故障診斷씨스템을 위한 知識獲得에 관한 硏究

柳 吉 洙

A Study on Knowledge Acquisition for a Machine Trouble-Shooting Consultation System

Keel-Soo Rhyu

〈目 次〉

Abstract

Knowledge based on expert systems are used in many domains, and incorporating domain specific knowledge into a program is difficult because of the many forms knowledge takes. Thus the process of extracting knowledge from an expert and transfering it to a program is an important and difficult problem. This process is termed knowledge acquisition, and this knowledge acquisition is a bottleneck in the construction of expert systems.

Recently, knowledge acquisition has used four methods. One, the expert interacts with a knowledge engineer or program to build the system. Two, the ex-

pert conversant with computer technology interacts directly with the expert system via an intelligent editing program, for example TEIRESIAS(Davis 1976). Three, an induction program is used for automatic knowledge acquisition, but this method has not been implemented for expert systems to date. Four, a method that may become feasible in the future, is acquiring the knowledge directly from text books.

In this study, the knowledge acquisition system for a machine trouble-shooting consultation system is studied. The system consists of the meta-rules(META-R) to control the system, the rules (RULE-F) for extracting production rules from the data-base, the rules (RULE-I) for generating production rules from the input data of an expert, a data-base based on frame theory, and a knowledge-base for storing the rules produced. The proposed architecture has been tested on a marine engine trouble-shooting system.

1. 序 論

知識베이스를 사용한 엑스퍼트씨스템이 종래부터 여러 分野에 응용되어 왔으며, 그러한 專門分野는 제각기 가지고 있는 知識이 다르기 때문에 表現方法도 다르다. 그러기 때문에 共通分野에 있어서 汎用性을 가진 씨스템의 구축에 관해서도 많은 연구가 진행되어 왔다. 또한 그러한 엑스퍼트씨스템은 專門知識이 복잡한 관계로 대부분의 씨스템이 知識表現이나 推論能率을 향상시키기 위해서는 유연한 프로덕션루울을 이용한 루울型씨스템으로 구축되어 있다. 이와 같이 知識工學의 研究者를 위해서 또는 專門家를 위해서 知識을 표현하기 쉬운 방법이 종래부터 연구되어 왔지만, 현재에도 知識獲得問題는 엑스퍼트씨스템을 구축하는데 있어서 보틀넥크(bottleneck)인 것이 現狀이다.

최근 知識獲得의 방법으로는

- 知識工學의 研究者가 專門家와 對話에 의해서 知識을 獲得하는 전형적인 방법[6) 7) 8) 10) 14)]
- 專門家가 인텔리젠트에디트(intelligent editor)를 매개체로 직접 엑스퍼트씨스템과 對話하면서 知識베이스를 구축하는 방법, [11) 15) 16)]
- 인간이 과거의 경험으로부터 부분적으로 知識을 獲得하는 것과 같이 歸納프로그램에

의해서 專門知識을 自動的 또는 半自動的으로 獲得하는 방법,[17]

• 장래에 실현가능할 것으로 보는, 씨스템이 직접 교과서로부터 知識을 獲得하는 방법

등이 있다.

본 稿에서는 기계의 故障診斷씨스템을 위한 知識獲得方法으로서 프레임(frame)型의 데이타베이스(data-base)로부터 프로덕션루울(production rule)을 생성하는 방법, 專門家로부터 입력된 動作, 狀態, 故障 등의 데이타를 프로덕션루울로 변환시키는 방법에 대하여 고찰하며 실험대상으로는 船舶의 主機關을 선택했다.

2. 씨스템의 構成

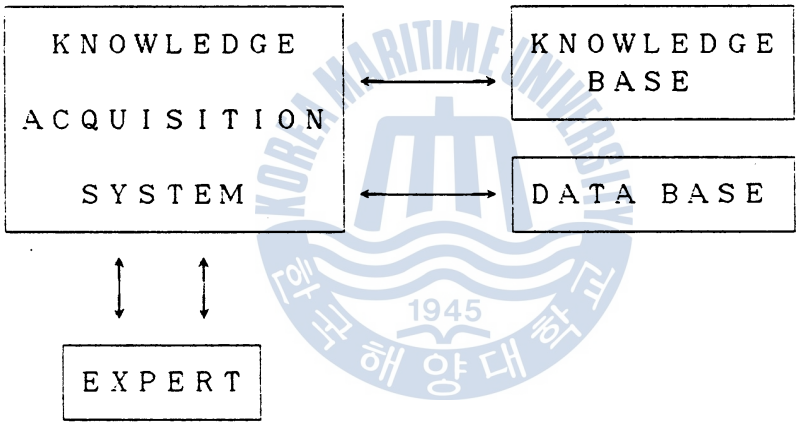

Fig. 1 The structure of an expert system

일반적인 엑스퍼트씨스템은 推論機構와 知識베이스에 의해서 구성되어 있으나 본 稿에서는 그림 1과 같이 데이타베이스(DB), 지식베이스(KB), 知識獲得機構로 구성했다. 또한 制御知識을 특수한 專門知識의 事實이나 關係로부터 분리해서 抽象的으로 나타나는 것은 씨스템의 설계를 보다 알기 쉽게 하고 설명하기 쉽게 하는 것과 동시에 다른 領域에 있어서도 그 知識베이스를 구축하는데 있어서 유용하게 사용될 수 있으며, 制御戰略의 성질을 연구하기 위해서도 유용한 출발점이 될 수 있다. 이와 같은 장점을 살리기 위하여 知識獲得機構는 그림 2와 같이 META-R, RULE-F, RULE-I로 구성했으며 META-R은 制御知識을 포함한 메타루울의 集合, RULE-F는 데이타베이스로부터 프로덕션루울을 생성하기 위한 루울의 集合, RULE-I는 專門家로부터 입력을 프로덕션루울로 변환하기 위한 루울의 集合으로 구성했다.

then (frame－name condition ? ac))

라는 루울을 생성하고 {part－of}, {lubricating}, {cooling} 슬로트의 값에 관한 루울을 생성시킨다.

즉 현재 주목하고 있는 프레임이 機械의 部品인 경우 그것의 狀態가 운전중이며, 그 부품을 윤활 또는 냉각시키기 위한 액체의 狀態가 非正常이라면 그 부품도 非正常이 된다는 루울을 생성함과 동시에 {part－of}, {lubricating}, {cooling}의 슬로트값에 관한 루울의 생성을 진행시킨다는 의미이다. 이와 같은 루울의 內部表現은 그림 5와 같다.

```
(part-rule01
  (f_exist-frame $frame a-kind-of-part)
  (= $part-of (f_search-slot part-of))
  (= $lubricating (f_search-slot lubricating))
  (= $cooling (f_search-slot cooling))
  (= $state (f_search-slot state))
  ->
  (= $current-rule
     (f_make (if (and ($part-of state run)
                      (or ($frame state $state)
                          ($lubricating condition abnormal)
                          ($cooling condition abnormal)))
                 then ($frame condition abnormal))))
  (f_comp $current-rule)
  (f_index $current-rule)
  (f_product-rule $part-of)
  (f_product-rule $lubricating)
  (f_product-rule $cooling))
```

Fig. 5 Internal representation of RULE－F

3.2 專門家의 입력에 의한 루울의 생성

3.1절에서도 언급한 바와 같이 故障씨스템을 위한 知識生成을 주목적으로 하기 때문에 전문가로부터 입력이 故障狀態를 나타내는가 또는 動作이나 正常狀態를 나타내는가를 분리하여, 실제로 이 분리작업은 메타루울이 하고 있지만, 動作이나 正常狀態를 나타내는 경우에는 크게 6종류 즉 機械類, 펌프類, 탱크類, 밸브類, 액체類 그리고 部品類로 나누어 각각의 특징에 의해서 非正常의 狀態로 변환시켰다. 예를 들어 다음과 같은 루울이 그 중에 하나이다.

if 條件節이 run 狀態를 포함하고
結論節도 run 狀態를 포함한다.

then (if (and (結論節이 run 狀態)
(條件節이 逆狀態))
then (結論節의 frame - name condition ? ac))

라는 루울을 생성한다. 이와 같은 루울의 內部表現은 그림 6에 나타나 있다.

```
(pr-ss-nn02
        (f_ssvt $condition state run)
        (f_ssvt $action state run)
        ->
        (f_mr (if (and (f_sv $action)
                       (f_rsv $condition))
              then (f_acmv $action))))
```

Fig. 6 Internal representation of RULE-I

4. 메타루울

메타루울은 본 씨스템의 중심부이며 專門家와의 對話에 의해서 입력된 데이타를 분리하고 이 분리된 데이타를 가지고 데이타베이스로부터 프로덕션루울을 생성하기 위한 루울(RULE-F), 動作이나 狀態로부터 프로덕션루울을 생성하기 위한 루울(RULE-I), 데이타베이스, 知識베이스를 제어한다.

이 메타루울을 작성하기 위한 作用要素로는 다음과 같은 것을 이용했다.

- (左邊 → 右邊) : 左邊을 실행하여 성공하면 右邊을 실행한다.
- (= var1 var2) : var2 를 作業記憶(working memory)으로부터 찾아서 그 값이 존재하면 그 값을, 그렇지 않으면 var 2 자신을 var 1의 값으로서 作業記憶에 저장한다.
- (= var1 pred) : pred 를 평가하여 그 값을 作業記憶에 var 1의 값으로 저장한다.
- (NOT pred) : pred 를 평가한 값이 nil (無) 이라면 성공한다.
- (pred var 1 · ·) : pred 를 평가하여 그 값이 존재하면 성공한다.
- (DO var PR) : var 의 값을 作業記憶에서 찾아내어 그 값의 각각의 要素에 대하여 이름이 PR 인 메타루울을 적용시킨다.
- (EVERY var pred) : var 의 값을 作業記憶에서 찾아내어 그 값의 각각의 要素를 순

번에 의하여 pred에 적용시켜 전체 요소가 성공하면 EVERY도 성공한다.

- (SOME var pred) : var 의 값을 作業記憶에서 찾아내어 그 값의 각각의 要素를 순번에 의하여 pred에 적용시켜 하나의 要素라도 성공하면 SOME도 성공한다.

이와 같은 作用要素를 사용한 메타루울의 內部表現은 그림 7에서 보여주고 있으며, 메타루울의 주요한 기능은 다음과 같다.

```
(mr execute-product-rule
        (= $prules (f_epr $active-rules))
        ->
        (do $prules verification-PR))
```

Fig. 5 Internal representation of RULE-F

4.1 입력된 데이타의 분리

專門家로부터의 입력이 데이타베이스로부터의 루울생성을 요구하는가, 事實을 나타내고 있는가 또는 動作이나 狀態를 나타내고 있는가를 인식할 필요가 있다. 입력이 DB인 경우에는 데이타베이스로부터의 루울생성을 요구하는 것으로 RULE-F를 기동시킨다. 입력이 INPUT인 경우에는 事實 또는 動作이나 狀態에 관한 입력의 예비신호이며 그 후의 입력이 if ~ then 형식을 취하고 있으면 動作이나 狀態에 관한 입력으로 간주하여 RULE-I를 기동시키고, if ~ then 형식이 아니면 事實에 대한 입력으로 간주하여 그 事實이 데이타베이스에 존재하고 있는가를 확인하여 존재하지 않을 때는 그 事實을 데이타베이스에 삽입함과 동시에 RULE-F를 기동시켜 입력된 事實에 대하여 생성할 만한 루울이 있는가를 조사하여 그러한 루울이 존재하면 루울을 생성시킨다.

動作이나 狀態를 나타내고 있을 때는 RULE-I를 기동시키기 전에 다음과 같은 것을 확인한다. 專門家가 입력하는 데이타는 "밸브를 연다", "펌프를 기동시킨다" 등의 動作이나 "냉각수의 압력은 2.0이다" 등의 狀態 또는 "주기관은 非正常이다" 등의 故障狀態가 있다. 故障狀態를 나타낸 입력에 대해서는 입력 그 자체를 知識으로 인정하며, 動作이나 狀態를 나타내는 입력에 대해서는 RULE-I를 적용시키거나 動作의 경우에는 프레임에 존재하지 않기 때문에 動作을 狀態로 변환시킬 필요가 있다.

예를 들어 "starting-air-valve를 open한다"고 하는 動作은 open하는 動作에 의해서 밸브의 狀態가 open 狀態로 변하기 때문에 (starting-air-valve state open)과 같이 內部變換시킨다. 그러나 機械類나 펌프類는 動作이 start-stop이고 狀態가 run-stop이기 때문에 "X를 Y한다"고 하는 입력에 대하여 다음과 같은 메타루울이 필요하

다.

If X가 frame－name 이고 동시에
X가 機械類 또는 펌프類이며,
Y가 start 이다.
then (X start run)으로 변환한다.

고 하는 메타루울에 의해서 動作으로부위 狀態로의 변환이 가능하다.

또한 메타루울은 RULE－F, RULE－I를 기동한 경우에 루울集合으로부터 적용 가능한 部分루울集合을 선출하여 이 部分集合으로부터 적용해야 할 하나의 루울을 선택하기 위하여 競合解消를 해야 한다. 본 씨스템에서는 競合解消를 하기 위하여 적용 가능한 部分루울集合으로부터 條件節의 길이가 가장 긴 루울, 즉 특정적인 사실을 가장 많이 포함하고 있는 루울을 선택했다.

4.2 非正常程度의 결정

생성된 루울에(frame－name condition ? ac)와 같은 事實이 포함되어 있으면, 즉 abnormal 인가 critical 인가가 아직 결정되지 않은 루울에 대해서는 다음과 같이 결정하는 방법을 택했다.

(frame－name condition ? ac)와 같은 事實이 結論節에 존재할 때에는 이 事實의 frame－name을 데이타베이스로부터 찾아내어 그 프레임에 존재하는 need－of 슬로트의 값 중에, 생성된 루우트의 條件節에 있는 각 事實의 frame－name과 동일한 값이 존재하는 가를 확인하여 존재할 경우에는 그 값의 必要性(CF)에 의하여 abnormal 또는 critical을 결정한다. 본 씨스템에서는 이 CF가 0에서 1사이의 값을 취하고 0.6미만인 값에 대하여는 abnormal을 취하고 0.6이상인 경우에는 critical을 취했다. 또한 need－of 슬로트가 존재하지 않을 경우에는 結論節에 ? ac가 속해있는 事實의 frame－name과 條件節에 있는 각 事實의 frame－name과의 必要性을 專門家에게 질문하도록 작성했다.

(frame－name codition ? ac)와 같은 事實이 條件節에 존재할 때는 이 事實의 frame－name을 데이타베이스로부터 찾아내어 그 프레임에 존재하는 purpose－of 슬로트의 값 중에 생성된 루울의 結論節에 있는 각 事實의 frame－name과 동일한 값이 존재하는 가를 조사하여 위와 같은 방법으로 결정한다.

4.3 생성된 루울의 意味的인 檢證

생성된 루울의 知識베이스에 저장하기 전에 意味的 또는 論理的으로 正當한 가를 檢證할 필요가 있다.

예를 들어 意味的 檢證을 하기 위하여 專門家로부터 입력된 데이타에 의하여 다음과 같은 루울이 생성되었다고 가정하자.

if (fuel－oil pressure (>$ 2.0 ?V))
then (fuel－oil－booster－pump condition ? ac)

이 루울은 "fuel－oil－booster－pump를 기동하여 fuel－oil 압력이 2.0이면 fuel－oil－booster－pump는 正常이다."라고 하는 입력에 의하여 생성된 루울중의 하나이다. 위의 루울은 fuel－oil－booster－pump가 非正常이 되기 위한 조건으로는 fuel－oil 압력에 대한 제한이 물론 필요하지만 fuel－oil－booster－pump가 기동되어 있을 경우에 한정되므로 條件節에 (fuel－oil－booster－pump state run)이라는 事實이 필요하다. 이와 같이 意味的으로 不正한 루울은 루울을 檢證하기 위한 메타루울에 의하여 지적되며 專門家에게 확인시킨다. 다음과 같은 메타루울이 이상의 檢證을 한다.

If 結論節에 있는 事實의 frame－name이 機械類 또는 펌프類이며, 그 事實이 (frame－name condition ? ac)이다.
then 條件節에 (frame－name state run)을 포함해야 한다.

4.4 생성된 루울의 論理的인 檢證

意味的인 檢證까지 성공한 루울은 知識베이스에 저장하기 전에 知識베이스에 저장되어 있는 루울과 비교하여 동일한 루울 또는 不必要한 루울이 있는가를 확인할 필요가 있다. 不必要한 루울 즉 하나의 루울의 條件節 또는 結論節이 다른 루울의 部分集合인가를 확인하기 위해서 다음과 같은 論理에 기초를 두었다.

i) a 이라면 b 이다. A
a 이라면 b and c 이다. B
루울A (루울B)가 생성되었을 경우 루울B (루울A)가 知識베이스에 존재하고 있으면 루울B를 우선하고 루울A를 不必要한 루울로 정의한다.

ii) a and b 이라면 c 이다. C
a 이라면 c 이다. D
루울C와 루울D에 대해서는 루울D를 우선시키고 루울C를 不必要한 루울로

로 정의한다.

이와 같이 생성된 루울과 知識베이스에 존재하는 루울을 비교하여 생성된 루울이 不必要한 루울이라고 정의되면 이 생성된 루울은 知識베이스에 저장될 필요가 없으며, 역으로 知識베이스에 있는 루울이 不必要한 루울이라고 판정되면 이 不必要한 루울을 知識베이스로부터 제거하고 생성된 루울을 知識베이스에 저장한다.

5. 프로그램의 實行例

본 씨스템은 SORD M685 上의 UTILISP 言語로 작성되어 있으며, 실험대상으로는 선박 주기관을 선택하여 이 기관에 적절한 RULE-F, RULE-I, 데이타베이스를 작성했다.

프로그램의 實行例는 부록에 실려있는 것과 같이 專門家와의 對話例가 주기관에 대한 知識獲得과정을 묘사하고 있다. 이 對話例는 간단한 시뮬레이션을 나타내고 있다. 실제로 선박 주기관의 故障診斷을 위한 知識을 획득하기 위해서는 專門家가 RUEL-F, RUEL-I, 데이타베이스를 보다 충실히 작성하는 것에 의하여 이루어질 수 있을 것이다.

씨스템은 (EXECUTE)에 의하여 기동되며 기동된 씨스템은 專門家에게 db, input, kb, q 중 하나를 요구한다. 이 입력이 db인 경우에는 데이타베이스로부터 루울을 생성하기 위한 루울 RULE-F를 기동시켜 생성시킨 루울수를 표시하고 다시 입력을 요구한다. 입력이 input인 경우에는 그 입력이 事實인가, 動作 또는 狀態를 나타내는 데이타인가를 구별하여 각각 RULE-F, RULE-I를 기동시킨다.

부록의 實行例는 데이타베이스로부터의 루울생성, 입력이 사실인 경우의 루울생성, 입력이 動作인 경우의 루울생성의 순으로 專門家와 對話를 나누고 있다. 또한 입력이 事實인 경우와 動作인 경우에는 각각 檢證과정이 묘사되어 있다.

6. 結 論

本 稿에서는 機械類를 중심으로 知識獲得에 관하여 고찰했다. 즉 專門家와 對話에 의하여 데이타 베이스로부터 루울을 생성하거나 직접 專門家의 입력을 루울로 변환시키며, 생성된 루울을 意味的 또는 論理的으로 檢證하는 方法에 대하여 고찰했다.

실험대상으로는 선박 주기관을 선택하여 이 기관에 관한 RULE-F, RULE-I, 데이타베이스를 작성했고, 특수한 專門分野로부터 制御知識을 制御하도록 했기 때문에 선박 주

plications. Proceedings IJCAI－77 , pp.920～928. 1977.

13．佐川外，機關故障診斷(シンポジウム)，日本船用機關學會誌，vol．14，No．11，1975 pp.905～950.

14．平 井：機械・裝置類の故障診斷へのルールベーズドシステムの適用について，電氣學會．システム・制御研究會資料 SC－82－43 (1982).

15．Clancey, W.J., The Advantages of Abstract Control Knowledge in Expert System Design, Proceedings of AAAl－83, 74－78, 1983.

16．Politakis, P. and Weiss, S., Using Empirical Analysis to Refine Expert System Knowledge Bases, Artificial Intelligence 22 (1984), pp.23～84.

17．Blum, R.L., Representation of Empirically Causal Relationships, Proceedings IJCAI－83, pp.268～271, 1983.

18．Thompson, T.F. and Wojcik, R.M., MELD: An Implementation of a Meta-Level Architecture for Process Diagnosis, The First Conference on Artifical Intelligence Applications", Sponsored by IEEE/AAAI, pp. 321～330, 1984.

19．Gallanti, M., Guida, G., Spampinato, L. and Stefanini, A., Representing Procedural Knowledge in Expert Systems: An Application to Process Control, Proceedings IJCAI－85, pp.345～352, 1985.

附錄

```
> (execute)
     Welcome to ship model!

Do you wish:DataBase,INPUT,KnowledgeBase,or Quit?
USER> db
81 rules are generated by using your data-base.

Do you wish:DataBase,INPUT,KnowledgeBase,or Quit?
USER> input
Please, give an input.
USER> (cylinder-oil-lubricator (part-of main-engine) (a-kind-of pumps))
Really this input?
 (cylinder-oil-lubricator (part-of main-engine) (a-kind-of pumps))
USER> yes

The following rule was produced by input.
   Input: (cylinder-oil-lubricator (part-of main-engine) (a-kind-of pumps))
   Product rule: (if (and (cylinder-oil-lubricator state run) (cylinder-oil condition abnormal))
                     then
                     (cylinder-oil-lubricator condition abnormal))

One rule in KB is similar to the new rule by input.
  Similar rule: ((R0055 if
                        (and (main-engine state run) (cylinder-oil-lubricator state stop))
                        then
                        (cylinder-oil-lubricator condition abnormal)))

This rule was indexed in KB.

The following rule was produced by input.
   Input: (cylinder-oil-lubricator (part-of main-engine) (a-kind-of pumps))
   Product rule: (if (and (main-engine state run) (cylinder-oil-lubricator state stop))
                     then
                     (cylinder-oil-lubricator condition abnormal))

The following rules in KB are similar to the new rule by input.
   Similar rules: ((R0086 if
                          (and (cylinder-oil-lubricator state run) (cylinder-oil condition abnormal))
                          then
                          (cylinder-oil-lubricator condition abnormal))
                  (R0055 if
                          (and (main-engine state run) (cylinder-oil-lubricator state stop))
                          then
                          (cylinder-oil-lubricator condition abnormal)))
This rule is existing in KB.
     (if (and (main-engine state run) (cylinder-oil-lubricator state stop))
         then
         (cylinder-oil-lubricator condition abnormal))

This rule isn't indexed because of existence of the equal rule in KB.
```

```
Do you wish:DataBase.INPUT.KnowledgeBase.or Quit?
USER> input
Please. give an input.
USER> (if (and (fuel-oil-booster-pump start) (starting-lever start-position))
        then (main-engine start))
Really this input?
 (if (and (fuel-oil-booster-pump start) (starting-lever start-position))
     then
     (main-engine start))
USER> yes

The following rule was produced by input.
   Input: (if (and (fuel-oil-booster-pump start) (starting-lever start-position))
              then
              (main-engine start))
   Product rule: (if (and (main-engine state run) (fuel-oil-booster-pump state stop))
                     then
                     (main-engine condition ?ac))

The rule produced was modified as follow.
   Modified rule: (if (and (main-engine state run) (fuel-oil-booster-pump state stop))
                      then
                      (main-engine condition critical))

The rule similar to new rule isn't existing in KB.

What do you think this rule:<good>.<partly-good>.<bad>?
USER> good
The following rule was indexed in KB.
Index rule: (if (and (main-engine state run) (fuel-oil-booster-pump state stop))
                then
                (main-engine condition critical))


The following rule was produced by input.
   Input: (if (and (fuel-oil-booster-pump start) (starting-lever start-position))
              then
              (main-engine start))
   Product rule: (if (and (main-engine state run) (starting-lever state /0-position))
                     then
                     (main-engine condition ?ac))
Is starting-lever frame adaptable for need-of-slot of main-engine frame?
USER> yes
What CF:<numerical(0-1)>?
USER> 0.7
The rule produced was modified as follow.
   Modified rule: (if (and (main-engine state run) (starting-lever state /0-position))
                      then
                      (main-engine condition abnormal))
```

```
The following rules in KB are similar to the new rule by input.
   Similar rules: ((R0019 if
                          (and (main-engine state run) (starting-Lever condition abnormal))
                          then
                          (main-engine condition abnormal))
                      .
                      .
                  (R0005 if
                          (and (main-engine state run) (cylinder condition abnormal))
                          then
                          (main-engine condition abnormal)))
One rule in KB is similar to the new rule by input.
  Similar rule: ((R0053 if
                        (and (main-engine state run) (starting-Lever state /0-position))
                        then
                        (starting-Lever condition abnormal)))

What do you think this rule:<good>,<partly-good>,<bad>?
USER> good
The following rule was indexed in KB.
Index rule: (if (and (main-engine state run) (starting-Lever state /0-position))
                then
                (main-engine condition abnormal))

Do you wish:DataBase,INPUT,KnowledgeBase,or Quit?
USER> q
Do you want to continue?
USER> no

    Return to LISP
>
```