**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**HỆ CHUYÊN GIA**

**Đề tài: SỬ DỤNG NGÔN NGỮ PROLOG XÂY DỰNG**

**HỆ CHUYÊN GIA CHẨN ĐOÁN HỎNG XE MÁY**

**Sinh viên thực hiện : NGUYỄN BÍCH NGỌC**

**LƯU QUANG NGUYÊN**

**LÊ CÔNG NGUYỆN**

**Giảng viên hướng dẫn : ĐỖ TRUNG TUẤN**

**Ngành : CÔNG NGHỆ THÔNG TIN**

**Chuyên ngành : CÔNG NGHỆ PHẦN MỀM**

**Lớp : D12CNPM1**

**Khóa : 2017 – 2022**

***Hà Nội, tháng 10 năm 2020***

**PHIẾU CHẤM ĐIỂM**

Sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ ký** | **Điểm** |
| Nguyễn Bích Ngọc  Mã SV: 17813100620 |  |  |
| Lưu Quang Nguyên  Mã SV: 1781310061 |  |  |
| Lê Công Nguyện  Mã SV: 1181020069 |  |  |

Giảng viên chấm:

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1 : |  |  |
| Giảng viên chấm 2 : |  |  |

**LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, sự giúp đỡ dù ít hay nhiều, dù là trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập, chúng tôi đã nhận được rất nhiều sự quan tâm, giúp đỡ của Thầy Cô, gia đình và bạn bè.

Chúng tôi xin chân thành gửi lời cảm ơn sâu sắc tới các thầy, các cô đã tận tình giảng dạy và truyền đạt cho chúng tôi những kiến thức, kinh nghiệm quý báu trong suốt thời gian học tập và rèn luyện.

Sau cùng, chúng tôi xin kính chúc các thầy cô trong Khoa Công Nghệ Thông Tin dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Chúng tôi xin chân thành cảm ơn!

***Sinh viên thực hiện***

***Nguyễn Bích Ngọc***

***Lưu Quang Nguyên***

***Lê Công Nguyện***

# LỜI MỞ ĐẦU

Thế giới ngày nay phát triển mạnh mẽ với các hoạt động vô cùng đa dạng và phức tạp đòi hỏi khả năng giải quyết vấn đề ở mức độ trí tuệ nhân tạo ngày càng cao. Lĩnh vực trí tuệ nhân tạo nói chung và hệ chuyên gia nói riêng góp phần tạo ra các hệ thống có khả năng trí tuệ của con người, có được tri thức tiên tiến của các hệ chuyên gia để giải quyết các vấn đề phức tạp trong cuộc sống. Hệ chuyên gia được thu hút mạnh mẽ vì những ưu điểm sau: Các chương trình hệ chuyên gia ngày càng tỏ ra hữu hiệu và tiện lợi đáp ứng nhu cầu thực tế, Các chương trình hệ chuyên gia ngày càng tỏ ra có tính khả thi cao, Hệ chuyên gia không có tính đơn lẻ, phù hợp với nhiều cá nhân. Ở Việt Nam, phương tiện di chuyển chủ yếu trên đường bộ thường là xe máy, xe ô tô, xe bus… trong đó, xe máy là phương tiện chiếm đa số. Chúng tôi tìm hiểu và thực hiện đề tài “Sử dụng ngôn ngữ Prolog xây dựng hệ chuyên gia chẩn đoán hỏng xe máy”, thực hiện tìm hiểu những thuật toán suy diễn và xây dựng hệ chuyên gia nhằm chẩn đoán “bệnh” của xe máy. Chúng tôi đã cố gắng tìm hiểu và hoàn thành đề tài, tuy nhiên, do thời gian và sự hiểu biết về xe máy còn có hạn nên trong quá trình xây dựng hệ chuyên gia còn thiếu sót, mong thầy tiếp tục góp ý để đề tài của chúng tôi hoàn thiện hơn.

# MỞ ĐẦU

## Lý do chọn đề tài

Việc sửa chữa xe máy đời mới ngày nay không chỉ đơn thuần là kinh nghiêm. Có rất nhiều lỗi nếu chỉ dùng kinh nghiệm rất khó để tím ra nguyễn nahn và nếu có tìm ra thì cũng phải mất khá nhiều thời gian. Vấn đề quan trọng nhất là phải tìm đúng nguyên nhân. Khin đã định đúng “bệnh” của xe thì mới có phương pháp sửa chữa đúng đắn; nhằm đảm bảo việc sửa chữa nhanh chóng, hiệu quả; tránh được những tổn thất ngoài dự kiến cho xe và chủ sở hữu do sửa chữa mò, sửa chữa không có phương pháp gây ra. Thực hiện sửa chữa theo xu hướng hiện đại, đó là sử dụng một “Hệ chuyên gia chuẩn đoán hỏng xe máy” để chuẩn đoán lỗi giúp cho việc sửa chữa đạt được mục tiêu chính xác nhất cũng như người sử dụng xe biết được chính xác xe của mình bị “bệnh” gì. Vậy nên, để hoàn thành môn học, nhóm em xin chọn đề tài: “***Sử dụng ngôn ngữ prolog xây dựng hệ chuyên gia chẩn đoán hỏng xe máy***”.

## Cấu trúc của tiểu luận

Chương 1: Tổng quan về Hệ chuyên gia.

Chương 2: Hệ chuyên gia chẩn đoán hỏng xe máy.

Chương 3: Thực nghiệm.

***Sinh viên thực hiện***

***Nguyễn Bích Ngọc***

***Lưu Quang Nguyên***

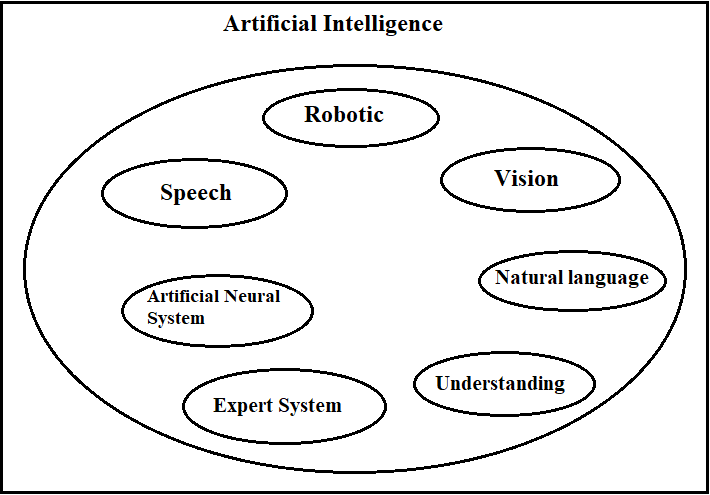
***Lê Công Nguyện***

# **Chương 1. Về hệ hệ chuyên gia**

## 1.1.Một số khái niệm

Hệ chuyên gia (Expert System) là một chương trình máy tính thông minh sử dụng tri thức (knowledge) và các thủ tục suy luận (inference procedures) để giải những bài toán tương đối khó khăn đòi hỏi những chuyên gia mới giải được, hay nói cách khác là thay con người giải quyết vấn đề về mọi lĩnh vực.

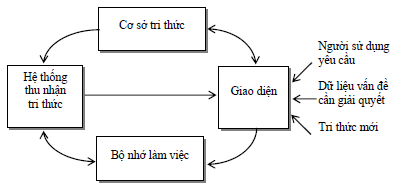
Hệ chuyên gia là một trong những lĩnh vực ứng dụng của trí tuệ nhân tạo (Artificial Intelligence) như hình dưới đây:



### Hình 1.1. Một số lĩnh vực ứng dụng của trí tuệ nhân tạo

Một hệ chuyên gia gồm ba thành phần chính: cơ sở tri thức, máy suy diễn, hệ thống giao tiếp với người sử dụng.

## 1.2. Về hệ thống chuyên gia



### Hình 1.2. Sơ đồ hệ chuyên gia

Đặc trưng của hệ chuyên gia:

* Hiệu quả cao. Khả năng trả lời với mức độ tinh thông bằng hoặc cao hơn so với chuyên gia trong cùng lĩnh vực
* Thời gian trả lời thỏa đáng. Thời gian trả lời bằng hoặc nhanh hơn so với chuyên gia để đi đến cùng một quyết định
* Độ tin cậy cao. Không thể xảy ra sự cố hoặc giảm sút độ tin cậy khi sử dụng
* Dễ hiểu. Hệ chuyên gia giải thích các bước suy luận một cách dễ hiểu và nhất quán, không giống như cách bí ẩn như hộp đen

Ưu điểm của hệ chuyên gia:

* Luôn phổ cập. Là sản phẩm chuyên gia, được phát triển không ngừng với hiệu quả sử dụng không thể phủ nhận
* Giảm giá thành
* Giảm rủi ro
* Tính thường trực. Bất kể lúc nào cũng có thể khai thác và sử dụng trong khi đó con người có thể mệt mỏi hoặc nghỉ ngơi
* Đa lĩnh vực
* Khả năng giảng giải rõ ràng và chi tiết câu trả lời
* Tính ổn định, suy luận có lý và đầy đủ mọi lúc mọi nơi
* Có thể truy cập như một cơ sở dữ liệu thông minh.

## 1.3. Kết luận

Hệ chuyên gia là một chương trình máy tính biểu diễn và lập luận dựa trên tri thức trong một chủ đề thuộc lĩnh vực cụ thể nào đó, qua việc giải quyết vấn đề hoặc đưa ra lời khuyên.

# **Chương 2. Hệ chuyên gia chẩn đoán hỏng xe máy**

## 2.1 Giới thiệu đề tài

Ngày nay, công nghệ thông tin là một lĩnh vực quan trọng trong đời sống xã hội. Song song cùng với đó, giao thông vận tải cũng phát triển không kém. Cùng với đường sắt, đường thủy, đường hàng không, giao thông đường bộ đóng một vai trò quan trọng trong giao thông cả nước. Chỉ cần đi ra đường, chắc hẳn chúng ta rất dễ dàng bắt gặp một chiếc xe máy chạy trên đường. Nếu xe buýt là phương tiện công cộng phổ biến thì xe máy là phương tiện cá nhân chiếm đa số.

Những sự cố luôn tồn tại với các phương tiện giao thông, xe máy cũng là phương tiện nên gặp sự cố , hỏng hóc là điều không thể tránh khỏi. Với những người không phải chuyên nghiệp, thật khó khăn để biết được chiếc xe máy của mình bị hỏng hóc, sự cố ở bộ phận nào. Do đó, với mong muốn giúp chẩn đoán hỏng xe máy một cách tự động dựa vào những sự cố xuất hiện khi đi xe máy nên chúng tôi thực hiện đề tài: “**Sử dụng ngôn ngữ Prolog xây dựng hệ chuyên gia chẩn đoán hỏng xe máy**”.

## 2.2 Mục tiêu đề tài

Xây dựng hệ chuyên gia chẩn đoán hỏng xe máy dựa trên những dấu hiệu về sự cố mà người sử dụng lựa chọn.

## 2.3 Phạm vi và giới hạn

* Hệ chuyên gia chỉ chẩn đoán hỏng xe máy ở bộ phận nào và đưa ra gợi ý kiểm tra.
* Người sử dụng: Bao gồm tất cả những người sử dụng xe máy, người sửa chữa xe máy.

## 2.4 Phương pháp thực hiện

* Tìm hiểu về các hệ cơ sở tri thức, hệ chuyên gia.
* Tham khảo các hệ chuyên gia có liên quan đến đề tài.

## 2.5 Nội dung thực hiện

* Kiến trúc tổng quát của hệ chuyên gia.
* Biểu diễn tri thức trong hệ chuyên gia.
* Thiết kế cài đặt hệ chuyên gia chẩn đoán hỏng xe máy.

## 2.6 Các luật dùng trong chương trình

**1**. NẾU người lái xe máy đạp nổ động cơ hoặc bật nút khởi động

VÀ động cơ không nổ

THÌ hãy kiểm tra bộ phận đánh lửa của bộ khởi động.

Sự kiện:

E1: Đạp khởi động

E2: Bấm nút đề (start)

E3: Không nổ, không chạy

E4: Kiểm tra điện

Luật:

**R1**: (E1 v E2) ^E3 -> E4

**2**. NẾU bộ điện từ khởi động và người lái xe mô tô đạp động cơ hoặc nhấn vào nút khởi động

VÀ động cơ vẫn không quay

THÌ hãy kiểm tra mức sạc của ắc quy

Sự kiện:

E1: Bộ điện chạy

E2: người lái đạp động cơ

E3: bấm nút đề

E4: động cơ không quay

E5: Kiểm tra mức sạc của ắc quy

Luật:

R2: ((E1 và E2) v E3) ^ E4 -> E5

**3**. NẾU pin đã được sạc đầy

VÀ động cơ vẫn không quay

THÌ hãy kiểm tra cáp jumper.

Sự kiện:

E1: pin đã sạc đầy

E2: động cơ không quay

E3: kiểm tra cap jumper

Luật:

R3: E1^E2->E3

**4**. NẾU người lái tắt động cơ

VÀ đèn pha vẫn sáng

THÌ có vấn đề về điện, nên tháo cực ắc quy.

Sự kện:

E1: tắt động cơ

E2: đèn vẫn sáng

E3: kiểm tra cực ắc quy

Luật:

R4: E1^E2->E3

**5**. NẾU pin đã được sạc

VÀ động cơ vẫn quay chậm khi đánh lửa

THÌ hãy thay pin mới.

Sự kiện:

E1: pin đã được sạc

E2: động cơ quay chậm

E3: thay pin mới

Luật:

R5: E1 ^ E2 -> E3

|  |  |  |
| --- | --- | --- |
| **Mã Luật** | **Chẩn đoán hỏng xe máy** | |
| **Giả thiết** | **Kết luận** |
| **R1** | **IF** đạp khởi động  **OR** bấm nút đề (start)  **AND** không nổ, không chạy | **THEN** hỏng điện, kiểm tra điện |
| **R2** | **IF** bộ điện chạy  **AND** người lái đạp động cơ  **AND** động cơ không quay  **OR** bấm nút đề  **AND** động cơ không quay | **THEN** ắc quy thiếu điện, kiểm tra mức sạc của ắc quy. |
| **R3** | **IF** ắc quy được sạc đầy  **AND** động cơ không quay | **THEN** hỏng cap kết nối, kiểm tra cáp jumper |
| **R4** | **IF** tắt động cơ  **AND** đèn vẫn sáng | **THEN** hỏng cực, kiểm tra cực ắc quy |
| **R5** | **IF** ắc quy được sạc đầy  **AND** động cơ quay chậm | **THEN**  hỏng ắc quy, thay ắc quy mới |

# **Chương 3. Thực nghiệm**

## 3.1.Về ngôn ngữ Prolog

### 3.1.1 Nhà sáng lập ngôn ngữ Prolog

Prolog là một ngôn ngữ lập trình máy tính thế hệ thứ 5 được phát triển bởi Alain Colmerauer đầu những năm 70. Cái tên “Prolog” bắt nguồn từ cụm từ “PROgramming in LOGic”. Nó được sử dụng phổ biến trong trí tuệ nhân tạo như là một công cụ để xây dựng Hệ chuyên gia (ES), Giao diện ngôn ngữ tự nhiên (Natural Language Interface). Một trong những phiên bản PROLOG khá nổi tiếng là Turbo Prolog, do hãng Borland phát triển trong những năm 80.



*Hình 3.1. Tác giả của Prolog, Alain Colmerauer*

### 3.1.2 Các yếu tố cơ bản của Turbo Prolog

Trong một chương trình Prolog, ta cần khai báo các yếu tố sau đây: đối tượng, quan hệ giữa các đối tượng, sự kiện và các luật.

**Đối tượng**

Gồm có các hằng và biến. Hằng mang giá trị cho sẵn ở đầu chương trình hoặc trong quá trình viết ta đưa vào; Các biến có giá trị thay đổi sẽ được gán giá trị khi chạy chương trình. Tên biến là một ký tự hoa hoặc một chuỗi ký tự, bắt đầu bằng một ký tự hoa.

Có một loại biến đặc biệt gọi là biến tự do, biến này không có tên và người ta dùng ký hiệu \_ (dấu gạch dưới) thay cho tên biến.

**Quan hệ giữa các đối tượng**

Quan hệ giữa các đối tượng được dùng dưới hình thức vị từ.

Ví dụ: Thich(X,Y) là vị từ diễn tả câu “X thích Y” trong ngôn ngữ tự nhiên.

Blue(car) là vị từ diễn tả câu “Car is blue”.

Như vậy các vị từ sẽ bao gồm tên của vị từ và các đối số của nó. Các đối số được đặt trong ngoặc và phân cách nhau bởi dấu phẩy.

**Sự kiện và luật**

Sự kiện là một vị từ diễn tả một sự thật.

Ví dụ: “2 là một số nguyên tố” là một sự kiện vì nó diễn tả sự thật 2 là một số nguyên tố.

Luật là vị từ diễn tả quy luật suy diễn mà ta công nhận đúng. Luật được trình bày dưới dạng một mệnh đề.

Ví dụ để suy diễn số nguyên N bất kỳ là một số nguyên tố ta viết:

“N là một số nguyên tố nếu N>0, M là số nguyên tố nào đó, M<N và N không chia hết cho M”.

**Cấu trúc của một chương trình Prolog**

Một chương trình Prolog thường gồm có 3 hoặc 4 đoạn cơ bản: clauses, predicates, domains và goal. Phần goal có thể bỏ đi, nếu ta không thiết kế goal trong chương trình, thì khi thực hiện, hệ thống sẽ yêu cầu ta nhập goal vào.

**Phần Domains**

Đây là phần định nghĩa kiểu mới dựa vào các kiểu đã biết. Các kiểu được định nghĩa ở đây sẽ được sử dụng cho các đối số trong các vị từ. Nếu các vị từ sử dụng đối số có kiểu cơ bản thì có thể không cần phải định nghĩa lại các kiểu đó. Tuy nhiên để cho chương trình sáng sủa, người ta sẽ định nghĩa lại cả các kiểu cơ bản.

Cú pháp: <**danh sách kiểu mới**>**=**<**kiểu đã biết**> hoặc <**danh sách kiểu mới**>**=**<**danh sách kiểu đã biết**>

Trong đó các kiểu mới phân cách nhau bởi dấu phẩy, còn các kiểu đã biết phân cách nhau bởi dấu chấm phẩy.

**Ví dụ:**

Domains

ten, tac\_gia, nha\_xb, dia\_chi = string

nam, thang, so\_luong = integer

dien\_tich = real

nam\_xb = nxb(thang, nam)

do\_vat = sach(tac\_gia, ten, nha\_xb, nam\_xb); xe(ten, so\_luong); nha(dia\_chi, dien\_tich)

Trong ví dụ trên, ta đã định nghĩa các kiểu mới, trong đó các kiểu mới ten, tac\_gia, nha\_xb, dia\_chi dựa vào cùng một kiểu đã biết là string; các kiểu mới nam, thang, so\_luong dựa vào cùng một kiểu đã biết là integer; kiểu mới dien\_tich dựa vào kiểu đã biết là real; kiểu mới năm\_xb dựa vào kiểu nxb được xây dựng từ các kiểu đã biết là thang, nam; còn kiểu do\_vat lại dựa vào các kiểu sach, xe, nha mà các kiểu này lại dựa vào các kiểu đã biết.

**Phần Predicates**

Đây là phần bắt buộc phải có. Trong phần này chúng ta cần phải khai báo đầy đủ các vị từ sử dụng trong phần Clauses, ngoại trừ các vị từ mà Turbo Prolog đã xây dựng sẵn.

Cú pháp: <**Tên vị từ**> **(**<**danh sách các kiểu**>**)**

Các kiểu là các kiểu cơ bản hoặc là các kiểu đã được định nghĩa trong phần domains và được viết phân cách nhau bơi dấu phẩy.

Ví dụ:

Predicates

so\_huu (ten, do\_vat)

so\_nguyen\_to(integer)

Trong ví dụ trên ta khai báo hai vị từ. Trong đó vị từ so\_huu (ten, do\_vat) để chỉ một người có tên là ten sẽ sở hữu môt do\_vat nào đó. Còn vị từ so\_nguyen\_to(integer) để xét xem một số integer nào đó có phải là số nguyên tố hay không.

**Phần Clauses**

Đây là phần bắt buộc phải có dùng để mô tả các sự kiện và các luật, sử dụng các vị từ đã khai báo trong phần predicates.

Cú pháp:

<**Tên vị từ**>**(**<**danh sách các tham số**>**)** <**kí hiệu**>

<**Tên vị từ 1**>**(**<**danh sách các tham số 1**>**)** <**kí hiệu**>

… … …

<**Tên vị từ N**>**(**<**danh sách các tham số N**>**)** <**kí hiệu**>

Trong đó: Tên vị từ phải là các tên vị từ đã được khai báo trong phần predicates. Các tham số có thể là các hằng hoặc biến có kiểu tương thích với các kiểu tương ứng đã được khai báo trong các vị từ ở trong phần predicates; các tham số được viết cách nhau bởi dấu phẩy. Các kí hiệu bao gồm:

**:-** (điều kiện nếu).

**,** (điều kiện và).

**;** (điều kiện hoặc).

**.** (kết thúc vị từ)

Ví dụ:

Clauses

so\_nguyen\_to(2):- !.

so\_nguyen\_to(N):- N>0,

so\_nguyen\_to(M),

M<N,

N MOD M <>0.

so\_huu(“Nguyen Van A”, sach(“Do Xuan Loi”, “Cau truc DL”, “Khoa hoc Ky thuat”, nxb(8,1985))).

Chú ý: Nếu trong các tham số của một vị từ có biến thì biến này phải xuất hiện ít nhất 2 lần trong vị từ đó hoặc trong các vị từ dùng để suy diễn ra vị từ đó. Nếu chỉ xuất hiện một lần thì bắt buộc phải dùng biến tự do.

Ví dụ: Để diễn tả sự kiện: Tổ hợp chập 0 của N (N bất kỳ) bằng 1, ta không thể viết Tohop(N,0,1) vì biến N chỉ xuất hiện đúng một lần trong vị từ này, do đó ta phải viết Tohop(\_,0,1) .

**Phần Goal**

Bao gồm các mục tiêu mà ta yêu cầu Turbo Prolog xác định và tìm kết quả. Đây là phần không bắt buộc phải có. Nếu ta viết sẵn trong chương trình thì đó gọi là goal nội; Nếu không, khi chạy chương trình Turbo Prolog sẽ yêu cầu ta nhập goal vào, lúc này gọi là goal ngoại.

Cú pháp phần goal giống như cú pháp phần clauses. Tức là ta đưa vào một hoặc một số các vị từ.

Nếu tất cả các tham số của vị từ là hằng thì kết quả nhận được là Yes (đúng) hoặc No (sai). Nếu trong các tham số của vị từ có biến thì kết quả trả về sẽ là các giá trị của biến.

Ngoài các phần chủ yếu nói trên, ta có thể đưa vào các phần liên quan đến khai báo hằng, các tập tin liên quan hoặc chỉ thị dịch.

Ví dụ:

Constants

Pi = 3.141592653

**Một số ví dụ về chương trình prolog**

**Ví dụ:**Xét xem một số N có phải là số nguyên tố hay không.

domains

so\_nguyen = integer

predicates

so\_nguyen\_to(so\_nguyen)

Clauses

so\_nguyen\_to(2):- !.

so\_nguyen\_to(N):- N>0,

so\_nguyen\_to(M),

M<N,

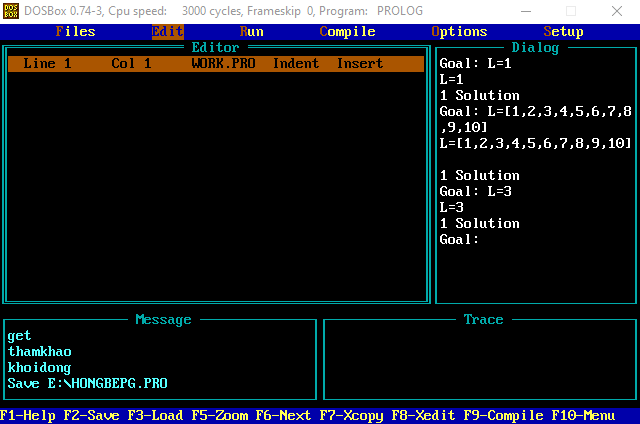
N MOD M <>0.

goal

so\_nguyen\_to(13).

## 3.2. Phần mềm sử dụng

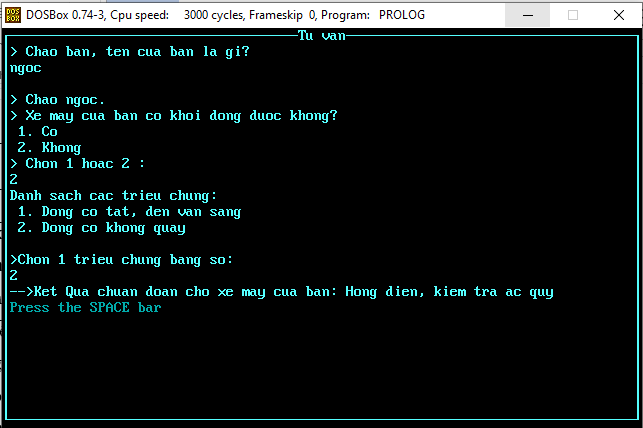
Chương trình sử dụng ngôn ngữ prolog và phần mềm DOSBox phiên bản 0.74-3 để cài đặt chương trình chuẩn đoán hỏng xe máy.



### Hình 3.2.Giao diện của phần mềm DOSBox.

## 3.3. Chương trình

Chương trình đã có thể mô phỏng được hệ chuyên gia, cũng như biểu diễn các luật được đề ra và đã có thể in ra được bệnh của xe máy thông qua các “triệu chứng” cụ thể.



### Hình 3.3.Giao diện kết quả của chương trình

# **Kết luận**

***1.Kết quả làm được***

Xây dựng một hệ chuyên nói chung và hệ chuyên gia về chẩn đoán hỏng xe máy nói riêng là một công việc đòi hỏi nhiều thời gian và công sức. Do bị giới hạn về mặt thời gian, bài tập này chỉ hoàn thành ở mức cơ bản, mang một qui mô nhỏ. Trong quá trình thực hiện đồ án, chúng tôi đã thực hiện được những công việc sau:

+ Lọc được các triệu chứng qua việc sừ dụng xe máy

+ Cài đặt bài toán bằng phần mềm DOSBox và ngôn ngữ Prolog.

+ Hiển thị kết quả chuẩn đoán hỏng xe máy.

***2. Hướng phát triển đề tài***

+ Tiếp tục thu thập dữ liệu để củng cố kho cơ sở tri thức cho bài toán.

+ Mở rộng qui mô chẩn đoán của chương trình, theo chiều sâu (xác định nhiều chi tiết lỗi hơn) .

+ Sử dụng các công cụ mạnh hơn Swi-Prolog như WinProlog, Visual Prolog nhằm xây dựng giao diện trực quan cho chương trình.

# **Tài liệu tham khảo**

* 1. *Đỗ Trung Tuấn, Hệ chuyên gia, NXB. Giáo dục, 1999*

1. *Đỗ Trung Tuấn, Trí tuệ nhân tạo, NXB. Đại học quốc gia Hà Nội, 2002.*
2. *Hayes-Roth, Frederick; Waterman, Donald; Lenat, Douglas Addison-Wesley.* [*Building Expert Systems*](https://archive.org/details/buildingexpertsy00temd)*, 1983.*
3. [*Feigenbaum, Edward A.*](https://en.wikipedia.org/wiki/Edward_Feigenbaum)*;*[*McCorduck, Pamela*](https://en.wikipedia.org/wiki/Pamela_McCorduck)*, Addison-Wesley. The fifth generation (1st ed.), 1983.*
4. *Douglas B. Lenat. "On the thresholds of knowledge", 1992.*