Censored Production Rules to General Rule Structure Converter

Nabil M. Hewahi
Computer science Department
Islamic University of Gaza
Palestine, Israel
nhewahi@mail.iugaza.edu

Abstract:- Standard rule structure of the form IF condition THEN action is a very simple structure and can be written easily by inexpert people . The standard rule structure is widely used in expert systems and various applications. One of the main problems of the standard structure is that it can’t represent real time systems where conclusions should be obtained within time limits. Censored Production Rule (CPR) is a rule that can handle real time applications in which given more time, we are more certain. The General Rule Structure (GRS) also serves real time application with ability to give more certain and specific answers given more time. GRS has also other advantages that makes it important to expedite the process of inference. The problem of GRS is its complexity in terms of structure and the need of an expert in a narrow domain to construct the relations between the rules. To avoid having an expert and still with the abilities of GRS, we propose a converter that converts a CPR into a GRS, we shall call this rules converter CR-GRCON. This will eliminate the need of having an expert to make GRS structure and any person can provide the rules to the system which will convert them to GRS structure.

Keyword:- Rule-based systems, Variable precision systems, Real time system

1 Introduction

The standard rule structure is very well known in the area of expert systems. The structure of standard rule structure is (<IF Condition HEN Action>). As an extension of standard production rule, Michalski and Winston[7] proposed the Censored Production Rule (CPR) to exhibit Variable Precision Logic (VPL) in which certainty varies which certainty varies, while specificity stays constant. The form of CPR is as follows:

IF Condition THEN Action UNLESS Censor

Where the censor is the exception condition. Such rules are employed in situations in which the conditional statement ‘IF Condition Then Action’ holds frequently and the assertion Censor holds rarely. The censors are only checked whenever time permits. The more time we have, the more censors can be checked and the more we are certain from the conclusion. If any censor holds we can not take the action of the rule. Such rules are used in real time systems. As an example to CPR

IF Working-Day THEN John-in-office UNLESS John-is-sick, John-on-leave

To address the various problems and shortcoming with CPRs system, Bharadwaj and Jain [1] have introduced a concept of Hierarchical Censored Production Rule (HCPR). HCPR is a CPR augmented with specificity and generality information, which can be made to exhibit variable precision in the reasoning such that both certainty of belief in conclusion and specificity may be controlled by the reasoning process. Such a system has numerous applications in situations, where decision must be taken in real time and with uncertain information.

The HCPR structure and its formal shape are shown below

(<Action
IF condition
Unless censor
GENERALITY general-information
SPECIFICITY specific-information>)

(<A
{concept/decision/head}
IF B[b1,b2,….,bm]{preconditions (AND) conditions}
UNLESS C[c1,c2,….,cn] {censor conditions (OR conditions)}
GENERALITY [G] {general information}
SPECIFICITY[a1,a2,….,ak]{specific information} >)

The general information G in HCPR is the clue about the next general concept related to the concept A in hierarchy. Hence, general information is useful in the backward chaining of inference. The specificity information is the clue about the next set of more specific concepts(goals,decision, or actions) in a knowledge base, which are the most relevant and which are the most likely to be satisfied after successful execution of that HCPR. The set of more specific information has XOR relation between its members (only one decision in the set is true at a time). Hence, specific information is useful in forward chaining of inference.

A few related HCPRs can be linked together to form HCPR-Tree. Each HCPR in the tree has a parent HCPR (except the root), which is its generality defined in the GENERALITY clause and children HCPRs (except the leaves), which is its specificity defined in SPECIFICITY clause. The HCPRs in the higher levels give more general results, whereas, lower level HCPRs give more specific results [1] . Afew related HCPRs are given below and it is shown how they are linked in a HCPR-tree structure, as depicted in Fig.1. (This represents a rule-base to find answers to queries of the type 'What is X doing?', when supplied with relevant input data):

{level 0}
Is-in-city(X,Y)
IF [Lives-in-city(X,Y)]
UNLESS [Is-on-tour(X)]
GENERALITY[ ]
SPECIFICITY[Is-at-home(X), Is-outside-home(X)]

{level 1}
Is-at-home(X)
IF [Time(night)]
UNLESS[Is-doing-overtime(X), Works-in-night-shift(X)]
GENERALITY[Is-in-city(X,Y)]
SPECIFICITY[ ]

Is-outside-home(X)
IF [Time(day)]
UNLESS [Is-sick(X)]
GENERALITY[Is-in-city(X,Y)]
SPECIFICITY[Is-working-outdoor(X),Is-entertaining-outdoor(X)]

{level 2}
Is-working-outdoor(X)
IF [Day(working)]
UNLESS[National-holiday,Is-unemployed(X)]
GENERALITY[Is-outside-home(X)]
SPECIFICITY[]

Is-entertaining-outdoor(X)
IF [Day(Sunday)]
UNLESS [Met-an-accident(X)]
GENERALITY[Is-outside-home(X)]
SPECIFICITY[ ]

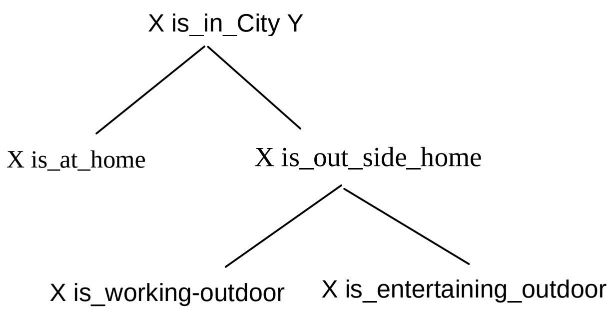

Fig. 1. A HCPR-tree for life queries

In HCPR system, any HCPR is a more specific case of its parent. Thus, the root HCPR represents the most general concept. Once this is verified, we can descend to its children for more specific information, depending on our requirement and resources available. If any HCPR we reach is blocked, either due to one of preconditions being false or one of one of the censor conditions being true, there is no need for further search of its children. As the concept becomes more specific, the number of elements of its IF-set increases. However, we are not required to list all such elements along with the HCPR. This is because the total inheritance is an inherent feature of the HCPR-tree structure; each HCPR inherits the entire IF-set of its parent and thus of all its ancestors.

Hewahi [6] developed the General Rule Structure (GRS) which is based on the ideas of variable precision logic rules and ripple down rules developed by Compton and Richards [2]. GRS can be application independent and comprehensible (understandable). The term 'general ' used in GRS indicates that the developed rule structure takes care of generality in terms of application and reasoning process (forward and backward chaining). GRS can be used for real time applications, control applications (from the characteristics of VPL) and standard rule-based expert system applications. Based on HCPR, GRS can give more certain and specific answers, whenever time permits. GRS can be easily used in directing the system to decide, which rule should be checked next if the currently checked rule is failed. One of the main advantages of the GRS structure is the simplicity to train it; therefore, the system can determine the most commonly used rules. This would reduce the time consumed for finding the proper rule to be fired. The general form for GRS is as below

(<Action
IF[b1,b2,…..,bk] (w)
{AND conditions, (w is the weight of the GRS and related to the concept of training and omitted for our purpose). }
UNLESS (c1,c2,…,cn) {OR conditions}
GENERALITY (g1,g2,….,gm) {XOR relation}
SPECIFICITY (a1,a2,…..,am)
{XOR relation-the maximum number of specificities is m or less }
ALTERNATIVELY(al1,al2,…,alm)
{XOR relation-the maximum number of alternatives is
m or can be less}
>)

It is to be noticed that the only one term added to HCPR to form GRS is the ALTERNATIVELY, which is responsible to direct the system to which rule (GRS) it should try if the current tried one is failed. The meaning of SPECIFICITY and GENERALITY in GRS are the same as HCPR but the usage is different. Very few studies have been done on GRS[8][9]. To understand the meaning of all the terms of GRS, Figure 1(this figure is taken from [6]), is a collection of GRSs forming a GRAPH and two GRAPHs form a SET (at least two GRAPHS form a SET). The rule structure of Fig. 2 would be as below :

1. Is-in-city(X,Y)
IF [Lives-in-city(X,Y)]
UNLESS [Is-on-tour(X)]
GENERALITY []
SPECIFICITY[Is-at-home(X)]
ALTERNATIVELY [Is-in-city(X,Z)]

2. Is-in-city(X,Z)
IF [Lives-in-city(X,Z)]
UNLESS [Is-on-tour(X)]
GENERALITY [ ]
SPECIFICITY[Is-at-office(X)]
ALTERNATIVELY [ ]

3. Is-at-home(X)
IF [Time(night)]
UNLESS [Is-doing-overtime(X),
works-in-night-shift(X)]
GENERALITY [Is-in-city(X,Y)]
SPECIFICITY[]
ALTERNATIVELY [Is-at-office(X)]

4. Is-at-office(X)
IF [Day-between-Mon-Wed]
UNLESS [Is-sick(X)]
GENERALITY [Is-in-city(X,Y),Is-in-city(X,Z)]]
SPECIFICITY[Is-in-comp-room(X)]
ALTERNATIVELY [( ).Is-at-field(X)]

5. Is-in-comp-room(X)
IF [Time-between-9-12]]
UNLESS [ ]
GENERALITY [Is-at-office(X)]
SPECIFICITY[ ]
ALTERNATIVELY [Is_at_finance_office(X)]

6. Is-in-finance-office(X)
IF [Time-between-1-5]
UNLESS [ ]
GENERALITY[Is-at-office(X,Y)]
SPECIFICITY[ ]
ALTERNATIVELY [ ]

7. Is-in-field(X)
IF [Day-between-Th-Fri]
UNLESS [ ]
GENERALITY [Is-in-city(X,Z)]
SPECIFICITY[ ]
ALTERNATIVELY [ ]

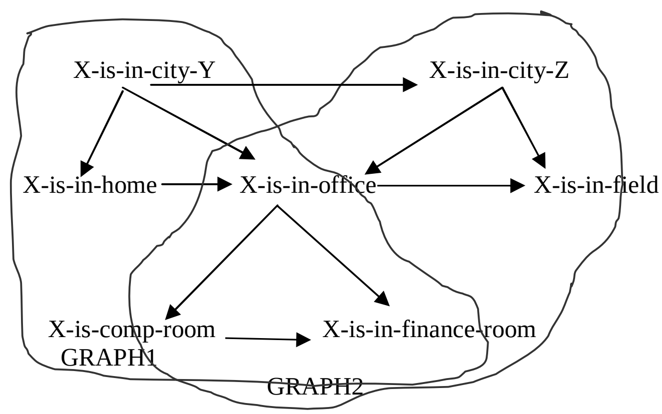

Figure2 .Two GRAPHs forming one SET. In this case there is a common subgraph in both the GRAPHs.

To clarify some of the GRSs meaning, we explain some of them. GRS number 1, says, X is in city Y if X lives in city Y and X is not in tour. If GRS number 1 fails, the next GRS to be tried is GRS number 2, otherwise, the next GRS to be tried is the specificity of GRS number1, which is GRS number 3. The GRS number 4 says, if the day is between Monday and Wednesday, then X is at office unless X is sick. The alternate GRS that can be checked next if GRS number 4 fails depends on the path we follow. If the system reaches to GRS number 4 from GRS number 1, then no alternate GRS, whereas if it comes from GRS number 2, the alternate GRS is GRS number 7.
It is still in general a little work has been done on GRSs[4][5][6].

2 Research Goal

Because GRS is an important structure that can be useful in real time applications as well as standard application, it is good to use it to be a general structure. One of the main limitations of GRS systems is the need to an expert to build the tree structure, which is considered to be more complex than simply the standard rule structure or even CPR. Because CPR has a simple form and can be written without the need of an expert. We propose a system that takes CPRs as its inputs and produces GRSs as its outputs. This converter (CR-GRCON) will eliminate the need of expert to form the structures of the GRSs. This process is depicted in Figure 3.

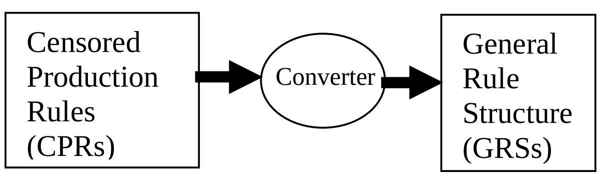

Figure 3. Converter that converts CPRs to

The general idea for constructing the GRSs is to start with having several groups, each will have initially a supposed root for GRS GRAPH. Gradually the related GRSs are formed in one group until all the relations are considered. This means we may start with n groups and end up with m groups where n >= m. n might be equal to m in cases where there is no further relations between the groups after constructing the roots. This says we have separate CPRs which is usually not common.

3 The Proposed Converter

In this section, we state the general steps of the proposed converter CR-GRCONV. The general GRS structure used is a little bit different from the above explained shape, this is done to serve the proposed converter:

[rule name] IF [condition]
THEN [action]
UNLESS[censors]
GENERALITY[general information]
SPECIFICITY[specific information]
ALTERNATEVILY [alternative rule]

The CR-GRCON main cases and general algorithm steps are:

1. Check firstly all the CPRs with only one condition. This gives an impression that those rules could be root nodes for GRS GRAPHs.

1.1 Put all the CPRs with the same condition in a separate group regardless of its action. We shall have n number of graphs, G_i is the ith group which might contain one or more CPRs. The number of groups is the number of different actions.

1.2 Each group G_i with only one CPR, do the following

1.2.1 If there is any other group G_j with the same condition and different action, form a CPR that handle both the conditions with their censors. This is explained as below

G_i : IF working-day
THEN John-in-office
UNLESS John-is-sick

G_j : IF working-day
THEN David-in-office
UNLESS David-is- on-leave

We shall get a rule of the form

IF working-day
THEN John-in-office+David-in-office

UNLESS John-is-sick,David-is-on-leave

1.2.1.1 Remove G_i and replace the CPR in G_j with the formed rule.

1.2.1.2 After forming all the CPRs with the conditions in 1.2. and applying 1.2.1 and 1.2.1.1. We shall have one GRS with all the formed CPRs with the same condition. The GENERALITY, SPECIFICITY and ALTERNATEVILY will have empty clauses.

1.2.2 If there is any other group G_j with the same action and different condition, form two GRSs, one for each CPR. Make one GRS as alternate to the other one. For example

G_i : IF temperature-is-high
THEN Summer
UNLESS unusual-weather

G_j: IF sunny THEN summer
UNLESS winter-season

The two formed GRSs are as below

```
[SUMMER] IF [temperature-is-high]
         THEN [summer]
         UNLESS[unusual-weather]
         GENERALITY[]
         SPECIFICITY[]
         ALTERNATEVILY
         [SUMMER-1]

[SUMMER-1] IF [sunny]
         THEN [summer]
         UNLESS [winter-season]
         GENERALITY[]
         SPECIFICITY[]
         ALTERNATEVILY []
```

1.2.2.1 Remove the two CPRs and Put the two formed GRS in one group, say G_i and remove G_j (our aim is finally to have in each group one SET or at least one GRAPH).

1.2.2.2 After forming all GRSs using steps of 1.2, 1.2.2, we shall have many SETs or GRAPHS, each in one group.

1.2.3 Otherwise (not as 1.2.1 or 1.2.2), construct a GRS for the CPR with empty clauses for the GENERALITY, SPECIFICITY and ALTERNATEVILY.

2. After finishing step 1, we shall have GRSs in each of the groups. Some groups might have incomplete or complete SETS or GRAPHs.

2.1 For each GRS in group G_i has a condition which is a subcondition of another GRS in G_j, the GRS in G_j will be the specificity of G_i. For example

G_i:

```
[EX1] IF [Lives-in-city(X,Y)]
      THEN[Is-in-city(X,Y)]
      UNLESS [Is-on-tour(X)]
      GENERALITY []
      SPECIFICITY[EX3]
      ALTERNATIVELY [EX4]

[EX3] IF [Time(day)]
      THEN[Is-at-office(X)]
      UNLESS[Is-sick(X)]
      GENERALIY[EX1]
      SPECIFICITY[]
      ALTERNATEVILY[]
```

G_j:

```
[EX2] IF [Lives-in-city(X,Y), Time(night)]
      THEN [Is-at-home(X)]
      UNLESS[Is-t-tour(X),
             Is-doing-overtime(X),
             works-in-night-shift(X)]
      GENERALITY [ ]
      SPECIFICITY[]
```

ALTERNATIVELY []

The GRS in Gj will be removed and the Gi will have the following rule structures:

G_i:

```
[EX1] IF  [Lives-in-city(X,Y)]
      THEN[Is-in-city(X,Y)]
      UNLESS [Is-on-tour(X)]
      GENERALITY []
      SPECIFICITY[EX3]
      ALTERNATIVELY [EX4]
[EX2] IF [Time(night)]
      THEN [Is-at-home(X)]
      UNLESS [Is-doing-overtime(X),
              works-in-night-shift(X)]
      GENERALITY [EX1 ]
      SPECIFICITY[]
      ALTERNATIVELY []
[EX3]   IF [Time(day)]
      THEN[Is-at-office(X)]
      UNLESS[Is-sick(X)]
      GENERALIY[EX1]
      SPECIFICITY[]
      ALTERNATEVILY[EX2]
```

We notice that all the GRSs in Gi and Gj are becoming in Gi to form one complete structure. The GENERALITY clause of EX2 will have EX1 because EX1's condition is a sub condition of EX2. Moreover, The ALTERNATIVELY clause of EX3 (which the specificity of EX1) was empty and becoming EX2. This is occurring because EX2 is now a more specific GRS of EX1, but should be tried if EX3 fails. It is to be observed that if EX1 is not having a specificity information, EX2 would be the specificity information of EX1.

3. An important case which might rarely occur is the case where one GRS is shared between to GRS GRAPHS as GRS number 4 in section 1. To make the idea clear, assume the following GRSs within the groups G_i and G_j :

G_i:

```
[EX1] IF [A]
      THEN [B]
      UNLESS[]
      GENERALITY[]
      SPECIFICITY[EX2 ]
      ALTERNATIVELY[]
[EX2] IF [C]
      THEN [M]
      UNLESS[]
      GENERALITY[EX1]
      SPECIFICITY[EX7]
      ALTERNATEVILTY [EX3]
[EX3] IF [G]
      THEN [Z]
      UNLESS[]
      GENERALITY[ ]
      SPECIFICITY[ ]
      ALTERNATIVELY[]
```

G_j:

```
[EX4] IF [E]
      THEN [R]
      UNLESS[]
      GENERALITY[ ]
      SPECIFICITY[EX5]
      ALTERNATIVELY[]
[EX5] IF [N]
      THEN [Y]
      UNLESS[]
      GENERALITY[EX4 ]
      SPECIFICITY[ ]
      ALTERNATIVELY[EX6
[EX6] IF [C]
      THEN [M]
      UNLESS[]
      GENERALITY[EX4]
      SPECIFICITY[ ]
      ALTERNATIVELY[]
```

The EX2 is removed and both the groups G_i and G_j will be in one group. The GRSs which change are those below:

```
[EX1] IF [A]
      THEN [B]
```

UNLESS[]
GENERALITY[ ]
SPECIFICITY[EX6]
ALTERNATIVELY[]

[EX6] IF [C]
THEN[M]
UNLESS[]
GENERALITY[EX4,EX1]
SPECIFIVITY[EX7]
ALTERNATIVELY[(),EX3]

4 Experimental Results

CR-GRCON was tested under many different cases and was tested by several people to check its ability for conversion. As a complete system Haddawy [3]'s estimator knowledge base for CPRs system is used to test our proposed system. The knowledge base used is concerned with equipments necessary for heating system. Any heating system may consist of seven main components, room thermostat, duct sensor, pipe sensor, valve, panel controller, heating coil, and fan. The system is to learn selecting the proper equipment. At the beginning the number of CPR were 74 and the converter produced 19 GRS GRAPHs covered all the cases.

5 Conclusions

In this paper we have presented CR-GRCON system converter to convert a censored production rule based system to general rule structure based system. This system was proposed to get rid of needing an expert to form the GRS structures and to get use of the advantages of GRS structure which can handle real time applications as well as standard applications. Some of the main cases included in CR-GRCON were discussed. CR-GRCON has gone under many tests with different cases by various individuals. One important system converted using CR-GRCON is the Haddawy 's knowledge base based on censored production rules and used for heating system equipment selection. Some of the future directions is (1). Because GRS is still new structure, it would be good to applying it to various applications and explore its behavior (2) incorporate CR-GRCON into a software (expert system tool) based on GRS structure to be used directly by expert system builders .

References

[1] K.Bharadwaj and N.Jain," Hierarchical censored production rules(HCPR) system", Data and Knowledge Engineering, Vol.8,1992,19-34.
[2] P.Compton and D.Richards,"Taking up the situated cognition challenge with ripple down rules",International Journal of Human Computer Studies, special issue on situated cognition 49,1998,895-926.
[3] P.Haddawy, A variable precision logic inference system employing the Dempster-Shafer uncertainty calculus , MS thesis (UILU-ENG-86-1777), University of Illinois at urbana champaign, 1987.
[4] N.M.Hewahi,"Principles on integrating general rule structure (GRS) based system and neural networks, Accepted for The 2004 International Conference on Artificial Intelligence (IC-AI'04),Las Vegas, USA, June 21-24,2004
[5] N.M.Hewahi,"Credit assignment scheme for general rule structure", Proceedings of the 1st Indian Inter. Conf. On Artificial Intelligence, Hyderabad, India,2003,833-842.
[6] N.M.Hewahi, "A general rule structure", Journal of Information and Software Technology,44, 2002, 451-457.
[7] R.Michalski And P.Winston, Variable precision logic, Artificial Intelligence,29,1986,121-145.