of a ring represents chemist's intelligence to a certain extent.

REFERENCES AND NOTES

(1) (a) Frèrejacque, M. *Bull. Soc. Chim. Fr.* **1939**, *6*, 1008. (b) Plotkin, M. *J. Chem. Doc.* **1971**, *11*, 60. (c) Zamora, A. *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 40. (d) Bersohn, M. *J. Chem. Soc., Perkin 1* **1973**, 1239. (e) Esak, A. *J. Chem. Soc., Perkin 1* **1975**, 1120. (f) Schmidt, B.; Fleischhauer, J. *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 204. (g) Gasteiger, G.; Jochum, C. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 43. (h) Roos-Kozel, B. L.; Jorgensen, W. L. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 101. (i) Hendrickson, J. B.; Grier, D. L.; Toczko, A. G. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 195.

(2) Corey, E. J.; Wipke, W. T. *Science (Washington, D.C.)* **1969**, *166*, 178.

(3) Corey, E. J.; Petersson, G. A. *J. Am. Chem. Soc.* **1972**, *94*, 460.

(4) Wipke, W. T.; Dyott, T. M. *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 140.

(5) Fugamann, R.; Dolling, U.; Nickelsen, H. *Angew. Chem., Int. Ed. Engl.* **1967**, *6*, 723.

(6) The present paper tentatively introduced the terms *abnormal* and *abnormality* in order to describe a ring containing an atom other than C, N, O, S, and P. This convention intends to simulate a chemist's supposition that lies in his/her processes of ring perception.

(7) Multi-tied rings are not adopted here because of saving various arrays that are concerned with rings detected. However, another algorithm that considers both tied and multi-tied rings for covering conditions would be adopted for some purposes.

(8) Programs for detecting all rings have been reported. See: (a) ref 4. (b) Kudo, Y. In *Organic Synthesis and Computers* (R&D Report No. 39); CMC: Tokyo, 1983; p 49. (c) Balaban, A. T.; Filip, P.; Balaban, T.-S. *J. Comput. Chem.* **1985**, *6*, 316. We have used Kudo's algorithm after rewriting with FORTRAN 77.

(9) The comparison of the present program with various programs reported, especially with respect to their execution times, will be published later.

(10) (a) Ryan, A. W.; Stobaugh, R. E. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 22. (b) Freeland, R. G.; Funk, S. A.; O'Korn, L. J.; Wilson, G. A. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 94.

(11) By some extension, the present algorithm can be applied to the perception of ring-opening, ring-closure, and rearrangement reactions. See: Fujita, S. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 1. See also: Fujita, S. *J. Chem. Inf. Comput. Sci.* **1986**, *26*, 205; **1987**, *27*, 115.

Knowledge Representation Using an Augmented Planning Network: Application to an Expert System for Planning HPLC Separations

A. L. ANANDA* and S. M. FOO

Department of Information Systems and Computer Science, National University of Singapore, 10 Kent Ridge Crescent, Singapore 0511

HARI GUNASINGHAM

Department of Chemistry, National University of Singapore, 10 Kent Ridge Crescent, Singapore 0511

Received March 11, 1987

The augmented planning network (APN), which is modeled on the augmented transition network (ATN), is proposed as a convenient structure for representing the knowledge of an expert system for planning HPLC separations. The advantage of the APN is realized in its inherent flexibility in handling preconditions, meta-knowledge (or control knowledge), and procedural inferencing. The internal structure of the APN is also presented.

INTRODUCTION

The planning and optimization of separations in high-performance liquid chromatography (HPLC) have been the subjects of considerable interest. A number of quantitative approaches, including statistical techniques, have been employed, but these have been largely unsatisfactory. Although the theory of HPLC as it stands at present is well developed, when one is dealing with real separations, its application requires insight and experience. Also, experimental variability can pose limitations.

In the past two years, we have been concerned with developing an expert system approach to the problem.[1,2] The aim of the HPLC expert system is to guide the chemist in the selection of sample preparation technique, column (stationary phase), eluent (mobile phase), and detection technique by making use of heuristic knowledge derived from experts in the area of HPLC or from the literature. The motivation is that heuristic knowledge is based on experience that cuts across theoretical limitations.

The key element of the HPLC expert system then is the knowledge base, and, in our conception, the primitive elements of the knowledge base are rules. In the design of the expert system, we have adopted a hierarchical strategy where the selection process is divided into modular subtasks as shown in Figure 1. An important benefit of problem reduction in this way is that only those sections of the knowledge base relevant to the particular subtask being executed need be searched for a solution to the subtask. However, because the sections can be by themselves quite large, it is important that an efficient representational structure is chosen.

In the earlier design, the hierarchical structure is reduced to an AND/OR decision tree.[1] Each node of the tree is a rule that can be invoked if the conditional part of the rule is satisfied and the extraction of a solution was derived by a heuristic search of this tree. Although the simple decision tree is a convenient structure in that it is easily implemented, it has a number of limitations. The most significant is the tendency for the tree to proliferate as the number of possible solutions increases.

In this paper we seek to show that the augmented transition network (ATN) formalism is a more efficient structure for representing the knowledge base of the HPLC expert system. In the search for a solution, the exploration of alternatives is natural to the ATN's network form.

The ATN model has been widely used to represent grammars for question-answering systems such as the LUNAR program.[3] It has also been used in speech-understanding systems,[4] in the design of a separable transition-diagram compiler,[5] for modeling learning, and even for processing visual information.[6] ATTENDING[7] uses the ATN model as the basis of its critiquing analysis of anesthetia plans for patients.

A KNOWLEDGE REPRESENTATION MODEL FOR THE HPLC EXPERT SYSTEM

For this application, networks are called *augmented planning networks* (APNs). Each APN is composed of *states* (represented by circles) connected by *arcs*. Each network has an *initial state* from which a *path* can be traced through the

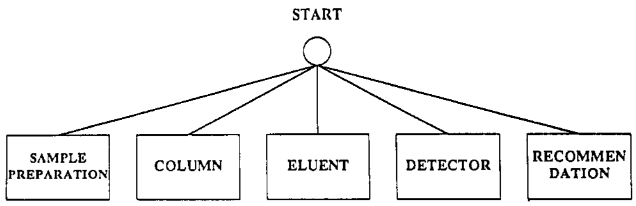

Figure 1. Hierarchical modular structure of the HPLC problem.

network by following successive arcs. An arc labeled POP is chosen to indicate the end of the path in the network. An arc in the network can be a *terminal* or a *nonterminal* arc. In this particular network model, a terminal arc is labeled with the name of an elemental object (e.g., name of a particular compound, column, eluent, or technique of detection). On the other hand, a nonterminal arc is labeled with the name of a nonelemental object (e.g., name of a general separation technique or even the name of a subtask) such that tracing of a path through the subnetwork indicated by the arc name is necessary before the path is continued in the original network. The ability to label arcs with entire subnetworks essentially constitutes a procedural facility.

In Figure 2, those arcs labeled with capital letters are nonterminal and the others are all terminal. Hence, for example, at state S2, the subnetwork COLUMN in Figure 2b has to be traced before control is returned to the higher level network in Figure 2a. Figure 2c depicts a network that actually contains all the rules inferring the same goal, namely, the determination of the type of the column. This way of grouping the rules into a single network based on the goal further modularizes the knowledge.

Note that Figure 2c is a network of a lower level, as it represents the rules in the most explicit manner possible. One can, for example, easily observe that if the conditions "compound = prednisolone" and "compound = prednisone" are satisfied, the value "silica" will be chosen for the column type. This is a rule that will be used in the HPLC expert system for determining the column type, and it is formally stated as

Rule 32. **If** compound = dexamethasone **then** column type = silica

Another such rule that is depicted in Figure 2c is:

Rule 34. **If** compound = prednisolone **and** compound = prednisone **then** column type = silica

From Figure 2a it can be seen that the hierarchical, modular structure conceived for the HPLC expert system in Figure 1 is easily translated into the APN format without any loss of information.

INITIALIZATION is a module for establishing the value of critical parameters that will be made use of in the later modules. These items include sample and compound. SAMPLEPREPARATION suggests the method to prepare the sample for the chromatographic experiment; COLUMN, ELUENT, and DETECTOR recommend the column(s), eluent, and detector to be used, respectively. RECOMMENDATION presents the solutions found from the previous modules. MODIFICATION is utilized only after the first cycle, when an initial plan has been suggested. It suggests refinements to a previous plan in the event an acceptable separation is not obtained.

In a typical decision tree, paths from an initial state successively diverge, whereas for the APN, paths that diverge at one point may later rejoin. This allows greater economy of expression in defining decisions to be made and, equally important, limits the proliferation of the number of possible states and arcs.

By way of illustration, we can consider the simple domain of selecting a column for the separation. In the module COLUMN (see Figure 2b), either an exact match is found, in which case a lower level task, EXACT-COLMATCH (not shown in Figure 2), which simply obtains a suitable column from the data base, is executed, or, alternatively, in the event an exact match does not exist, the choice must be inferred.

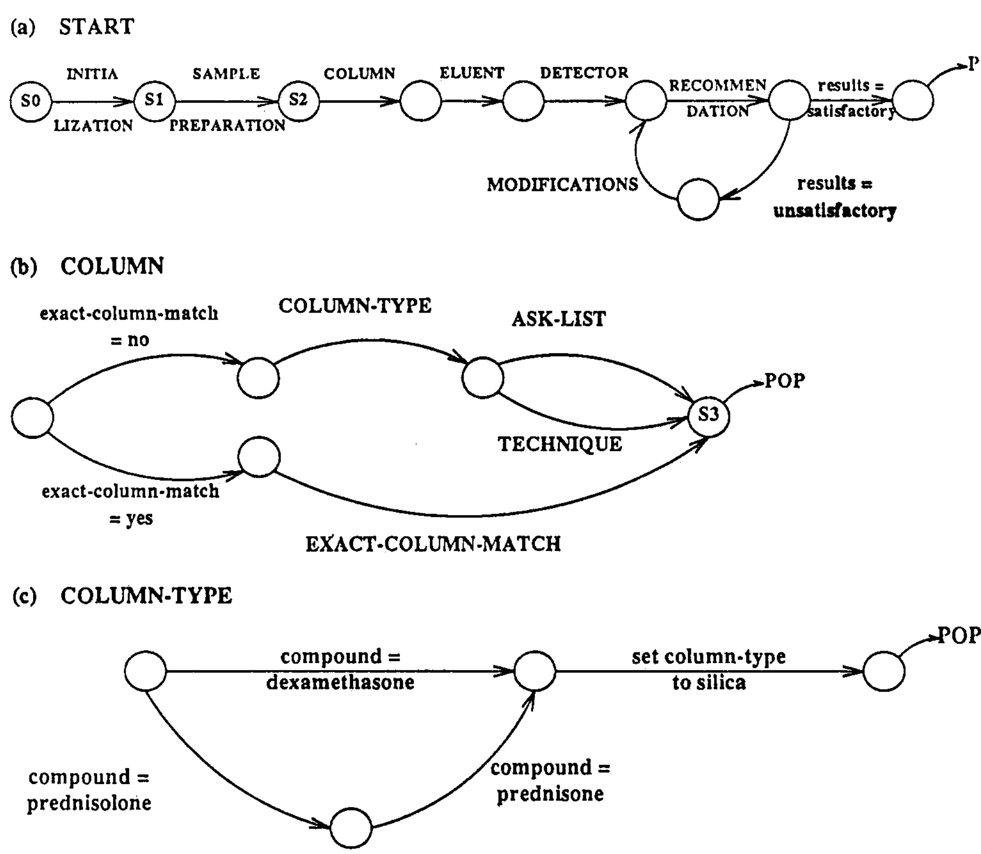

Figure 2. Some APN's used in the HPLC expert system.

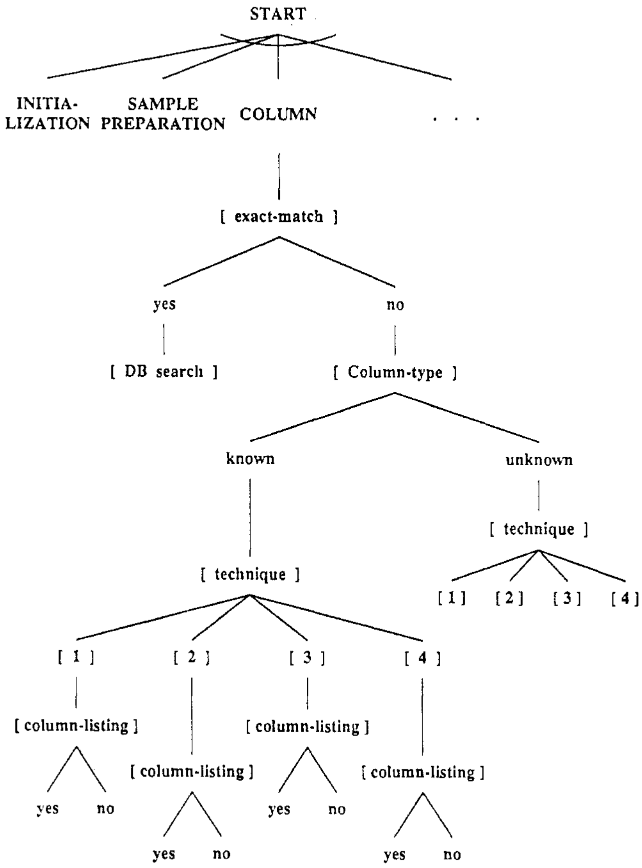

Figure 3. Proliferating decision tree.

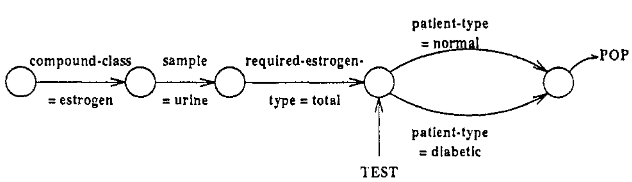

Figure 4. APN depicting rule 11 and rule 12.

Hence, a divergence occurs at this point. Eventually, however, the paths rejoin at S3.

To represent the same information in a decision tree, a simple structure is obviously impossible, as each divergence will cause the growth of new subtrees. The corresponding AND/OR decision tree is shown in Figure 3.

In the HPLC application, it may be beneficial to incorporate procedural calls in certain inferencing rules. In these procedures, tests or actions may be invoked. This enables a drastic reduction in the number of rules that otherwise have to be explicitly included in the rule base. This is especially true if the conditions of certain rules overlap considerably. For example, consider the following rules:

Rule 11. **If** compound class = estrogen **and** sample = urine **and** total estrogen required **and** patient = normal **then** use acid hydrolysis

Rule 12. **If** compound class = estrogen **and** sample = urine **and** total estrogen required **and** patient = diabetic **then** use enzyme hydrolysis

The above two rules may be represented in the APN shown in Figure 4. TEST is an embedded procedure for obtaining the value of patient type, which can be derived from the patient data base and the laboratory records.

Another case in which procedure embedding is useful is in the assignment of confidence factor as illustrated in the following example. Consider the following rules:

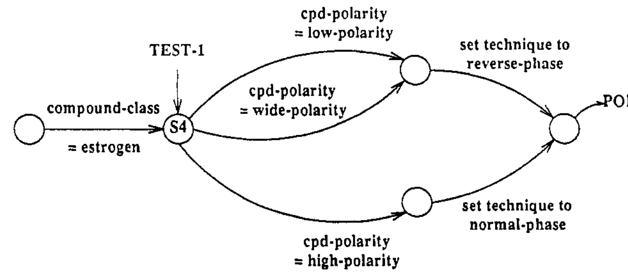

Figure 5. APN depicting rule 5, rule 6, and rule 7.

Rule 5. **If** compound class = estrogen **and** cpd polarity = low polarity **then** technique = reverse phase **cf 80.**

Rule 6. **If** compound class = estrogen **and** cpd polarity = high polarity **then** technique = normal phase **cf** 70.

Rule 7. **If** compound class = estrogen **and** cpd polarity = wide polarity **then** technique = reverse phase **cf 60.**

In this case, an embedded procedure, TEST-1 (see Figure 5), is employed to test for the value of cpd polarity at S4 before fixing the confidence factor for the conclusion.

In the above rules it is implicit that the value of the variable, cpd polarity, must already contain some valid value for comparison. A precondition is necessary to check the availability of such a valid value. If cpd polarity has not yet been determined, then an embedded procedure is called to do so. Hence, the building-in of a precondition is actually an augmentation of an arc in the APN involved. Without use of the embedded procedure, many more rules would have been needed to obtain the value of cpd polarity.

In an APN, an arc may be augmented by being associated with an action routine that can set and test contextual flags to guide the traversing of the APN. For example, consider the following rule in the knowledge base of the HPLC expert system:

Rule 46. **If** at least 2 compounds to be separated differ in number of aryl groups they possess **then** activate polarity rules.

In this rule, the condition is first tested and if satisfied, the arc that leads to a subnetwork has to be activated, enabling the firing of the polarity rules. In other words, an action routine activates a certain control switch in the polarity rules so that they have priority over other rules. Similarly, other rules that are made inappropriate may be deactivated.

OVERVIEW OF THE GENERAL STRUCTURE OF APN

The APN is a single-entry and single-exit network consisting of nodes that represent the states and arcs that represent the conditions of the rules. A node in such a network acts either as a control point at which a procedure may be invoked to test a precondition before travelling through any of the enumerating arcs from that node or as a point at which conclusions are reached and values assigned. Furthermore, a node may be classified as a start node or an intermediate node. Every start node consists of a nodal description that keeps track of the information of the current rule.

The data structure for representing the APN internally is shown in Figure 6. The main features in an APN are discussed below.

(a) APN Identifier. This is unique for each APN and, as the name implies, serves as an identifier for the APN. For example, it may be an index to the variable for which conclusion is sought.

(b) APN Attributes. These include the following: (1) Activity status of the APN. The status of the APN can be either inactive, executable, or at a higher priority over the other

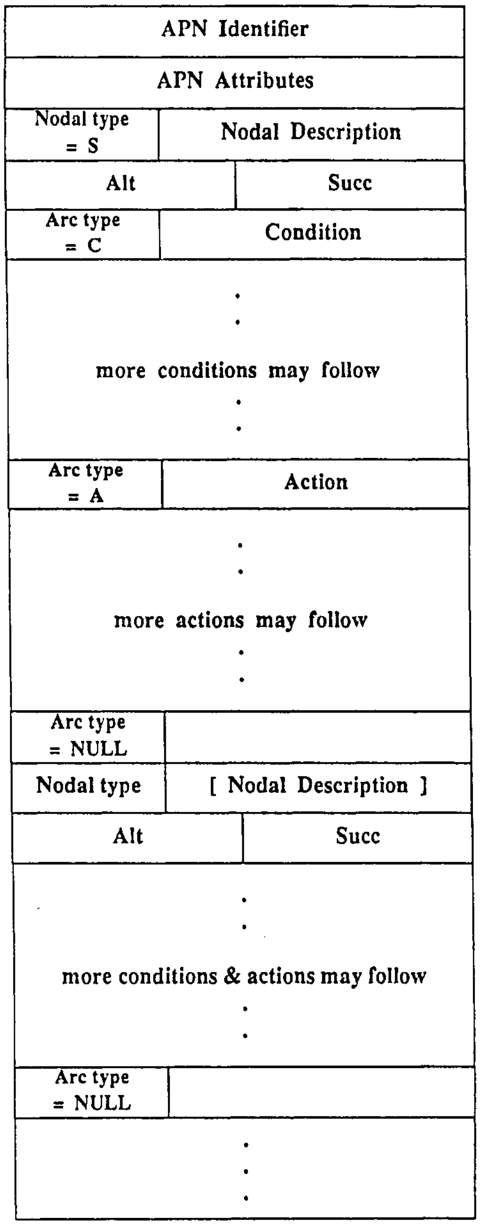

Figure 6. Internal representation of an APN.

APNs to be executed. At the beginning of a planning session, the status of each APN is initialized to executable. (2) A counter. This field is used to indicate the number of times the APN has been executed.

(c) Nodal Type and Nodal Description. The nodal type may be S (for start node) or I (for intermediate node). The nodal description of an S-node includes information on the following: (1) The rule number. One application of this field is to serve as an index to a text file for extracting the appropriate explanatory text for display when requested at any point of time during the consultation session. (2) Activity status of the rule. This has the same function as the activity status at the APN level. (3) A counter. This is to keep track of the number of times the rule (or APN path) has been executed.

(d) Alt and Succ. Attached to each node are the two fields, Alt and Succ. *Alt* contains the pointer to any alternative path that may exist at the current network. *Succ* contains the pointer to the next node to be executed if the current condition is satisfied. Functionally, Alt and Succ form the linkages within the APN and establish all the possible solution paths to the problem in question. NULL at the Alt and Succ fields indicates the absence of an alternative and a successor, respectively.

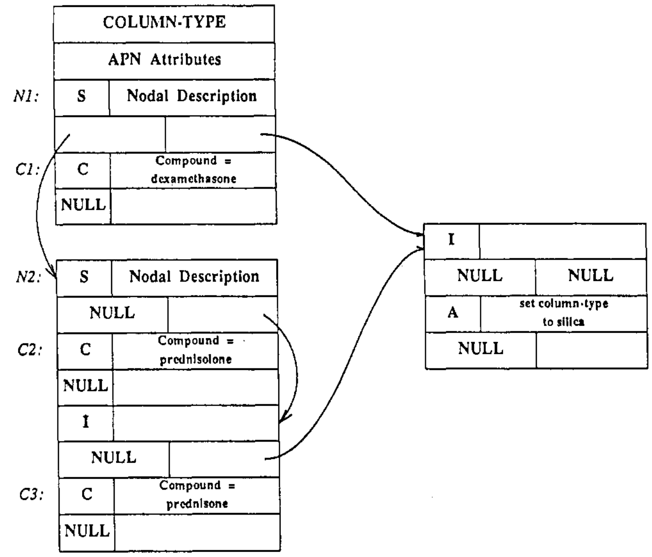

Figure 7. Corresponding internal structure of Figure 2c.

(e) Arc Type. Arc type may take the value C to mean a condition, the value A to indicate an action, or a null value. Arcs labeled null are sometimes used in the graphical representation of an APN so as to conform to the single-entry and single-exit definition of the APN.

(f) Condition. This is the condition to be tested before an arc is selected in the graphical representation of the APN.

(g) Action. An action is a procedure that the driver invokes to perform a task such as assigning a value to the variable for which a solution is sought. It may also be a procedure for setting some status or counter or simply for performing the task of exiting from the current APN.

EXAMPLE OF AN APN

Consider a simple example of determining the columns for separation. The procedure flows according to Figure 2a. Assuming that the system has finished the task of determining the preparation technique for the sample and is now at the state S2, the next task to be performed is to find the column(s) to be used. The inference engine will then load all APNs (and hence all rules) that will help determine the columns to be used.

At this point, the immediate goal is the column, and thus the COLUMN APN is first invoked. During the execution of the COLUMN APN, the system will come to a point where the COLUMN-TYPE APN needs to be invoked if there is no exact match for columns, that is, if there are not exact past experiences of the problem.

The target now is column type, and the rules involved are rule 32 and rule 34. The corresponding graphical APN is shown in Figure 2c, and the internal representation is depicted in Figure 7. The problem at hand is stated as follows: *to separate the compounds, prednisolone and prednisone, in a given sample.* As the driver traverses down the COLUMN-TYPE APN (see Figure 7), it first does a condition match at C1 and, on failing to find a successful match, takes the alternative path to N2, indicating a branch to a new rule. The nodal description at N2 becomes the current rule information, whereas the one at N1 is treated as the previous rule's information. This stacking of nodal description is necessary for handling user queries at any point of time as well as for some system use. When the driver concludes that condition C2 is satisfied, it then takes the successor address and does a condition match at C3. The success of this condition calls for the invocation of an action to set the value of column type to "silica". After performing the task, the system checks that both the Alt and Succ fields are NULL and hence makes a

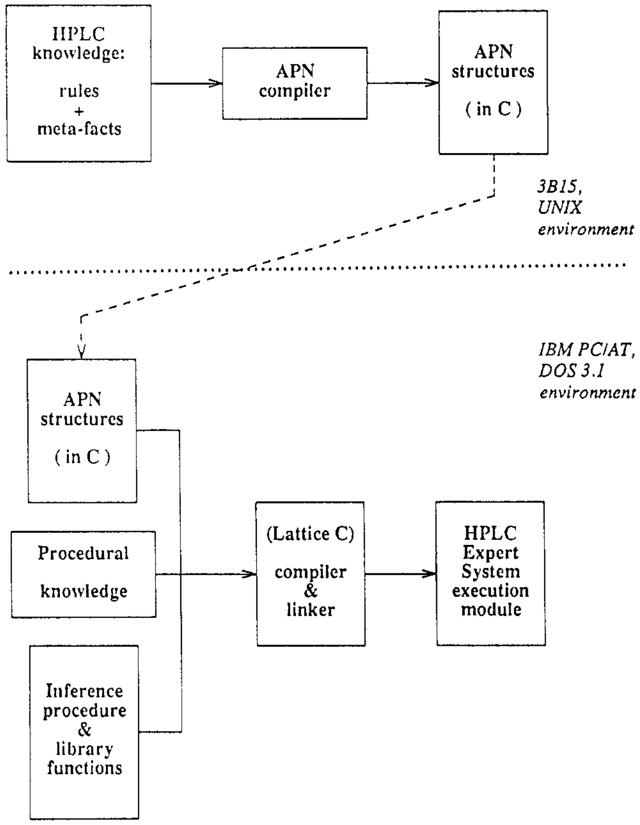

Figure 8. Overview of the implementation method of the HPLC expert system.

return to the higher level APN, COLUMN, where the COLUMN-TYPE APN was called.

IMPLEMENTATION

The knowledge representation using APN's that is described in the previous sections has been adopted for the HPLC expert system. The main objective of the development effort is twofold: first, to develop an expert system for HPLC in a laboratory network environment where many chemists could share a common knowledge base; second, to develop a portable expert system that can be used on microcomputers.

In order to implement the APN structure, a compiler was developed. The compiler accepts rules and facts in a structured format and transforms them into the APN structures in C language. In the initial system, a commercially available expert system shell, M.1, was employed to input the rules and facts. The APN structures are transferred to the Lattice C environment on an IBM PC/AT, and they are then further compiled along with user-supplied procedural knowledge. A schematic diagram of the compilation process is shown in Figure 8.

The major part of the development work on the compiler has been carried out in the UNIX System V environment on AT&T's 3B15 machine. The lex and yacc utility on UNIX were used extensively. An extensive run-time support system with debugging and editing features is being developed to facilitate the user in monitoring the inferencing path and also to control to some degree the execution sequence by modifying the working storage parameters. The user interface is being developed by using the GKS (Graphics Kernel System) graphics software package with Lattice C as the binding language on a microcomputer.

ACKNOWLEDGMENT

The provision of a university research grant, RP 33/85, for this work is gratefully acknowledged.

REFERENCES

(1) Gunasingham, H.; Srinivasan, B.; Ananda, A. L. "Design of a Prolog-based Expert System for Planning Separations of Steroids by High-Performance Liquid Chromatography". *Anal. Chim. Acta* **1986**, *182*, 193–202.
(2) Ananda, A. L.; Gunasingham, H. "Development of a Cybernetic Chromatograph". *Proceedings of the Interdepartmental Seminar on Artificial Intelligence, Nov 1985*; Department of Information Systems and Computer Science, National University of Singapore, Publication No. TR11/85; pp 56–72.
(3) Woods, W. A. "Progress in Natural Language Understanding: An Application to Lunar Geology". *Proc. AFIPS Conf. 42*; American Federation of Information Processing Press: Virginia, 1973; pp 441–450.
(4) Wolf, J.; Woods, W. A. "The HWIM Speech Understanding System". In *Trends in Speech Recognition*; Lea, W., Ed.; Prentice-Hall: Englewood Cliffs, NJ, 1980; pp 316–339.
(5) Conway, M. E. "Design of a Separable Transition-Diagram Compiler". *Commun. ACM* **1963**, *6*(7), 396–408.
(6) Tropf, H.; Walter, I. "An ATN Model for 3-D Recognition of Solids in Single Images". *Proc. IJCAI* **1983**, *2*, 1094–1098.
(7) Miller, P. L. *A Critiquing Approach to Expert Computer Advice: Attending*; Pitman: London, 1984.