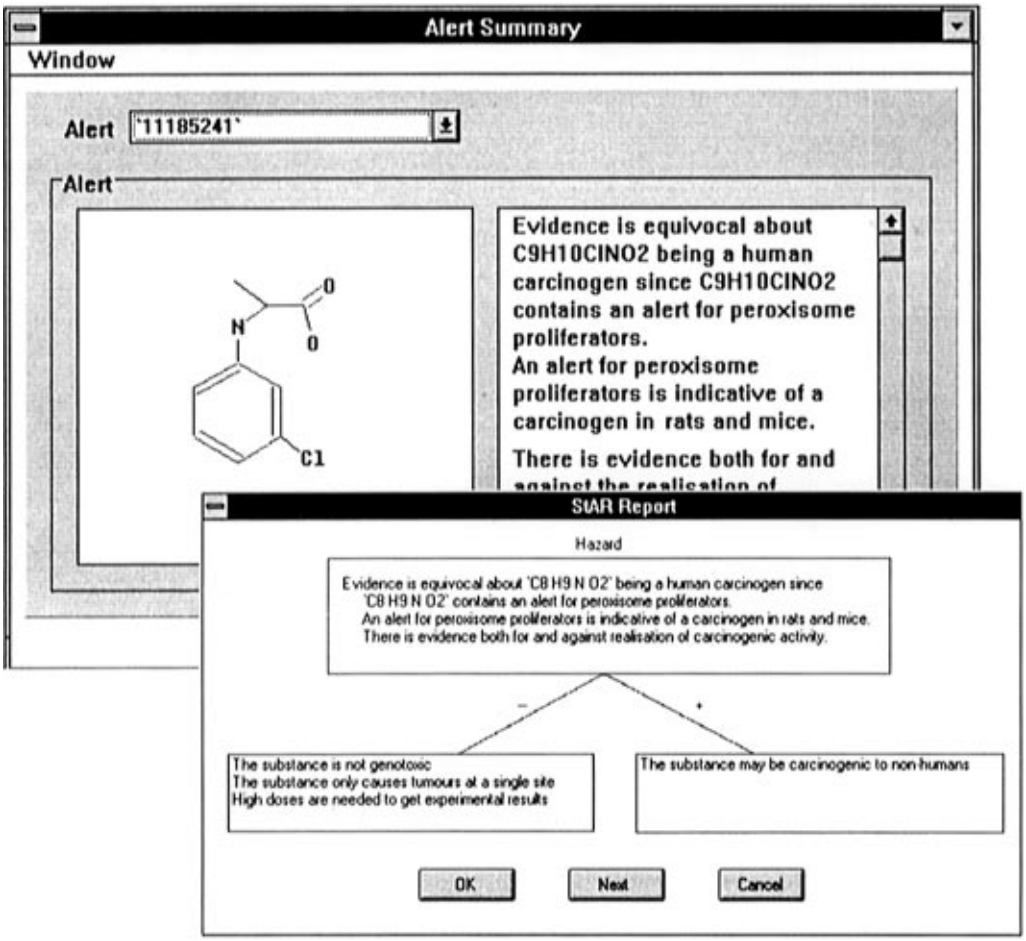

Figure 6. The StAR Carcinogenicity Risk Advisor can already be used to assess the carcinogenicity of a limited class of compounds.

tools would be of great help to control the huge amount of knowledge processed by these systems. Reaction systems are by no means limited to CADOS; reaction prediction systems, e.g., metabolism prediction systems, could also benefit from this approach.[21]

ACKNOWLEDGMENT

The StAR project is supported by the Department of Trade and Industry and the Engineering and Physical Sciences Research Council.

APPENDIX: THE STAR STRUCTURAL LANGUAGE

1. Atoms and Bonds. Items fundamental to the structural language include simple or generic definitions of atoms and bonds. Atoms and bonds can be generically-defined as a set of specific items, for example, halogen atom representing chlorine, bromine, fluorine, and iodine or any type of bond including the set of all specific bonds.

1.1. Atoms. 1.1.1. Specific Atoms. A specific atom is by definition one of the basic atom type of the periodical table. Note that hydrogens can be explicitly specified. As the patterns that are entered are to be considered as substructures, explicit hydrogens are meaningful and therefore kept.

1.1.2. Generic Atoms. Generic atoms represent sets of atoms or atomic groups with similar properties. These properties can be chemical, e.g., aromaticity or structural, e.g., alkyl group. The language not only allows commonly-used generic atoms, metals, halogens, aromatic atoms, heteroatoms, and alkyls, but also allows less common generic entities of great utility for pattern description, non-hydrogen atoms, any atom types, nonfunctionalized atoms, *etc*. This list is expandable (and expanding) on request by the rule writers (Figure 7).

List of generic atoms currently available:
"X": halogen atom
"A": any atom except H
"Q": any atom except C, H (heteroatoms)
"M": metal
"Ar": aromatic atom
$Alkyl_1$, $Alkyl_2$, $Alkyl_3$, $Alkyl_4$: "alkyl" groups

It is also possible to give attributes to atoms such as "in chain", "in ring", "at ring junction".

About Alkyl Groups. The elusive nature of what rule writers define as "alkyl" deserves some explanation. The chemist is generally quite clear about what "alkyl" means, a saturated hydrocarbon group. However, when ask what it should mean, rule writers say that it depends on the case. On this account, the StAR language currently offers four "alkyl" group definitions:

a. "$Alkyl_1$" is the alkyl group in the strict sense of the term; a saturated hydrocarbon group.
b. "$Alkyl_2$" is a saturated or unsaturated hydrocarbon group.
c. "$Alkyl_3$" is a group with no functionalities on the atoms at and α to the anchor point.

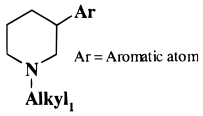

Figure 7. The use of generic atoms in patterns allows the description of families of compounds.

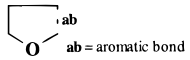

Figure 8. Example of a pattern containing a generic bond defined as aromatic.

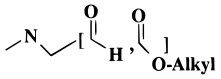

Figure 9. Example of a pattern containing a G-group defined by a list of two fragments.

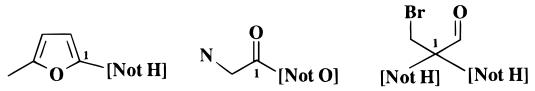

Figure 10. Patterns including restrictions on the environment of specific atoms.

d. "$Alkyl_4$" is a group with no functionalities, other than multiple C−C bonds, on the atoms at and α to the anchor point.

1.2. Bonds. 1.2.1. Specific Bonds. The list of bonds currently available includes those commonly used, single, double, and triple bonds, but the list is to be extended to those bonds more specific to inorganic chemistry: dative, ionic, and π bonds.

1.2.2. Generic Bonds. Apart from specific bond types, a number of generic bond types are also defined grouping bonds with similar chemical or structural properties: aromatic, any type, and ring junction (Figure 8). As for the atom type set, the set of generic bond types available will be expanded when needed. It is also possible to give attributes to bonds such as "in chain" or "in ring".

1.3. G-Groups. Typically, a G-group is a fragment (or substructure) associated with two anchor points, one on the fragment attachment point and one on the pattern (G-atom position). The two anchor positions define how the fragment is attached to the main pattern. The possibility to define lists of fragments is a powerful mean for Markush structure representation and is therefore of very common usage (Figure 9) (see section "3.Lists"). The use of generic atoms, e.g., alkyls, aromatic, *etc*., is allowed in G-group definitions. However, only one level of G-group definition is supported, i.e., it is not permitted to nest a G-group in another G-group definition. This restriction is intended to limit the complexity of the Markush structures to keep the internal structure processing of the system manageable.

2. NOT Operator. Often, the expert applies structural restrictions to general patterns to obtain more accurate pattern descriptions, e.g., by restricting the number of hydrogens on an atom. The StAR structural language allows imposition of conditions on the attributes of specific atoms (Figure 10).

3. Lists. The definition of atoms, bonds, and groups in the form of lists allows the general description of substructures dissimilar by (at least) one functional group or bond type (Figure 11).

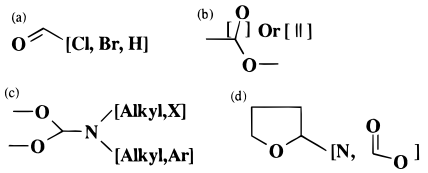

Figure 11. Patterns containing atom lists (a), bond lists (b), generic atom lists (c), and lists of atoms and groups (d).

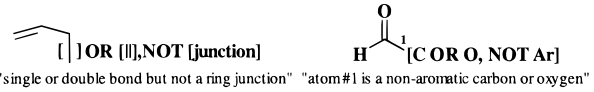

Figure 12. Patterns combining OR and NOT operators.

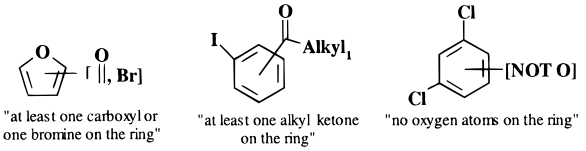

Figure 13. Patterns containing variable points of attachment.

4. Operators. The StAR language will support the combination of OR and NOT operators to accurately define types and item properties (Figure 12).

5. Variable Points of Attachment. The presence or absence of a specific atom at precise locations of a pattern structure can have dramatic effects on its activity. Positional isomers can present similar activities, e.g., *o*-, *m*-, and *p*-chlorotoluene can be all active. In such situations, to avoid splitting the pattern (e.g., one for each positional isomers), the StAR structural language supports the use of variable points of attachment (Figure 13).

REFERENCES AND NOTES

(1) Combes, R. D.; Judson, P. The Use of Artificial Intelligence Systems for Predicting Toxicity. *Pesti. Sci.* **1995**, *45*, 179−194.
(2) Corey, E. J.; Cramer, R. D.; Howe, W. J. Computer-Assisted Analysis of Complex Synthetic Problems. Methods and Procedures for Machine Generation of Synthetic Intermediates. *J. Am. Chem. Soc.* **1972**, *94*, 440−59.
(3) Corey, E. J.; Johnson, A. P.; Long, A. K. Computer-Assisted Synthetic Analysis. techniques for Efficient Long-Range Retrosynthetic Searches Applied to the Robinson Annulation Process. *J. Org. Chem.* **1980**, *45*, 2051−7.
(4) The Logic of Chemical Synthesis; Corey, E. J., Cheng, X. M., Eds.; Wiley: New York, 1989.
(5) Ihlenfeldt, W. D.; Gasteiger, J. Computer-Assisted Planning of Organic Syntheses: The Second Generation of Programs. *Angew. Chem. Int. Ed. Engl.* **1995**, *34*, 2613−2633.
(6) Ridings, J. E.; Barratt M. D.; Cary, R.; Earnshaw, C. G.; Eggington, C. E.; Ellis, M. K.; Judson P. N.; Langowski, J. J.; Marchant, C. A.; Payne, M. P.; Watson, W. P.; Yih, T. D. Computer Prediction of Possible Toxic Action from Chemical Structure - An Update on the DEREK System. *Toxicology* **1996**, *106*, 267−279.
(7) Long, A.; Combes, R. D. Using DEREK to Predict the Activity of some Carcinogens Mutagens Found in Foods. *Toxicol. in Vitro.* **1995**, *9*, 563−569.
(8) Richard, A. Application of SAR Methods to Non-Congeneric Data Bases Associated with Carcinogenicity and Mutagenicity: Issues and Approaches. *Mutation Res.* **1994**, *305*, 73−97.
(9) Ugi, I; Bauer, J.; Bley K.; Dengler A.; Dietz, A.; Fontain, E.; Gruber, B.; Herges, R.; Knauer, M.; Reitsam, K.; Stein, N. Computer-Assited Solution of Chemical Problems - The Historical Development and the Present State of the Art of a New Discipline of Chemistry. *Angew. Chem., Int. Ed. Engl.* **1993**, *32*, 201−227.
(10) Klopman, G. *J. Math. Chem.* **1991**, *7*, 187
(11) Weininger, D.; Smiles, D. 3. Depict. Graphical Depiction of Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 237−243.
(12) Woo, Y.; Lai, D. Y.; Argus, M. F.; Arcos, J. C. Development of Structure−Activity Rules for Predicting Carcinogenic Potential of Chemicals. *Toxic. Lett.* **1995**, *79*, 219−228.

(13) Hopkinson, G. A. Computer-Assisted Organic Synthesis Design; Ph.D. Thesis, Leeds University, 1985.
(14) Myatt, G. J. Computer Aided Estimation of Synthetic Accessibility; Ph.D. Thesis, Leeds University, 1994.
(15) Judson, P. N.; Fox, J; Krause, P. J. Using New reasoning Technology in Chemical Information Systems. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 621−624.
(16) Krause, P. J.; Ambler S. J.; Elvang-Goranssøn, M.; Fox, J. A Logic of Argumentation for Reasoning under Uncertainty. *Comput. Intelligence* **1995a**, *11*, 1.
(17) Dethlefsen, W.; Lynch, M. F.; Gillet, V. J.; Downs, G. M.; Holliday, J. D.; Barnard, J. M. Computer Storage of Generic Chemical Structures in Patents. 11. Theorical Aspects of the Use of Structure Languages in a Retrieval System. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 233−253.
(18) 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.* **1989**, *29*, 172−187.
(19) Mata, P.; Lobo, A. M.; Marshall, C.; Johnson, A. P. The CIP Sequence Rules; Analysis and Proposal for a Revision. *Tetrahedron Asymmetry* **1993**, *4*, 657−668.
(20) Holliday J. D.; Lynch M. F. Computer Storage of Generic Chemical Structures in Patents. 16.The Refined Search: An Algorithm for Matching Components of Generic Chemical Structures at the Atom-Bond Level. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 1−7.
(21) Klopman, G.; Rosenkranz, H. S. Toxicity Estimation by Chemical Substructure Analysis: The Tox II Program. *Toxic. Lett.* **1995**, *79*, 145−155.

CI960094P